Energy-Aware Evolutionary Optimization for Cyber-Physical Systems in Industry 4.0

Xu Gong

Promotoren: prof. dr. ir. W. Joseph, prof. dr. ir. L. Martens
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

UNIVERSITEIT
GENT

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. B. Dhoedt
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2017 - 2018

ISBN 978-94-6355-105-2
NUR 958, 959
Wettelijk depot: D/2018/10.500/23

Promoters
Prof. Dr. Ir. Luc MARTENS
Prof. Dr. Ir. Wout JOSEPH

Chair
Prof. Dr. Ir. Patrick DE BAETS (Ghent University – EEMMeCS)

Other members of the examination board (in alphabetical order):
Prof. Dr. Ing. Johannes COTTYN (Ghent University – ISyE)
Dr. Ir. Toon DE PESSEMIER (Ghent University – INTEC)
Dr. Xiang LI (Singapore Institute of Manufacturing Technology)
Prof. Dr. Ir. Mario PICKAVET (Ghent University – INTEC)
Prof. Dr. Ing. Tony WAUTERS (KU Leuven, Belgium)

Acknowledgment

I would like to thank my two supervisors, Prof. Luc Martens and Prof. Wout Joseph, to provide me the opportunity to focus on this PhD study at Ghent University & imec during four years. I owe my sincere gratitude to them and my mentor Dr. Toon De Pessemier, for their help and advice during this challenging yet interesting journey. Over the past four years, it has been my great honor to collaborate with (in alphabetical order) Quentin Braet, Didier Colle, Dirk Deschrijver, Tom Dhaene, Jeroen Hoebeke, Ying Liu, Niels Lohse, David Plets, Prashant Singh, Emmeric Tanghe, Jens Trogh, Marlies Van der Wee, and Sofie Verbrugge for joint publications.

Many other colleagues and project/academic collaborators have also provided inspirations and/or help in various ways to enrich this PhD study. I sincerely thank (in alphabetical order): Ivo Couckuyt, Margot Deruyck, Wouter Haerick, Jetmir Haxhibeqiri, Bart Jooris, Bart Lannoo, Xiang Li, Wei Liu, Michael Mehari, Vincent Sercu, Thomas Sys, David Van Den Dooren, Kris Vanhecke, Leen Verloock, and Yizhi (George) Zhao.

A number of companies have also provided various help during my PhD study, in terms of offering industrial environments, machines/devices, empirical data, know-how, feedback on my research, etc. I would thus thank (in alphabetical order): Delta Engineering, Egemin, Nervia Plastics, Objective, Siemens, System Insights, and Volvo.

I also thank all the other colleagues in the WAVES-WiCa group for the nice working atmosphere that they have created. They are (in alphabetical order): Sam Aerts, Reza Aminzadeh, Aliou Bamba, Sander Bastiaens, Said Benaissa, Joachim David, Simon Dooms, Brecht Hanssens, Frederic Heereman, Karien Hemelsoen, Rodney Martinez Alonso, Michel Matalatala Tamasala, Denys Nikolayev, Liu Ning, Mostafa Pakparvar, Nico Podevijn, Amine M. Samoudi, Ke Shen, Sergei Shikhantsov, Thomas Tarnaud, Arno Thielens, Matthias Van Den Bossche, Isabelle Van der Elstraeten, Günter Vermeeren, Miao Yang, and Marwan Yusuf.

Last but not least, I thank my parents, my grandparents, my fiancée, and her parents for their long-term support in my PhD study although they are far in the China. I also thank my Belgian landlord Rose, who creates a homelike atmosphere.

Gent, March 2018
Xu Gong

Table of Contents

List of Figures

List of Tables

List of Acronyms

0-9

2D	Two-dimensional
3D	Three-dimensional
5G	Fifth Generation

A

ACO	Ant Colony Optimization
AEAP	As Early As Possible (schedule)
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
ALAP	As Late As Possible (schedule)
AMOMA	Adaptive Multi-Objective Memetic Algorithm
AP	Access Point
APS	Automated Planning and Scheduling
ASCII	American Standard Code for Information Interchange

C

CapEx	Capital Expenditure
CC	Correlation Coefficient
CI	Computational Intelligence
CPP	Critical Peak Pricing
CPU	Central Processing Unit

CSV	Comma-Separated Values
CTS	Convergence-oriented Tabu Search

D

DAM	Day-Ahead Market (for energy purchase)
DES	Discrete-Event Simulation
DoE	Design of Experiments
DR	Demand Response
DSM	Demand Side Management
DTS	Diversity-oriented Tabu Search

E

EA	Evolutionary Algorithm
EC	Evolutionary Computing
EBM	Extrusion Blow Molding
ERP	Enterprise Resource Planning

F

FJSSP	Flexible Job Shop Scheduling Problem
FMS	Flexible Manufacturing System
FSM	Finite State Machine

G

GA	Genetic Algorithm
GAOD	Genetic Algorithm based Over-Dimensioning
GATPC	Genetic Algorithm based Transmit Power Control
GHG	Greenhouse Gas
GHOD	Greedy Heuristic based Over-Dimensioning
GP	Grid Point
GRASP	Greedy Randomized Adaptive Search Procedure

H

HetNet	Heterogeneous Networks
HMI	Human-machine interface
HPC	High-Performance Computing
HVAC	Heating, Ventilation, and Air Conditioning

I

ICT	Information and Communications Technology
IIoT	Industrial Internet-of-Things
I/O	Input/output
IoT	Internet-of-Things
IP	Integer Programming
IWLAN	Industrial Wireless Local Area Network

J

JSON	JavaScript Object Notation

L

LP	Linear Programming
LTE	Long-Term Evolution

M

MA	Memetic Algorithm
MaOP	Many-objective Optimization Problem
MC	Memetic Computation
MEP	Mean Electricity Price
MES	Manufacturing Execution System
MHS	Material Handling System
MIP	Mixed Integer Programming
MOEA	Multi-Objective Evolutionary Algorithm

MOP	Multiobjective Optimization Problem
MRP II	Manufacturing Resource Planning
MWL	Maximum Work Load (of a machine)
M2M	Machine-to-Machine

N

NC	Numerical Control
NP	Non-deterministic Polynomial
NSGA-II	Nondominated Sorting Genetic Algorithm-II
NSGA-III	Nondominated Sorting Genetic Algorithm-III

O

OD	Over-Dimensioning

P

PC	Personal Computer
PDA	Personal Digital Assistant
PL	Path Loss
PLC	Programmable Logic Controller
PSO	Particle Swarm Optimization

Q

QoS	Quality of Service

R

RCL	Restricted Candidate List
REM	Radio Environment Map
RF	Radio Frequency
RFID	Radio-Frequency IDentification

RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RTP	Real-Time Pricing
RTPC	Random Transmit Power Control
Rx	Receiver (of wireless signals)

S

SCADA	Supervisory Control And Data Acquisition
SEC	Specific Energy Consumption
SOP	Single-objective Optimization Problem

T

TDMA	Time Division Multiple Access
ToUP	Time-of-Use Pricing
TS	Tabu Search
TSP	Traveling Salesman Problem
TPC	Transmit Power Control
TWL	Total Work Load (of machines)
Tx	Transmitter (of wireless signals)

V

VRP	Vehicle Routing Problem

W

WCDMA	Wideband Code Division Multiple Access
WiFi	Wireless Fidelity
WirelessHART	Wireless Highway Addressable Remote Transducer protocol
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

WWAN	Wireless Wide Area Network

X

XML	eXtensible Markup Language

Nederlandse Samenvatting
–Dutch Summary–

Industrie 4.0 is een wereldwijd opkomend onderwerp sinds de officiële presentatie op Hannover Messe (één van 's werelds grootste industriële beurzen in Hannover, Duitsland) in 2011. In dit initiatief raken internet- en productietechnologieën, die gewoonlijk afzonderlijk worden ontwikkeld, verweven. Dit omvat een reeks opkomende technologieën, zoals cyber-fysische systemen (CPS), industrial Internet-of-Things (IIoT), big-data analyse, edge computing en cloud computing. Deze technologieën maken de industrie 4.0 evolutie mogelijk op 3 fronten: (1) heterogene en gedistribueerde monitoring van fysieke omgevingen en / of objecten, (2) accurate en inzichtelijke analyses van massieve gestructureerde en ongestructureerde data, (3) goed geïnformeerde en efficiënte optimalisatie en controle van fysieke systemen. Door deze Industrie 4.0 updates krijgen industriële bedrijven een beter inzicht in mogelijke verbeteringen van hun interne operationele efficiëntie, reducties van productiekosten en afvalproductie, en het aanbieden van zeer gepersonaliseerde producten met een zeer kleine oplage, met een nog betere productkwaliteit en lagere prijzen voor hun klanten.

Dit proefschrift onderzoekt Industrie 4.0 vanuit het perspectief van energiebewust evolutionair rekenen. Het industriële energieverbruik neemt ongeveer 1/3 van het totale energieverbruik in een samenleving in beslag. Met de introductie van "smart grids" in de verwerkende industrie, moeten fabrieken hun productie aanpassen aan variabele elektriciteitsprijzen ("demand response" of DR), om hun elektriciteitskosten te verlagen en hun reputatie op het gebied van duurzaamheid te promoten. Evolutionair rekenen is een subdomein van kunstmatige intelligentie ("Artificial Intelligence" of AI). Het bevat diverse op de natuur geïnspireerde metaheuristieken voor snelle en hoogwaardige optimalisatie. Het onderzoek van dit proefschrift wordt in twee delen opgesplitst. Het eerste deel concentreert zich

op energiebewuste productieplanning onder variabele elektriciteitsprijzen, om de energiekosten en de productiekosten voor fabrieken te verlagen, zonder de conventionele productie KPI aan te tasten. Het tweede deel onderzoekt hoe dichte, economische, industriële draadloze netwerken gepland kunnen worden en hoe het zendvermogen van deze netwerken kan worden geregeld voor interferentiebeperking, zodat het hele draadloze netwerk robuust kan blijven in ruwe industriële omgevingen.

Hoofdstuk 1 van dit proefschrift geeft een inleiding met achtergrondinformatie. Hoofdstukken 2, 3 en 4 behandelen het onderwerp van energiebewuste productieplanning onder volatiele elektriciteitsprijzen. Deze drie hoofdstukken behandelen hoofdzakelijk drie problemen. Het eerste probleem is de integratie van energiebewustzijn in conventionele productieplanningsmodellen, rekening houdend met de mogelijke afweging van andere belangrijke productieaspecten, zoals "makespan" en arbeid. Het tweede probleem is de toepassing en vernieuwing van evolutionaire algoritmen om deze nieuwe productieplanningsmodellen op een snelle, hoogwaardige en zeer schaalbare manier op te lossen. Het derde probleem is de "wat-als" en statistische analyse van de economische prestaties van deze nieuwe modellen en algoritmen, zodat de gebruikers (bijv. fabrieksmanager) uitgebreid inzicht hebben in wanneer en hoe deze modellen en algoritmen economisch concurrerend kunnen worden.

Hoofdstuk 2 presenteert de methode om energiebewuste productieplanning uit te voeren op het niveau van één machine. Dit hoofdstuk introduceert eerst de FSM-methode (Finite-State Machine) om het energieverbruik van een machine te modelleren aan de hand van gemonitorde energiegegevens op de werkvloer, welke kunnen worden gecorreleerd met andere productiegegevens (bijv. producttype en -hoeveelheid, alsook machineveranderingen). Vervolgens formuleert dit hoofdstuk een energiebewust productieplanningsmodel voor één machine onder wisselende elektriciteitsprijzen. Productieactiviteiten worden op intelligente wijze verschoven van duurdere periodes naar lager geprijsde periodes met als doel lagere energiekosten. Daarna stelt dit hoofdstuk een simulatie-optimalisatieraamwerk voor dat is gebaseerd op de simulatie van discrete gebeurtenissen en een genetisch algoritme (GA). Dit raamwerk kan niet alleen het eerder geformuleerde planningsmodel op een schaalbare manier oplossen (in termen van het aantal tijdssloten in de planning en productieopdrachten), maar ook snel herplannen bij onvoorziene gebeurtenissen (bijv. machinestoring, dringende bestellingen, en geannuleerde bestellingen) om zo de productieplanning flexibel te maken voor de dynamiek op

de werkvloer. Ten slotte worden de reducties in energiekosten van deze energiebewuste productieplanningsmethode statistisch geëvalueerd in een casestudy van een vermaalmachine, waarbij conventionele productieplanning / verzendingsmethoden worden gebruikt als benchmarks. De gemiddelde energiekostenreductie van deze methode blijkt te liggen tussen 6% en 19% onder real-time prijsstelling (RTP). De wisselwerking tussen de energiekostenreductie en de "makespan" wordt ook statistisch geanalyseerd.

Hoofdstuk 3 gaat verder dan hoofdstuk 2 door arbeidsaspecten te integreren in het eerdere energiebewuste productieplanningsmodel voor één machine. Er wordt een continue heuristische accumulatieheuristiek voorgesteld om het type en het aantal menselijke werknemers te berekenen op basis van de status van het energieverbruik van de machine. Daarom zijn werknemers, ploegendiensten en machinebewerkingen gekoppeld, en is geïntegreerde energie- en arbeidsbewuste productiesimulatie mogelijk gemaakt. Naast het gebruik van een conventioneel GA voor optimalisatie van één objectief, wordt een adaptief memetisch algoritme (MA) voorgesteld voor multi-objectieve optimalisatie (MOO), dat AMOMA wordt genoemd. AMOMA heeft als doel een snelle convergentie naar het Pareto-front zonder verlies van diversiteit. Het integreert synergistisch in de veel gebruikte NSGA-II de convergentie- en diversiteitgestuurde tabu-zoekopdrachten. Tijdens een zoektocht naar een approximatieset van het Pareto-front, coördineert het adaptief de exploratie en exploitatie in de oplossingsruimte, waarbij gebruik wordt gemaakt van de feedback van kruisdominantie en stagnatie van deze zoekopdracht. Gebruikmakend van de empirische gegevens (bijv. energieverbruik en levenscycluskosten) van een Belgische fabrikant van plastic flessen, toont een uitgebreide sensitiviteitsanalyse het economisch belang aan van het gezamenlijk modelleren van energieverbruik en arbeid in een algoritme voor productieplanning. De superieure MOO-prestaties van AMOMA worden ook aangetoond door uitgebreide benchmarking, in termen van het aantal niet-dominerende oplossingen, convergentie en diversiteit. De wisselwerking tussen de energiekosten en de arbeidskosten wordt bovendien gekwantificeerd.

Hoofdstuk 4 vormt een aanvulling op het werk van de hoofdstukken 2 & 3 in twee dimensies. Ten eerste schaalt het geïntegreerde energie- en arbeidsbewuste productieplanning van één enkele machine naar de flexibele productieomgeving (job shop) met jobrecirculatie, wat één van de meest complexe configuraties op de werkvloer is. Ten tweede maakt het de uitbreiding van bi-objectieve optimalisatie naar multi-objectieve optimalisatie (d.w.z. het aantal objectieven

is groter dan drie). Daarvoor wordt een meerlagig energie- en arbeidsbewust productie-simulatieraamwerk voor de werkvloer voorgesteld om snel en nauwkeurig de meerdere objectieven te berekenen. Een heterogene chromosoomrepresentatie, een voorwaarts en achterwaarts decoderingsschema, evenals heterogene cross-over en mutatievormen worden voorgesteld om de recent geïntroduceerde NSGA-III aan dit nieuwe probleem aan te passen. Door middel van uitgebreide numerieke experimenten wordt de hoge schaalbaarheid van dit simulatieraamwerk met discrete-gebeurtenissen aangetoond. De effectiviteit van deze op maat gemaakte NSGA-III wordt bewezen voor multi-objectieve optimalisatie. De sterke en relatief zwakke afwegingen worden aangetoond tussen de "makespan" en de energiekosten, en tussen de energiekosten en de arbeidskosten, respectievelijk. Daarentegen wordt een harmonieuze relatie onthuld tussen de totale werkbelasting en de maximale werkbelasting.

Hoofdstukken 5 & 6 presenteren het onderzoek naar planning en zendvermogencontrole van dichte en robuuste industriële draadloze netwerken. Deze twee hoofdstukken bestuderen drie verschillende kwesties. De eerste kwestie is de wiskundige modellering van het dekkingsprobleem bij draadloze netwerken in ruwe industriële omgevingen. De tweede kwestie is het zoeken naar een optimale of bijna optimale oplossing voor dit probleem door een op maat gemaakt GA toe te passen. De derde kwestie is het experimenteel valideren van het voorgestelde model en GA met een echt systeem in de echte industriële omgeving en het uitvoeren van een numerieke demonstratie en benchmarking. Op deze manier kunnen netwerkmanagers en fabrieksmanagers betere en geautomatiseerde beslissingen nemen over robuuste IoT-netwerkplanning, implementatie, en configuratie in industriële omgevingen, met vaak harde en ruwe omstandigheden voor draadloze dekking in vergelijking met de gewone kantoor- en woonomgevingen.

Hoofdstuk 5 introduceert eerst de holistische methode voor robuuste draadloze dekking in ruwe industriële binnenomgevingen, die bestaat uit mobiele metingen, overdimensionering (hoofdstuk 5) en herconfiguratie (hoofdstuk 6). Vervolgens richt het zich op het presenteren van overdimensionering met betrekking tot het wiskundige model, het gulzige heuristische overdimensioneringsalgoritme (greedy heuristic based over-dimensioning, GHOD), en het op het GA gebaseerde overdimensioneringsalgoritme (GAOD). In dit nieuwe planningsmodel voor draadloze netwerken zijn twee volledige dekkingslagen verzekerd, terwijl de implementatiekosten worden geminimaliseerd en een minimale scheidingsafstand wordt gegarandeerd tussen twee aangrenzende draadloze knooppunten om te voorkomen dat

ze worden overschaduwd door hetzelfde obstakel (bijv. machines, robots en metalen rekken). GHOD wordt voorgesteld als een state-of-the-art heuristiek voor benchmarking. In GAOD zijn genetische operatoren, parallellisme en versnellingsmaatregelen ontworpen om een conventioneel GA aan te passen om dit model op te lossen, zelfs bij een grote probleemomvang (met betrekking tot het aantal draadloze knooppunten, de grootte van een industriële binnenomgeving, de ruimtelijke resolutie van het raster, en het aantal dekkingslagen). Voor kleine en grootschalige OD-problemen gebaseerd op industriële gegevens, is aangetoond dat GAOD 20% -25% zuiniger is dan benchmarkalgoritmen voor OD in dezelfde omgeving. De effectiviteit van GAOD wordt verder experimenteel gevalideerd met een echt implementatiesysteem.

Hoofdstuk 6 presenteert de herconfiguratiefase van de holistische methode die is geïntroduceerd in Hoofdstuk 5. Een ander methodeoverzicht vanuit het perspectief van het systeem wordt eerst gegeven. Vervolgens wordt een wiskundig model geformuleerd voor de beschrijving van het probleem van de zendvermogenssturing voor een dicht draadloos lokaal netwerk (WLAN) in een metaaloverheersende industriële binnenomgeving. Met andere woorden, Hoofdstuk 6 neemt de overdimensionerende uitvoer van Hoofdstuk 5 als zijn invoer, waarbij wordt gedacht aan het optimaal herconfigureren van de zendkracht van een overgedimensioneerd WLAN, zodanig dat de netwerkinterferentie wordt geminimaliseerd. Als oplossingsmethode wordt een GA op maat gemaakt (genaamd GATPC) met aangepaste codering, populatie-initialisatie, cross-over, mutatie, en parallellisme. De GATPC is experimenteel gevalideerd in een kleinschalig IWLAN dat wordt ingezet in een echte industriële binnenomgeving. Het wordt verder numeriek gedemonstreerd en gebenchmarkt op zowel kleine als grote schaal, met betrekking tot de effectiviteit en de schaalbaarheid. Bovendien onthult een sensitiviteitsanalyse de relatie tussen de geproduceerde interferentie en de mate van geschiktheid van GATPC in overeenstemming met het variërende doeldekkingspercentage evenals het aantal en de oriëntatie bij plaatsing van dominante obstakels.

Tot slot worden in hoofdstuk 7 algemene conclusies getrokken en worden de mogelijkheden voor toekomstige onderzoeksrichtingen kort beschreven.

English Summary

Industry 4.0 is an emerging global topic since its initial presentation at Hannover Messe (one of the world's largest industrial exhibition at Hannover, Germany) in 2011 and its final implementation recommendations published by the Industry 4.0 Working Group of the German National Academy of Science and Engineering (acatech) in 2013. In this initiative, Internet and manufacturing technologies, which are conventionally under separate development, get interweaved. This involves a set of emerging technologies, such as cyber-physical system (CPS), industrial Internet-of-Things (IIoT), big data analysis, edge computing, and cloud computing. These technologies generally serve as threefold enablers for Industry 4.0: (1) heterogeneous and distributed monitoring of physical environments and/or objects, (2) accurate and insightful analytics of massive structured and unstructured data, (3) well-informed and efficient optimization and control of physical systems. By getting updated toward Industry 4.0, manufacturing enterprises are ultimately envisioned to improve their internal operational efficiency, decrease their production cost and waste, and provide low-volume highly-personalized products to clients with enhanced product quality, shorter lead time, and affordable prices.

This dissertation investigates Industry 4.0 from the perspective of energy-aware evolutionary computation of cyber-physical systems (CPSs). The manufacturing industry is a large energy consumption sector, occupying around 1/3 of the total energy consumption in a society. The evolutionary computation (EC) or evolutionary algorithm (EA) is a subfield of artificial intelligence (AI) and computational intelligence (CI). It comprises diverse nature-inspired metaheuristics for fast and high-quality optimization. A CPS typically has a physical-to-digital-to-physical loop through environment perception, automated decision making, and dynamic control. The investigation of this thesis is performed in two parts. The first part focuses on energy- and labor-aware production scheduling under time-varying electricity prices and labor wage, in order to reduce the energy cost and the labor cost

for factories, without affecting the conventional production metrics. The second part studies how to plan dense yet economical industrial wireless networks and how to perform transmit power control of these networks for interference mitigation, so that the whole wireless network can remain robust in harsh industrial environments.

While Chapter 1 of this dissertation gives a background introduction, Chapters 2-4 cover the topic of energy-aware production scheduling under volatile electricity prices. These three chapters essentially deal with three issues. The first issue is the integration of energy awareness to conventional production scheduling models while considering the potential trade-off with other important production aspects, such as makespan and labor. The second issue is the tailoring and novel design of evolutionary algorithms, in order to solve these novel production scheduling models in a fast, high-quality, and highly scalable manner. The third issue is the what-if and statistical analysis of the economic performance of these novel models and algorithms, so that the users (e.g., factory manager) can have comprehensive understandings on when and how to make these models and algorithms economically competitive.

Chapter 2 presents the method to perform energy-aware production scheduling at the single-machine level. It first introduces the finite-state machine (FSM)-based method to model the energy consumption behavior of a machine from monitored power data on the shop floor, which may be correlated with other production data (e.g., product type and quantity, as well as machine changeover). It then formulates an energy-aware single-machine production scheduling model under volatile electricity prices. Production loads are intelligently shifted from higher-priced periods to lower-priced periods for energy cost reduction. Afterward, it proposes a simulation-optimization framework which is based on discrete-event simulation and a genetic algorithm (GA). This framework can not only solve the former formulated scheduling model in a scalable way (in terms of the number of scheduling time slots and jobs), but also perform fast rescheduling upon unforeseen events (e.g., machine failure, urgent order, and cancelled order) to make a production schedule adaptive to dynamics on the shop floor. Finally, the energy cost reduction performance of this energy-aware production scheduling method is statistically evaluated in the case study of a surface grinding machine, taking conventional production scheduling/dispatching methods as benchmarks. The average energy cost reduction ratio of this method is demonstrated to remain between 6% and 19% under real-time pricing (RTP). The trade-off between the energy cost reduction ratio

and the makespan is also statistically revealed.

Chapter 3 goes further beyond Chapter 2 by integrating labor awareness to the former energy-aware single-machine production scheduling model. A continuous-time shift accumulation heuristic is proposed to calculate the type and number of human workers based on machine power consumption states. Consequently, human workers, work shifts, and machine operations are linked, and an integrated energy- and labor-aware production simulation is enabled. Besides using a conventional GA for single objective optimization, an adaptive memetic algorithm (MA) is proposed for multiobjective optimization (MOO), which is named AMOMA. The AMOMA aims to rapidly converge toward the Pareto trade-off front without loss in diversity. It synergistically integrates in the widely-used NSGA-II the convergence- and diversity-driven tabu searches (CTS and DTS). During a search for a Pareto front approximation set, it adaptively coordinates the exploration and exploitation in the solution space, leveraging the feedback of cross-dominance and stagnation of this search. Using the empirical data (e.g., energy consumption and lifecycle cost) from a Belgian plastic bottle manufacturer, extensive sensitivity analyses highlight the economic importance of jointly modeling energy consumption and labor in a production scheduling algorithm. The superior MOO performance of AMOMA is also demonstrated through extensive benchmarking, in terms of the number of nondominated solutions, convergence, and diversity. The trade-off between the energy cost and the labor cost is additionally quantified.

Chapter 4 complements the work of Chapter 2 and Chapter 3 in two dimensions. Firstly, it scales integrated energy-and labor-aware production scheduling from the single machine to the flexible job shop with job recirculation, which is one of the most complex shop floor configurations. Secondly, it extends from bi-objective optimization to many-objective optimization (i.e., the number of objectives is larger than three). To this end, a multi-layer energy- and labor-aware shop floor production simulation framework is proposed for fast yet accurate calculation of multiple objectives; a heterogeneous chromosome representation, a forward and backward solution decoding scheme, as well as heterogeneous crossover and mutation are proposed to tailor the recently-introduced NSGA-III to this novel problem. Through extensive numerical experiments, the high scalability of this discrete-event simulation framework is demonstrated. The effectiveness of this tailored NSGA-III is proved for many-objective optimization. Strong and relatively weak contradiction is shown between the makespan and the total energy cost, and between the total energy cost and the total labor cost, respectively. In contrast,

a harmonious relation is revealed between the total workload and the maximal workload.

Chapter 5 and Chapter 6 present the investigation of planning and transmit power control of dense and robust industrial wireless networks. These two chapters basically study three issues. The first issue is mathematical modeling of a wireless coverage optimization problem in harsh industrial environments. The second issue is searching for an optimal or near-optimal solution to this problem applying a tailored GA. The third issue is to experimentally validate the proposed model and GA with a real system in the real industrial environment, and perform numerical demonstration and benchmarking. In this way, network managers and plant managers can effectively have enhanced and automated decision making on robust IoT network planning, deployment, and reconfiguration in industrial environments, which are harsh for wireless coverage compared to the common office and residential environments.

Chapter 5 first introduces the holistic method for robust wireless coverage in harsh industrial indoor environments, which comprises mobile measurement, over-dimensioning (Chapter 5), and reconfiguration (Chapter 6). It then focuses on presenting over-dimensioning regarding the mathematical model, the greedy heuristic based over-dimensioning (GHOD) algorithm, and the GA based over-dimensioning (GAOD) algorithm. In this novel wireless network planning model, two full coverage layers are ensured while the deployment cost is minimized and a minimal separation distance is guaranteed between two adjacent wireless nodes to prevent them from being shadowed by the same obstacle (e.g., machines, robots, and metal racks). The GHOD is proposed as a state-of-the-art heuristic for benchmarking. In the GAOD, genetic operators, parallelism, and speedup measures are designed to tailor a conventional GA to solve this model even in a large problem size (regarding the number of wireless nodes, the size of an industrial indoor environment, the spatial resolution or grid size, and the number of coverage layers). In small- and large-size realistic OD problems (using the industrial data), the GAOD is demonstrated to be 20%-25% more economical than benchmark algorithms for OD in the same environment. The effectiveness of GAOD is further experimentally validated with a real deployment system.

Chapter 6 further presents the reconfiguration phase of the holistic method introduced in Chapter 5. Another method overview from the perspective of the system is first given. Then a mathematical model is formulated to describe the transmit power control problem for a dense wireless local area network (WLAN) in a

metal-dominating industrial indoor environment. In another word, Chapter 6 takes the over-dimensioning output of Chapter 5 as its input, considering to optimally reconfigure the transmit power of an over-dimensioned WLAN, such that the network interference is minimized. As the solution method, a GA is tailored (named GATPC) in terms of solution encoding, population initialization, crossover, mutation, parallelism, and speedup measures. The GATPC is experimentally validate in a small-scale IWLAN that is deployed in a real industrial indoor environment. It is further numerically demonstrated and benchmarked on both small- and large scales, regarding the effectiveness and the scalability. Moreover, a sensitivity analysis reveals the relation between the produced interference and the qualification rate of GATPC according to the varying target coverage percentage as well as the number and the placement direction of dominant obstacles.

Finally in Chapter 7, overall conclusions are drawn and opportunities for future research directions are briefly described.

1 Introduction

1.1 Industry 4.0

The term "Industry 4.0" ("Industrie 4.0" in Dutch and German) was first used at the Hanover Fair (Hanover Messe in Germany) in 2011, which is one of the largest industrial exhibition in the world. A comprehensive collection of consolidated recommendations for implementing this strategic initiative was then published in 2013 by the Industrie 4.0 working group of the German National Academy of Science and Engineering (acatech) [1]. Briefly, Industry 4.0 refers to the fourth industrial revolution (Figure 1.1) that is fundamentally driven by the introduction of Internet technologies into the manufacturing industry. More specifically, Industry 4.0 encompasses a set of relevant emerging technologies, including cyber-physical system (CPS), industrial Internet of Things (IIoT), cloud computing, edge computing, big data, etc. Despite some overeager marketing messages, it is still a future vision with the current three evolutions [2]:

(1) Communication infrastructure in production systems is more affordable and widely introduced. This is useful for various purposes such as engineering, configuration, operations, service, and diagnostics.

(2) Field devices, machines, factories, and even products are connected to a

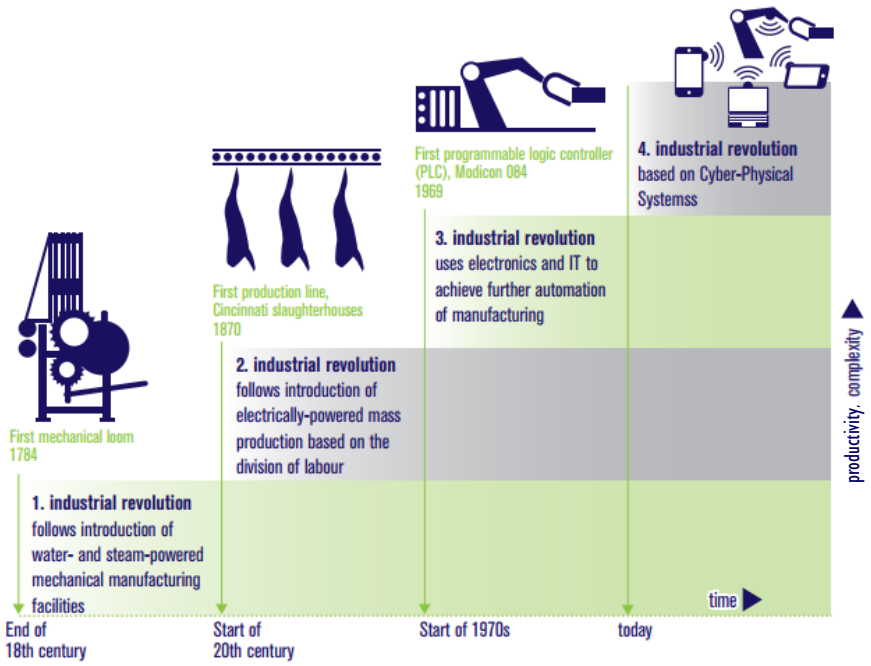


Figure 1.1: An overview of the four industrial revolutions according to the Industrie 4.0 working group of the German National Academy of Science and Engineering (acatech) [1]

network and available as data objects in the network. Therefore, they are searchable, explorable, and analyzable in the network.

(3) Field devices, machines, factories, and even products are able to store documents and knowledge about themselves as a virtual living representation in the network, with individual identifiers. This stored information is updatable. Furthermore, diverse functionality acts for the physical objects, e.g., negotiation and exploration, creating various applications on top of these virtual representations and data.

By gradually implementing Industry 4.0, the manufacturing industry is expected to evolve toward a distributed organization of production, with connected goods (products with communication capability), low-energy processes, collaborative robots, as well as integrated manufacturing and logistics [3]. One driving application scenario is to form a network of geographically distributed factories with flexible adaptation of production capabilities and sharing of resources and assets to improve order fulfillment [3]. Therefore, manufacturers are envisioned to enjoy many advantages such as innovative applications and services, new technologies and advanced features, increased operational benefits, and reduced instal-

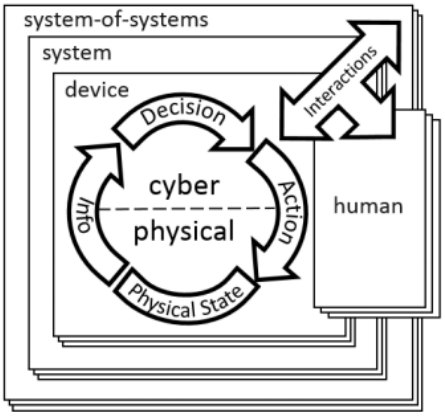


Figure 1.2: Generic architecture of a cyber-physical system (CPS) according to the definition of the American National Institute of Standards and Technology (NIST) [5]

lation costs [4]. They are thus able to provide highly-personalized products and services to customers with high quality and affordable prices.

Although the term "Industry 4.0" originates from Germany, analogous country-wide initiatives or platforms subsequently emerge around the world in recent years. This is illustrated by the "Industrial Internet" in US, the "Made in China 2025" in China, the "Robot Revolution Initiative" in Japan, and the "Future Industry" ("Industrie du Futur" in French) in France. Consequently, Industry 4.0 and its alternative versions in different countries are foreseen to reshape the functioning of our society socially, environmentally, and economically.

1.2 Cyber-Physical Systems in Industry 4.0

In many real-world systems, computational and physical resources are interconnected: embedded computers and communication networks receive outside-world data from sensors and govern actuators that operate in the physical reality, creating a smart control loop capable of adaptation, autonomy, and improved efficiency [6]. Such systems are widely defined as CPSs [7]. According to the definition of the US National Institute of Standards and Technology (NIST) [5], a CPS integrates computation, communication, sensing, and actuation with physical systems to fulfill time-sensitive functions with varying degrees of interaction with the environment, including human interaction.

The conceptual model [5] of a CPS is presented in Figure 1.2. A CPS may be an individual device, a system compromising multiple cyber-physical devices, or a system of systems consisting of multiple systems that compromise multiple

devices. A CPS must contain the decision flow with at least one of the flows for information or action. The information flow represents the digital measurement of the physical state of the physical world, while the action flow influences the physical state of the physical world.

Common examples of CPSs are industrial control systems, computerized vehicles or self-driving cars, wireless sensor networks, smart grids, and almost all devices encompassed by the Internet of Things (IoT). Unlike conventional embedded systems, which are standalone, CPSs focus on networking of multiple devices [8]. Compared to information and communications systems, CPSs have a unique challenge in combining the discrete information and communications technology (ICT) framework (which is adherent to rigid specifications) and a continuous physical system (which is often neither easily modeled nor completely understood) [6].

This dissertation investigates two CPSs in the context of Industry 4.0. The first CPS is a production system that receives electricity from the smart grids with dynamic prices, collects power consumption data from manufacturing machines, frames its production load into labor shifts, and adapts its production load to the time-varying electricity prices and labor wage. In this way, the energy cost and the labor cost for production are minimized. Such an electricity price adaptation mechanism on the shop floor is widely known as industrial demand response (DR). The second studied CPS is a wireless communication system in an industrial indoor environment (shop floor or warehouse). It is first deployed based on the perceived wireless signal propagation pattern such that the deployment cost is minimized and network redundancy is created. It then senses its real-time quality of service (QoS) and adapts the transmit power to the monitored obstacle shadowing effects in this environment. Compared to the cyber-physical production system, the energy cost is minor and a less sensitive factor for such a cyber-physical wireless communication system. Therefore, the objective for the transmit power control of the latter CPS is to minimize the interference in the network, which is a crucial performance metric for the latter CPS.

1.2.1 Production System under Demand Response

Generally, there are two categories of DR schemes [9]: (1) the price- or time-based DR, and (2) the incentive-based DR. In the price- or time-based DR, consumers cannot affect the electricity price by their energy consumption. Instead, an energy provider attempts to adjust the load by dynamic pricing, e.g., real-time pricing, critical peak pricing, time-of-use pricing, such that consumers are encouraged to adapt to the time-varying electricity price curve. There are two types of consumers for price-based DR scheme [10]: (1) the price-taking consumers, and (2) the price-anticipating consumers. The former ones are motivated to adapt their energy con-

sumption to the electricity price, whereas the latter can change the electricity price by their energy consumption. The latter refers to large energy consumers, such as industrial facilities, commercial buildings, and data centers.

While the residential sector receives higher attention in the DR research due to the high flexibility and schedulability of residential activities (e.g., dishwashers and washing machines), the DR of larger industrial consumers has less research outcome due to the triggered influence on industrial performance, although the impact of the latter on the electricity grid is sizable [11]. As highlighted in [11] and [12], there is little research on implementing price-based DR in industrial facilities. Even for the DR research on industrial facilities, most of these studies focus on HVAC (heating, ventilation, and air conditioning) controls rather than production systems, due to the difficulty of defining schedulable loads which industrial consumers would be willing to shift considering their production environments and constraints.

More specifically, the challenges of a production system under DR are fivefold. Firstly, the energy consumption behavior of a machine is conventionally unknown due to a lack of energy monitoring on the shop floor. Secondly, production constraints are often various, including machine interdependency, machine changeover/setup for processing different parts/products, due date, labor shifts, a potential split of a job or a changeover, operation sequence of a job, etc. Thirdly, diverse unforeseen events may occur from time to time on the shop floor, causing disruptions to a running production system, e.g., machine failure, rush order, and canceled order. Fourthly, production loads are often linked with human workers, which makes the adaptation to dynamic electricity prices more economically sensitive. Fifthly, the resolution of production planning and scheduling problems often encounters poor scalability and intractability, due to the prevalent use of rigorous mathematical formulation and problem resolution by a commercial off-the-shelf solver (e.g., IBM CPLEX and FICO Xpress).

1.2.2 Wireless Communication System in Harsh Industrial Indoor Environments

A wireless communication system usually comprises a transmitter system (Tx) at one location and a receiver system (Rx) at another location. The signal is sent out by the Tx, propagates through a medium (e.g., air), and is received by the Rx. The received signal strength at the Rx is weaker than the output signal strength at the Tx, due to the attenuation in this medium. A location of a Rx is considered as covered when there is a sufficient probability of correct signal reception during a sufficient percentage of the time for a given reception scenario. Therefore, for a certain wireless communication system and propagation environment, coverage can be defined according to four aspects [13]: (1) location probability (e.g., 95% of

the locations at a given distance from the Tx required to have wireless coverage), (2) time probability (e.g., a coverage requirement at a given location during 95% of the time), (3) throughput requirement (e.g., 5 Mbps), and (4) scenario (e.g., coverage required on the shop floor). The coverage range of such a system can be defined as the distance from the Tx to the most distant location where the coverage is provided. Ranges for a wireless communication system will increase or decrease when any one of the four coverage aspects changes.

The most important advantage of using wireless connections in manufacturing industry is that machines and devices are movable and more easily connectable without any cabling constraints. Although the wired networks with higher reliability has conventionally outweighed the flexible wireless networks, the emerging development of wireless technologies and the recent adoption of the IoT and CPS concepts in manufacturing industry are changing the situation. In this new trend, from production processes up to a supervisory control and data acquisition (SCADA) systems, industrial communication solutions are heterogeneous but optimized to fulfill different requirements for diverse applications [3]. They include fieldbuses, industrial Ethernet, and industrial wireless networks. The industrial wireless networks further encompass diverse wireless communication standards, e.g., 5G, IIoT, and WLAN (wireless local area network).

The challenges of deploying and operating a wireless network in a harsh industrial indoor environment are threefold regarding coverage. Firstly, the manufacturing industry is a more conservative sector for wireless applications, compared to an office or residential environment which has already been deployed with heterogeneous wireless networks. Consequently, few existing wireless network planning tools specifically take an industrial indoor environment into account. Secondly, various obstacles are prevalent on a shop floor or in a warehouse, e.g., machines, robots, vehicles, and racks, many of which even move over time. Such an industrial indoor environment is thereby harsh for remaining good or desired wireless coverage, in contrast to an office or residential environment which is relatively static and has less obstacles. Thirdly, as there are more obstacle shadowing effects in an industrial indoor environment, an intuitive approach to tackle this problem is to manually set up a redundant wireless coverage layer for robustness. However, this would significantly increase the deployment cost due to a lack of an optimization process.

1.3 Evolutionary Algorithm

A common fundamental property shared by the former two cyber-physical systems (CPSs) (Section 1.2.1 and Section 1.2.2) is the need for automated and enhanced decision makings. As these decision makings occur in a rather dynamic physical

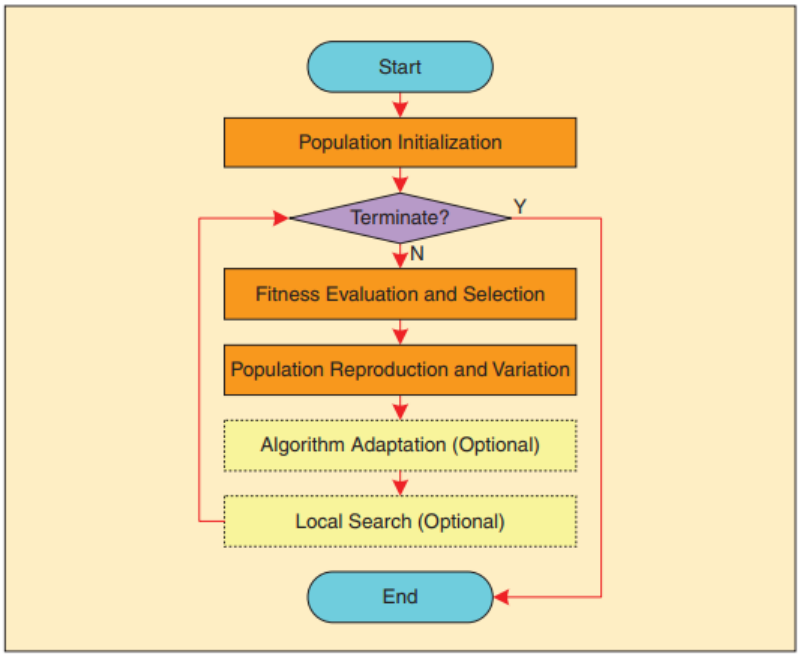


Figure 1.3: Generic framework of evolutionary algorithms (EAs) or evolutionary computing (EC) [14]

environment, it is crucial to rapidly obtain a high-quality solution of a problem in hand. As an important branch of artificial intelligence (AI) [15] and computational intelligence (CI) [16], evolutionary algorithms (EAs) intrinsically satisfy this decision making requirement. EAs are generic population-based optimization algorithms [17]. As presented in Figure 1.3, an EA uses mechanisms inspired by biological evolution, e.g., natural selection, reproduction, and variation. Candidate solutions are represented as individuals in a population. After the initialization, a population makes the evolution by repetitively applying these genetic operators. During each evolution, an EA may additionally use an algorithm adaptation procedure and a local search to enhance its optimization performance [14]. The solution or solution set improves its quality with the population evolution, such that it becomes optimal or near-optimal at the end of an EA instance [18, 19]. In literature, EAs are frequently treated the same as evolutionary computation (EC) [14].

An EA is a stochastic process to search for an optimal or near-optimal solution of an optimization problem. Firstly, a population is stochastically initialized to guarantee well-spread sampling of the solution space, though some initial individuals may be dedicatedly produced to guide the evolutionary search toward the promising area. Secondly, individuals are often selected for breeding the offspring according to their selection probability, which is positively correlated with the fit-

ness value of an individual. Thirdly, the recombination and mutation operators randomly choose one or multiple crossover loci and mutation genes, respectively, for breeding the offspring. Due to this stochastic characteristic, the population diversity is preserved, such that an evolutionary search can naturally escape local optima. Therefore, an EA is widely known as a global search algorithm [20, 21].

1.3.1 Single- and Multi-Objective Optimization

Among the rich set of EA variants, one of the most popular EA types is the genetic algorithm (GA) [22]. In a GA, the fitness function is the objective function that this GA aims to minimize or maximize based on the model of the optimization problem. The fitness value of a solution indicates its quality to solve this model. For this reason, a GA is usually used to solve single-objective optimization problems (SOPs) or multi-objective optimization problems (MOPs) based on scalarization of multiple objectives using the weighted-sum method [23]. Furthermore, a GA employs elitism to preserve the best solution(s) in a parent generation to the offspring, such that the convergence curve (i.e., the best fitness value of a generation vs. the number of generations) of a GA search cannot get deteriorated with the population evolution.

Another popular EA type is the multi-objective evolutionary algorithm (MOEA). While a GA essentially aims for SOPs and is widely known to be problematic in weight assignment when converting a MOP to a SOP using the weighted-sum method, a MOEA is intrinsically designed for MOPs, taking advantage of multiple solutions in a single run due to the population in an EA. A highly representative MOEA is the nondominated sorting genetic algorithm-II (NSGA-II) [24]. In the NSGA-II, a fast nondominated sorting procedure classifies a population into a set of ranked solutions based on Pareto dominance, where the rank indicates the nondomination relation of these solutions. A parameterless crowded-comparison operator differentiates solutions first by the nondomination rank and then by a crowding distance (indicating the spread) if the nondomination rank of two solutions is equal. The selection process combines the parent and offspring populations and selects the best N solutions using this crowded-comparison operator. In this way, elitism is preserved with the population evolution and the uniform spread of a nondominated solution set is also guaranteed.

1.3.2 Many-Objective Optimization

However, additional common difficulties are increasingly known when applying a Pareto dominance-based EA (e.g., NSGA-II) for many-objective optimization problems (MaOPs), of which the number of objectives exceeds three. These difficulties are roughly classified into five categories [25]: (1) difficulties in the search

for Pareto optimal solutions, (2) difficulties in the approximation of the entire Pareto front, (3) difficulties in the presentation of obtained solutions, (4) difficulties in the choice of a single final solution, and (5) difficulties in the evaluation of search algorithms.

The nondominated sorting genetic algorithm-III (NSGA-III) [26] is one representative EA that has been recently proposed for MaOPs aiming to solve part of these difficulties. It follows the NSGA-II framework that emphasizes nondominated solutions in a population. However, unlike in NSGA-II, it also emphasizes population members that are in some sense associated with each of the well-spread reference points, which are supplied upon the start of a NSGA-III instance and adaptively updated with the population evolution. To this end, the crowding distance operator in NSGA-II is replaced by the following sequential procedure: (1) determination of reference points on a hyper-plane, (2) adaptive normalization of population members, (3) association operation, and (4) niche-preservation operation. Despite this newly-introduced procedure for elitist selection and diversity maintenance, the NSGA-III does not require setting any new parameter other than the usual GA parameters, e.g., population size, crossover and mutation probabilities, and termination criterion. Although the number and the location of reference points remain to be determined for a NSGA-III instance, they are not algorithmic parameters for the following two reasons. Firstly, the number of reference points is directly related to the desired number of trade-off points and is usually roughly set as the population size. Secondly, the location of reference points depends on the preference information that the user or the decision maker is interested in to achieve in the obtained solution set.

1.3.3 Memetic Computation

Although basic EAs or MOEAs are well-known as effective and efficient global search algorithms, they are not good at local search. This weakness leads to a number of shortcomings, such as slow convergence to the Pareto front, no efficient termination criterion, and a lack of a theoretical convergence proof [27]. While a global search explores in the solution space and attempts to localize the potential regions, a local search exploits these potential regions by incorporating domain-specific knowledge on how a solution can be locally improved. Such a hybrid of global and local searches in an EA or MOEA is known as a memetic algorithm (MA) [28, 29], where each local search algorithm is named a meme serving a "brick" of knowledge on a problem.

Nowadays, although the memetic computation (MC) concept goes far beyond the initial definition of MAs [29], active research is still observed on leveraging this concept for unconstrained and constrained optimization [30–32]. A crucial problem of designing and implementing a MA is to determine how the introduced

memes interact during an optimization process. The coordination of these memes can be roughly classified into four categories [33]: (1) adaptive hyper-heuristic, where memes are coordinated by heuristic rules, (2) meta-Lamarckian learning, where the success of a meme biases its activation probability with the population evolution, (3) self-adaptive and co-evolution, where memes, either directly encoded in the solutions or evolving in parallel to these solutions, take part in the evolution, (4) fitness diversity-adaptive, where an indicator of the population diversity is used to select and activate the most appropriate memes.

1.3.4 Challenges

Despite these developments of EAs, threefold challenges are identified regarding decision makings for CPSs in Industry 4.0. Firstly, most research on EAs is either focused on the theoretical algorithm design without any deep domain knowledge, or applying an existing EA to solve a problem in a specific domain without any algorithmic innovation. There remains huge research potential in tailoring an EA for an important problem or a class of problems, e.g., how to choose the most appropriate chromosome representation, crossover, mutation, and termination criterion for such a problem or a class of problems, how to make the best use of domain knowledge to enhance the efficiency of an evolutionary search, how to enhance the scalability of an EA when facing a large or hyper-large problem size, how to speedup an EA, how to enlarge the application or problem domain of an EA, and so on. Secondly, although many EAs are proposed in literature as a generic optimization framework, new advanced evolutionary search paradigms are still desired to solve a problem or a set of problems in a faster and higher-quality manner. Thirdly, when applying an EA for a real-world problem in CPSs in Industry 4.0, the optimization performance bottleneck of an EA easily lies in the fitness calculation or solution evaluation, since such a problem is usually highly constrained and large-scale, and has a number of objectives and interdependent variables. Therefore, it remains a challenge in how to build an efficient model for such a real problem, how to efficiently calculate this model, and how to couple this model to an EA to ensure the overall efficiency of an evolutionary search.

1.4 Outline

This dissertation has two parts around energy-aware evolutionary computation for CPSs in Industry 4.0, aiming to tackle these technical challenges described in Section 1.2 and Section 1.3. The fundamental purpose of both parts is to automate and enhance the CPS-level decision makings in factories. The first part (Chapters 2-4) is on energy-aware production scheduling. This is a promising approach to implement demand response in factories while considering the labor shift planning of

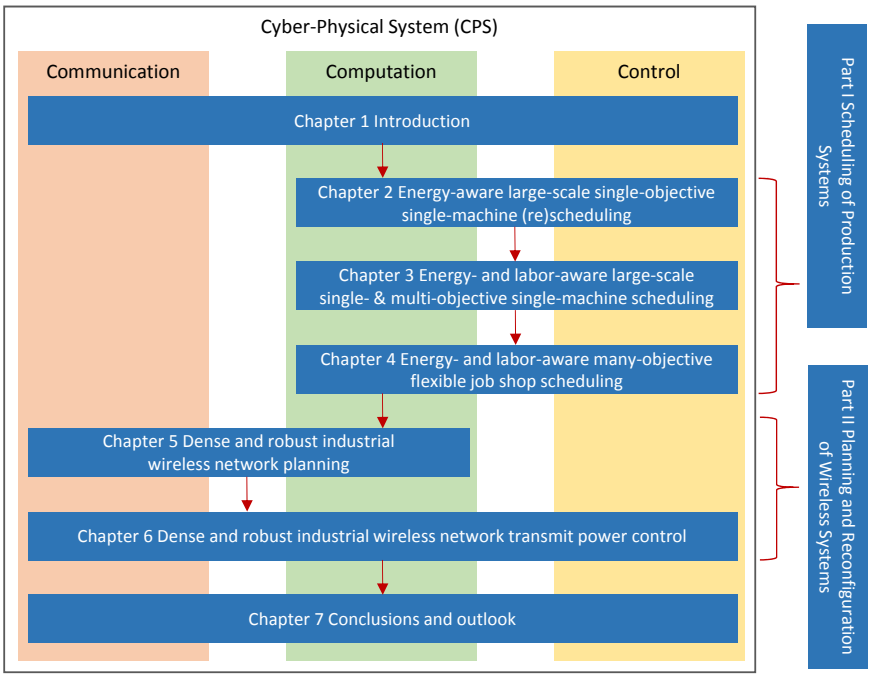


Figure 1.4: Structure of this thesis around the three pillars of a cyber-physical system (CPS): the communication, the computation, and the control.

human workers. The second part (Chapters 5-6) is on planning and transmit power control of dense and robust industrial wireless networks. This leads to a decision support system for factories to deploy and reconfigure IoT networks.

The main chapters in this thesis are presented in Figure 1.4 and briefly introduced below. The research of each chapter contributed to at least one peer-reviewed publication in an international journal. The first part on energy-aware production scheduling additionally led to five presented international conference papers. These papers can be found in the publication list in Section 1.5.

• Chapter 2 proposes an energy-aware single-machine production scheduling algorithm under dynamic electricity prices, which is scalable to a large problem size (regarding the number of time slots and jobs) and helps factories to reduce the energy cost for production. It further statistically analyzes the economic performance of this algorithm from the perspective of a factory.

• Chapter 3 quantitatively analyzes the economic importance of jointly considering energy and labor in a production scheduling algorithm under dynamic electricity prices. It additionally proposes an adaptive multi-objective memetic algorithm (AMOMA) to fast produce a set of nondominated solutions without a loss in the diversity of solutions, considering the revealed trade-off between the energy

cost and the labor cost.

• Chapter 4 scales energy-aware production scheduling under dynamic electricity prices from a single machine to a flexible job shop considering the possible job recirculation. It also extends from single- and bi-objective optimization to many-objective optimization. It then quantifies the relations among multiple production metrics, i.e., makespan, total energy cost, total labor cost, total workload, and maximal workload.

• Chapter 5 first proposes an over-dimensioning model and an efficient GA, which help factories to deploy robust wireless networks in harsh industrial environments in an economical way. It then validates this model and GA with a system deployed in a real industrial environment, as well as numerically demonstrate and benchmark their superior economic performance.

• Chapter 6 proposes a transmit power control model and an efficient GA, which help network managers in the factories to optimally reconfigure dense wireless networks for network interference mitigation while remaining network robustness in harsh industrial environments. Analogously, it validates this model and GA with a system deployed in a real industrial environment, and numerically demonstrate and benchmark their superior interference mitigation performance.

1.5 Publications

The proposed methods and obtained results during this PhD research have been published in or submitted to scientific journals[1], and presented at international conferences[2] on Industry 4.0. The research work on energy-aware production scheduling was of interest for Singapore Institute of Manufacturing Technology within ASTAR (Singapore Agency for Science, Technology, and Research) as well as universities and companies (e.g., Palo Alto Research Center in California, System Insights in India and US, Technical University of Braunschweig in Germany, and Comillas Pontifical University of Madrid in Spain), and received a FWO (Research Foundation - Flanders, Belgium) academic stay grant in 2017. The following list provides an overview of these publications.

[1]The listed journal publications are recognized as "A1 publications", according to the definition used by Ghent University: A1 publications are articles listed in the Science Citation Index Expanded, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

[2]The listed conference publications are recognized as "P1 publications", according to the definition used by Ghent University: P1 journals are articles listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

1.5.1 Publications in International Journals

[1] **X. Gong**, T. De Pessemier, W. Joseph, and L. Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*, Journal of Cleaner Production. vol. 113, pp. 508-522, 2016 (IF: 5.715).

[2] **X. Gong**, J. Trogh, Q. Braet, E. Tanghe, P. Singh, D. Plets, J. Hoebeke, D. Keschrijver, T. Dhaene, L. Martens, and W. Joseph. *Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories*, IET Science, Measurement & Technology, vol. 10, no. 4, pp. 375-382, 2016 (IF: 1.263).

[3] **X. Gong**, M. Van der Wee, T. De Pessemier, S. Verbrugge, D. Colle, L. Martens, and W. Joseph. *Integrating labor awareness to energy-efficient production scheduling under real-time electricity pricing: An empirical study*, Journal of Cleaner Production, vol. 168, pp. 239-253, 2017 (IF: 5.715).

[4] **X. Gong**, D. Plets, E. Tanghe, T. De Pessemier, L. Martens, and W. Joseph. *An efficient genetic algorithm for large-scale planning of dense and robust industrial wireless networks*, Expert Systems with Applications, vol. 96, pp. 311-329, 2018 (IF: 3.928).

[5] **X. Gong**, D. Plets, E. Tanghe, T. De Pessemier, L. Martens, and W. Joseph, *An efficient genetic algorithm for large-scale transmit power control of dense and robust wireless networks in harsh industrial environments*. Applied Soft Computing, 65(C), pp. 243-259, 2018 (IF: 3.541).

[6] **X. Gong**, Y. Liu, N. Lohse, T. De Pessemier, L. Martens, and W. Joseph. *Integrated energy-cost-efficient and labor-aware production scheduling: a memetic algorithm for multi-objective optimization*. IEEE Transactions on Industrial Informatics, under revision, 2017 (IF: 6.764).

[7] **X. Gong**, T. De Pessemier, L. Martens and W. Joseph. *Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation*. Journal of Cleaner Production, submitted, 2018 (IF: 5.715).

1.5.2 Publications in International Conferences

[1] **X. Gong**, T. De Pessemier, W. Joseph, and L. Martens, "An energy-cost-aware scheduling methodology for sustainable manufacturing," *Procedia CIRP (In-*

ternal Academy for Production Engineering), 22nd CIPR Conference on Life Cycle Engineering, vol. 29, pp. 185-190, Sydney, Australia, April 2015.

[2] **X. Gong**, T. De Pessemier, W. Joseph, and L. Martens, "A stochasticity handling heuristic in energy-cost-aware scheduling for sustainable production," *Procedia CIRP, 23rd CIRP Conference on Life Cycle Engineering (LCE)*, vo. 48, pp. 108-113, Berlin, Germany, May 2016.

[3] **X. Gong**, T. De Pessemier, W. Joseph, and L. Martens, "A power data driven energy-cost-aware production scheduling method for sustainable manufacturing at the unit process level," *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1-8, Berlin, Germany, September 2016.

[4] **X. Gong**, M. Van der Wee, T. De Pessemier, S. Verbrugge, D. Colle, L. Martens, and W. Joseph, "Energy- and labor-aware production scheduling for sustainable manufacturing: a case study on plastic bottle manufacturing," *Procedia CIRP, 24th CIRP Conference on Life Cycle Engineering (LCE)*, vol. 61, pp. 387-392, Kamakura, Japan, March 2017.

[5] **X. Gong**, T. De Pessemier, L. Martens, and W. Joseph, "Energy-efficient and labor-aware production scheduling based on multi-objective optimization," *Computer Aided Chemical Engineering, 27th European Symposium on Computer Aided Process Engineering (ESCAPE)*, vol. 40, pp. 1369-1374, Barcelona, Spain, October 2017.

1.5.3 Awards

[1] FWO (Research Foundation - Flanders, Belgium) grant (V416217N) for a long academic stay outside the Europe, 2017.

[2] CWO (Scientific Research Committee, Ghent University) Mobility Fund, 2017.

[3] CWO Mobility Fund, 2016.

References

[1] Henning Kagermann, Wolfgang Wahlster, and Johannes Helbig. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0 - Final report of the Industrie 4.0 Working Group*. Technical report, acatech, the German National Academy of Science and Engineering, 2013.

[2] R. Drath and A. Horch. *Industrie 4.0: Hit or Hype? [Industry Forum]*. IEEE Industrial Electronics Magazine, 8(2):56–58, June 2014.

[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite. *The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0*. IEEE Industrial Electronics Magazine, 11(1):17–27, March 2017.

[4] P. Haller and B. Genge. *Using Sensitivity Analysis and Cross-Association for the Design of Intrusion Detection Systems in Industrial Cyber-Physical Systems*. IEEE Access, 5:9336–9347, 2017.

[5] Edward Griffor, David Wollman, and Christopher Greer. *Framework for Cyber-Physical Systems*. Technical report, US National Institute of Standards and Technology, May 2016.

[6] S. Zanero. *Cyber-Physical Systems*. Computer, 50(4):14–16, April 2017.

[7] V. Jirkovský, M. Obitko, and V. Mařík. *Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration*. IEEE Transactions on Industrial Informatics, 13(2):660–667, April 2017.

[8] N. Jazdi. *Cyber physical systems in the context of Industry 4.0*. In 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, pages 1–4, May 2014.

[9] R. Deng, Z. Yang, M. Y. Chow, and J. Chen. *A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches*. IEEE Transactions on Industrial Informatics, 11(3):570–582, June 2015.

[10] K. Ma, G. Hu, and C. J. Spanos. *A Cooperative Demand Response Scheme Using Punishment Mechanism and Application to Industrial Refrigerated Warehouses*. IEEE Transactions on Industrial Informatics, 11(6):1520–1531, Dec 2015.

[11] Ahmed Abdulaal, Ramin Moghaddass, and Shihab Asfour. *Two-stage discrete-continuous multi-objective load optimization: An industrial consumer utility approach to demand response*. Applied Energy, 206(Supplement C):206 – 221, 2017.

[12] X. Huang, S. H. Hong, and Y. Li. *Hour-Ahead Price Based Energy Management Scheme for Industrial Facilities*. IEEE Transactions on Industrial Informatics, 13(6):2886–2898, Dec 2017.

[13] David Plets. *Characterization and optimization of the coverage of digital wireless broadcast and WLAN networks*. PhD thesis, Ghent University, 2011.

[14] J. Zhang, Z. h. Zhan, Y. Lin, N. Chen, Y. j. Gong, J. h. Zhong, H. S. H. Chung, Y. Li, and Y. h. Shi. *Evolutionary Computation Meets Machine Learning: A Survey*. IEEE Computational Intelligence Magazine, 6(4):68–75, Nov 2011.

[15] Michael Negnevitsky. *Artificial intelligence: a guide to intelligent systems*. Pearson Education, 2011.

[16] Andries P. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley, 2 edition, 2007.

[17] S.P. Leo Kumar. *State of The Art-Intense Review on Artificial Intelligence Systems Application in Process Planning and Manufacturing*. Engineering Applications of Artificial Intelligence, 65:294 – 329, 2017.

[18] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[19] Ying Liu, Haibo Dong, Niels Lohse, and Sanja Petrovic. *A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance*. International Journal of Production Economics, 179(Supplement C):259 – 272, 2016.

[20] P. Yang, K. Tang, and X. Yao. *Turning High-dimensional Optimization into Computationally Expensive Optimization*. IEEE Transactions on Evolutionary Computation, PP(99):1–1, 2017.

[21] Daniel Molina Cabrera. *Evolutionary algorithms for large-scale global optimisation: a snapshot, trends and challenges*. Progress in Artificial Intelligence, 5(2):85–89, May 2016.

[22] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998.

[23] X. D. Xue, K. W. E. Cheng, T. W. Ng, and N. C. Cheung. *Multi-Objective Optimization Design of In-Wheel Switched Reluctance Motors in Electric Vehicles*. IEEE Transactions on Industrial Electronics, 57(9):2980–2987, Sept 2010.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.

[25] H. Ishibuchi, N. Akedo, and Y. Nojima. *Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems*. IEEE Transactions on Evolutionary Computation, 19(2):264–283, April 2015.

[26] K. Deb and H. Jain. *An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints*. IEEE Transactions on Evolutionary Computation, 18(4):577–601, Aug 2014.

[27] K. Sindhya, K. Miettinen, and K. Deb. *A Hybrid Framework for Evolutionary Multi-Objective Optimization*. IEEE Transactions on Evolutionary Computation, 17(4):495–511, Aug 2013.

[28] Chi-Keong Goh, Yew Soon Ong, and Kay Chen Tan, editors. *Multi-Objective Memetic Algorithms*. Springer-Verlag Berlin Heidelberg, 2009.

[29] X. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan. *A Multi-Facet Survey on Memetic Computation*. IEEE Transactions on Evolutionary Computation, 15(5):591–607, Oct 2011.

[30] V. A. Shim, K. C. Tan, and H. Tang. *Adaptive Memetic Computing for Evolutionary Multiobjective Optimization*. IEEE Transactions on Cybernetics, 45(4):610–621, April 2015.

[31] A. Maesani, G. Iacca, and D. Floreano. *Memetic Viability Evolution for Constrained Optimization*. IEEE Transactions on Evolutionary Computation, 20(1):125–144, Feb 2016.

[32] A. Gupta, Y. S. Ong, L. Feng, and K. C. Tan. *Multiobjective Multifactorial Optimization in Evolutionary Multitasking*. IEEE Transactions on Cybernetics, 47(7):1652–1665, July 2017.

[33] Ferrante Neri and Carlos Cotta. *Memetic algorithms and memetic computing optimization: A literature review*. Swarm and Evolutionary Computation, 2(Supplement C):1 – 14, 2012.

Part I

Scheduling of Production Systems

2

Energy-Aware Single-Machine Production Scheduling

Currently, it is a trend for electricity prices to become volatile. This is due to the distributed generation and storage of renewable energy, deployment of smart grids, and emerging energy management concepts such as "Internet of energy" [1] and "Energy Union" [2]. On the other hand, the manufacturing industry occupies a significant part of energy consumption in a society. It thus deserves a deep investigation on how to reduce the energy cost of manufacturing enterprises without affecting the production objectives and/or breaking the production constraints.

To this end, this chapter proposes a genetic method to minimize the energy cost[1] and improve the energy efficiency of manufacturing unit processes. Finite state machines have been used to build the transitional state-based energy model of a single machine. A mixed integer programming (MIP) model has been formulated for energy-cost-aware job order scheduling on a single machine. A generic

[1] While the energy cost generally includes the cost for electricity, water, compressed air, natural gas, etc., it specifically denotes the electricity cost in Chapters 2-4 of this thesis. Extension to other types of energy cost is possible by the method proposed in Chapters 2-4, i.e., state-based energy model, discrete-event simulation, and scheduling/optimization.

algorithm has been implemented to search for an energy-cost-effective schedule at volatile electricity prices with the constraint of due dates. As a result, plant managers can have an energy-cost-effective production schedule which is associated with machine energy states along time, and also time-indexed energy simulation of the schedule. Compared to most of the static scheduling approaches, unforeseen events have been further handled through reactive rescheduling during the execution of a schedule on a machine. This facilitates to investigate how unforeseen events on a shop floor affect the performance of energy-cost-aware scheduling. Validation of this method has been performed using empirical data of the case study, including the power measured from a grinding machine, and the real-time pricing and time-of-use pricing tariffs. The proposed method has been demonstrated to be both energy-efficient and energy-cost-efficient even at the presence of unforeseen events, where energy efficiency is promoted by optimal decision making on machine idling and off states for energy conservation, and energy cost efficiency is contributed by optimized job sequencing and timing under volatile electricity prices for energy cost reduction. As a joint focus on energy efficiency and demand response, this method shows its effectiveness to reduce greenhouse gas emissions during peak periods, and to promote energy-efficient, demand-responsive, and cost-effective manufacturing processes.

2.1 Introduction

Traditionally, utilities called upon peak power generation to meet rising demand from energy consumers in a real-time manner. Those peak power generators were usually thermal power plants producing high emissions of greenhouse gas (GHG). As a consequence, the stability of the power grid was threatened and the environment was seriously polluted. The demand side management (DSM) [3] is a set of interconnected and flexible programs including energy efficiency and demand response. It enables energy users of all types to highly take their own initiatives in maintaining the stability of the power grid. Environmental sustainability and economic saving are thus achieved. As to industrial energy users, energy efficiency is focused on approaches to reduce the energy consumption without declining the production outputs, while demand response encourages a temporary change in their electricity consumption in response to market or supply conditions [4]. In summary, energy efficiency can be seen as load reduction, and demand response can be viewed as load shift [5]. Both energy efficiency and demand response are among the major measures to implement smart grids [6].

In order to match the power supply and demand during time, various energy charging policies are made in different countries, e.g., time-of-use pricing (ToUP), real-time pricing (RTP), and critical peak pricing (CPP). In ToUP tariff, two types

of periods are generally defined: "on-peak" and "off-peak". The kWh energy charge during on-peak periods is evidently higher than that during off-peak periods, such as more than twice [7]. RTP can be commonly found in countries whose energy market is highly developed. For instance, on Belpex, the electricity spot market in Belgium, users can buy a certain amount of electricity in two different submarkets, namely the day-ahead market (DAM) and the continuous intraday market. The DAM enables users to purchase electricity, of which the prices vary every hour and are known 24 hours in advance, and which will be delivered the day after. The continuous intraday market provides the industry with hourly-dynamic or multi-hourly-dynamic electricity prices up to five minutes before delivery. With an increasing amount of energy provision by volatile energy sources such as wind turbines, the RTP complies with the principle of demand and response. Time periods with surpluses of energy and low grid demands, result in low electricity prices, while periods with only little energy from renewable energy sources and high grid demands, lead to high electricity prices [8]. Besides the non-event days during which the default ToUP is applied, CPP has mid-peak and critical-peak periods on critical event days. During the two types of peak periods, the electricity price is set much higher, in order to reflect the marginal cost of electricity generation. For example, in the Korean CPP pilot, the critical peak price and the mid-peak price are about 4.8 times and 3 times higher than the peak price and the off-peak price on non-event days, respectively [9].

The industry plays a key role in the society's overall energy consumption and GHG emissions. It thus exhibits a high potential for reducing both energy and GHG. In Taiwan, the industry occupies approximately 53.8% and 48.3% of its whole energy consumption and GHG emissions, respectively, by taking 2010 as base year. The total energy reduction potential in its industry is assessed as 5.3% of the national energy use per year. The maximal GHG emissions reduction of Taiwan's six most energy-intensive industrial sectors is estimated as 6.4% of the national GHG emissions [10]. Therefore, it remains meaningful to investigate the energy consumption of production processes, in order to achieve better energy efficiency and energy cost efficiency in industry.

Under the scope of energy efficiency, the energy modeling of unit production processes does not consider the impact of volatile electricity prices. In the scope of demand response, the limited energy-cost-aware production scheduling investigations tend to have weak capacities of modeling the energy consumption as well as performing an effective scheduling according to dynamic electricity prices. The conversion from energy consumption amount in kWh to energy consumption cost should be more explicit for decision-makers to get clear conscious of the economic benefit brought by improved energy awareness. Therefore, a more advanced production scheduling algorithm should be developed, which is both energy-aware and energy-cost-aware corresponding to energy efficiency and demand response,

respectively. Consequently, the industry is able to use this algorithm to take advantage of lower-priced periods for extensive production or for storing energy for subsequent use during higher-priced periods.

In this chapter, a generic method is proposed to perform energy modeling, simulation, and optimization for a unit manufacturing process. The novelty includes: (1) a joint connection of energy efficiency and demand response is carried out to fully explore the industrial energy saving potentials within the DSM; (2) built on finite-state machines (FSMs), the energy model is extensible and enables detailed energy simulation; (3) by using a genetic algorithm (GA), the energy-cost-aware scheduler assigns the job sequence such that electricity pricing peaks are avoided and valleys are taken advantage of; (4) the power measurement on a surface grinding machine and two real dynamic electric tariffs fully demonstrate the applicability and effectiveness of the proposed method; (5) the energy consumption of a unit process can be predicted according to the energy-cost-aware scheduling solution.

The rest of this chapter is organized as follows. Section 2.2 provides a literature review revealing the problem. Section 2.3 gives an overview of the proposed data-driven energy-aware production scheduling method. Section 2.4 introduces the generic method to build an energy model from empirical data. Section 2.5 formulates the investigated problem in a MIP model. Section 2.6 presents a tailored GA to solve this problem. Section 2.7 introduces the reactive rescheduling framework to handle unforeseen events during the execution of a schedule on the shop floor. Section 2.8 describes the case study of scheduling a surface grinding machine under different electricity pricing schemes. Section 2.9 provides the discussions and the conclusions.

2.2 Literature Review

The studied issues in this chapter include energy modeling for a unit manufacturing process and energy-cost-aware scheduling of a single machine. The former research investigates how to increase the knowledge of machine energy consumption, which paves the way for reducing energy consumption. Thereby, it is within the scope of energy efficiency. The latter takes variable electricity prices into consideration and shifts the production along the time course such that low electricity prices are made use of as many as possible. This is part of the principal measures taken by industrial end-users to implement demand response. The rest of this section will discuss the state of the art in these two fields.

2.2.1 Energy Modeling for a Unit Process

Prior to energy modeling, electrical energy metering in complex manufacturing facilities is necessary to provide industrial enterprises higher levels of quantifica-

tion and visibility in their energy consumption. Both voltage and current need to be measured at either low or high sampling rates, in order to calculate power consumption and to produce more complex power quality statistics such as sags, peaks, and harmonics [11, 12]. An energy management framework can be further established to promote energy awareness in manufacturing processes [13]. On the basis of the measured power, empirical energy models can be built for estimating the energy consumption related to the production. The rest of this sub-section focuses on energy modeling at the level of a unit process.

Gutowski et al. [14] used an exergy framework to examine the energy requirement for a wide range of unit processes such as milling, injection molding, and grinding. Specific energy consumption (SEC) was defined to describe the energy needed for processing one unit of material. The process rate was demonstrated through empirical experiments as the key variable influencing the energy requirement of a unit process. This relatively early finding pointed out the complexity of industrial energy consumption, but there was no systematic approach to energy modeling and simulation.

In [15] an energy model was built for single machines via discrete state chart and transitions between states. In their model, operational states are defined by the functionality a specific machine has, and each state is associated with an energy consumption profile. A stochastic extension of the model is further provided to complement its stochastic simulation capacity. To achieve a global energy consumption optimization, they proposed to adjust the process parameters related to each state, but they did not further demonstrate this proposition.

The energy consumption of milling machine tools was characterized during their use stage [16]. The best fitted model is found with a 95% confidence level. It could then be used to estimate the total energy consumed during cutting. The effect of workpiece material on power demand was also studied. However, this empirical energy model was specifically for milling processes and no concrete energy saving measures were given.

In the framework of CO2PE! Initiative [17, 18], the two energy estimation methodologies for unit processes are screening approach and in-depth approach, respectively. The screening approach relies on publicly available data and engineering calculations for energy use. In the in-depth approach, different production modes are identified by the time study, and the power consumption for each mode is measured during the power study. The energy consumption of a unit process can then be estimated through multiplying the power by the duration of an operation. Optimization of energy consumption or energy cost is out of their scope.

An empirical energy modeling method was developed in [19] to predict energy consumption of unit processes. This industrial environment oriented method comprises four stages, namely design of experiments (DoE), physical experiments, statistical analysis, and model validation. The case study on an extrusion pro-

cess proved its ability to accurately predict energy consumption of unit processes. Briefly, their work also focuses on energy modeling.

In the approach proposed in [20], power measurements are not necessarily needed. A single machine tool is described by several functional modules which further consist of various components. In the Hardware-in-the-Loop-Simulation (HiL-Simulation), a physical machine controller is connected to the simulation model so that the programmable logic control (PLC) or numerical control (NC) signals, which contain power-on states, axis speeds, machine tool movement path, process operations, etc., are coupled with the functional modules and components to enable continuous energy simulation of a machine tool. In their case study of a coolant pump, various component configurations were tried to gain higher energy efficiency.

In addition to estimating the machine energy requirement within the work of [20], the HiL-Simulation model was further developed for real-time monitoring of the energy demand of a machine and its functional modules in production environments [21]. This energy monitoring system is claimed to raise the awareness of machine tool manufacturers and operators with regard to the machine energy consumption and to clearly show the consequences of their actions towards energy efficiency. Energy optimization measures based on components and operating states were finally discussed but not fully demonstrated.

While these machine energy modeling studies focus on energy consumption measurements and modeling, they do not explicitly investigate how to further make use of these measurements and models to optimize energy consumption of machines.

2.2.2 Energy-Aware Production Scheduling

The traditional manual production schedule becomes increasingly difficult in modern factories, where the production environment gets increasingly complicated. For instance, in semiconductor manufacturing, the dynamic job arrival, job recirculation, shift bottlenecks, and lengthy fabrication process are all involved. A wafer fabrication process typically contains over 500 processing steps [22]. Moreover, multiple types of products can be produced by the same line [23]. The product variety is even increasing to satisfy the rapid changes at marketplaces [24]. Furthermore, the volatile electricity price implies the need of frequent and short-term scheduling of plant operations, such as at a day-to-day time frame [4]. Therefore, it turns evident to foresee that an automated production scheduling will be widely deployed in modern or future factories.

With energy monitoring systems increasingly implemented on shop floors, the visibility is enhanced in the energy consumption behaviors of production activities. It is then feasible to add energy awareness to the conventional production sched-

ulers which are part of manufacturing execution systems. The electricity price can be further input into the scheduler, to facilitate its energy cost awareness.

A production planning control software was developed in [25]. It schedules the production on the basis of not only the usual planning criteria, i.e., deliver date, short lead-time, high resource utilization, and low inventory, but also their newly introduced objective of reducing peak power. The electricity price was not explicitly considered, but a decrease of peak consumption was calculated to implicitly bring a cost reduction. As one of the key results given by this software, the 24-hour power load forecast for a plant has a 15-minute time step, which can only give a coarse estimation of energy consumption.

A multi-agent based distributed evolutionary algorithm was used in [8] to search for a multi-process schedule with a minimized energy cost. This approach makes use of the potential for rearranging process steps to shift loads to low-priced periods. However, they did not mention the details on machine energy consumption, i.e., the variable energy consumption along time, and the detailed machine startup/shutdown operations when encountering machine idle periods.

Energy consumption and tardiness were jointly considered in [26] for multi-machine scheduling. Two heuristics were designed respectively based on the "earliest due date" rule and the "weighted shortest processing time" rule, and developed a particle swarm optimization algorithm. Nevertheless, both the energy consumption and energy cost were not clearly described. They simply assumed a higher machine speed would bring a shorter job makespan, while the corresponding energy consumption and energy cost would increase.

A new ant colony optimization metaheuristic was devised in [27]. It takes into account both makespan and energy cost to carry out hybrid flow shop scheduling. The ToUP mechanism and different machine processing speeds were considered. However, all the test data were randomly generated including the ToUP price and machine power consumption values. In addition, only two machine energy consumption states were assumed, i.e., processing and standby. The time aspect of scheduling results was unclearly described either.

The energy consumption and the energy cost of manufacturing systems were minimized in [28] while the production target was respected. This problem was formulated and its near-optimal solution was searched by particle swarm optimization. The effects of the summer and winter ToUP pricing profiles on the scheduling result were also investigated. Nevertheless, machine transition states between off and producing, i.e., startup and shutdown, were ignored, and the power consumption value was theoretically assumed.

In [29] the ToUP tariff was adopted in the time-indexed integer programming formulation to conduct production scheduling. This scheduling minimizes the energy cost while maintaining reasonable trade-offs with production throughput and CO_2 emission reduction, respectively. However, the concerned machines had only

on- and off-modes, which turns out to be too simple for energy modeling. Furthermore, both energy profiles and ToUP tariff values were theoretically supposed.

A bi-objective model was built in [30]. The Non-dominant Sorting Genetic Algorithm (NSGA-II) was used to minimize total energy consumption and total weighted tardiness on shop floors. However, only limited energy states were introduced in their energy model, i.e., idle, runtime, and cutting. Besides, the electricity price was not considered to convert energy consumption into a more meaningful energy cost.

In [31] the machine tool selection and operation sequence in job shops were optimized, in order to save energy consumption following the trade-off with makespan. Nonetheless, the volatile electricity price was not taken into account, either. A mathematical model was further formalized for the tri-objective job shop scheduling [32]. By using NSGA-II, the energy consumption and the energy cost were reduced while the total weighted tardiness remained acceptable, when the Rolling Blackout policy was applied. A trade-off was found between total weighted tardiness and total energy cost. Nevertheless, both the involved energy model was simplified and the electricity price was theoretical.

A GA was used in [33] to optimize the production scheduling of a single machine. Their schedule takes into account the dynamic electricity price to minimize the related energy cost. However, they only focused on determining when each job would start, and ignored scheduling the actual job sequence, which caused the job sequence on the same machine to be always fixed. Besides, they used a limited number of machine states, i.e., idle, processing, and shutdown, as well as presumed power values to model the energy consumption of a machine. This, together with the theoretical values for electricity price, caused a gap between their work and the industrial application.

Furthermore, an unexpected event that take place during the execution of a schedule is a practical issue on a shop floor. Its occurrence can be and should be handled by the scheduler. Unforeseen events on a shop floor include machine failure, starvation or blockage of a production unit, cancellation or change of a customer order, etc. Each event has its corresponding statistical distribution to occur. For instance, a machine breakdown is often modeled by the Weibull distribution. These events are seen as disturbance to a production schedule, since they interrupt the execution of the original schedule. A right-shift rescheduling policy [30, 34] is often used to deal with similar situations: the originally scheduled job sequence stays unchanged, and the queuing jobs are postponed for an amount of time to just accommodate the duration of an unforeseen event.

Despite these recent investigations on energy-aware production scheduling, a complete state-based energy model is seldom integrated to the scheduling model. The scheduling problem size is usually small regarding the number of time slots and jobs. The impact of unforeseen events on the energy cost efficiency as well as

the statistical energy cost efficiency of energy-aware production scheduling methods have never been analyzed.

2.3 Method Overview

The proposed data-driven production scheduling method [35] is presented by the flow chart in Figure 2.1, where the mapping between this method and the cyber-physical system (CPS) is also indicated. It is composed of 6 sequential steps. Step 1 and the rest 5 steps are executed in the physical space and cyber space, respectively. The scheduling solution provided by step 6 is sent back to the target machine for execution. This thus creates the typical closed loop of a CPS. While the whole loop is involved in Chapters 2-4, the focus is the optimization-based decision-making (Step 6 in Figure 2.1). The following paragraphs will introduce each step in detail.

Step 1 Energy consumption of a unit process/machine is empirically measured on the shop floor. This measurement can be either short term, midterm, or long term. A short-term measurement enables a fast reveal of the machine's energy consumption behavior from several hours up to one day. Usually one type of product/part is processed during this short term. The midterm measurement is conducted from several days to several months. A long-term measurement, i.e., permanent onsite monitoring, facilitates a complete understanding of the machine's energy consumption along with diverse types of products/parts flowing through the machine. However, a long-term measurement is more expensive from the economic perspective, since energy monitoring facilities (e.g., power meters and sensors) must be purchased, installed, and maintained.

Step 2 To enable centralized data management in the pyramid plant organization structure, the collected energy data are integrated into common industrial IT systems, e.g., MES (manufacturing execution system), ERP (enterprise resource planning), APS (advanced planning & scheduling), and manufacturing resource planning (MRP II). A few IIoT (industrial Internet of things) platforms specifically meet this data integration requirement, e.g., ThingWorx, Siemens' Mindsphere, and GE's Predix. Various data formats can be used for the integration, e.g., XML (extensible markup language), CSV (comma-separated values), and JSON (Java script object notation). Besides, MTConnect [36] is emerging as a more structured XML-based format for unified communication among sensors, equipment, and other hardware in manufacturing via standardized interface.

Step 3 A complete process power profile is identified from the measured power data. A power profile can be characterized by a set of power states. Each state has its power consumption and retention time. This can be illustrated by the power profile of a 4kW CO_2 laser cutting machine tool [37], which encompasses

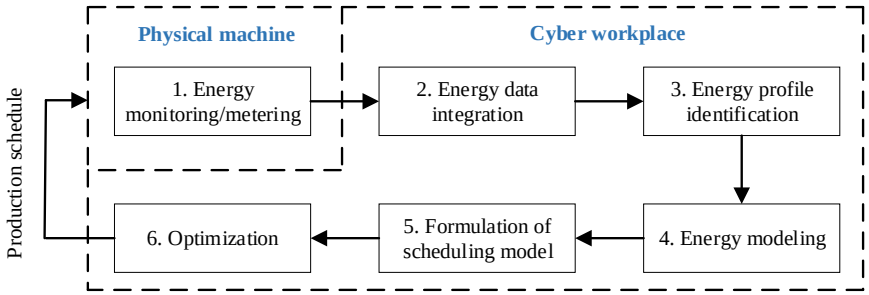


Figure 2.1: Method overview for data-driven energy-aware production scheduling from the perspective of cyber-physical systems

power states of machine tool startup, laser source startup, production ready, cutting at 3 different power levels, and machine tool shutdown. The power states of a process can be extracted from the collected power data, or identified by clarifying the machine's operational states which can be obtained from the machine's specification or from the machine's controller (e.g., a programmable logic controller or PLC). Once all the power states are identified, the power and retention time of each state can be statistically obtained based on the collected power data. For instance, the power of each state can be averaged from all the corresponding power samples. Besides, the time study and power study of the in-depth approach, which is proposed in [17], serve as a systematic way to identify a complete power profile. In general, a machine should be operated such that all the power states can be involved during the measurement for ensuring a complete power profile identification.

Step 4 The identified machine power profile is joint with a state-based energy consumption model. Rationalized transitions are established between these states by a case study. A generic state-based energy consumption model of a unit process can be found in [38], which includes common states such as *Off*, *Startup*, *Ready* (for production), *Production*, *Standby*, and *Shutdown*. This generic model was further applied to a surface grinding process and implemented by FSM (finite-state machine), which includes within the production state more specific sub-states for this process, i.e., *Grinding* and *Dressing*. The energy consumption behavior of machine changeover and/or maintenance can also be mapped with states in the model given that empirical energy data is available. For instance, in an extrusion blow molding process, the power consumption of a changeover can be modeled by one of the *ProheatIdle* and *PreheatIdle* states. The calculation of process energy consumption can be performed by accumulating the power along with the time-indexed retention and transition of power states.

Step 5 A mathematical model is formulated to rigorously describe the energy consumption model based production scheduling problem. For production

planning/scheduling, this model can often be formulated by MIP. A MIP (mixed-integer programming) model can be solved by a standard optimization problem solver, e.g., IBM CPLEX and FICO Xpress. Although upper/lower bounds can be obtained by these solvers, the common shortcomings in using them for a complex scheduling problem are known as long CPU time, poor extensibility, and intractability [39]. Therefore, it is of practical importance to design a tailored algorithm for an energy-aware production scheduling problem, which leads to the next step.

Step 6 One of the prevalent metaheuristics, e.g., GA, ant-colony, and particle swarm, is implemented according to the MIP model. Besides using a decent metaheuristic, hybridization of different metaheuristics can often enhance the optimization performance. For instance, a synergistic combination of NSGA-II and one or more local searchers lead to a memetic algorithm (MA) [40], which is a synergy of evolutionary approach with individual learning or local improvement procedures. Due to the mutually complementary strengths of NSGA-II and local searcher(s), it can achieve superior performance in both exploration and exploitation. In literature, it is common to apply a metaheuristic or MA to solve a scheduling problem, because metaheuristics essentially find optimal or near-optimal solutions within a reasonable time span, which complies with the intrinsic requirement of scheduling in practice [41].

2.4 Generic Energy Modeling

The generic energy model for a single machine is described by the finite state machine (FSM). The FSM is commonly used to represent discrete events and logic systems. It comprises five basic elements: a set of states, state transitions, external inputs, initial state, and final state. It can be depicted by either formulas or graphs. The graphical representation is preferred herein, since it is more intuitive.

Only the normal energy consumption mode is considered in this generic model. The energy saving mode is out of scope, as there are currently a rather limited number of machines supporting this functionality [42]. As presented in Figure 2.2, the generic FSM energy model consists of four main states: (1) *Off*, (2) *Startup*, (3) *Unowned*, and (4) *Shutdown*. The composite state Unowned indicates that no energy management policy is owned by the machine. It further contains two substates: *Ready* and *Production*.

The initial state of each simulation is *Off*. It indicates the machine is powered off and consumes no energy. Upon receiving the event “Power on”, the state transition is triggered from *Off* to *Startup*. At *Startup*, machine sub-units are sequentially powered on instead of all sub-units being powered on at one time. This complies with the measured startup energy profiles of different production ma-

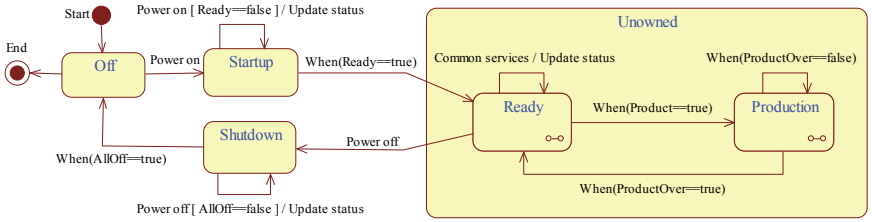


Figure 2.2: Generic state-based machine energy consumption model

chines [43, 44]. Following the completion of power-on operation, the machine updates its status list, which contains the power on/off state of each sub-unit. This self-transition continues until all the sub-units are powered on and the Boolean signal "Ready" becomes true.

Triggered by the "Ready" signal which turns true, the machine passes to the composite state *Unowned*. The entrance sub-state of *Unowned* is *Ready*, signifying that the machine is ready for production. The signal event "Common services" triggers a self-transition at *Ready*. The self-transition terminates by updating the machine status. "Common services" are to be defined according to the case study, e.g., to check the input material's availability. Once a production schedule is given, the Boolean signal "Product" changes from false to true. This then triggers the state transition to *Production*. The machine stays at this state until it completes the current production. When the signal event "ProductionOver" becomes true, it triggers the state transition back to *Ready*.

The signal event "Power off" occurring at *Ready* triggers the state transition towards *Shutdown*. At *Shutdown*, the machine powers off its sub-units also in a consecutive manner. This continues until the machine updates its status list such that all the sub-units are powered off and the Boolean signal "AllOff" becomes true. This finally drives the machine back to *Off*. The final state of each simulation is by default set to *Off*. However, upon an unforeseen event, a simulation run can terminate at any one of the states.

Based on the measured power consumption, different energy consumption states can be identified. The state identification method is similar to the time study of the in-depth approach proposed by [17]. So the time span of each state can be determined. Exceptionally, the duration of *Ready* can be arbitrary, as it is a state for staying idle. In the model proposed by this chapter, two types of *Ready* durations are thus defined, i.e., default duration and customized duration. The default duration for *Ready* stands for the necessary internal machine time for an immediate transition from *Startup* to *Production*, or from *Production* to *Shutdown*. The customized duration is determined by the production schedule, which can be an arbitrary value no shorter than the default duration.

Furthermore, each machine power state is associated with a mean power value.

The machine energy consumption E during a simulation can thereby be estimated by Equation (2.1):

$$E = \sum_{s \in S} \sum_{t \in T_s} P_s \cdot t \tag{2.1}$$

where s is a machine state, S is the set of machine states, t is a time period during which the machine stays at the state s, Ts is the set of periods during which the machine stays at the state s, and P_s is the mean power consumption of the state s. This general mapping enables a quick denotation of the fundamental energetic performance. Moreover, based on the identified energy profiles of machine states, the energy model can be further expanded to provide machine energy consumption details and machine energy consumption-related performance indicators.

2.5 Problem Formulation

Considering volatile electricity prices, the proposed scheduler aims to assign the job sequence and timing as well as machine states, such that this machine processes all these jobs with a minimal energy cost and without breaking the due time. Jobs are independent of each other without any precedence relationship, such that an arbitrary job sequence can be generated (the job precedence may be added if needed in practice, which consequently reduces the size of solution space for optimization). This job scheduler is a discrete-time system, since it is built upon the FSM energy model. Its basic time step is quite flexible depending on the applied case, especially on the frequency of the measured energy data injected into the energy model. The inputs of this scheduler are variable electricity prices, job IDs and production durations, a pre-fixed due time, and the energy model introduced in Section 2.4. The outputs include the job sequence, the start time and end time of each job, the machine operation following the completion of each job, and also a detailed energy and cost audit for the current scheduling solution. The machine operation can be "immediately start the next job", "shut down", or "stay idle".

A mathematical model [38] is formulated below for this problem. The objective function will be first given, followed by a bunch of relations or constraints. For the sake of conciseness, each machine state is assigned a unique integer index. As shown in Table 2.1, the last item "others" is specially retained for any case study where the generic FSM energy model needs to be extended. The involved parameters are presented in Table 2.2.

Equation (2.2) defines the objective function. It determines the three types of decision variables, i.e., the job sequence (π), the job start time (ST_j), and the machine power states (s), such that the energy cost for processing all the jobs before the due date is minimized.

Table 2.1: Machine power state indexing

Machine power state	Index
Off	1
Startup	2
Ready	3
Production	4
Shutdown	5
Others	6

$$\min_{\pi, ST_j, s} \left(\sum_{j=1}^{N_J} C_j + \sum_{j=1}^{N_J - 1} \left(\alpha_j \cdot CR_j + (1 - \alpha_j) \cdot CSD_j \right) \right) \tag{2.2}$$

Equation (2.3) calculates the energy cost for processing a scheduled job. Equation (2.4) calculates the energy cost for the machine to stay at the *Ready* state between the job on the *j*-th scheduled position and the next job (on the $(j+1)$-th scheduled position). Equation (2.5) obtains the energy cost for the machine to be shut down between the job on the *j*-th scheduled position and the next scheduled job. The relevant cost can be further divided into three parts: (1) the cost for staying at *Ready* during a default duration, (2) the cost for powering off a machine, and (3) the cost for powering on the machine after staying powered off and just before the start of the next scheduled job.

$$C_j = \sum_{ts=STS_j}^{ETS_j} EP_{ts} \cdot \left(\beta_{ts} \cdot \sum_{t=ST_j}^{ET_j} \sum_{s=1}^{N_s} \left(P_s^t \cdot t \right) \right), \; j \in [1, 2, ..., N_J] \tag{2.3}$$

$$CR_j = \sum_{ts=ETS_j}^{STS_{j+1}} EP_{ts} \cdot \left(\beta_{ts} \cdot \sum_{t=ET_j}^{ST_{j+1}} \left(P_3 \cdot t \right) \right), \; j \in [1, 2, ..., N_J - 1] \tag{2.4}$$

Table 2.2: Nomenclature of the proposed energy-aware production scheduling model (non-italic: input variables, italic: variables to be determined by the model)

Parameter	Notation
C_j	Energy cost for the *j*-th job
CR_j	Energy cost for *Ready* state after the *j*-th job
CSD_j	Energy cost for a machine power-off after the *j*-th job
D	Time duration of one electricity pricing slot
D_j	Processing duration for the job with ID *j*
D_j^i	Processing duration for the *i*-th job with ID *j*
DT	Common due time for all the jobs to be scheduled
EP_{ts}	Electricity price during the *ts*-th pricing time slot
ET_j	End time for the *j*-th scheduled job
ETS_j	End time in slots for the *j*-th scheduled job
N_J	Total number of jobs to be scheduled
N_s	Total number of machine power states
P_s	Power consumption of the machine state *s*
P_s^t	Power consumption of the machine state *s* at time *t*
s	Machine power state
ST_j	Absolute start time of the *j*-th scheduled job
STS_j	Start time discretized in time slots for the *j*-th scheduled job
t	Absolute time
ts	Time in electricity pricing slots
T_s	Start time of the scheduling span
TO	Time duration for the machine to to stay off
TR	Default time duration for the machine to stay ready
TSD	Time duration to shut down a machine
TSU	Time duration to start up a machine
α_j	Machine operation Boolean indicator
β_{ts}	Time mapping Boolean indicator
π	Job sequence

$$
\begin{aligned}
CSD_j = & \sum_{ts=ETS_j}^{\lfloor (ET_j+D_3-T_s)/D \rfloor} EP_{ts} \cdot \left(\beta_{ts} \cdot \sum_{t=ET_j}^{ET_j+D_3} (P_3 \cdot t) \right) \\
& + \sum_{ts=\lfloor (ET_j+D_3-T_s)/D \rfloor}^{\lfloor (ET_j+D_3+D_5-T_s)/D \rfloor} EP_{ts} \cdot \left(\beta_{ts} \cdot \sum_{ET_j+D_3}^{ET_j+D_3+D_5} (P_5 \cdot t) \right) \\
& + \sum_{\lfloor (ET_j+D_3+D_5-T_s)/D \rfloor}^{\lfloor (ET_j+D_3+D_5+TO+D_2-T_s)/D \rfloor} EP_{ts} \cdot \left(\beta_{ts} \cdot \sum_{t=ET_j+D_3+D_5+TO}^{ET_j+D_3+D_5+TO+D_2} (P_2 \cdot t) \right), \\
& j \in [1, 2, ..., N_J - 1]
\end{aligned}
\tag{2.5}
$$

Equation (2.6) determines the machine to stay at *Ready* if the idle period between two adjacent scheduled jobs is shorter than the breakeven duration (duration to power off and immediately power on a machine), or if the cost for powering off is more expensive. Equation (2.7) requires the duration of the first scheduled job to consider the time for the machine to start up, to pass by *Ready* for the default duration, and to execute the job. Equation (2.8) defines that the duration of an intermediate job should consist of a default duration of *Ready* at the beginning and then the job execution time. The default duration of *Ready* before the actual job execution is considered as necessary machine time to receive and read the next production schedule. Equation (2.9) ensures the duration of the last scheduled job to include the default duration of *Ready*, the job execution time, the default duration for the machine to pass by *Ready*, and finally the time span for shutting down.

$$\alpha_j = \begin{cases} 1, \text{ if } (ST_{j+1} - ET_j) \leq (D_3 + D_5 + D_2) \text{ or } CR_j \leq CSD_j \\ 0, \text{ otherwise} \end{cases}, \quad j \in [1, 2, ..., N_J - 1] \tag{2.6}$$

$$ET_1 = ST_1 + TSU + TR + D_j^1,\ i = 1,\ j \in [1, 2, ..., N_J] \tag{2.7}$$

$$ET_i = ST_i + TR + D_j^i,\ i \in [2, 3, ..., N_J - 1],\ j \in [1, 2, ..., N_J] \tag{2.8}$$

$$ET_{N_J} = ST_{N_J} + TR + D_j^{N_J} + TR + TSD,\ i = N_J,\ j \in [1, 2, ..., N_J] \tag{2.9}$$

Equation (2.10) guarantees that each job is scheduled only once and thus all the jobs can be scheduled. Equation (2.11) restricts that the machine can have only one state at one point of time. Equation (2.12) uses the flooring function to decide at which pricing slot the discrete time is located. Equation (2.13) calculates the duration for staying at the *Off* state between two jobs. Equation (2.14) makes sure that only one job is executed at one time on respecting the scheduled job sequence, and preemption is prohibited. Equation (2.15) requires enough time to complete all the jobs and the machine shutdown before the due time.

$$D_j = \sum_{k=1}^{N_J} D_j^k,\ j \in [1, 2, ..., N_J] \tag{2.10}$$

$$P_s^t = P_s = \sum_{k=1}^{N_s} P_k^t,\ s \in [1, 2, ..., N_s] \tag{2.11}$$

$$ts = \lfloor (t - T_s)/D \rfloor \tag{2.12}$$

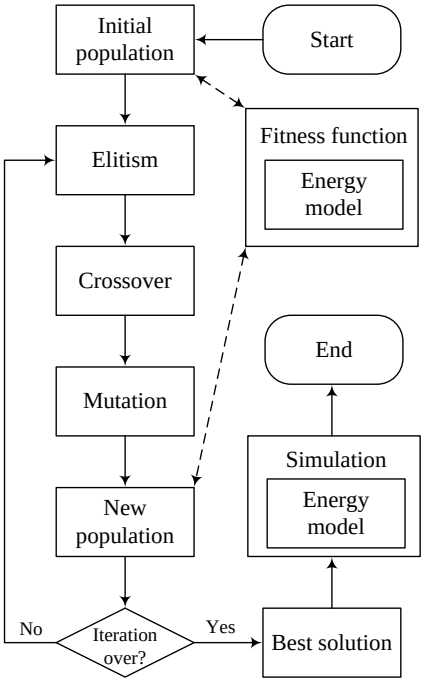


Figure 2.3: Implementation of a genetic algorithm (GA) as the solution method

$$TO = \begin{cases} 0, \text{ if } ST_{j+1} - ET_j \leq D_3 + D_5 + D_2 \\ ST_{j+1} - (ET + D_3 + D_5 + D_2), \text{ otherwise} \end{cases}, \quad (2.13)$$

$$j \in [1, 2, ..., N_J - 1]$$

$$ST_j < ET_j, ET_i + TR \leq ST_{i+1}, j \in [1, 2, ..., N_J], i \in [1, 2, ..., N_J - 1] \quad (2.14)$$

$$ET_{N_J} + TR + TSD \leq DT \quad (2.15)$$

2.6 Genetic Algorithm

As presented in Figure 2.3, a GA [38] is used to search for the optimized solution to this scheduling problem, which is described by Equations (2.2)-(2.15). A gene contains the information of a certain job including job ID, job duration, workpiece number, job start time, energy cost for executing this job, and idle/off machine operation after the completion of this job. A chromosome is a complete scheduling solution including the job sequence, all the jobs with their detailed information,

Table 2.3: Illustration of the one-point crossover

Case	Parent1	Parent2	Locus	Offspring1	Offspring2
1	(**2**, 4, 1, 3, 5)	(***5***, 1, 4, 2, 3)	1	(**2**, 5, 1, 4, 3)	(***5***, 2, 4, 1, 3)
2	(**2**, **4**, 1, 3, 5)	(***5***, ***1***, 4, 2, 3)	2	(**2**, **4**, 5, 1, 3)	(***5***, ***1***, 2, 4, 3)
3	(**2**, **4**, **1**, 3, 5)	(***5***, ***1***, ***4***, 2, 3)	3	(**2**, **4**, **1**, 5, 3)	(***5***, ***1***, ***4***, 2, 3)
4	(**2**, **4**, **1**, **3**, 5)	(***5***, ***1***, ***4***, ***2***, 3)	4	(**2**, **4**, **1**, **3**, 5)	(***5***, ***1***, ***4***, ***2***, 3)

and the machine operation following each job. The crossover and mutation are two important operations on genes in a GA, on which the GA performance largely depends. The crossover creates child solutions from parent chromosomes. The mutation prevents falling all solutions into a local optimum of the solved problem. Besides, the elitism is implemented to ensure the best solutions of a generation can be always retained into the next generation. The fitness function (Figure 2.3) containing the energy model evaluates each solution within a population. Therefore, a solution is the input of the energy model. The output is the energy cost for the input solution, which is stored as the solution's fitness. When the maximal iteration number is reached, the best solution is selected and simulated which finally provides detailed energy simulation information.

The permutation encoding [45] was used to represent a scheduling solution and to facilitate the crossover and mutation. Supposing there are two solutions represented by different orders of job IDs, i.e., parent1 is (2, 4, 1, 3, 5) and parent2 is (5, 1, 4, 2, 3), the single-point crossover and swap-based mutation are then illustrated in Table 2.3 and Table 2.4, respectively. Genes of two chromosomes are exchanged while ensuring that there is no job ID repetition in each chromosome (i.e., solution). For instance, in case 1 in Table 2.3, job2 in offspring1 comes from parent1, and the rest jobs in offspring1 come from parent2 by following the job order in parent2 while skipping job2 in parent2.

The permutation encoding indeed only includes the decision making on job sequencing. Once the job sequence is determined, the job timing (start time of a job while the job duration is an input variable) is randomly generated while respecting this job sequence and other important production constraints, e.g., the due date. Once the job sequence and timing are determined, my optimization algorithm will check the inter-job duration as well as the duration before and after the whole production. The decision making on machine power-off or idling mode for such a duration depends on the minimal energy cost that the selected mode requires in this duration (including the energy cost for transitioning to and recovering from the mode). Therefore, the multiple decision variables are sequentially determined in this problem by considering its problem structure.

Regarding the decoding method (mapping a chromosome to a schedule in this problem), FSM-based discrete-event simulation was used. In this simulation envi-

Table 2.4: Illustration of the swap-based mutation

Case	Solution	Mutated solution
1	(**2**, 4, **1**, 3, 5)	(**1**, 4, **2**, 3, 5)
2	(2, **4**, 1, 3, **5**)	(2, **5**, 1, 3, **4**)
Others	...	...

ronment, the determined decision variables can be read from a chromosome, e.g., job sequence timing and machine power-off/idling. The production is then executed following these decision variables to calculate the corresponding objective value or fitness value.

2.7 Rescheduling Framework

The production planning and scheduling hierarchy commonly exists in a manufacturing enterprise. The production planning determines when and how to produce in the medium term, by considering customer orders (e.g., product type, quantity, delivery time, etc.), as well as material and resource availability. In comparison, the production scheduling assigns production resources in the short term on the shop floor, by taking the production plan as general input constraints (e.g., job release time, due dates, etc.).

When this hierarchy is applied in a production environment, there are a variety of uncertainties, which may take place in a stochastic manner [46, 47]. In energy-cost-aware production scheduling, it is essential to have, as many as possible, time periods during which the electricity price is low and the resources are available for production. These time periods are referred here as *golden periods*. For a specific schedule, a longer length of golden periods within the scheduling time span will create a higher energy cost saving potential, in comparison to a conventional production schedule which has no awareness of the volatile electricity prices.

Due to this nature, the energy cost effectiveness of a schedule is sensitive to unforeseen events, since they may exert an influence on the length of the golden periods within the scheduling time span. Based on the influence which may be negative, positive, or neutral, the following taxonomy is made on unforeseen events [48].

Negative Influence Unforeseen events are susceptible to decrease the length of the golden periods. These unforeseen events are listed as (1) machine failure, buffer overflow, temporal blackout of power supply, and a late arrival of materials; (2) rework of some products or parts, increased product demand of a customer order, and an urgent new order; (3) an advanced due date; etc. The unforeseen events in (1) lead to machine or resource unavailability for the scheduled production when

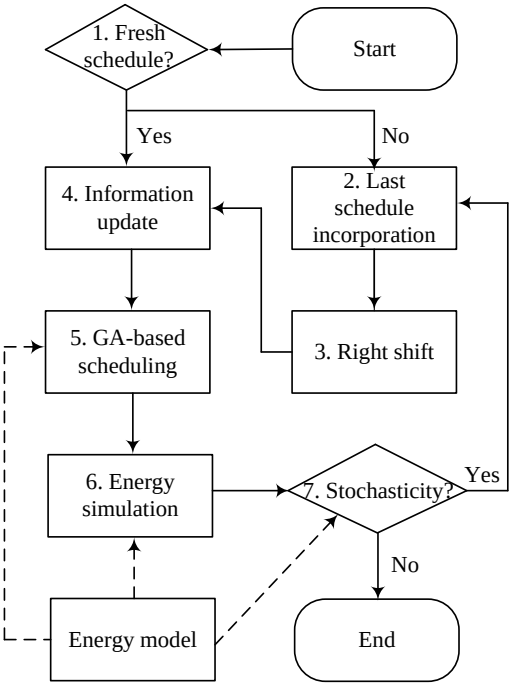


Figure 2.4: Reactive rescheduling framework based on simulation-optimization

they have a time overlap with jobs in an original schedule. The unforeseen events in (2) require the insertion of additional jobs into an original schedule. The unforeseen events in (3) may remove some golden periods.

Positive Influence Unforeseen events are likely to increase the length of the golden periods. These unforeseen events are, e.g., (1) decreased product demand of a customer order and cancellation of a customer order; (2) a postponed due date; etc. The unforeseen event in (1) directly releases more golden periods in the original schedule, thus adding more time to the scheduling. The unforeseen event in (2) may provide extra golden periods.

Neutral Influence Unforeseen events of which the trend to increase or decrease the length of golden periods is not evident. Within the demand side management [14], the variability of the electricity price, e.g., real-time pricing (RTP), is specifically viewed as such type of unforeseen event for energy-cost-aware production scheduling. The electricity price is volatile such that the length of golden periods fluctuates if a scheduling time span shifts in time.

The proposed heuristic to handle unforeseen events during the execution of a production schedule is presented in Figure 2.4. The key operations are indicated by different numbers. (1) The fresh schedule operation decides whether the next scheduling is run on the basis of a former schedule. In the case of unexpected

events, a former schedule is the one that is interrupted by an unforeseen event. (2) If a former schedule is involved, the interrupted schedule is taken into consideration for the next scheduling. The considered information includes a) the time when the former schedule is interrupted by an unforeseen event (i.e., the start time of the unforeseen event), b) the duration of the unforeseen event, c) the already executed jobs in the former schedule, or the non-executed jobs that need to be reconsidered in the next scheduling, and d) the job that is being executed, but is not yet accomplished, upon the occurrence of the last unforeseen event. (3) The next scheduling (i.e., rescheduling) can be performed starting from the time when the last disruption terminates (i.e., the start time plus the duration of the disruptive unforeseen event). The right-shift operation postpones all the upcoming jobs after the termination of this disruption. Depending on the specific production, an interrupted job has to be totally reproduced (i.e., a non-resumable job), or only its non-executed part remains to be produced during the next schedule (i.e., a resumable job). So for an interrupted non-resumable job, its whole part is right-shifted. For an interrupted resumable job, its non-executed part is right-shifted. In comparison to the existing right-shift policy, rescheduling of job orders with the volatile electricity price is involved in the following steps, in order to remain energy-cost-effective. (4) Input information is updated and loaded in the scheduler, e.g., the electricity price, input jobs, the start time, the due time, GA configurations, etc. (5) The scheduling is carried out by using the GA. (6) For the output optimized schedule, an energy simulation is conducted to have a detailed energy report of this schedule. (7) If any disruption is involved, it will invoke another rescheduling. Otherwise, the whole procedure terminates.

The energy model is coupled with operation5 (O5) and operation6 (O6) to make the scheduling and modeling energy-aware. It is also associated with operation7 (O7) to incorporate disruption awareness into its time progression. The sequential steps “Start-O1-O4-O5-O6-O7-End” result in a conventional scheduling procedure, which is static. The cyclic steps “O2-O3-O4-O5-O6-O7-O2-O3...” set up a dynamic scheduling procedure to deal with disruptive unforeseen events.

2.8 Case Study of a Surface Grinding Machine

The proposed energy-aware production scheduling and rescheduling methods were implemented in a case study of a surface grinding machine (Paragon RC-18CNC) under two real electricity pricing mechanisms (RTP and ToUP).

2.8.1 Energy Modeling of a Surface Grinding Machine

The power measurement on this grinder was performed with a clamp-on power meter (Yokogawa CW240). Connected between the power supply and the grinder,

the power meter records the grinder's overall power consumption every second. The major energy consumers of this grinder are listed in Table 2.5.

Table 2.5: Major energy consumers of the surface grinding machine Paragon RC-18CNC

Energy consuming unit	Function
Grinding wheel	Grind the workpiece (Each grain of abrasive on the grinding wheel's surface cuts a small chip from the workpiece via shear deformation)
Regulating wheel	Rotate the workpiece and pull it through the operation so as to control workpiece rotational speed and feed rate
Hydraulic pump	Transport the liquid press to subsystems for mechanical control
Coolant pump	Move coolant for cooling the workpiece, grinding wheel and regulating wheel
Others (computer, light, hydraulic oil cooler, automatic lubricator etc.)	Miscellaneous functions

This grinder's complete energy profile is illustrated in Figure 2.5. At *Startup* state, its power consumption first has sharp peaks at around 15 kW, and then experiences a periodic drop-down and rise-up between 4.8 kW and 2 kW, which should be due to the power-off and power-on of the first coolant pump. At *Ready* state, the grinding wheel rotates at a fixed peripheral speed of 2000 m/min without the touch of a workpiece or the dresser, which results a nearly constant power consumption of 6 kW. The Production state is further divided into Grinding and Dressing sub-states. Dressing is responsible for sharpening and regularizing the grinding wheel shape, and cleaning the impurities coming from the chips. The second coolant pump should be powered on when the state transitions from *Ready* to *Grinding*. At *Grinding*, each evident peak corresponds to grinding one workpiece. The grinder passes from *Grinding* to *Dressing* about every 350 seconds. The second coolant pump should be powered off during the dressing cycle. At *Shutdown*, the main power consumers are powered off rapidly, which leads to the chute of the power curve; then the grinder stays in a constant power level for more than five minutes (Figure 2.5). As the grinder is computer-numerically controlled, this is interpreted as a compulsory duration for the numerical system to perform shutdown work, e.g., storing data to non-volatile memory. The state-based energy audit result for the grinder is listed in Table 2.6. Some states have an obvious difference between their maximum and average powers, e.g., *Startup* and *Shutdown*,

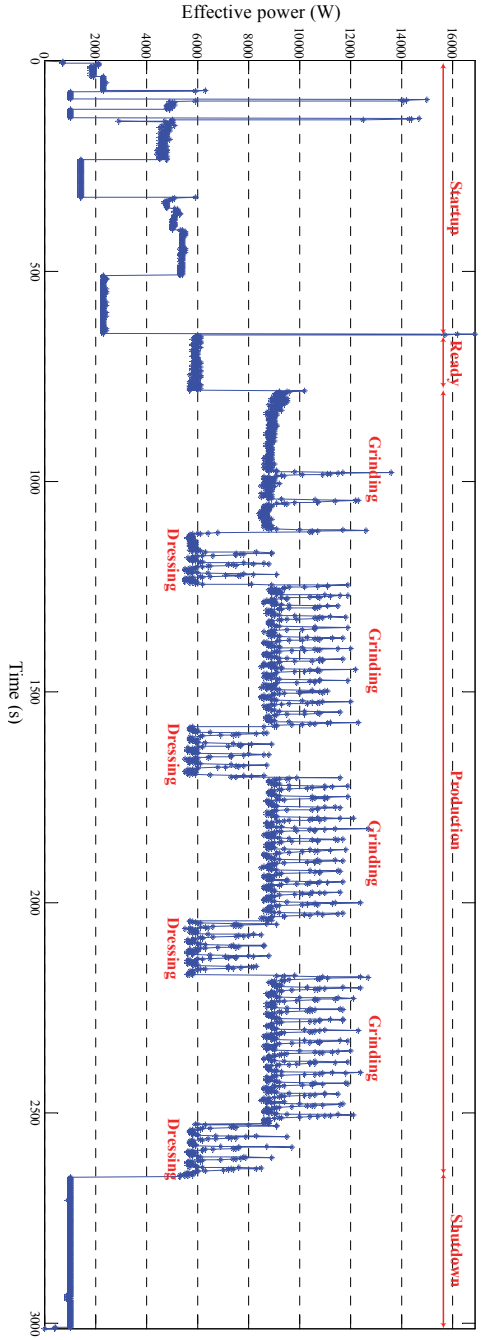


Figure 2.5: A complete energy consumption profile of the investigated surface grinding machine

Table 2.6: Energy audit for the surface grinding machine

Machine state	Maximum power (kW)	Mean power (kW)	Cycle time (s)
Startup	16.90	3.55	652
Ready	6.10	5.93	25 (default)
Grinding	12.07	9.49	25
Dressing	8.95	6.72	125
Shutdown	5.30	1.00	362

while others have steady power profiles, e.g., *Ready*.

Based on the identified states, the generic energy model (Section 2.4) was applied to this specific case, as shown in Figure 2.6. It was implemented in Java. Compared to the generic model, the *Production* state further contains *Grinding* and *Dressing*. Although the actual grinding operation has to be interrupted periodically, the dressing operation should be carried out not only to avoid the occurrence of abnormalities on the grinding wheel's surface but also to guarantee high product quality. The dresser is assumed to be in good condition when the grinder starts a new job. So when the machine stays at *Ready*, its next state is either *Grinding* or *Shutdown*.

2.8.2 Scheduling under Real-Time Pricing (RTP)

The energy-cost-aware job scheduling model is expected to work such that the total energy cost for the scheduled production is minimized under the dynamic pricing mechanism. It is coupled with the energy model built in Section 2.8.1. Therefore, it can not only get full knowledge of the energy-related information, but also output a scheduling solution for the energy-related simulation.

Table 2.7: Grinding jobs for scheduling

Job ID	Number of steel workpieces	Required production time (grinding + dressing)
1	100	3375 (56m15s)
2	200	6750 (1h52m30s)
3	300	10125 (2h48m45s)
4	400	13500 (3h45m)
5	500	16875 (4h41m15s)

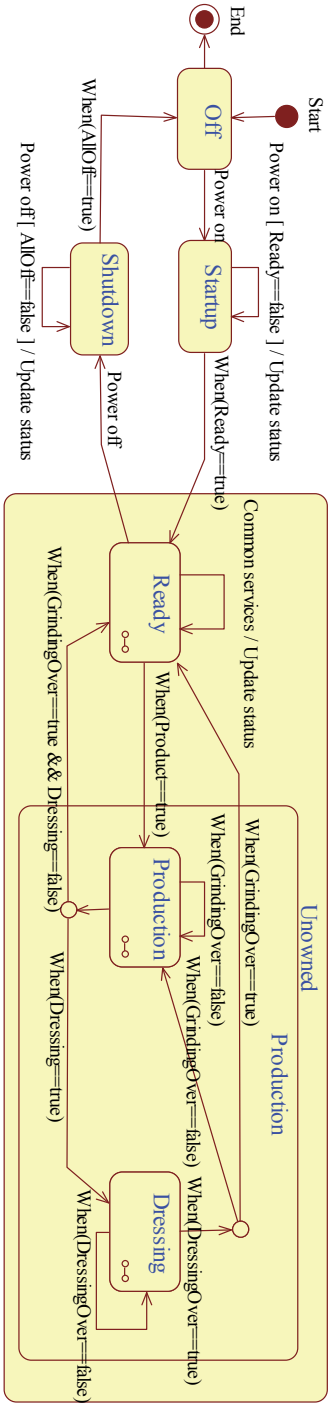


Figure 2.6: Specific state-based energy model for surface grinding process

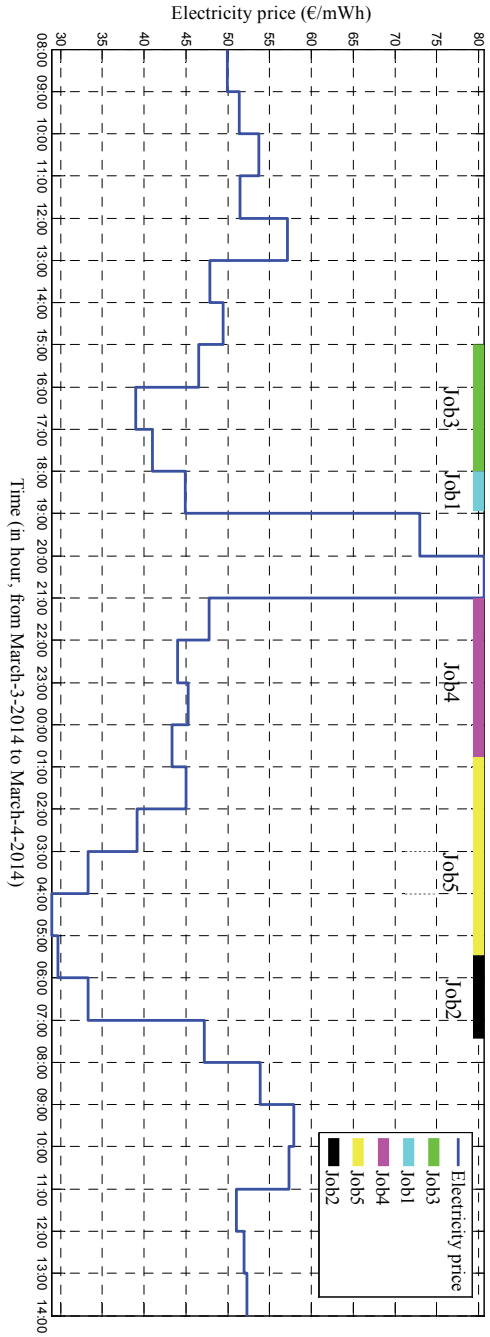


Figure 2.7: Real-time pricing (RTP) data from the Belgian electricity spot market and the optimized job schedule (the scheduling step is one second)

2.8.2.1 Optimization based on a genetic algorithm (GA)

A number of assumptions were first made. (1) The concerned work shifts last from 8 am on March-3-2014 to 2 pm on March-4-2014. (2) At Belpex, since the exchanged power volume on the day-ahead market (DAM) is significantly greater than that on the continuous intraday market (Belpex, 2013), the RTP data was taken from the DAM (Figure 2.7). (3) The concerned steel workpieces are of the same type as that in the measurement. (4) The grinder runs the same numerical control (NC) program, which means that it keeps the same energy consumption behavior as that identified in Section 2.8.1. Moreover, the machine always grinds 14 workpieces then conducts one cycle of dressing operation during a continuous grinding process. (5) If the machine grinds less than 14 workpieces upon finishing the current job, it will grind another 14 workpieces for the next job before it performs another dressing operation. This is denoted as "non-memory dressing". (6) If the grinder stays idle or off before the start of one job, the start time of this job is always set at the very start of a certain hour, e.g., 9 am and 11 pm. (7) The grinding jobs are shown in Table 2.7. Totally five jobs are considered to take an example. An extension to a larger number of jobs is possible (Section2.8.3).

The tailored GA (Section 2.6) was implemented in Java. The population size was set to 80. This means that each generation has 80 individuals. The elitism rate was 0.15, which means the top 15% of individuals were retained from one generation to the next. The crossover and mutation rates were fixed at 95% and 3%, respectively. The maximal iteration was 100. These configurations were set based on empirical experiments. Chapter 3 will further present the design of experiment to achieve an optimized parameter setting of an evolutionary algorithm.

The optimized job schedule found by the GA is shown in Table 2.8. The time step in this schedule is one second, since the measured power data has a frequency of one hertz. The scheduler's stability is proven by the fact that there is no time overlap between jobs, and in the case of consecutive jobs, the next job strictly starts from the end time point of its precedent job. The machine operation is also given to indicate the machine behavior following each job.

This schedule is depicted in Figure 2.7, which evidently demonstrates its high effectiveness. The electricity price changes at different hours, but stays the same within one hour. The highest pricing peak appears in the evening from 7pm to 9pm on March-3, while the lowest pricing valley falls in the early morning from 3am to 7am on March-4. This scheduling solution can not only effectively avoid high-priced periods, e.g., the aforementioned highest pricing peak, but also allocate as many as possible the jobs to low-priced periods, e.g., from 4pm to 7pm on March-3 and the aforementioned pricing valley. While this is a simple case to illustrate the energy-aware production scheduling idea, more complex cases will be presented in Section 2.8.3, Chapter 3, and Chapter 4.

This optimized schedule is further compared with some other scenarios with-

Table 2.8: Optimized energy-aware production schedule

Job ID	Job start time (March 2014)	Job end time (March 2014)	Machine operation following the current job	Machine states following the current job
3	3d:15h:0m:0s	3d:18h:0m:2s	Immediately start the next job	*Ready*, *Grinding + Dressing*
1	3d:18h:0m:2s	3d:18h:56m:42s	Shut down	*Ready*, *Shutdown*, *Off*
4	3d:21h:0m:0s	4d:0h:45m:25s	Immediately start the next job	*Ready*, *Grinding + Dressing*
5	4d:0h:45m:25s	4d:5h:27m:5s	Immediately start the next job	*Ready*, *Grinding + Dressing*
2	4d:5h:27m:5s	4d:7h:26m:27s	Shut down	*Ready*, *Shutdown*, *Off*

Table 2.9: Energy cost comparison between the energy-aware production schedule and benchmark scenarios

Scenario	Electricity price (euro/mWh)	Number of machine startups and shutdowns	Cost (euro)	Energy cost saving rate[a]
Maximal pricing	80.69	5	10.7	52%
Average pricing	48.24	3	6.3	19%
As-early-as-possible schedule	Hourly dynamic	1	5.8	12%
As-late-as-possible schedule	Hourly dynamic	1	5.5	7%
Energy-aware schedule	Hourly dynamic	2	5.1	-

[a]The energy-aware production schedule compared to the corresponding benchmark scenario which has no energy awareness.

out energy awareness (Table 2.9). The maximum pricing scenario uses the highest electricity price during the entire scheduling period and performs the maximal possible number of machine startups and shutdowns (i.e., a machine shutdown and startup between each pair of adjacent jobs). Similarly, the average pricing scenario utilizes the mean value electricity price during the entire scheduling period and intermediate number of machine startups & shutdowns (3 startups & shutdowns in Table 2.9). As a classical production schedule, the as-early-as-possible schedule drives the machine to grind all the jobs consecutively from the beginning of the work shifts without staying idle or powered off between any jobs. In comparison, the backward schedule plans the start time of jobs from the due time. This leads to the as-late-as-possible schedule. These scenarios all have the same job sequence of the energy-aware production schedule. As is shown by Table 2.9, the cost reduction effect of an energy-aware production schedule is obvious with the cost saving rates varying from 7% to 52%.

2.8.2.2 Energy Simulation of the Obtained Schedule

As the scheduler is coupled with the specific energy model (Section 2.8.1), detailed energy related statistics can be further given by the energy simulation of the optimized schedule, including the accumulated time duration, the electric consumption, and the cost at the level of machine states, and also the aggregated information at the machine level (Table 2.10). The main electricity consumer among states can be identified as *Grinding*, which takes up nearly 80% of the total electric consumption and cost, followed by *Dressing* at nearly 20%. This type of table enables machine operators and decision-makers to have a clear view over the energy related details of the machine.

In addition, the power consumption of this grinder during the simulation can be represented over time, as shown in Figure 2.8. The grinder's energy consumption states are correspondingly indicated above the power curve. Figure 2.8 can be zoomed at one second, which is illustrated by the left arrow, showing that the scheduling time slot is one second. There are in fact two power consumption curves: the green one indicates the a priori estimation, which is predicted according to the given job schedule independent of the simulation environment, while the dark blue one is the a posteriori display based on the power data collected during the simulation. The complete overlap of the two curves demonstrates the correct functionality of the proposed model to perform energy modeling and simulation. Given the production schedule and the electricity price in the coming days, this representation can also serve as an accurate power consumption prediction. Besides, the accurate power consumption behavior can be stored and compared with unrecognized power consumption patterns. This facilitates machine failures to be detected in the early stage of abnormal events on a shop floor.

Table 2.10: Energy consumption details of the energy-aware production schedule at levels of machine and power state

	Time consumption		Energy consumption		Energy cost	
	Amount (s)	Percentage[a]	Amount (kWh)	Percentage[a]	Amount (euro)	Percentage[a]
Off	55172	51.1%	0	0	0	0
Startup	1304	1.2%	1.29	1.0%	0.08	1.6%
Ready	175	0.2%	0.29	0.2%	0.01	0.2%
Grinding	37500	34.7%	98.85	79.0%	3.97	78.5%
Dressing	13125	12.2%	24.50	19.6%	0.98	19.4%
Unowned	50800	47.0%	123.64	98.8%	4.96	98.0%
Shutdown	724	0.7%	0.20	0.2%	0.01	0.2%
Grinder	10800	-	125.13	-	5.06	-

[a]The percentage of each power state over the aggregated amount.

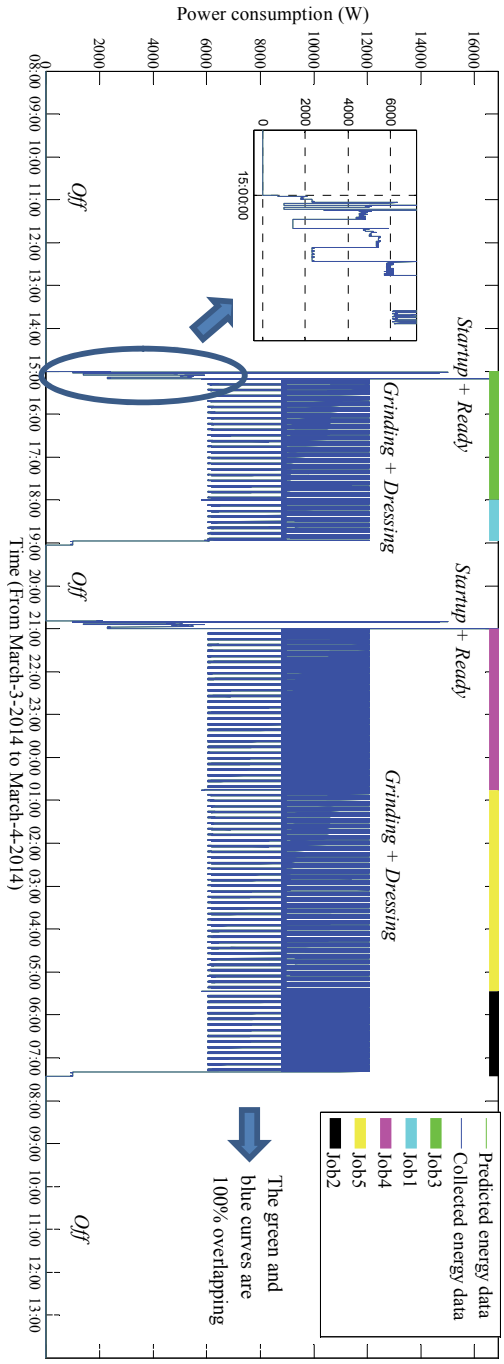


Figure 2.8: Energy consumption prediction of the surface grinding machine (prediction step is 1 second)

2.8.3 Scheduling under time-of-use pricing (ToUP)

The ToUP tariff was taken from a Belgian plastic bottle manufacturer, which buys energy from the spot market once a month. All the other assumptions are the same as those in the above case. In order to further demonstrate the effectiveness of the proposed energy-cost-aware scheduler, the investigated period is extended to one week, i.e., from 8am on March-3-2014 to 8am on March-10-2014. Compared to the number of each job in Table 2.7, the number of jobs in this experiment rises by 7 times (35 jobs). The scheduling step is one second, which is equal to the former case.

As shown in Figure 2.9, this electricity price has two levels: on-peak and off-peak within every 24 hours, at 61.1 euro/mWh and 39.6 euro/mWh, respectively. The off-peak period lasts from 9 pm to 6 am of the next day, which has only nine hours within a day. Hence, the obtained job schedule makes use of these periods as many as possible, while keeping energy-related overheads as small as possible. The energy-related overheads can be extra energy consumed by a frequent machine switch on/off, or/and by a long-term machine idle state. In this schedule, there are some short off-peak slots that are allocated for the idle state, instead of the production-related states, i.e., Grinding and Dressing, which are more energy intensive. This can be illustrated by the idle periods of 14m35s between Job9 and Job11 during the Wednesday off-peak, and 1h3m20s between Job6 and Job34 during the Thursday off-peak. In total, 90% of the off-peak periods are allocated for production-related states. Besides, the machine operation, which is scheduled to follow each job, is indicated next to each job in the legend of Figure 2.9. This is similar to Table 2.8 in the RTP case. The machine operation will invoke the corresponding machine state transition in the state-based energy model. Coupled with the energy model, the scheduler thus assigns the proper machine states along with jobs.

In Figure 2.9, the obtained schedule is revealed as a near-optimal solution, instead of the optimal solution. This can be explained by two reasons. Firstly, there are 604,800 time slots in this demonstration. This turns out to be a large number for scheduling, in comparison with maximum dozens or hundreds of time slots in similar work [30, 33]. Secondly, this complies with the intrinsic characteristic of metaheuristics (such as the GA used in this work): fast resolution of a hard optimization problem without guaranteeing the real optimum. Furthermore, This near-optimal schedule is obtained after a 2593-second GA search. Figure 2.10 depicts the convergence trend of this GA search. In the first 20 generations, the total energy cost of the best scheduling solution rapidly drops from 45.18 euro to 42.86 euro. Then, from the 21st generation to the 67th generation, the cost experiences a steady decrease down to 42.63 euro. After that, the cost stays quite stable with only a slight decrease, and reaches 42.61 euro at the 250th generation.

Based on the energy simulation of the optimized job schedules in the two elec-

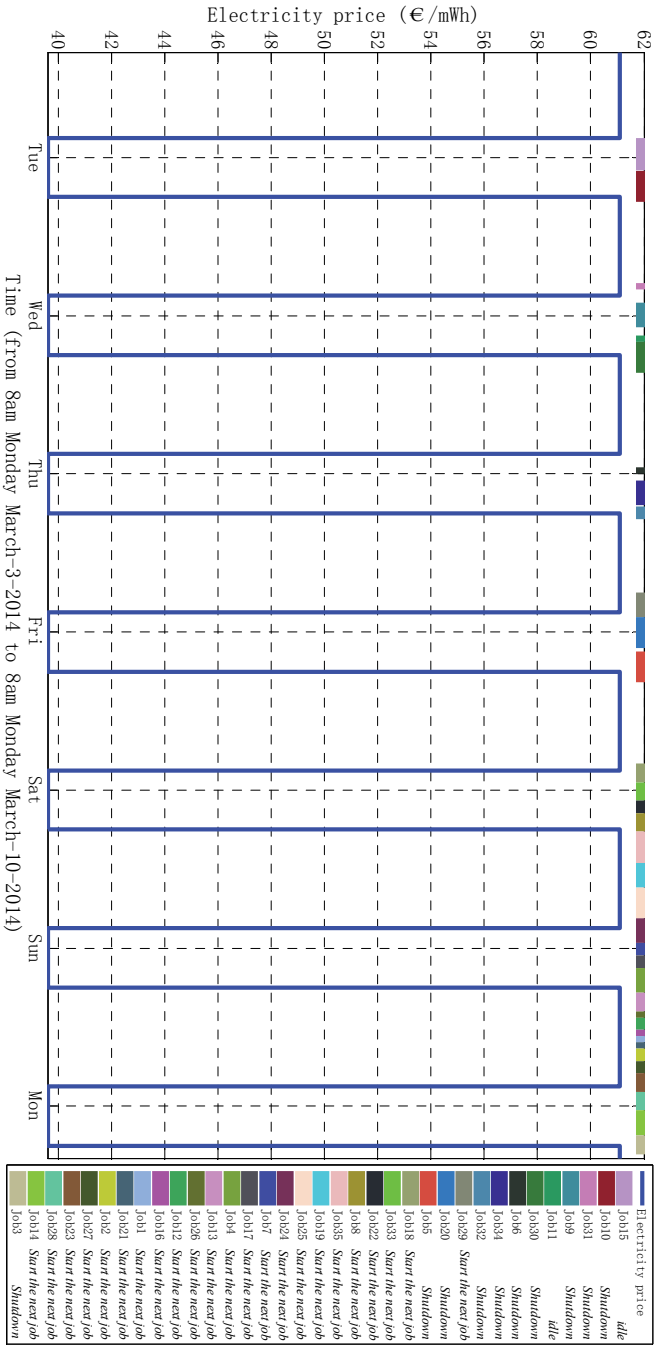


Figure 2.9: Energy-aware production schedule under ToUP (time-of-use electricity pricing)

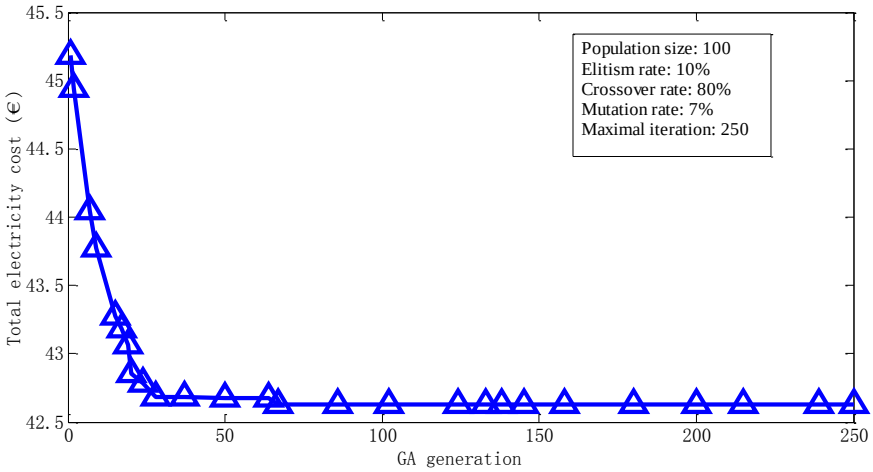


Figure 2.10: Convergence trend of the genetic algorithm-based search

tric tariffs, a comparison is further conducted between their energy consumption efficiency, energy cost efficiency, and productive energy rate (Table 2.11). The energy consumption efficiency (Table 2.11) indicates that, for producing one workpiece, the two optimized schedules consume almost the same amount of energy under RTP and ToUP, respectively. The energy cost efficiency (Table 2.11) shows that, for one workpiece, it consumes a lower energy cost (17% difference) under RTP than under ToUP. The productive energy rate (Table 2.11) reveals the percentage of the consumed electricity which directly contributes to the added value of workpieces. It is the same (79%) in the two cases, although the time duration and job quantity are different. This can be explained as follows. First, the grinder is scheduled to be powered off during most time periods when there is no need for grinding. Second, the dressing operation is accompanied with the grinding operation periodically. So the energy consumed by grinding and dressing proportionately increases with the growth of job number. This type of table can not only

Table 2.11: Energy metrics of scheduling in different electricity pricing schemes

	RTP tariff	ToUP tariff
Energy consumption efficiency[a]	0.0834 kWh/piece	0.0833 kWh/piece
Energy cost efficiency[b]	0.0034 euro/piece	0.0041 euro/piece
Productive energy rate[c]	79%	79%

[a]Total energy/total amount of products.
[b]Total energy cost/total amount of products.
[c]Productive energy/total energy.

provide machine energy related KPIs to decision-makers, but also help them to get an accurate insight into the effect of different electric tariffs on the energy related KPIs. Therefore, the knowledge of the energy consumption and energy cost contributes to a more informed decision on production activities on a shop floor.

2.8.4 Trade-Off between Energy Cost and Makespan

An intuitive observation on the energy-aware production scheduling method is that a longer makespan for the same production would release more free durations and increase the opportunity to reduce the energy cost. To identify and quantify the potential trade-off relationship between the energy cost and the makespan, a Monte Carlo simulation [49] was performed. Three electricity pricing schemes are involved: RTP, ToUP, and critical peak pricing (CPP). All the pricing data are taken from [49].

While the RTP and ToUP are introduced in former subsections, CPP is an overlay on the ToUP. It imposes a much higher rate during a period called critical peak in an event day, when the electricity use is significantly high [50]. CPP event days are usually determined based on a day-ahead maximum temperature forecast at specific locations, since peak demands usually occur in a hot summer day or a cold winter day. The utility notifies its customers by 3 PM, on a day-ahead basis if a CPP day is to take place the next day. There are a high price and a moderate price during a CPP period. The high price can be five times as high as the on-peak price of a normal ToUP tariff, and the moderate price can be almost three times as high as the off-peak price. A CPP period often lasts from noon to the early evening.

The Monte Carlo simulation was performed to statistically sample the relationship between energy cost and makespan under each pricing scheme. In this experiment, one million schedules were randomly produced under each of these three electricity pricing schemes, and each pair of energy cost and makespan was recorded. The trade-off between the energy cost saving rate (R_{ECS}, compared to the classical as-early-as-possible schedule or AEAP schedule) and the makespan prolongation rate (R_{MP}) is revealed in Figure 2.11. Both rates are calculated compared to the AEAP schedule. Among the three trade-off curves, the one under CPP has the lowest slope and covers an obviously larger range of R_{ECS} (0.6%-69%), because of its two significantly higher priced levels during its predefined critical period. However, the other two trade-off curves under RTP and ToUP, which have more or less the same slope and the same coverage, include a small negative interval on the horizontal axis. This demonstrates that there are a few schedules even more expensive than the AEAP schedule, or the AEAP schedule is among the most expensive schedules in terms of energy cost. In CPP, the AEAP schedule is uniquely the most costly one, since it covers the entire critical period on the same day (Figure 3 in [49]). Vertically, the R_{MP} ranges of the three trade-off curves

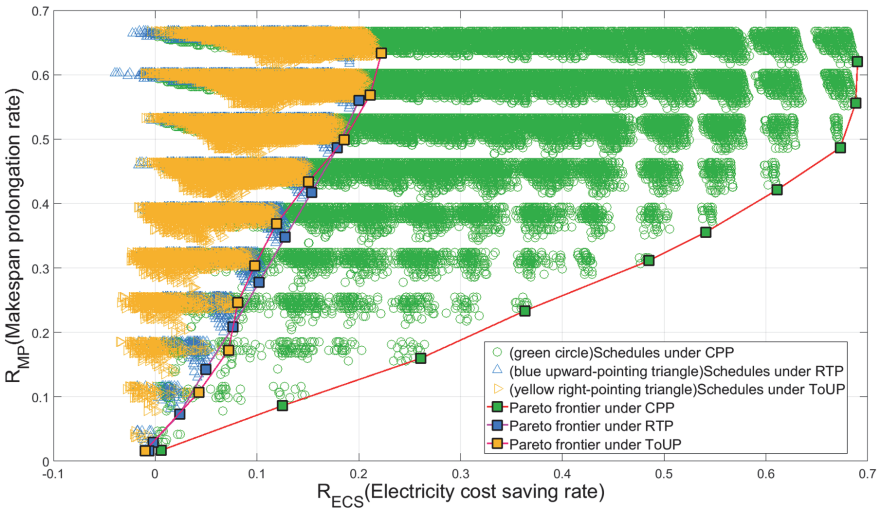


Figure 2.11: Trade-off between the energy cost saving rate and the makespan prolongation rate

vary between 0 and 70%. There is no negative rate, as the AEAP schedule already has the shortest makespan.

2.8.5 Rescheduling upon Unforeseen Events

Two types of unforeseen events were simulated to demonstrate the effectiveness to handle unforeseen events when the machine processes jobs according to the energy-aware production schedule (Section 2.7), and further to enable the analysis of how unforeseen events affect the energy-cost-effective performance of the proposed scheduling method. The case of scheduling under RTP in Section 2.8.2 was taken as a baseline case.

2.8.5.1 Unforeseen Events with Negative Influence

A random machine failure is simulated. It is a representative unforeseen event that has negative impact on the energy-cost efficiency of an energy-aware production schedule (Section 2.7). The time for a machine failure follows the Weibull distribution [51], whose probability density function is described by Equation (2.16).

$$f(t;\alpha,\beta) = \begin{cases} \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \cdot e^{-\left(\frac{t}{\alpha}\right)^{\beta}}, x \geq 0 \\ 0,\ x < 0 \end{cases} \tag{2.16}$$

In this investigation, the shape parameter α in Equation (2.16) equals one,

meaning that the machine failure rate is constant along time; the scale parameter β equals 10,000 in order to adapt the time generation interval to the investigated period (30 h, i.e., 108,000 s). One machine failure occurs during the simulation of a complete schedule. The duration of a machine failure is one hour, meaning that it takes one hour for the machine to recover from the machine failure. The machine stays powered off during the machine failure period. At the presence of a machine failure, the interrupted job is resumable, and the rest of it should be placed in the first position in the reschedule. Otherwise, it is not possible to separate the jobs.

As the baseline case is a fresh schedule, the scheduler has the same configuration, e.g., the job number and duration, RTP data, etc. It first goes through the heuristic steps "Start-O1-O4-O5" (Figure 2.4) to get the original optimized schedule (Figure 2.12). This optimized schedule is then simulated in O6. Upon the machine failure which occurs at 15h29m35s on March-3-2014, the energy simulation terminates, and the scheduler continues to go through the steps "O7-O2-O3-O4-O5" (Figure 2.4) to reschedule the order and start time of the upcoming jobs (i.e., Job5, Job1, Job2, and Job4 in Figure 2.12), and also to reschedule the start time of the non-executed part of Job3. As shown by Figure 2.12, the jobs are successfully rescheduled by making use of the low-priced periods and avoiding the high-priced periods. The total energy cost (5.2 euro) is then comprised of the cost for producing Job3's executed part in the original schedule, and the cost for running the whole reschedule. It slightly increases by 2% in comparison to the energy cost of the baseline case. The reason is that the machine failure takes up some low-priced periods, such that there is not fully sufficient low-priced periods to accommodate the reschedule after the machine failure (e.g., the last part of Job3 after 19h on March-3-2014). The energy cost rising rate caused by the machine failure will get higher, if the energy cost consumed by the repair activity during the machine failure period is considered.

The second simulated unforeseen event is an urgent customer order. Five new small jobs (Table 2.12) arrive at 17h on March-4, which is during the execution of Job3 in the original schedule. They have the same due time as the original jobs (i.e., 2 pm on March-4-2014). The reschedule is thus triggered. Job3 continues to be executed, while all the upcoming jobs (the non-executed original jobs plus the new jobs numbered from 6 to 10) are rescheduled by going through the steps "O7-O2-O3-O4-O5-O6" (Figure 2.4). The start time of the reschedule is the time when Job3 is finished. As presented by Figure 2.13, the rescheduled jobs effectively make use of the low-priced periods, while avoiding the high-priced periods (i.e., 19h to 21h on March-3, and 9h to 11h on March-4). The total energy cost for all the jobs is 6.35 euro. Compared to the baseline case, it rises 25%, while the number of workpieces increases 20%. The reason why the energy cost has a higher increasing rate is that some higher-priced periods have to be used to accommodate the rising job volume. However, the difference between the two rising rates is relatively

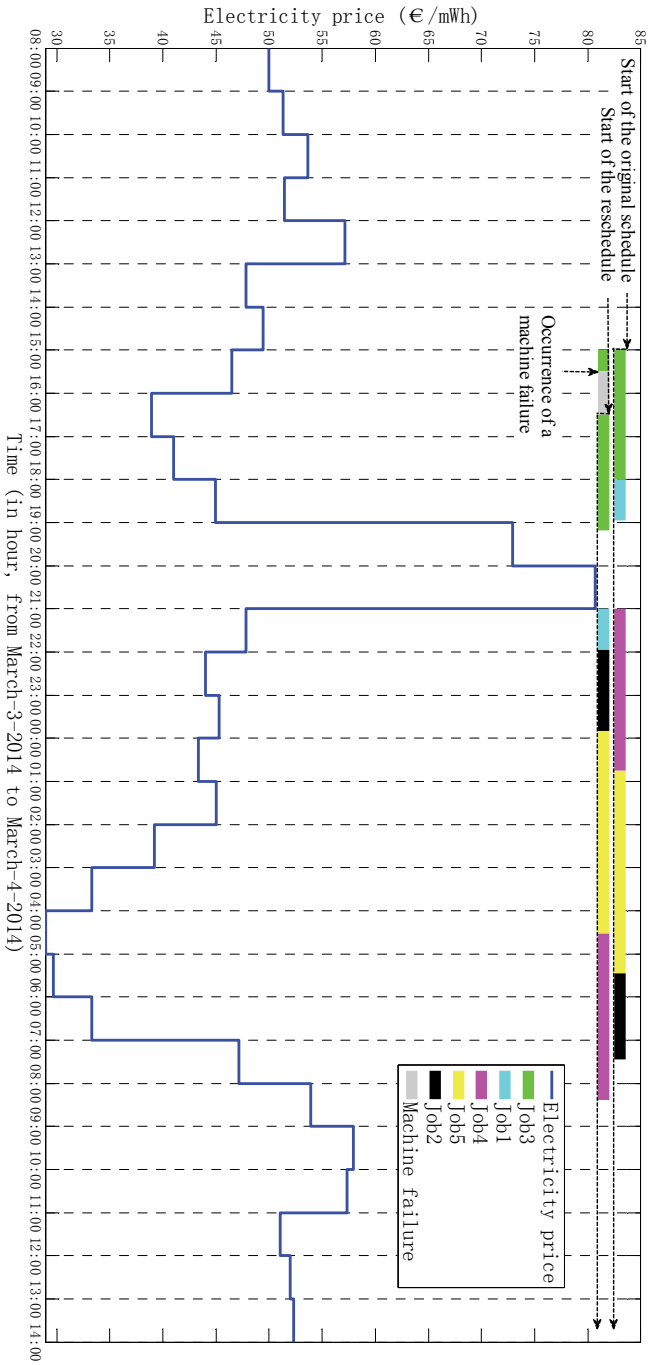


Figure 2.12: Energy-aware production rescheduling upon the occurrence of a machine failure during the execution of a schedule on this machine

Table 2.12: New grinding jobs that urgently arrive during the execution of a production schedule

Job ID	Number of steel workpieces	Required production time (grinding + dressing)
6	80	2625 (43m45s)
7	70	2375 (39m35s)
8	60	2000 (33m20s)
9	50	1625 (27m5s)
10	40	1250 (20m50s)

small (i.e., 5%), in comparison to the large variation of the electricity price around its mean level (i.e., 16%). This further indicates the energy-cost-effectiveness of the proposed method.

2.8.5.2 Unforeseen Events with Positive Influence

Cancellation of a customer order is investigated as an example of unforeseen events that have positive influence on the energy cost effectiveness of a production schedule. If an order is canceled during the execution of a job, this job is supposed to continue to be completed before the start of the updated schedule. Job5 is assumed to be canceled at 3 h during the execution of the baseline schedule. The same baseline schedule and scheduling time span in Section 2.8.5.1 are used in this investigation.

The updated schedule (Figure 2.14) is obtained by following the steps "O6-O7-O1-O2-O3-O4-O5" in the heuristic (Figure 2.4). The corresponding energy cost is 3.80 euro. For comparison, the AEAP schedule is assumed to follow the same schedule (i.e., the same job sequence and job start time) even when job5 is canceled. Its energy cost is 4.11 euro. The energy cost reduction ratio is thus 7.5%, indicating an interesting potential for energy cost saving.

In addition, if the baseline schedule is executed by keeping its job sequence and start time with the same order cancellation, the energy cost is 3.89 euro. It is higher than the energy cost of the updated schedule (3.80 euro), due to the low-priced periods released by job5. This demonstrates the robust energy-cost-effectiveness of the full rescheduling policy.

2.8.5.3 Unforeseen Events with Neutral Influence

The variability of the electricity price is viewed as a special unforeseen events that has neutral influence on the energy cost effectiveness of a schedule (Section 2.7). To this end, the RTP data from Belpex during a whole year (4-Nov-2014 to 3-Nov-2015) was used. The time duration of energy-cost-aware scheduling is set to

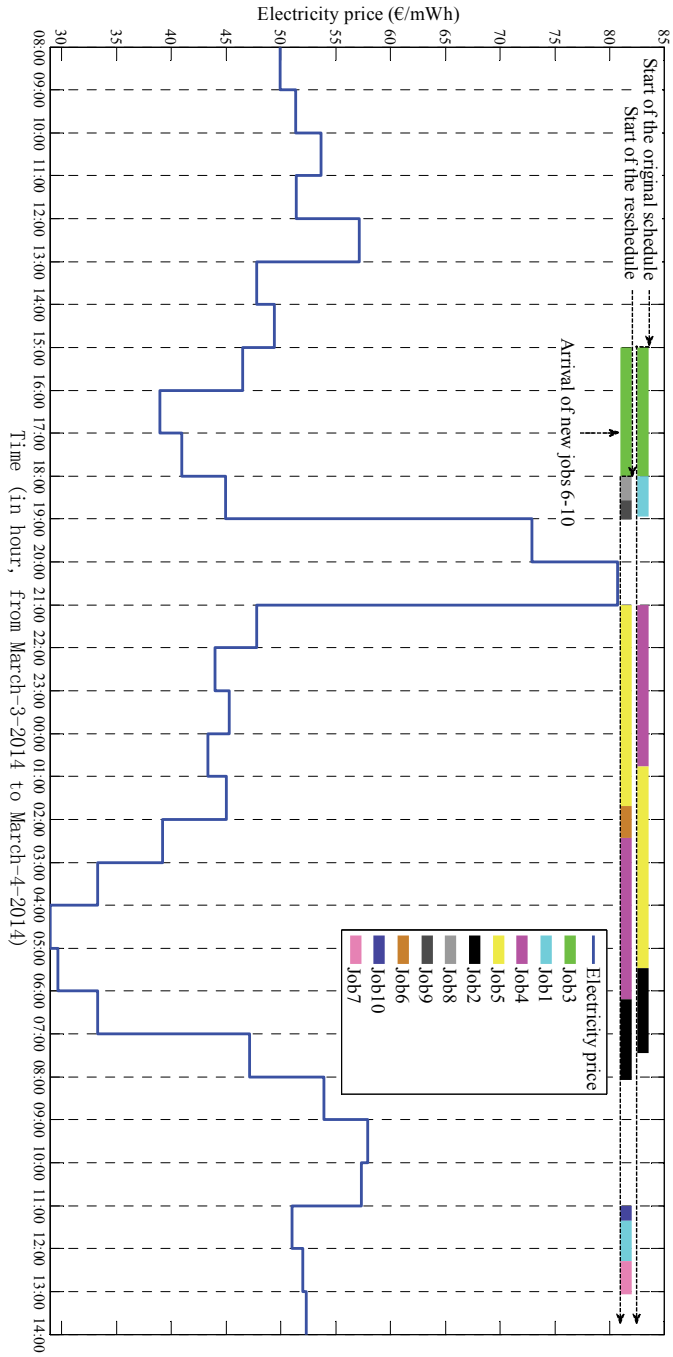


Figure 2.13: Energy-aware production rescheduling upon the urgent arrival of new jobs during the execution of a schedule on the machine

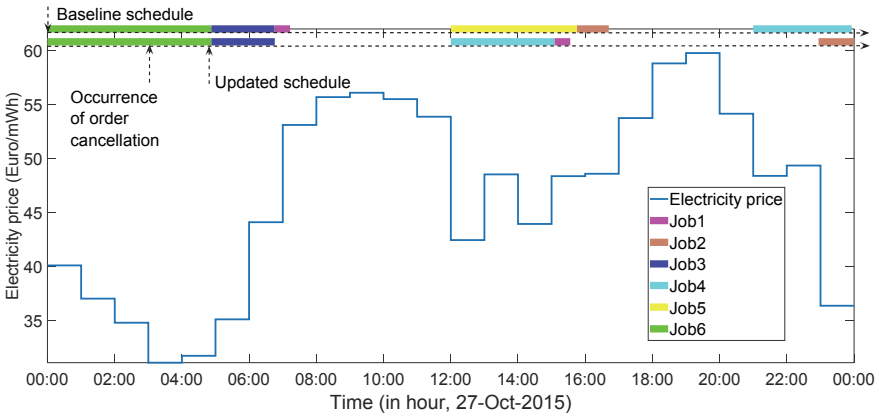


Figure 2.14: Energy-aware production rescheduling upon an urgent order cancellation during the execution of a schedule on the machine

24 hours within the same day. So there are in total 365 sets of electricity prices to exhibit the variability of the electricity price during different scheduling time periods while having the same time span length.

As a result, 365 energy-cost-aware schedules for 356 different days are obtained by iteratively following the closed loop "O1-O4-O5-O6-O7-O1-O4..." in the heuristic (Figure 2.4). In comparison, the energy costs of the schedules AEAP, "as-late-as-possible" (ALAP), "start-from-nAM" (SFnAM) are respectively calculated on each day. An SFnAM schedule means that the entire production starts at nAM of a day (n = 1, 2, ..., 9) , and continues without idling between jobs until all the jobs are completed. They serve as intermediate cases between the AEAP and ALAP schedules, in terms of the start time of the whole production.

As indicated by the box plot in Figure 2.15 and by the corresponding statistics in Table 2.13, the schedule provided by the proposed scheduling method (i.e., optimized schedule) achieves an energy cost saving/reduction ratio which is averaged between 6% and 19%. The schedule tends to save more energy cost if the benchmark schedule starts the whole production in the early morning (6AM to 9 AM). This is explained by the electricity price profile which tends to rise from the valley during this period, and is followed by peaks (e.g., the electricity price profile in Figure 2.14).

The maximal energy cost reduction ratio tends to reach 50% or even more (Figure 2.15). The minimal ratio tends to be slightly higher than 0 in most cases and to be slightly lower than 0 in a few cases (i.e., schedules starting from the night, i.e., 0 AM, 1 AM, 2AM, and 3 AM). The latter phenomenon is again explained by the electricity price profile, which usually has a large valley from the beginning (e.g., the price profile in Figure 2.14). These schedules starting from the night

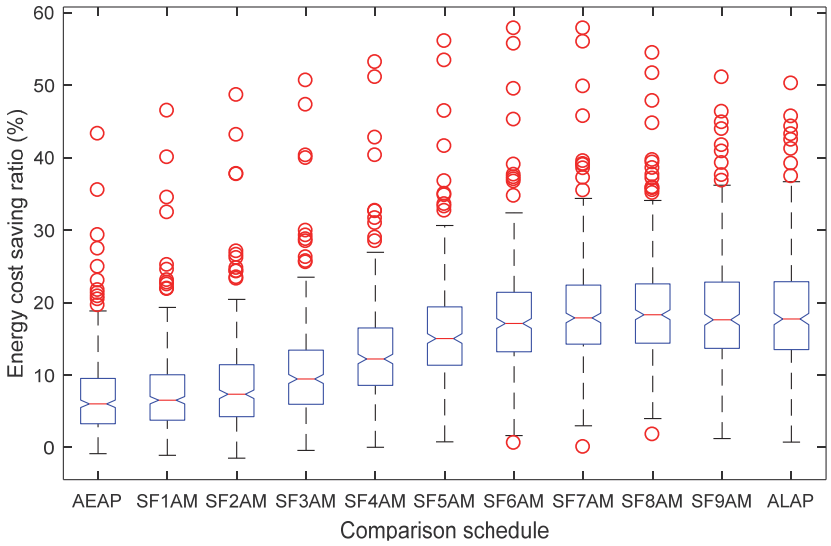


Figure 2.15: Statistical energy cost saving ratio of the energy-aware production scheduling method under RTP (real-time electricity pricing) at 365 different days. The central red line, upper edge, and lower edge of each box are median, 75th percentile (Q3), and 25th percentile (Q1), respectively. The red circles are outliers. The whiskers (black dash lines) are set to 1.5(Q3-Q1). AEAP: as-early-as-possible schedule, SFnAM: start-from-nAM schedule (n=1, 2, ..., 9), ALAP: as-late-as-possible schedule.

thus naturally make use of the valley to achieve a low energy cost, which is comparable to the energy cost of the optimized schedule. Furthermore, the 25% best ratio tends to achieve 23%, while the 75% best ratio has the trend to approach 15% (Figure 2.15). All the outliers are beyond the 1.5 interquartile range of Q3 (75th percentile), except the three that are around 0. Overall, these statistics demonstrate the high potential of the proposed scheduling method for helping factories to reduce their energy cost of production.

2.9 Discussions and conclusions

As the final section of this chapter, the strengths, weaknesses, and application of this proposed scheduling method will be first discussed. Then the conclusion and outlook will be performed.

Table 2.13: Statistical energy cost reduction ratio of the proposed energy-cost-aware production scheduling method (Q3: 75th percentile, Q1: 25th percentile)

Schedule	Max(%)	Min(%)	Median(%)	Q3	Q1	Outlier(%)
AEAP	43.24	-0.86	6.02	9.53	3.28	3.02
SF1AM	46.43	-1.09	6.52	10.02	3.77	3.02
SF2AM	48.59	-1.47	7.34	11.43	4.25	3.30
SF3AM	50.60	-0.41	9.45	13.43	5.96	3.02
SF4AM	53.13	0.02	12.21	16.48	8.57	2.75
SF5AM	56.03	0.77	15.03	19.39	11.34	2.75
SF6AM	57.79	0.59	17.11	21.41	13.21	3.02
SF7AM	57.80	0.04	17.87	22.41	14.26	3.02
SF8AM	54.39	1.77	18.31	22.58	14.39	3.85
SF9AM	51.03	1.23	17.62	22.82	13.67	2.47
ALAP	50.19	0.74	17.72	22.88	13.50	2.20

2.9.1 Discussions

Compared to the existing energy-aware production scheduling research, the work presented in this chapter fills the gap in three aspects. Firstly, it proposes a systematic method to perform empirical energy modeling and simulation. This makes energy-aware production scheduling more concrete when a lot of empirical shop floor energy data are available in the era of Internet of Things. Secondly, it formulates a scheduling model that can consider a wide range of electricity pricing schemes and explicitly convert energy consumption to energy cost for production. By being linked to the financial issue, this is more realistic in contrast to most existing research that focuses on energy efficiency and sustainability which are somewhat vague and abstract for manufacturers. Thirdly, it proposes a rescheduling framework such that unforeseen events/disruptions can be handled in a reactive and energy-cost-effective manner. This makes the proposed method more practical when applying to a shop floor that may have various disruption when producing according to the production schedule.

However, this proposed work also has some observed weakness. Firstly, although the convergence trend analysis has been performed in Figure 2.10, the CPU time of the proposed scheduling method is not thoroughly investigated. In practice, the time is a crucial factor for production scheduling, especially when reactive rescheduling is performed to handle disruptions. Secondly, the reactive rescheduling risks to increase the nervousness of a cyber-physical production system. If disruptions frequently occur, a production schedule has to be frequently altered such that a production system has to be frequently set up or reconfigured. This may reduce the life time of a production system. Thirdly, only energy cost

is minimized in this work. Other important cost parts for production, e.g., labor cost, are not yet considered. As energy cost is a novel metric introduced for the conventional production scheduling model, it would be more realistic to take into account its potential interaction and constraint with more conventional production metrics besides makespan and due date, e.g., tardiness and earliness. The first and third weakness will be tackled in Chapter 3.

This proposed work can be applied to automatically design production schedules for a single manufacturing/re-manufacturing machine and a production line that has one major energy consumption process. The more volatile electricity prices for a factory, the higher contribution this work will make.

2.9.2 Conclusions and Outlook

Under real-time electricity pricing, an energy-aware production modeling, simulation, scheduling, and rescheduling method is proposed in this chapter. Its energy cost effectiveness has been demonstrated using empirical energy data from a surface grinding machine and two real electricity price schemes. An empirical sensitivity analysis shows that in average it can reduce the energy cost for production on a single machine by 10% compared to conventional production schedules without energy awareness.

The extension work has been performed in this PhD thesis and will be presented in Chapter 3 and Chapter 4. (1) In terms of practical problem modeling, more conventional yet important production aspects, e.g., labor, will be jointly modeled with energy consumption/cost, in order to make the investigation more realistic and unlock more production cost reduction potential. (2) Regarding problem complexity and scale, the single-machine scheduling configuration will be extended to multi-machine configurations, and the number of scheduling time slots and jobs will increase to an even larger scale. (3) Concerning solution algorithm design, more efficient optimization algorithms can be proposed, instead of simply tailoring a conventional genetic algorithm to the model. The efficiency of an optimization algorithm can be evaluated in terms of solution quality (the degree to minimize/maximize the objective), required CPU time, and scalability.

References

[1] C. C. Lin, D. J. Deng, W. Y. Liu, and L. Chen. *Peak Load Shifting in the Internet of Energy With Energy Trading Among End-Users*. IEEE Access, 5:1967–1976, 2017.

[2] Matúš Mišík. *On the way towards the Energy Union: Position of Austria, the Czech Republic and Slovakia towards external energy security integration*. Energy, 111(Supplement C):68 – 81, 2016.

[3] Linas Gelazanskas and Kelum A.A. Gamage. *Demand side management in smart grid: A review and proposals for future direction*. Sustainable Cities and Society, 11:22 – 30, 2014.

[4] Hubert Hadera, Iiro Harjunkoski, Guido Sand, Ignacio E. Grossmann, and Sebastian Engell. *Optimization of steel production scheduling with complex time-sensitive electricity cost*. Computers & Chemical Engineering, 76:117 – 136, 2015.

[5] Brandon Davito, Humayun Tai, and Robert Uhlaner. *The smart grid and the promise of demand-side management*. Technical report, McKinsey & Company, 2010.

[6] Jesus A. Cardenas, Leopoldo Gemoets, Jose H. Ablanedo Rosas, and Robert Sarfi. *A literature survey on Smart Grid distribution: an analytical approach*. Journal of Cleaner Production, 65:202 – 216, 2014.

[7] C. A. Babu and S. Ashok. *Peak Load Management in Electrolytic Process Industries*. IEEE Transactions on Power Systems, 23(2):399–405, May 2008.

[8] Tobias Küster, Marco Lützenberger, Daniel Freund, and Sahin Albayrak. *Distributed evolutionary optimization for electricity price Responsive manufacturing using multi-agent system technoloy*. International Journal on Advances in Intelligent Systems, 6(1&2), 2013.

[9] Dongsik Jang, Jiyong Eom, Moon Gyu Kim, and Jae Jeung Rho. *Demand responses of Korean commercial and industrial businesses to critical peak pricing of electricity*. Journal of Cleaner Production, 90:275 – 290, 2015.

[10] Shyi-Min Lu, Ching Lu, Kuo-Tung Tseng, Falin Chen, and Chen-Liang Chen. *Energy-saving potential of the industrial sector of Taiwan*. Renewable and Sustainable Energy Reviews, 21:674 – 683, 2013.

[11] Eoin O'Driscoll and Garret E. O'Donnell. *Industrial power and energy metering – a state-of-the-art review*. Journal of Cleaner Production, 41:53 – 64, 2013.

[12] Fadi Shrouf and Giovanni Miragliotta. *Energy management based on Internet of Things: practices and framework for adoption in production management*. Journal of Cleaner Production, 100:235 – 246, 2015.

[13] Konstantin Vikhorev, Richard Greenough, and Neil Brown. *An advanced energy management framework to promote energy awareness*. Journal of Cleaner Production, 43:103 – 112, 2013.

[14] Timothy Gutowski, Jeffrey Dahmus, and Alex Thiriez. *Electrical energy requirements for manufacturing processes*. In Electrical energy requirements for manufacturing processes, pages 623–628, 2006.

[15] Anton Dietmair and Alexander Verl. *A generic energy consumption model for decision making and energy efficiency optimisation in manufacturing*. International Journal of Sustainable Engineering, 2(2):123–133, 2009.

[16] Diaz Nancy, Redelsheimer Elena, and Dornfeld David. *Energy Consumption Characterization and Reduction Strategies for Milling Machine Tool Use*, pages 263–267. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[17] Karel Kellens, Wim Dewulf, Michael Overcash, Michael Z. Hauschild, and Joost R. Duflou. *Methodology for systematic analysis and improvement of manufacturing unit process life-cycle inventory (UPLCI)—CO2PE! initiative (cooperative effort on process emissions in manufacturing). Part 1: Methodology description*. The International Journal of Life Cycle Assessment, 17(1):69–78, Jan 2012.

[18] Karel Kellens, Wim Dewulf, Michael Overcash, Michael Z. Hauschild, and Joost R. Duflou. *Methodology for systematic analysis and improvement of manufacturing unit process life cycle inventory (UPLCI) CO2PE! initiative (cooperative effort on process emissions in manufacturing). Part 2: case studies*. The International Journal of Life Cycle Assessment, 17(2):242–251, Feb 2012.

[19] Wen Li, Sami Kara, and Bernard Kornfeld. *Developing unit process models for predicting energy consumption in industry: A Case of extrusion line*, pages 147–152. Springer Singapore, Singapore, 2013.

[20] Eberhard Abele, Christian Eisele, and Sebastian Schrems. *Simulation of the Energy Consumption of Machine Tools for a Specific Production Task*, pages 233–237. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[21] Philipp Eberspächer, Philipp Schraml, Jan Schlechtendahl, Alexander Verl, and Eberhard Abele. *A Model- and Signal-based Power Consumption Monitoring Concept for Energetic Optimization of Machine Tools*. Procedia CIRP, 15:44 – 49, 2014. 21st CIRP Conference on Life Cycle Engineering.

[22] Chen-Fu Chien, Chia-Yu Hsu, and Chih-Wei Hsiao. *Manufacturing intelligence to forecast and reduce semiconductor cycle time*. Journal of Intelligent Manufacturing, 23(6):2281–2294, Dec 2012.

[23] Saeede Ajorlou and Issac Shams. *Artificial bee colony algorithm for CONWIP production control system in a multi-product multi-machine manufacturing environment*. Journal of Intelligent Manufacturing, 24(6):1145–1156, Dec 2013.

[24] Bo Huang, Rongxi Jiang, and Gongxuan Zhang. *Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions*. Computers & Industrial Engineering, 71:21 – 26, 2014.

[25] Agnes Pechmann, Ilka Schöler, and Rens Hackmann. *Energy Efficient and Intelligent Production Scheduling – Evaluation of a New Production Planning and Scheduling Software*, pages 491–496. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[26] Kuei-Tang Fang and Bertrand M.T. Lin. *Parallel-machine scheduling to minimize tardiness penalty and power cost*. Computers & Industrial Engineering, 64(1):224 – 234, 2013.

[27] Hao Luo, Bing Du, George Q. Huang, Huaping Chen, and Xiaolin Li. *Hybrid flow shop scheduling considering machine electricity consumption cost*. International Journal of Production Economics, 146(2):423 – 439, 2013.

[28] Yong Wang and Lin Li. *Time-of-use based electricity demand response for sustainable manufacturing systems*. Energy, 63:233 – 244, 2013.

[29] Hao Zhang, Fu Zhao, Kan Fang, and John W. Sutherland. *Energy-conscious flow shop scheduling under time-of-use electricity tariffs*. CIRP Annals - Manufacturing Technology, 63(1):37 – 40, 2014.

[30] Ying Liu, Haibo Dong, Niels Lohse, Sanja Petrovic, and Nabil Gindy. *An investigation into minimising total energy consumption and total weighted tardiness in job shops*. Journal of Cleaner Production, 65:87 – 96, 2014.

[31] Yan He, Yufeng Li, Tao Wu, and John W. Sutherland. *An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops*. Journal of Cleaner Production, 87(Supplement C):245 – 254, 2015.

[32] Ying Liu, Haibo Dong, Niels Lohse, and Sanja Petrovic. *Reducing environmental impact of production during a Rolling Blackout policy – A multi-objective schedule optimisation approach*. Journal of Cleaner Production, 102:418 – 427, 2015.

[33] Fadi Shrouf, Joaquin Ordieres-Meré, Alvaro García-Sánchez, and Miguel Ortega-Mier. *Optimizing the production scheduling of a single machine to minimize total energy consumption costs*. Journal of Cleaner Production, 67:197 – 207, 2014.

[34] Wei-Wei Cui, Zhiqiang Lu, and Ershun Pan. *Integrated production scheduling and maintenance policy for robustness in a single machine*. Computers & Operations Research, 47:81 – 91, 2014.

[35] X. Gong, T. De Pessemier, W. Joseph, and L. Martens. *A power data driven energy-cost-aware production scheduling method for sustainable manufacturing at the unit process level*. In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–8, Sept 2016.

[36] *MTConnect Standard*, 2018.

[37] Karel Kellens, Goncalo Costa Rodrigues, Wim Dewulf, and Joost R. Duflou. *Energy and resource efficiency of laser cutting processes*. Physics Procedia, 56:854 – 864, 2014. 8th International Conference on Laser Assisted Net Shape Engineering LANE 2014.

[38] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[39] Ada Che, Yizeng Zeng, and Ke Lyu. *An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs*. Journal of Cleaner Production, 129(Supplement C):565 – 577, 2016.

[40] X. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan. *A Multi-Facet Survey on Memetic Computation*. IEEE Transactions on Evolutionary Computation, 15(5):591–607, Oct 2011.

[41] Michael L. Pinedo. *Scheduling - Theory, Algorithms, and Systems*. Springer, 5 edition, 2016.

[42] Nils Weinert and Christian Mose. *Investigation of Advanced Energy Saving Stand by Strategies for Production Systems*. Procedia CIRP, 15:90 – 95, 2014. 21st CIRP Conference on Life Cycle Engineering.

[43] Joost R. Duflou, Karel Kellens, Tom Devoldere, Wim Deprez, and Wim Dewulf. *Energy related environmental impact reduction opportunities in machine design: case study of a laser cutting machine*. International Journal of Sustainable Manufacturing, 2(1):80–98, 2010.

[44] A. Dietmair and A. Verl. *Energy Consumption Forecasting and Optimisation for Tool Machines*. MM Science Journal, 1:62–67, 2009.

[45] R. Lakshmi and K. Vivekanandan. *Performance analysis of a novel crossover technique on permutation encoded genetic algorithms*. In 2014 International Conference on Advances in Engineering and Technology (ICAET), pages 1–4, May 2014.

[46] T. Tolio, M. Urgo, and J. Váncza. *Robust production control against propagation of disruptions*. CIRP Annals, 60(1):489 – 492, 2011.

[47] Lixin Tang, Wenxin Liu, and Jiyin Liu. *A neural network model and algorithm for the hybrid flow shop scheduling problemin a dynamic environment*. Journal of Intelligent Manufacturing, 16(3):361–370, Jun 2005.

[48] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A stochasticity handling heuristic in energy-cost-aware scheduling for sustainable production*. Procedia CIRP, 48(Supplement C):108 – 113, 2016. The 23rd CIRP Conference on Life Cycle Engineering.

[49] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *An energy-cost-aware scheduling methodology for sustainable manufacturing*. Procedia CIRP, 29(Supplement C):185 – 190, 2015. The 22nd CIRP Conference on Life Cycle Engineering.

[50] Yong Wang and Lin Li. *Critical peak electricity pricing for sustainable manufacturing: Modeling and case studies*. Applied Energy, 175(Supplement C):40 – 53, 2016.

[51] S. Kotz N.L. Johnson and N. Balakrishnan. *Continuous univariate distributions*, volume 1. John Wiley, New York, 2 edition, 1994.

3

Energy- and Labor-Aware Single-Machine Production Scheduling

Sustainability is a crucial factor in future production systems for manufacturing enterprises to stay competitive [1]. It is the "development that meets the needs of the present, without compromising the ability of future generations to meet their own needs" [2]. When it is integrated in manufacturing enterprises, all dimensions of the triple bottom line should be followed: the economic, environmental, and social dimension [3]. Production scheduling is a promising industrial demand response (DR) approach for sustainable production [4]. It shifts flexible production loads to lower-priced periods to reduce energy cost for the same production task. However, the existing energy-aware production scheduling methods only focus on integrating energy awareness to conventional production scheduling models. They ignore the labor cost which is shift-based and follows an opposite trend of energy cost. For instance, the energy cost is lower during nights and weekends while the labor cost is higher.

To fill this gap, this chapter formulates a new and practical mixed integer linear programming (MILP) model that enables joint decision makings on energy conservation, as well as job and human worker scheduling on a single machine. Both

single-objective optimization and multi-objective optimization are investigated to solve this model. While the former is based on a genetic algorithm (GA), the latter is handled by a proposed adaptive multi-objective memetic algorithm (AMOMA) which aims to fast converge toward the Pareto front without loss in diversity. The AMOMA leverages the feedback of cross-dominance and stagnation in an evolutionary search and a prioritized grouping strategy. In this way, an adaptive balance remains between the exploration of the nondominated sorting genetic algorithm II (NSGA-II) and the exploitation of two proposed mutually-complementary local search operators, i.e., convergence-oriented tabu search[1](CTS) and diversity-oriented tabu search (DTS). An empirical case study is performed on an extrusion blow molding process in a plastic bottle manufacturer. Extensive sensitivity analyses demonstrate the economic importance of integrating labor awareness to energy-aware production scheduling. Extensive benchmarking proves the effectiveness and efficiency of AMOMA. Therefore, the modeling, solution algorithm, and empirical economic analytics provided in this chapter is foreseen to help factories perform automated and integrated energy- and labor-aware production scheduling and exploit the economic insights behind these decision makings.

Compared to the work presented in Chapter 2, this chapter has threefold additional contributions. (1) The energy-aware production scheduling model is enhanced by introducing the labor type and quantity, the work shift, the option of weekend production, machine changeovers, as well as multiple idle modes. (2) A continuous-time shift accumulation heuristic is proposed to synchronize power states and labor shifts in order to enable integrated energy and labor simulation. (3) The AMOMA is proposed which synergistically integrates convergence- and diversity-oriented tabu searches in the NSGA-II, respectively. It further adaptively coordinates the exploration and the exploitation by taking real-time feedback from a search. (4) An empirical study is performed in a Belgian plastic bottle manufacturer. On the one hand, extensive sensitivity analyses revealed a new understanding of energy-aware production scheduling: the energy cost and the labor cost should be jointly considered to reduce the overall production cost. On the other hand, extensive benchmarking proved that the proposed AMOMA can achieve fast Pareto front approximation while preserving diversity for this highly-constrained multi-objective optimization problem (MOP).

[1]Tabu search is a metaheuristic search method employing local search methods for mathematical optimization. Compared to a typical local search which is easy to be stuck in a local optimum, tabu search has two major enhancement aspects. First, at each step, worsening moves can be accepted if no improving moving is available. Second, prohibitions are introduced to discourage the search from coming back to previously-visited solutions.

3.1 Introduction

As introduced in detail by Section 2.1, the price- or time-based DR stimulates end users to adapt their electricity consumption patterns to time-sensitive electricity prices. In this way, end users could possibly reduce their energy cost and the grid stability can be enhanced by a better balanced demand-supply [5]. While price-based DR has earlier focused on residential or commercial applications, such as scheduling electrical loads of households [6] and electric vehicles charging [7], it penetrates later into the manufacturing industry, which is a major electricity consumer. The industrial price-based DR can be realized by energy-aware scheduling of production processes [8–10]. These methods perform energy-cost-effective production load shifting, by setting machines to off or standby modes in periods without active production. An energy-aware production scheduling method can simultaneously have economic, ecological, and societal impact.

From an economic perspective, energy-aware production scheduling reduces the energy cost, under a volatile energy price from the deregulated electricity markets [11]. For many power-intensive industries, the electricity cost accounts for 10-50% of the total product cost [12]. Therefore, the potential to save energy cost remains considerable.

From an ecological perspective, energy-aware production scheduling decreases greenhouse gas (GHG) emissions, of which manufacturing processes are known as the major source [13]. Some of GHG emissions are caused by unnecessary machine idling [14] and peak power consumption in the electricity grid [10], which are solvable by production scheduling.

From a societal perspective, energy-efficient production scheduling stabilizes the electricity grid by avoiding peak demand. This secures the power supply and delivery for local residents. Moreover, optimal energy utilization and reduced GHG emissions help enterprises meet sustainability compliance and regulations, improving an enterprise's reputation for public responsibility.

Production activities commonly involve human workers, which is a crucial factor that makes industrial DR more complex than residential or commercial DR. Unfortunately, to the author's knowledge and as highlighted in [15], the existing energy-aware production scheduling methods widely ignore the compromising interdependence of energy and labor costs. Shifting loads from a day to a night or a weekend for energy cost reduction is accompanied by an increased labor wage and thus a rising labor cost. Consequently, the actual production cost may rise.

According to the cost breakdown of a plastic bottle manufacturer investigated in this PhD study, this factory has a number of independent extrusion blow molding (EBM) machines, and the labor cost is over 3 times higher than the energy cost. As a result, the former is much more sensitive to load shifting than the latter. Although the portion of these two cost parts differs on a case-by-case basis, it is of economic

benefit to jointly consider both energy and labor costs in energy-aware production scheduling, and study it as a MOP, instead of only integrating energy awareness in conventional production scheduling algorithms in most existing studies.

Both electricity and labor costs were considered in the optimization of a multi-pass face milling process [16]. However, both costs were calculated using flat rates. These two cost parts were explicitly modeled in the flow shop scheduling problem under time-varying electricity and labor pricing [17]. Nevertheless, the following limitations are observed: (1) the labor rate does not vary among workers; (2) the production-prohibited period, which is often introduced by the labor shift, and its constrained influence on production operations are not modeled; (3) the exhaustive search is a rude solution method with poor scalability. A similar problem was investigated in [18] in a single objective optimization manner. Consequently, the trade-off relations in these cost parts and other important production metrics were not quantified, besides the aforementioned ignorance of production-prohibited periods.

Multi-objective evolutionary algorithms (MOEAs) are suitable to solve multi-objective production scheduling problems [19], as they are characterized by finding high-quality solutions in reasonable time, fitting the requirement of production scheduling [20]. Scalarization-based MOEAs [21] transform a MOP to a single-objective optimization problem by summing weighted objectives in one fitness function. Nonetheless, it is problematic to assign proper weights for Pareto front approximation. Comparatively, domination-based MOEAs have been widely proven to be effective, among which NSGA-II [22] is highly representative.

Beyond using a decent NSGA-II to produce nondominated solutions[2] [23], recent studies hybridize with one or more local searches to accelerate the convergence rate toward the Pareto front without loss in diversity [24]. While a genetic search explores the solution space for potential regions, a local search exploits these regions by incorporating domain-specific knowledge on how a solution can be further improved. Such a hybrid is named as a memetic algorithm (MA) [25]. In simple MAs, domain knowledge is only captured and incorporated once by human experts at the design phase [26]. Adaptive MAs additionally integrate knowledge on how an instance of MA (MA on the fly) is self-reconfigurable to better suit the problem when a search progresses [27]. Compared to employing MAs for many unconstrained optimization problems, few studies tackled multi-objective constrained problems by adaptive MAs [27, 28].

The remainder of this chapter is organized as follows. Section 3.2 gives the literature review and proposes research questions to be investigated in this chapter. Section 3.3 describes the energy- and labor-aware production scheduling problem at the unit process level. Section 3.4 presents the method to integrated energy and

[1] In multi-objective optimization, a solution to a problem is called nondominated or Pareto optimal, if none of the objective values can be improved without degrading one or more other objective values.

labor for integrated simulation. Section 3.5 describes the AMOMA in detail. Section 3.6 introduces the empirical data from a Belgian plastic bottle manufacturer as a case study in this chapter. Section 3.7 exhibits the sensitivity analysis results by genetic algorithm (GA)-based single-objective optimization. Section 3.8 demonstrates the effectiveness and efficiency of AMOMA in solving this problem in a multi-objective optimization manner. Section 3.9 discusses the results and draws the conclusions.

3.2 Literature Review

3.2.1 Energy-Aware Production Scheduling

The powering-on/-off mechanism is an intuitive idea to enhance energy efficiency via production scheduling. It prevents machines from consuming energy when there are no active production jobs. This idea was first described in [29]. Furthermore, a multi-objective genetic algorithm was utilized to minimize energy consumption and total completion time of a single machine [30]. In addition to reducing non-cutting energy consumption, the machining energy of machine tools was characterized in [31]. They minimized the joint non-cutting and cutting energy by sequencing the feature processing order of a part. Despite these efforts, the economic impact is vague, since energy consumption was not linked to the energy cost.

Shrouf et al. [32] considered the volatile electricity price from the spot market in a single-machine scheduling model. Production loads were shifted to low-priced periods. However, a lack of job sequencing capability locks the energy cost saving potential of this idea. The authors further proposed to use Internet of Things (IoT) technologies for industrial energy management [33], but gave no implication on how to link empirical energy data to the scheduling model.

These gaps were filled in [10]. Finite state machines (FSMs, or automata) were utilized to build an energy model whose power profiles were extracted from measurements. Job sequencing and reactive rescheduling upon disruptions during the execution of a schedule were also introduced in the scheduling model. The energy-cost-effectiveness was validated on a surface grinding process, and further demonstrated with various electricity pricing schemes [34], including time-of-use pricing (ToUP), real-time pricing (TRP), and critical peak pricing (CPP). Numerical experiments showed that a higher electricity cost saving ratio is contributed by prolongation of makespan. To specifically reduce the energy cost under ToUP, a greedy insertion heuristic was proposed in [35] for a single machine scheduling model, such that it yielded high-quality solutions within 10 seconds even for the instance with 5000 jobs. [36] further investigated the same scheduling problem under the cases of uniform and scalable machine speeds.

Energy-efficient production scheduling can be found in other shop floor configurations, though most of them are not explicitly linked to the energy cost. A parallel machine scheduling problem was investigated in [37]. Machines differ in energy consumption and discharged pollutants. The energy cost and pollutant clean-up cost were modeled as hard constraints, while the objective was to minimize the makespan. In [38], a flow shop scheduling problem was studied under ToUP electricity tariffs. They revealed the trade-off between reducing electricity cost and decreasing CO_2 emissions. A hybrid flow shop floor configuration was involved in [39], where the ant colony-based scheduling method shifted loads under ToUP. The electricity cost was minimized considering the trade-off with the makespan. A job shop energy-efficient scheduling problem was studied in [14]. Energy consumption was decreased by turning off underutilized machines, accounting for the trade-off with total weighted tardiness. A flexible job-shop scheduling problem was investigated in [40], where the optimization objective was to minimize the total completion time, maximize the total availability of the system, and minimize total energy cost of production and maintenance operations. In [41], an energy saving method was proposed for flexible job shops. This method optimizes not only the operation sequence for reducing idle energy consumption, but also the machine tool selection for decreasing the energy consumption for machining operations. A reactive rescheduling method was proposed in [42] to handle unforeseen events during the execution of a schedule. As an alternative method for disruptions handling, a dynamic game theory- and IoT-based two-layer scheduling method was proposed in [43]. Consequently, this method achieved real-time multi-objective flexible job shop scheduling . Upon a machine's active request for processes during an idle period, the real-time scheduling task pool outputs a schedule based on the real-time machine status, optimizing the makespan, total workload, and energy consumption.

Furthermore, some recent studies are observed to perform economic benefit analysis of energy-aware production planning and scheduling. In [15], both electricity consumption (kWh) and peak demand (kW) were combined to calculate the electricity cost of manufacturing systems. Using the formulated model, a saving of up to 24.8% of the per-product electricity cost was estimated by adopting the ToUP rates. The additional consideration of the human factor was highlighted as a future work direction, since a time-shifted schedule with extended night hours must be paid for with a premium. However, no concrete method was proposed. A preliminary case study revealed that although the incorporation of labor increased the energy cost by 9%, it reduced the joint energy and labor cost by 12%, due to the minor proportion (3%) of energy cost in this joint cost [44]. A two-dimensional energy performance measure was proposed in [45]. A sensitivity study was performed on energy-aware single- and multi-machine production planning according to the 3-year RTP and ToUP data as well as load profile data. This investigation

Table 3.1: Literature analysis of recent energy-efficient production scheduling methods

Reference	Shop floor configuration	Energy model		Labor model		Optimization objective		Problem size[a]	
		Assumed	Empirical	Shift	Personnel	Economic	Non-economic	Small	Large
[29]	Single-machine	✓		No	No		✓	✓	
[30]	Single-machine	✓		No	No		✓		✓
[31]	Single-machine		✓	No	No		✓	✓	
[32]	Single-machine	✓		No	No	✓		✓	
[10]	Single-machine		✓	No	No	✓			✓
[35]	Single-machine	✓		No	No	✓			✓
[36]	Single-machine	✓		No	No	✓		✓	
[37]	Parallel-machine	✓		No	No		✓	✓	
[38]	Flow-shop	✓		No	No	✓	✓	✓	
[39]	Hybrid flow-shop	✓		No	No	✓	✓	✓	
[14]	Job-shop	✓		No	No		✓	✓	
[40]	Flexible job-shop	✓		No	No	✓	✓	✓	
[41]	Flexible job-shop	✓		No	No		✓	✓	
[43]	Flexible job-shop	✓		No	No		✓	✓	

[a]The size of optimization problems is characterized by the number of jobs and time slots in this study. The size of problems solved by dispatching rules is excluded.

found that ToUP lead to lower total economic loss. The interrelationships between the production target, speed change, energy consumption, and electricity cost were investigated in [46]. Using the proposed multi-objective optimization method, a manufacturing system was demonstrated to be more eco-friendly without a substantial increase in the electricity cost.

3.2.2 Gaps in Energy-Aware Production Scheduling Research

Table 3.1 analyzes these representative studies and unveils the following gaps. Firstly, despite these emerging investigations, energy efficiency has never been jointly optimized with the labor (regarding the time associated with shifts as well as the type and quantity of personnel), despite the highlighted importance of labor consideration in production under dynamic electricity prices [15]. For instance, an 8-h shift was involved in [29]. But it rather defined the overall scheduling time span, instead of introducing multiple continuous shifts which are not only a practical constraint but also unlock more optimization potential for energy-cost-effective load shifting. As the makespan tends to be prolonged in these scheduling methods, the number of shifts and the period with higher labor wage would increase. Therefore, the reduced energy cost may be compensated by the rising labor cost.

Secondly, empirical power data has seldom been utilized, though IoT-enabled energy monitoring has penetrated the factories to enable empirical energy awareness and energy efficiency measures [33, 47]. The energy consumption is only an assumed constraint in a majority of these studies, such as on/off mode and unit energy consumption or cost for production operations, without complete or empirical modeling of the energy consumption behavior of a machine. Consequently, this simplifies the real problem. For example, to reduce the search space for an energy-efficient scheduling solution in [39], each operation was assumed to start immediately after the previous operation. This goes against the philosophy of energy-efficient production scheduling which may insert idle or off periods between operations, thereby removing many potential solutions in the search space.

Thirdly, energy efficiency is not often linked to economic benefits for factories, though such an explicit link embodies the impact of energy-efficient production scheduling. Fourthly, the problem size remains small regarding the number of jobs (smaller than 10) and time slots (several to dozens). Although some investigations on single-machine scheduling tried to handle a large problem size, studies on the other shop floor configurations ignored this scalability issue.

3.3 Energy- and Labor-Aware Scheduling Model

The problem is to perform cost-effective load shifting of a single machine under real-time pricing (RTP) and before a due time (DT), aiming to minimizing the

following bi-objectives:

$$\min_{\pi, STJ_i, s, pt, sh} (ELC,\ C_{max}) \tag{3.1}$$

where *ELC* is the sum of total energy cost (*TEC*) and total labor cost (*TLC*), and C_{max} is the makespan. Following the three-field notation [48], this problem is denoted as $\{1,\ \text{RTP}\}|\text{split}|\{ELC,\ C_{max}\}$, where '$\{1,\ \text{RTP}\}$' represents single-machine scheduling under RTP; 'split' indicates that a job/changeover can be split by production-prohibited periods, e.g., weekends during which the production stops; the last field specifies the objectives.

While the existing relevant models focus on enabling energy awareness in scheduling, energy- and labor-related decision variables are integrated in this model: (1) the job sequence (π), (2) the job start time (STJ_i^n) considering the production-prohibited periods which may split a job into multiple subparts, (3) the machine power states (s), (4) the number of each type of personnel (pt), and (5) the labor shift (sh). pi and STJ_i^n define job sequencing and timing, respectively; s assigns machine power states for job processing and idling (off and standby) between jobs; both pt and sh are adaptively determined according to the scheduled production. This integrated decision making is crucial to reduce the overall production cost, without neglecting the dependency between the TEC and the other important production metrics.

The production cost often consists of the machine depreciation cost, the material cost, the energy cost, the labor cost. It should be minimized in order to maximize the profit of a manufacturer. The number, type, release time, and due time of production jobs, which are predetermined by production planning [49], are the input variable for this scheduling problem. Therefore, the machine depreciation is considered as a fixed cost on the fixed short scheduling horizon with the fixed amount of production. The material cost increases linearly with the amount of production and cannot be influenced by the manufacturer [17]. Comparatively, the energy and labor costs are the key production cost parts whose variance is directly linked to the production. As a result, this energy- and labor-aware production scheduling problem enables profit maximization.

Compared to the prevalent residential DR studies [50, 51] that basically determine the simple operations (on/off) of household appliances in each scheduling time slot, this industrial DR model has threefold contributions. (1) Besides the decision-making of multiple machine operations (processing, on, off, and multiple idle modes), it integrates job sequencing and timing as well as human worker and labor shift planning. (2) The time granularity is reduced (second-scale or even smaller) for finer-grained scheduling and analytics. (3) A complete state-based energy model is employed for a more realistic consideration of the energy consumption behavior of a machine.

To investigate the difference between multi- and single-objective optimization in solving this problem, three single-objective functions are additionally defined:

$$\min_{s,\pi,STJ_i,pt}(ELC) \tag{3.2}$$

$$\min_{s,\pi,STJ_i,pt}(TEC) \tag{3.3}$$

$$\min_{s,\pi,STJ_i,pt}(TLC) \tag{3.4}$$

where Equation (3.2) aims to minimize the *ELC*, i.e., the joint *TEC* and *TLC*, Equation (3.3) minimizes the *TEC*, and Equation (3.4) minimizes the *TLC*.

Table 3.2 summarizes the symbols that are used in this mixed integer programming (MIP) model [18]. The following subsections will introduce this model.

Table 3.2: Nomenclature of the proposed energy- and labor-aware production scheduling model (italic: general variables or variables to be determined by the model, non-italic: input variables)

Parameter	Notation
C_{max}	Makespan of the entire production
CC_i	Electricity cost for performing the *i*-th machine changeover
CI_i	Electricity cost for performing the *i*-th machine idling
CJ_i	Electricity cost for processing the *i*-th job
D	Duration of the electricity pricing slot
D_{poff}	Duration to power off a machine
D_s	Duration of machine power state *s*
DC_i	Duration of the *i*-th machine changeover
DJ_i	Duration to process the *i*-th job
DT	Common due time of all jobs
ELC	Joint energy and labor cost for processing all jobs
EP_{ts}	Electricity price on the *ts*-th time slot
ETC_i^n	End time in δt of the *n*-th subpart of the *i*-th changeover
ETJ_i^n	End time in δt of the *n*-th subpart of the *i*-th job
$ETSC_i^n$	End time in electricity pricing slots of the *n*-th subpart of the *i*-th machine changeover
$ETSJ_i^n$	End time in electricity pricing slots of the *n*-th subpart of the *i*-th job
N_J	Number of jobs to be scheduled
NSC_i	Number of subparts of the *i*-th changeover

Table 3.2: Continuation of Table 3.2 on the previous page

Parameter	Notation
NSJ_i	Number of subparts of the *i*-th job
NSH	Number of labor shifts in the scheduling horizon without considering the match for production loads
P_p	Power consumption of the machine power state *Production*
P_s^t	Power consumption of the machine power state *s* at time *t* in δt
pt	Type of personnel required in shift *sh*
PT	Set of personnel types (e.g., operator and quality checker)
RT	Common release time of all jobs
S_c	Sequence of power states for a machine changeover
SH	Set of labor shift types on a weekday or a day on weekends
sh	Labor shift corresponding to time *t* in δt
SI_{ij}	SI_j following the *i*-th job
SI_j	Sequence of power states for switching to, staying at, and recovering from the *j*-th machine idle mode
S_o	Sequence of power states for switching to, staying at, and recovering from the *Off* state between contiguous jobs
STC_i^n	Start time in δt of the *n*-th subpart of the *i*-th changeover
STJ_i^n	Start time in δt of the *n*-th subpart of the *i*-th job
$STSC_i^n$	Start time in electricity pricing slots of the *n*-th subpart of the *i*-th changeover
$STSJ_i^n$	Start time in electricity pricing slots of the *n*-th subpart of the *i*-th job
t	Absolute time or clock time
ts	Time in electricity pricing slots
TEC	Total energy cost for processing the jobs
TLC	Total labor cost for processing the jobs
W_{sh}^{pt}	Labor wage of the personnel type *pt* in the shift *sh*
δsh	Duration of one labor shift
δt	Scheduling time slot
θ_{sh}^{pt}	Boolean indicator for the personnel type *pt* in the shift *sh*
λ	Boolean production-prohibited period indicator
β_{ts}	Boolean time slot indicator
π	Job sequence

3.3.1 Total Labor Cost (TLC)

The *TLC* depends on *sh* and calculated by Equation (3.5), where NSH is the total number of shifts in the scheduling time span without considering the match for production loads, W_{sh}^{pt} is the labor wage of the personnel type *pt* in the shift *sh*, and θ_{sh}^{pt} is the boolean indicator for the *pt* in *sh*. If a *pt* is required by an involved power state, it is included in the corresponding shift (i.e., $\theta_{sh}^{pt} = 1$). Otherwise, $\theta_{sh}^{pt} = 0$. In other words, once a human worker is required in a shift, this person will work and be paid for the whole shift, regardless of the actual workload.

$$TLC = \sum_{sh=1}^{\text{NSH}} \sum_{pt \in \text{PT}} \left(W_{sh}^{pt} \cdot \theta_{sh}^{pt} \right) \tag{3.5}$$

The duration of a shift (δsh, in hours) is defined in Equation (3.6). While δsh remains constant, the shift types in a day (SH) depend on weekdays and weekends. For instance, a day may have three shift types: the morning shift, the late shift, and the night shift. The labor wage in each shift type is often different on weekdays and weekends.

$$\delta sh = \frac{24}{|\text{SH}|} \tag{3.6}$$

The purpose of this labor model is to plan the type and the number of human workers in each shifts, such that the total labor cost can be predicted and its relation with the total energy cost can be quantified in a multi-objective optimization manner.

3.3.2 Total Energy Cost (TEC)

The RTP price varies in every pricing slot (D). Consequently, the *TEC* varies with load shifting. It comprises the energy cost for processing jobs, as well as performing machine changeovers and idling between jobs (Equation (3.7)).

$$TEC = \sum_{i=1}^{\text{N}_\text{J}} CJ_i + \sum_{i=1}^{\text{N}_\text{J}-1} (CC_i + CI_i) \tag{3.7}$$

The constraint of production-prohibited period, introduced by the consideration of labor, should be integrated in the energy model. If a period is prohibited for production ($\lambda = 1$), a job or changeover may be split by a production-prohibited period into at least two subparts with machine power-off&on in-between adjacent subparts; an energy cost for these power-off&on operations should also be considered (Equation (3.8) and Equation (3.9)). A job or a changeover has only one subpart if $\lambda = 0$.

$$
\begin{aligned}
CJ_i = & \sum_{n=1}^{NSJ_i} \left(\sum_{ts=STSJ_i^n}^{ETSJ_i^n} \mathrm{EP_{ts}} \cdot \sum_{t=STJ_i^n}^{ETJ_i^n} (\beta_{ts} \cdot \mathrm{P_p} \cdot t) \right) \\
& + \lambda \cdot \sum_{n=1}^{NSJ_i-1} \left(\sum_{ts=STSJ_i^n}^{ETSJ_i^n} \mathrm{EP_{ts}} \cdot \left(\sum_{t=ETJ_i^n}^{STJ_i^{n+1}} \sum_{s \in \mathrm{S_o}} \left(\beta_{ts} \cdot P_s^t \cdot t \right) \right) \right), \\
& i \in [1, 2, ..., \mathrm{N_J}]
\end{aligned} \tag{3.8}
$$

$$
\begin{aligned}
CC_i = & \sum_{n=1}^{NSC_i} \left(\sum_{ts=STSC_i^n}^{ETSC_i^n} \mathrm{EP_{ts}} \cdot \left(\sum_{t=STC_i^n}^{ETC_i^n} \sum_{s \in \mathrm{S_c}} (\beta_{ts} \cdot P_s^t \cdot t) \right) \right) \\
& + \lambda \cdot \sum_{n=1}^{NSC_i-1} \left(\sum_{ts=STSC_i^n}^{ETSC_i^n} \mathrm{EP_{ts}} \cdot \left(\sum_{t=ETC_i^n}^{STC_i^{n+1}} \sum_{s \in \mathrm{S_o}} (\beta_{ts} \cdot P_s^t \cdot t) \right) \right), \\
& i \in [1, 2, ..., \mathrm{N_J} - 1]
\end{aligned} \tag{3.9}
$$

The energy cost for standby encompasses the period when a machine switches to and stays at one of multiple idle modes, and returns to *Production* state, as defined in Equation (3.10). The absolute time is mapped to the electricity pricing slot by Equation (3.11) and Equation (3.12).

$$
\begin{aligned}
CI_i = & \sum_{ts=ETSJ_i^{NSJ_i}}^{STSC_i^1} \mathrm{EP_{ts}} \cdot \left(\sum_{t=ETJ_i^{NSJ_i}}^{STC_i^1} \sum_{s \in SI_{ij}} (\beta_{ts} \cdot P_s^t \cdot t) \right), \\
& i \in [1, 2, ..., \mathrm{N_J} - 1]
\end{aligned} \tag{3.10}
$$

$$
\beta_{ts} = \begin{cases} 1, \text{ if } t \in [ts \cdot \mathrm{D}, (ts+1) \cdot \mathrm{D}) \\ 0, \text{ otherwise} \end{cases} \tag{3.11}
$$

$$
ts = \lfloor (t - \mathrm{RT})/\mathrm{D} \rfloor, t \in [\mathrm{RT}, \mathrm{RT} + \delta t, ..., \mathrm{DT} - \delta t, \mathrm{DT}] \tag{3.12}
$$

3.3.3 Job and Changeover

Jobs can follow an arbitrary sequence with a common release time (RT) and a common due time (DT). Due to the constraint of production-prohibited period introduced by the labor model, a job may be split by a production-prohibited period and thus contains one or more sub-durations, as indicated in Equation (3.13).

$$
\mathrm{DJ_i} = \sum_{n=1}^{NSJ_i} (ETJ_i^n - STJ_i^n), i \in [1, 2, ..., \mathrm{N_J}] \tag{3.13}
$$

The last job must be completed before DT, considering the duration to power off a machine (Equation (3.14)).

$$ETJ_{N_J}^{NSJ_{N_J}} + \mathrm{D_{poff}} \leq \mathrm{DT}, j \in [1, 2, ..., \mathrm{N_J}] \tag{3.14}$$

As defined by Equation (3.15), a changeover is required between adjacent jobs and starts right before the upcoming job.

$$ETC_i^{NSC_i} = STJ_{(i+1)}^1, i \in [1, 2, ..., \mathrm{N_J} - 1] \tag{3.15}$$

Analogously, the duration of a changeover is the sum of potentially multiple sub-durations, as described in Equation (3.16).

$$\mathrm{DC_i} = \sum_{n=1}^{NSC_i} \left(ETC_i^n - STC_i^n\right), i \in [1, 2, ..., \mathrm{N_J} - 1] \tag{3.16}$$

3.3.4 Machine

A machine is assumed to have sufficient material supply and have no breakdown. in order to focus on static scheduling. For practical dynamic scenarios, e.g., shortage of material supply and machine failure during the execution of a production schedule, rescheduling is a common method [52]. The static scheduling techniques can be leveraged to enable the two popular rescheduling methods, i.e., full generation and repair of a schedule [52].

A machine cannot simultaneously process multiple jobs and does not allow any preemption (Equation (3.17)). As formulated in Equation (3.18), an idle mode is only applicable to an inter-job period that can accommodate it. Such an inter-job period is between the end of the last subpart of the i-th job ($ETJ_i^{NSJ_i}$) and the start of the first subpart of the i-th changeover (STC_i^1). If the production is prohibited during a period, e.g., the weekend, the machine must be powered off, as indicated in Equation (3.19).

$$ETJ_i^{NSJ_i} < STJ_{(i+1)}^1, i \in [1, 2, ..., \mathrm{N_J} - 1] \tag{3.17}$$

$$\sum_{s \in \mathrm{SI_i}} D_s \leq STC_i^1 - ETJ_i^{NSJ_i}, i \in [1, 2, ..., \mathrm{N_J} - 1] \tag{3.18}$$

$$P_s^t = 0, \text{ if } (\lambda = 1) \ \& \ (\forall t \in \text{weekends}) \tag{3.19}$$

3.4 Integrated Energy and Labor Simulation

The state-based energy modeling method was introduced in Section 2.4. When power monitoring is performed on a production machine, a set of power states can

be identified by mapping the power data to the machine functionality and operational states. Each power state has an empirical power profile. A power profile comprises an average duration and a mean power level. The machine transitions between power states over time. Once an empirical energy model is built and has the volatile electricity price as its input, it can calculate the energy consumption and cost, as well as predict the power consumption behavior over time.

An important issue on integrated energy and labor simulation is to coordinate machine power states and labor shifts. As both machine power states and labor shifts are correlated with time, dedicated coordination is needed for correct cost calculation. Multiple cases can be identified for this coordination in the state-based discrete-event simulation environment (Figure 3.1).

Figure 3.1a shows the simplest case, where the current power state (*s*) starts and ends in the current shift. Figure 3.1b presents a more complex situation, where *s* lasts so long that new shifts are needed for full accommodation. Besides, *s* may start exactly at the end of the current shift. If the duration of *s* is shorter than a shift, one new shift is added (Figure 3.1c). Otherwise, multiple new shifts are incorporated (Figure 3.1d). Figure 3.1e sketches a case, where a machine power-off is scheduled between two jobs. When the current job is finished, the machine may be completely shut down, stay off for an assigned period, and start up for the next job. This may need new shifts. Note that even if a shift contains one or more off periods, the entire shift duration is taken for labor cost calculation. Figure 3.1f depicts a case where production is not allowed on weekends. The machine is shut down in advance, such that the end of shutdown is the start of a weekend. An ongoing job may be split by this weekend.

Based on the above cases, a continuous-time shift accumulation heuristic (Algorithm 1) is proposed to accumulate the number of each personnel type with the power state transition over time. It applies to all power states except off, since no shifts are needed when a machine stays off.

Algorithm 1 uses three global variables. The first is current shift ($sh \in SH$), containing current time. A *sh* includes two sub-variables: end time (*sh.endTime*) and personnel types already required by *sh* (*sh.pt*). *sh.endTime* represents a critical time point ($\in$ *SBT*) for a shift switch (so *SBT* denotes a set of shift boundary time when a shift switch occurs). The second global variable is current power state (*state*), with three sub-variables: state's start time (*state.startTime*), state's end time (*state.endTime*), and the personnel types required by *state* (*state.pt*). The third global variable is the personnel type (*pt*) with one sub-variable: the accumulated number of this personnel type in the whole schedule (*pt.num*).

Algorithm 1 comprises two functional blocks. The first block (lines 1-20) accumulates the number of each personnel type (*pt*) and judges whether to switch a shift according to *state*. It first determines whether *state.startTime* is before *shift.endTime* (lines 1-11). If it is not this case, the launch of state then triggers

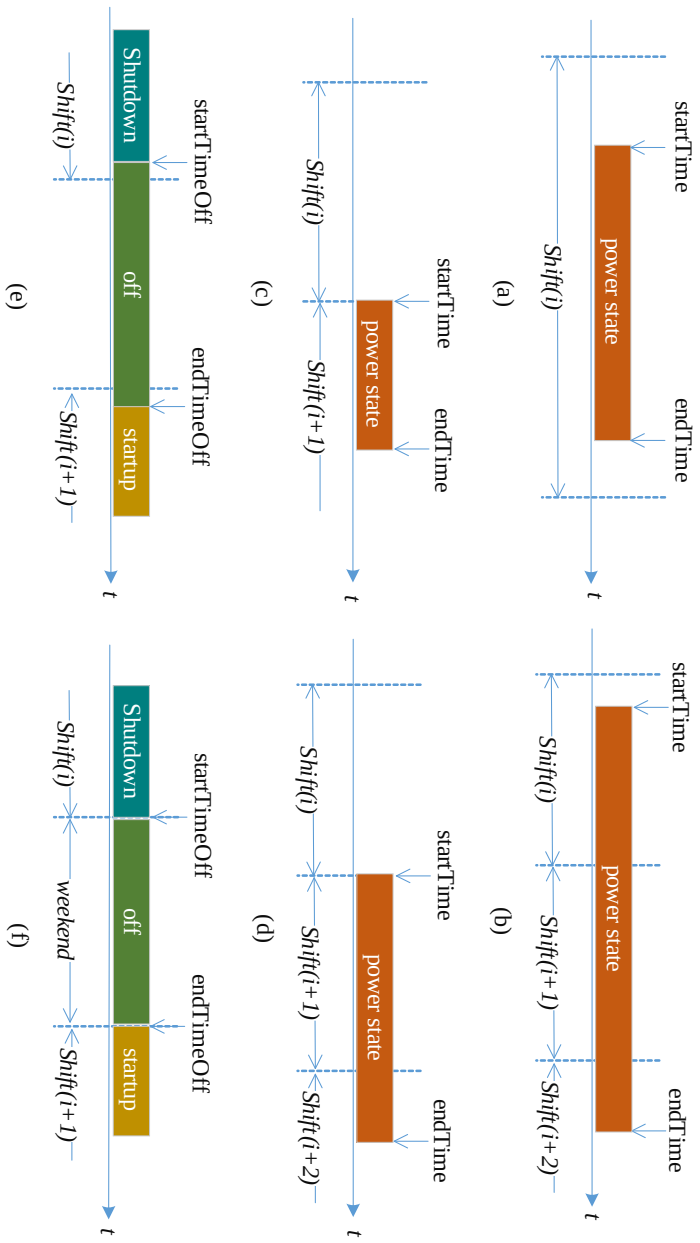


Figure 3.1: Coordination of current machine power state (s) and labor shift ($shift(i)$) over time in the discrete-event simulation. This coordination generally has 6 representative scenarios.

Algorithm 1 Continuous-time shift accumulation heuristic

Input: *state* (including *state.name*, *state.startTime*, *state.endTime*, and *state.pt*)
Output: *sh* (including *sh.name*, *sh.pt*, and *sh.endTime*)

1: **if** $state.startTime < sh.endTime$ **then**
2: **for** $pt \in state.pt$ **do**
3: **if** $pt \notin sh.pt$ **then**
4: $pt.num \leftarrow pt.num + 1$
5: $sh.pt \leftarrow sh.pt \cup pt$
6: **end if**
7: **end for**
8: $flagSwitchShift \leftarrow$ false
9: **else**
10: $flagSwitchShift \leftarrow$ true
11: **end if**
12: **if** $state.endTime \geq sh.endTime$ **then**
13: **for** $sh \in SH$ **do**
14: $numNewShift \leftarrow$ number of $sh.name \in [sh.endTime, SBT$ larger than and closest to $state.endTime]$
15: **for** $pt \in state.pt$ **do**
16: $pt.num \leftarrow pt.num + numNewShift$
17: **end for**
18: **end for**
19: $flagSwitchShift \leftarrow$ true
20: **end if**
21: **if** $flagSwitchShift =$ true **then**
22: $shift \leftarrow$ switch to a new shift according to $shiftEndTime$
23: $sh.endTime \leftarrow SBT$ larger than and closest to $state.endTime$
24: $sh.pt \leftarrow \oslash$
25: **for** $pt \in state.pt$ **do**
26: $sh.pt \leftarrow sh.pt \cup pt$
27: **end for**
28: **end if**

a shift switch. Otherwise, no shift switch is needed for a new shift. But it needs to check whether an additional *pt* is introduced by *state*. If it is the case, this *pt* is considered both in its accumulated number (*pt.num*) and in *sh* (*sh.pt*). The first functional block of Algorithm 1 then decides whether *state.endTime* is after *sh.endTime* (lines 12-20). If it is this case (Figure 3.1b and Figure 3.1d), it needs to account for the number of all additional shifts, which are needed for accommodating *state*. Besides, *sh* has to be updated as the last new shift.

The second functional block (lines 21-29) switches the shift when necessary, and initiates a new shift. *Sh.endTime* is compared with *SBT*. This results a corresponding shift type, which is taken as *sh*. After initialization (lines 24-25), *sh* incorporates all the required personnel types of *state*. Note that the rest duration of *sh* may accommodate other states after the end of *state*.

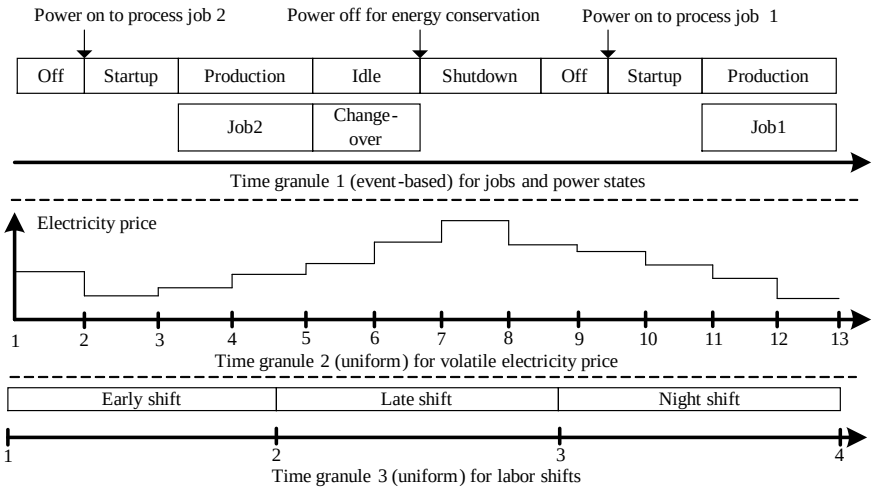


Figure 3.2: Parallel time axes to synchronize electricity prices, jobs, shifts, power states, and machine operations in discrete-event simulation of a schedule

Furthermore, a time-granule-based method is elaborated to synchronize various elements in discrete-event simulation of an integrated production and labor schedule. As illustrated in Figure 3.2, three parallel time axes with independent time granules are coupled. Time axis 1 synchronizes jobs and power states, where the time elapse is triggered by machine operations. Time axis 2 coordinates real-time electricity prices that can be obtained from the electricity spot market beforehand, with the time granule of pricing period. Time axis 3 matches labor shifts, of which the time granule is the duration of a shift.

3.5 Solution Algorithms

From the perspective of optimization, the integrated model (Section 3.3) increases the number of decision variables, the interaction between decision variables, and the complexity of constraints in the search space. This thus requires a more efficient optimization algorithm to search for the optimal or near-optimal solution.

3.5.1 Genetic Algorithm for Single-Objective Optimization

The genetic algorithm (GA) tailored for the energy-aware single-objective production scheduling problem (Section 2.6) can be used for the integrated energy- and labor-aware single-objective production scheduling problem. Readers are referred to Section 2.6 for the detailed design.

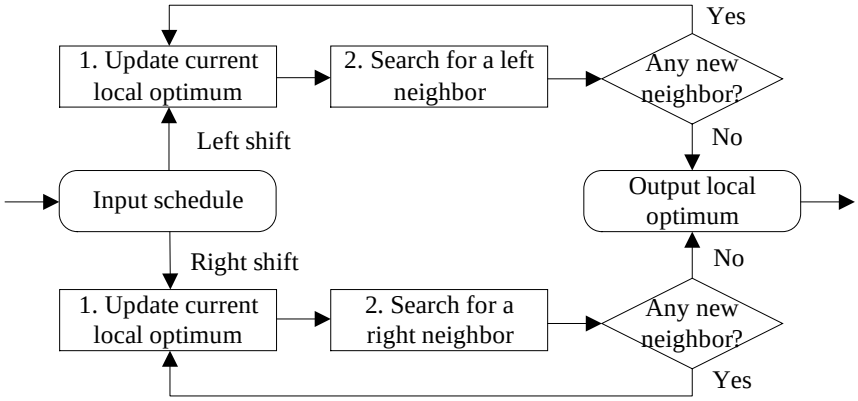


Figure 3.3: Illustration of a local search in a static memetic algorithm. Two hill climbing algorithms (left shift and right shift) are applied to each individual of a population in the NSGA-II.

3.5.2 Adaptive Memetic Algorithm for Multi-objective Optimization (AMOMA)

As introduced in Section 3.1, a conventional memetic algorithm (MA) statically applies one or multiple local searches to a multi-objective evolutionary algorithm (MOEA). Figure 3.3 illustrates such a static MA [26]. For every individual of a population in the NSGA-II [22], this MA always applies two hill climbers[3]for local improvement. The criterion of determining a local best can be defined, such as the production schedule with the least energy cost and the least labor cost if there are multiple solutions with the same least energy cost. A hill climber takes an input schedule upon start (step 1 in Figure 3.3). It encounters a new neighbor (step 2 in Figure 3.3) via shifting one job by one electricity pricing period toward the assigned direction and without altering the job sequence. It then updates the local optimum (step 1), and iterates this search process (step 1 and step 2) until it cannot find a new neighbor any more. The search direction can be left (backward over time) or right (forward over time), but remains fixed in a hill climbing. These two types of hill climbing (i.e., in two directions over time) are simultaneously applied to increase the opportunity for a higher-quality local optimum.

Beyond such a static MA, the AMOMA is proposed for multi-objective optimization of the integrated energy- and labor-aware production scheduling problem. It not only synergistically integrates in the NSGA-II [22] the convergence-

[1]Hill climbing is a local search algorithm for optimization. A descent hill climber starts with a certain solution for a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution. This process iterates until no further improvement is found.

and diversity-oriented tabu searches, respectively, but also adaptively coordinates the exploration and the exploitation during a search.

3.5.2.1 Exploration by Genetic Search

The exploration framework employs the prevalent genetic search NSGA-II, which uses domination as the fitness assignment strategy. It searches for potential regions in the solution space without having to guarantee the local optimum in each region. Thereby, exploration equals diversity preservation in AMOMA. As two local search operators are used (Section 3.5.2.2), the population size should stay sufficiently large (larger than that for pure exploration) to balance the computation resource for exploration and exploitation in a generation.

Several measures are taken on the crossover to preserve diversity. First, parents are chosen via a binary tournament selection, preserving the diversity to the maximal extent. Second, a one-point crossover is employed, such that a solution can change the job sequence (π) combing partial sequences of parents. As *TEC*, *TLC*, and C_{max} are sensitive to job shifting, crossover loci are randomly selected and offspring are randomly timed (without altering π). Third, the crossover rate remains high to introduce sufficient recombination of solutions and thus a higher opportunity for the genetic search to enter diverse areas.

A swap mutation is used for a solution to switch two of its randomly selected jobs. The diversity is preserved by two measures. First, random timing is performed on all jobs after a mutation (without altering π). Second, as a mutation follows a crossover, the mutation rate remains low, to maintain the effect of crossover and to avoid a pure random search.

Fully feasible solutions are produced to remove the need for repairing infeasible solutions. This is realized by assigning the start time of one job after another, according to the natural order of scheduled job positions. When timing a job, a random start time is generated from its maximal slack regarding the *DT*. Furthermore, redundancy is fully prevented to increase the diversity and also to reduce a waste of computation resources. A solution is considered redundant if it equals another one in the multi-objective space. Reproduction is iteratively performed upon redundancy until it is removed.

3.5.2.2 Exploitation by Multiple Memes

Memes are incorporated in the exploitation framework of AMOMA in two manners: preprocessing scheme and problem-specific local search operators. The former integrates a priori knowledge in population initialization, biasing the overall search from the start to promising regions. The latter is interwoven with genetic operators (Figure 3.4) and leverages domain knowledge to refine selected solutions.

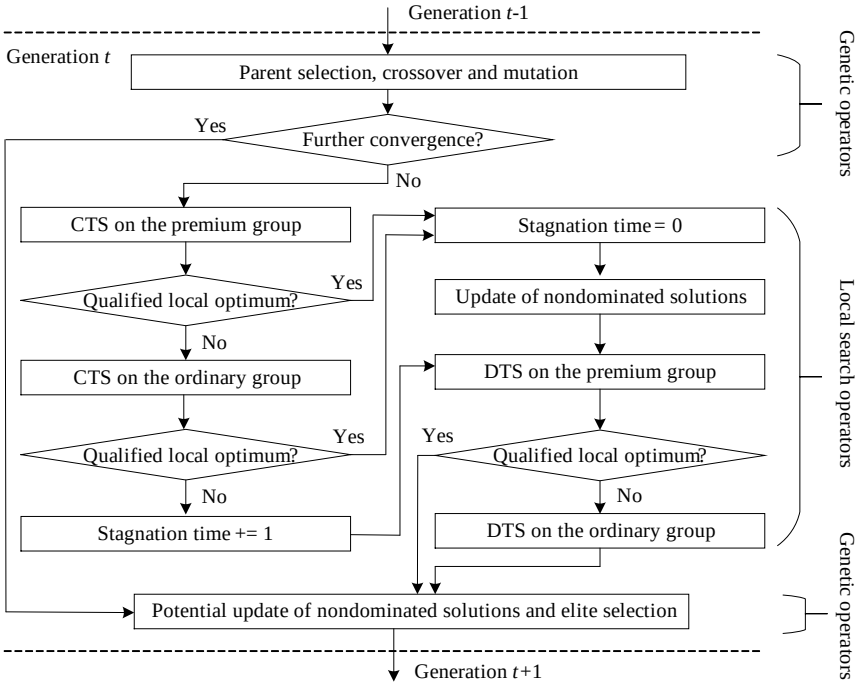


Figure 3.4: Adaptive coordination of genetic search and two local search operators, i.e., convergence- and diversity-oriented tabu searches (CTS and DTS)

Regarding preprocessing, two dispatching rules which match the problem $\{1, \text{RTP}\}|\text{split}|\{\textit{ELC}, \text{C}_{max}\}$ introduce specialized solutions in initialization. (1) "As-early-as-possible": all jobs are joint and start from the beginning such that C_{max} is minimized. (2) "As-late-as-possible": all jobs are joint and start late such that the last job ends at the DT and C_{max} is maximized.

Two tabu search (TS) algorithms are proposed for local refinement: convergence-oriented tabu search (CTS) and diversity-oriented tabu search (DTS). They are preferred over hill climbing, since they can escape a local optimum by temporarily accepting a deteriorated solution, and potentially leading to a superior solution. They are mutually complementary by stimulating the convergence and diversity of nondominated solutions, respectively.

To enable exploitation of all neighborhoods, a greedy termination criterion is defined: the longest free period is traversed, among free periods that are inter-job, before and after the entire production. Hence, the TS step (S_{ts}, basic time slot to define a neighborhood structure) determines the shared portion of genetic and local searches within a fixed time budget.

Table 3.3: Analysis of convergence- and diversity-oriented tabu search (CTS and DTS) in optimizing the joint energy and labor cost (ELC) and makespan (C_{max})

Tabu serach	ELC	C_{max}	Convergence	Diversity	Complexity[b]
CTS	$\downarrow$	–[a] or $\downarrow$	$\uparrow$	$\downarrow$	O(mn^2)
DTS	$\uparrow$	$\downarrow$	–	$\uparrow$	O(mn)

[a] –: no impact.
[b] m: number of time slots, n: number of jobs.

A CTS builds a neighborhood structure through backward moving a block of n ($n \geq 1$) contiguous jobs by one S_{ts}. In every TS iteration, blocks of jobs are constructed by starting from the first job and ending at every following job, restarting from the second job and ending at every following job, and so on until starting from the last job. In this way, all neighbors are checked. The best neighbor has the lowest *ELC*, and the shortest C_{max} in case of equal *ELC*. The aspiration criterion requires that a neighbor dominates at least one solution in the approximation set of latest generation t (NS_t). During the progress of a CTS instance, it is a soft criterion by only filtering neighbors when at least one qualified neighbor exists. At the end of a CTS instance, it acts as a hard criterion to ensure that the refined solution actually improves the convergence of nondominated solutions. As a CTS naturally reduces *ELC* without increasing C_{max}, it enhances the convergence while losing the diversity (Table 3.3) due to the concentration behavior of CTS.

In a DTS, a neighborhood structure is built via backward moving last n ($n \geq 1$) contiguous jobs by one S_{ts}. The last job must be included to reduce C_{max}. During each TS iteration, the best neighbor is the one that is not dominated by any other neighbors, and leads to the most evenly spread approximation set in case of multiple nondominated neighbors. The metric Δ is used to indicate this evenness:

$$\Delta = \frac{1}{\overline{d}} \sqrt{\frac{1}{|NS_t|} \sum_{x_i \in NS_t} (d_i - \overline{d})^2} \tag{3.20}$$

where d_i is the Euclidean distance between solution x_i and its nearest neighbor in NS_i , and $\overline{d}$ is the mean Euclidean distance. A smaller Δ implies a higher extent of spread in NS_i. Δ has a complementary role of the crowding distance in NSGA-II for diversity preservation. The former smoothens the approximated Pareto front, whereas the latter produces distant or extreme solutions by preferring less-crowded regions. The aspiration criterion requires that a neighbor is neither dominated by nor equal to any latest nondominated solutions. Analogously, it is a soft and hard criterion during and at the end of a DTS instance, respectively. As a DTS intrinsically reduces C_{max} and increases *ELC*, it strengthens the diversity without influencing the convergence (Table 3.3).

Figure 3.4 depicts the combination of genetic and local searches. The CTS and DTS are sequentially applied, between which the NS_t absorbs the convergence contribution of CTS while losing diversity (Table 3.3). To prevent premature convergence, the DTS then compensates by improving diversity (Table 3.3).

Both CTS and DTS are tabu list free. This is because they are mono-directional on the time span, naturally skipping previously-visited solutions. A tabu list is no more needed to forbid the search direction. Accordingly, this removes the extra work of tuning each tabu list size.

3.5.2.3 Coordination of Genetic and Local Searches

While preprocessing is evidently performed in initialization, three major issues lie in coordinating the genetic search and two local search operators: (1) when to trigger them during the progress of genetic search? (2) Which initial solutions and which refinement frequency? (3) When to terminate the entire search? Compared to simple hybrid-based MAs that take static measures, these issues are resolved by taking feedback from a search. The adaptive measure will be presented for each issue.

As preprocessing may lead to premature convergence upon the start of an entire search, local search operators are not utilized in the first N_f generations, where the N_f controls the frequency of pure genetic search. Afterward, they are launched once the NS_t that are outputted by the genetic search stops converging compared to the NS_{t-1} (nondominated solutions in the previous generation $t-1$). The cross-dominance metric λ [53] is employed to characterize this relative convergence:

$$\lambda = \frac{\Lambda_t}{(|NS_t| \cdot |NS_{t-1}|)} \tag{3.21}$$

where Λ_t denotes the number of dominance occurrences obtained by pairwise comparing the NS_t to the NS_{t-1}. λ equaling zero indicates that the NS_t do not converge any more after the NS_{t-1}.

Two groups of solutions with distinct priority can be initial solutions for local searches. A premium group includes the NS_t. An alternative group contains $|NS_t|$ solutions that are randomly selected from the rest population of generation *t* except the NS_t. The rationale for this prioritized and equally-sized grouping is that it is more promising to exploit the NS_t and exploitation on too many dominated solutions may unnecessarily waste computation resources. The alternative group is only used when a local searcher cannot find any qualified local optimum from the premium group. The alternative group thus introduces randomness and adaptively raises the frequency of refinement.

If the CTS cannot find a qualified local optimum from both premium and alternative groups, the entire search is considered to stagnate, and the stagnation time (T_{stag}) accordingly increases by one. The entire search terminates if T_{stag}

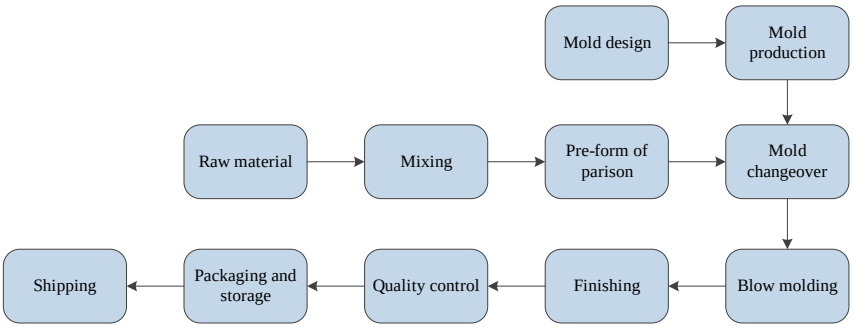


Figure 3.5: Value chain of the investigated plastic bottle manufacturer

reaches the preset maximal stagnation time (T_{max}) or the time budget is used up. Thereby, the T_{max} may terminate an AMOMA instance before a time budget is used up, which provides room for an AMOMA instance to go even faster than the expectation.

3.6 Empirical Data

A Belgian plastic bottle manufacturer was taken as empirical study. Overall, there are 17 extrusion blow molding (EBM) process lines on its shop floor, producing plastic bottles which vary from 40 mL to 5 L. The entire value chain is presented in Figure 3.5, where the mold changeover and blow molding process are related to production scheduling.

The overall factory data were collected through three site surveys. The power consumption of two EBM processes was monitored every 30 seconds for over one year, by installing Siemens PAC 3200 power monitors on the three major electricity consumers: main system, hydraulic system, and extruder.

3.6.1 Overall Factory Cost Data

A lifecycle cost breakdown of this plant was performed to clearly link the production scheduling to all the relevant cost parts [18]. Five lifecycle phases [18] are presented in Figure 3.6: planning, deployment, migration, operations, and teardown. The relevant cost parts are in orange.

The planning phase includes purchasing, manpower planning and long-term planning for a new EBM process. For purchasing energy, the planning phase will only account for the negotiation costs. The actual energy consumption is included in the operations phase and depends on the scheduled production. Manpower planning establishes a long-term vision for the number of employees.

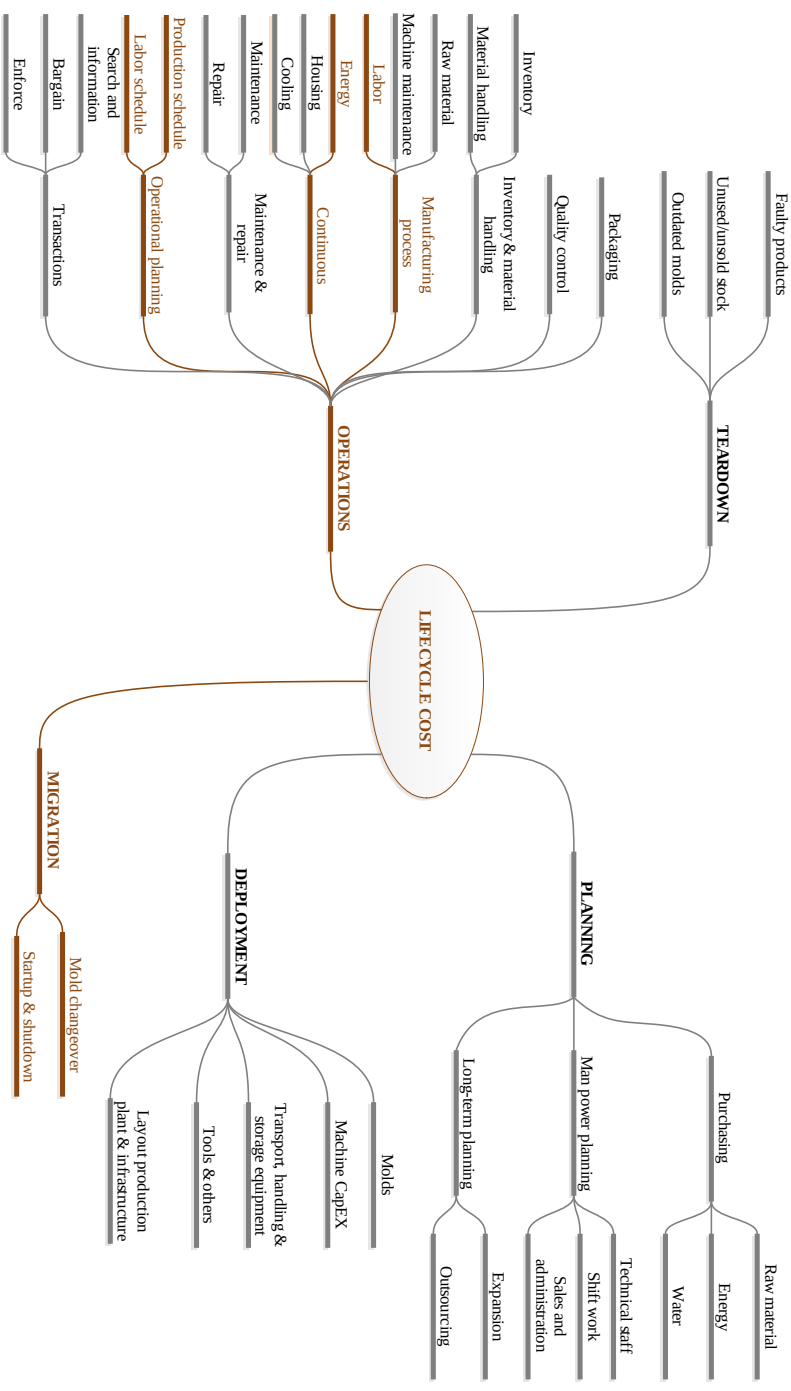


Figure 3.6: Lifecycle cost breakdown of the investigated plastic bottle manufacturer [18]. The cost parts related to production scheduling are in orange. The other unrelated cost parts are in black and grey.

Table 3.4: Overall factory cost parts [18]

Cost type	Normalized cost
Transparent plastic (per kg)	1
Color additives (per kg)	3-5
Machine CapEx (per hour)	1.35
Packaging equipment (per hour)	0.45
Packaging material (per hour)	2.80
Technical staff (per hour, daytime)	14
Technical staff (per hour, nighttime)	15.5
Technical staff (per hour, daytime on weekends)	19.3
Transport equipment (per hour)	0.20

The deployment phase consists of the activities needed to start the production: the necessary equipment (machines, tools and molds) should be provided and installed. The mold design and production costs (Figure 3.5 and Figure 3.6) are passed on to the customer, either through a premium percentage on the yearly volume or through a dedicated payment plan. They are hence independent of production scheduling. A similar reasoning can be followed for machines and tools.

The migration phase denotes the costs associated with setting up a new bottle type, where the mold changeover and machine shutdown & startup are involved. As they may influence both energy and labor costs, they are linked to production scheduling.

The operations phase includes day-to-day operational costs. The manufacturing process cost is the most important, including the raw material cost, labor cost, as well as machine CapEx and mold cost. Table 3.4 indicates the normalized cost types by comparing to the transparent plastic cost. Obviously, labor cost takes up an important part, demonstrating the necessity of integrating labor awareness to production scheduling for production cost minimization. Raw material cost include transparent plastic cost and color additive cost. As the production quantity (number of bottles) is prefixed, the raw material use (and hence cost) is constant, independent of production scheduling. As a result, it is logical that energy and labor cost parts are linked to production scheduling in this chapter.

Finally, the teardown phase represents the end-of-lifetime of EBM processes, including the processing of faulty products, unused/unsold stock and outdated molds. It is not associated with production scheduling.

3.6.2 Power Consumption Data

3.6.2.1 Energy Consumption Monitoring and Profiling

Through a site survey [54], the continuous EBM process under investigation was found to comprise of three major electricity consumers: main system, hydraulic system, and extruder. The main system demands general power supply for the process. This general demand steers a series of energy-intensive operations, e.g., mixing, cutting, grinding, and pushing the input materials (raw plastic granules, color granules, and recycled plastic chips), heating, and melting. The hydraulic system consumes electricity to provision major mechanical movements of the process, e.g., clamping and closing the mold, cutting the parison, moving extruder continuously pushes the melt plastic through a die.

A Siemens PAC3200 power meter was installed on these three consumers, respectively. The sampling interval was 30 seconds. The instantaneous effective power was captured every 30 seconds, and stamped with time and other essential information, e.g., power unit, sensor name, and product name. The raw data were in ASCII format, communicated throughout Modbus protocol, and captured by a cabinet with PLC as the data collector. The data collector was connected with a PC via Ethernet, in order to enable data management and visualization. A midterm power measurement campaign was carried out during about one year. A variety of plastic bottles were produced during this measurement period. Figure 3.7 shows the power data, where eight power states are identified: *Off*, *Startup*, *Idle*, *Preheat*, *PreheatIdle*, *Proheat*, *ProheatIdle*, and *Production*.

When the EBM machine is powered on, it goes through *Startup*, *Idle*, and *Preheat*, during which the plastic is heated in the barrel until 140 °C. It then stays at *PreheatIdle* and remains this temperature until an operator launches *Proheat*. Afterward, the temperature of plastic rises to a higher level between 140 °C and 200 °C, depending on the type of bottles to be produced. When the target temperature is achieved, the machine stays at *ProheatIdle*. Once a production command is given, it transitions to *Production* state for producing plastic bottles.

3.6.2.2 Energy Consumption Modeling

The identified power profiles of the three major consumers of the EBM process (Figure 3.7) were aggregated into one power profile for characterizing the power consumption behavior of the process. The aggregated profile and the required labor per state is demonstrated in Table 3.5. The rationalized transitional relation between states is further depicted in Figure 3.8. As an intuitive summary of the working procedure of this EBM process in perspective of power, Figure 3.9 illustrates the complete power consumption behavior of this process, based on the entire power profile which was identified from the measured data.

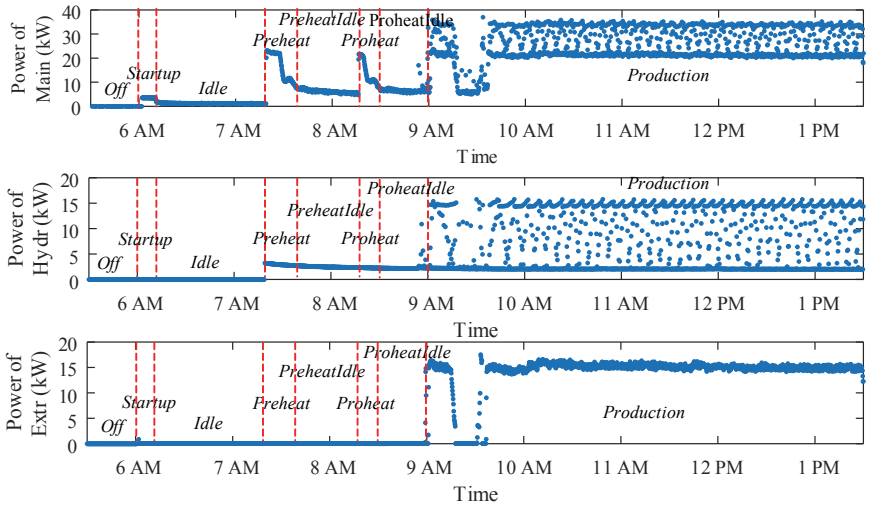


Figure 3.7: Measured power data and power profile identification of three major energy consumers of an extrusion blow molding machine (from the top to the bottom: main system, hydraulic system, and extruder)

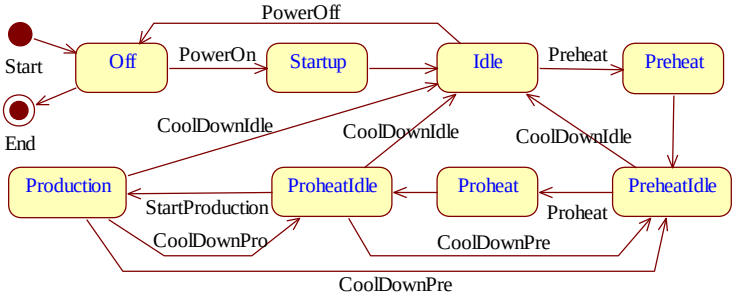


Figure 3.8: State-based energy model of an extrusion blow molding machine

Plastic bottles in various types were produced, which may cause significant discrepancy in the power consumption of the Production state. However, the standard variation of power and cycle time was found to be quite minor, taking up 2% and 1% of the corresponding mean values, respectively. One important reason for this is that there are only bottle color changeovers (e.g., silver → white) in the collected data. As a result, the EBM process does not need to significantly change its configurations (e.g., plastic temperature and mold). Therefore, the power and cycle time for producing one type of bottles is randomly selected (5L-silver-M50-UN-160g-Y1 in this case) as the power profile of the *Production* state (Table 3.5).

The power profiles shown in Table 3.5 were further integrated into the FSM-based energy model (Figure 3.8). The operations above the transitional arrow

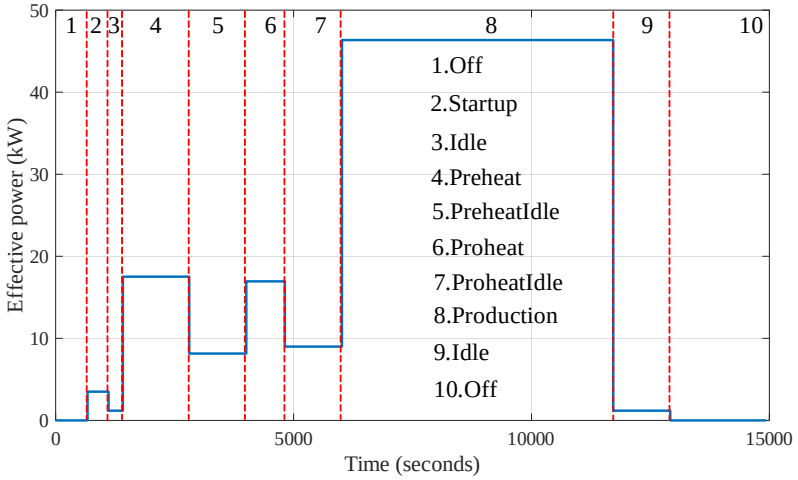


Figure 3.9: Complete power consumption behavior/power profile of the extrusion blow molding machine

Table 3.5: Power profile and required labor of an extrusion blow molding machine

State	Power (kW)	Duration (s)	Required personnel type
Off	0	≥ 0	None
Startup	3.51	442	Operator
Idle	1.19	≥ 0	Operator
Preheat	17.52	1395	Operator
PreheatIdle	8.15	≥ 0	Operator
Proheat	16.95	810	Operator
ProheatIdle	9.00	≥ 0	Operator for powering up Technician for a changeover
Production	46.35	17.92	Operator, technician, packer, and quality checker

in Figure 3.8 should be performed by operators. An arbitrary duration exists at *ProheatIdle*, *PreheatIdle*, *Idle*, and *Off* states, which therefore provides four idle modes for an additional decision making (a loose assumption is made on *ProheatIdle* that its retention time can be random without considering the power demand of cooling). In the case of an idle period between two adjacent jobs, it depends on the scheduler to optimally select one of these four idle modes. In the other cases, a constant duration of 1200 seconds is assumed for each retention time of *ProheatIdle*, *PreheatIdle*, and *Idle*.

The four idle modes are presented in Table 3.6, compared to the production

Table 3.6: Energy modes of the extrusion blow molding process and measured power consumption

Component (power in kW)	*Production* (46.35)	*ProheatIdle* (9.00)	*PreheatIdle* (8.15)	*Idle* (1.19)	*Off* (0)
Main	On (25.22)	On (6.51)	On (5.66)	On (1.19)	Off (0)
Hydraulic	On (6.28)	On (2.49)	On (2.49)	Off (0)	Off (0)
Extruder	On (14.84)	Off (0)	Off (0)	Off (0)	Off (0)

mode. At the production mode, the three major components are powered on. In comparison, the extruder is powered off at *ProheatIdle* and *PreheatIdle* states; both the extruder and hydraulic system are powered off at *Idle* state; all the components are powered off at *Off* state. The same component (main, hydraulic, and extruder) has distinct power demands at different idle modes (Table 3.6). The four idle modes are further illustrated in Figure 3.10. Given that a changeover must be conducted just following the end of a job, an idle period is the duration from the end of a changeover to the start of the next scheduled job. Since the idle period can be arbitrary in practice, and the four idle modes' power profiles are different, the required energy consumption (i.e., shadowing areas in the subplots in Figure 3.10) and energy cost are arbitrary and tend to be distinct, making it difficult to conduct human-based decision makings in a long term. This additionally highlights the need for an automated energy-cost-aware production scheduling method.

Besides, the color changeover data are mapped with the collected power data, which provides an insight into the power consumption and cycle time of the process depending on the type of the color changeover. As indicated by Table 3.7, the power consumption and cycle time not only vary among different color changeovers, but between two changeovers of the same type (i.e., white $\rightarrow$ dark blue). The reason could be that a changeover highly depends on a specific operator on the shop floor, which may need different time and set different process parameters to conduct a changeover. For simplicity, the mean power and cycle time in Table 3.7 are used as the power profile of a color changeover for scheduling. A changeover can be conducted at either *ProheatIdle* or *PreheatIdle* state. In this case study, it is assumed that the process always shifts to *ProheatIdle* for a changeover.

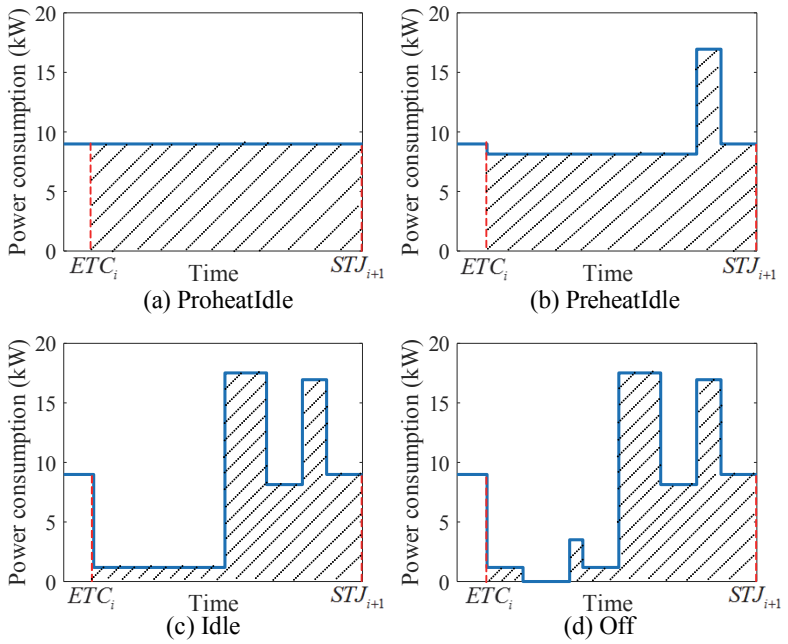


Figure 3.10: Four idling modes of the extrusion blow molding process that remains to be optimally selected between jobs (ETC_i: end time of the i-th changeover which is conducted at the end of the i-th scheduled job, STJ_{i+1}: start time of the (i+1)-th scheduled job)

Table 3.7: Power profile of a color changeovers of the investigated extrusion blow molding (EBM) process

Changeover	Power consumption of the process (kW)	Cycle time (s)
Silver → White	13.60	28022
White → Dark blue 1	15.22	25841
White → Dark blue 2	10.00	1157
Dark blue → White	9.03	6088
Silver → Dark blue	9.11	5439
Average	11.39	13309
Standard deviation	2.54	11272

Table 3.8: Three shifts and working days per week of the investigated plastic bottle manufacturer. The whole factory is shut down on weekends.

Early shift	Late shift	Night shift	Start of a week	End of a week
6 h – 14 h	14 h – 22 h	22 h – 6 h	6 h Monday	6 h Saturday

3.6.3 Labor and Electricity Price Data

As the labor aspect is considered, the empirical labor shift data were also collected. Table 3.8 indicates the three shifts of this plant. This factory is closed on weekends, meaning that all EBM lines have to be powered off before 6 am on Saturday and powered on again at 6 am on Monday.

Table 3.5 lists the personnel type required by each state of the EBM process. Specifically, *ProheatIdle* state has two cases. If the machine stays at *ProheatIdle* for powering up toward the *Production* state, only one operator is required. If the machine transitions from *Production* to *ProheatIdle* for a color changeover, an operator and a technician are required.

A workday comprises early shift (6 AM - 2 PM), late shift (2 PM - 10 PM), and night shift (10 PM - 6 AM). The labor compensation of a night shift rises by 10%. The factory is closed on weekends. The exact labor costs cannot be disclosed due to confidentiality. But all staff is paid on an hourly basis (euro/h), where a bonus is paid for night shifts, with a compensation rise of 10% compared to early and late shifts. The real-time pricing (RTP) data were taken from the Belgium electricity spot market [55], where the electricity price varies every hour and is known 24 hours in advance.

3.7 Single-Objective Optimization Experiments

To fill the gaps identified in Section 3.2.2, this section intends to integrate labor awareness to energy-efficient production scheduling by single-objective optimization [18]. The experiments aim to address the following questions: 1) Will the incorporation of energy and labor awareness in a production schedule help to reduce the energy cost for production? 2) Will the incorporation of energy and labor awareness in a production schedule decrease the total energy and labor cost for production? 3) What are the potential factors that will impact the total energy and labor cost, the energy cost, as well as the labor cost of an energy-efficient and labor-aware production schedule, and to which extent?

The life cycle cost analysis of this investigated plant (Section 3.6.1) showed that the labor cost takes up 10% of the total production cost, while the energy cost is limited to 3%. The raw material cost occupies over 50%, as the main cost driver.

The three single-objective functions, which are defined by Equation (3.2), Equation (3.3), and Equation (3.4), were used, respectively, with the same GA configuration on one computer (Intel i5-3470 CPU @ 3.20 GHz, 8 GB RAM). The time span was set to one week, corresponding to the scheduling horizon of this factory. The outcome schedule was named schedule1, schedule2 and schedule3, representing optimizing toward joint energy and labor cost (*ELC*), toward total energy cost (*TEC*), and toward total labor cost (*TLC*), respectively.

Random, as-early-as-possible (AEAP), and as-late-as-possible (ALAP) schedules were taken as benchmarks. The random schedule was generated by satisfying all the formulated constraints without any optimization. The latter two schedules are two rule-of-thumb scheduling strategies, which group all jobs together and starts production either as early or as late as possible. These schedules have neither energy nor labor awareness.

The scheduling time slot δt was one second. The scheduling time span was set to one week. This led to overall 604,800 δt if weekend production is allowed and 432,000 δt if weekend production is not allowed.

3.7.1 Impact of Energy and Labor Awareness

The GA search for schedules 1, 2, & 3 and the random scheduling was performed 100 independent times, respectively. This experiment was repeated at two different periods: 8 - 14 Aug. 2016 (P1) when the weekly mean electricity price (MEP) is low (28 euro/MWh), and 12 - 18 Oct. 2015 (P2) when the weekly MEP is high (104 euro/MWh). The production on weekends was disabled according to the labor shift of this facory (Table 3.8). The results are indicated in Table 3.9.

Schedule1 and schedule3 are effective in *ELC* minimization. They grossly achieve the same *ELC*, *TEC* and *TLC*, in both scheduling periods. Conversely, schedule2 and the random schedule have 17% higher *ELC*. The poor *ELC* performance of schedule2 is explained by the minor portion of *TEC* in ELC (3%).

As schedule1 optimizes *TLC* besides *TEC*, it avoids the situation where load shifting leads to obviously more labor shifts. For instance, shifting loads to the night may induce an additional early shift to complete the rest of production and the entire changeover. In comparison, schedule2 incurs 21% and 22% higher *TLC* in P1 and P2, compared to schedules 1 & 3, respectively.

Nonetheless, schedule2 is effective in *TEC* minimization. It achieves the smallest variation in *TEC* (0.8% in P1 and 2.0% in P2, Table 3.9). Compared to schedules 1 & 3, it reduces *TEC* by 20% and 50% in P1 and P2, respectively. Besides, its impact of *TEC* minimization rises when the MEP increases (20% in P1 vs. 50% in P2).

Last but not least, although both the genetic search and the random generation of a qualified solution are stochastic (such that the solution quality varies in

Table 3.9: Runtime and cost performance of schedules 1, 2, & 3, and a random schedule. P1: scheduling period1 (8 - 14 Aug. 2016). P2: scheduling period2 (12 - 18 Oct. 2015).

		Schedule 1		Schedule 2		Schedule 3		Random schedule	
		P1	P2	P1	P2	P1	P2	P1	P2
Runtime	Mean (s)	163	166	172	135	214	218	0	0
	Standard deviation (s)	6	4	4	4	5	4	0	0
Energy and labor cost	Mean (euro)	4938	5146	5930	6056	4936	5172	5908	6125
	Deviation around mean	0.5%	0.7%	5.5%	3.1%	0.3%	0.8%	5.6%	4.9%
Total energy cost	Mean (euro)	131	334	107	179	132	368	131	374
	Deviation around mean	1.2%	12.8%	0.8%	2.0%	1.5%	7.6%	4.0%	8.7%
Total labor cost	Mean (euro)	4807	4812	5822	5877	4803	4804	5778	5750
	Deviation around mean	0.5%	0.7%	5.6%	3.2%	0.3%	0.5%	5.8%	5.4%

different runs of the same search), the runtime and economic variation of each schedule is minor (Table 3.9). Therefore, a schedule obtained by a single run is representative. The following experiment results were then obtained based on one GA search.

3.7.2 Impact of Electricity Prices

The previous experiment was further performed at a rolling horizon of one week over 2007-2015, where full-year RTP data are available at the Belgium electricity spot market. Consequently, 441×3 schedules were obtained. The MEP was calculated (6h Monday - 6h Saturday of the same week considering the labor shift, Table 3.8) to indicate an averaged electricity price level per week.

3.7.2.1 Economic Sensitivity to Electricity Prices

Figure 3.11 shows the correlation between the cost parts (*ELC*, *TEC*, and *TLC*) of each schedule and weekly MEP. The Pearson correlation coefficient (CC) and p-value are indicated in each subplot. A CC value ($\in [-1, 1]$) close or equal to 1, 0, and -1 indicates strong positive correlation, no correlation, and strong negative correlation between two variables, respectively. A p-value should be within 5% to guarantee the general significance of the observed statistical behavior. A column in Figure 3.11 indicates different cost part of the same schedule. A row demonstrates the same cost part for the different schedules.

As demonstrated in Figure 3.11, both *ELC* and *TEC* of three schedules follow a positive linear relation with MEP, while *TLC* is insensitive to MEP. More specifically, schedule1's *ELC* has a linear relationship with MEP (CC is 0.68). This is explained by the strong positive linear correlation between *TEC* and MEP (CC is 0.97), and non-correlation between *TLC* and MEP (CC is 0.09). A similar phenomenon is observed in schedules 2 & 3, except that schedule2's *ELC* is weakly correlated with MEP (CC is 0.14). This weaker correlation is caused by schedule2's high *TLC* variation, which has no correlation with MEP. There is no correlation between *TLC* and MEP in schedule3, of which the CC is -0.07. A slightly negative CC indicates that TLC may occasionally gently decrease with rising MEP, which further exhibits the non-correlation between these *TLC* and MEP in schedule3.

Energy awareness reduces *TEC*'s sensitivity to the electricity price, although *TEC* of the same production still rises with increasing MEP. This is indicated by the slope of the fitted linear curve on the second row of Figure 3.11. A slope value equal to k ($k > 0$) infers that for every increased one euro in the electricity price, there will be additional k euro added to the corresponding cost. An analogous interpretation applies to the decrease case. For *TEC*, schedule2's slope is the

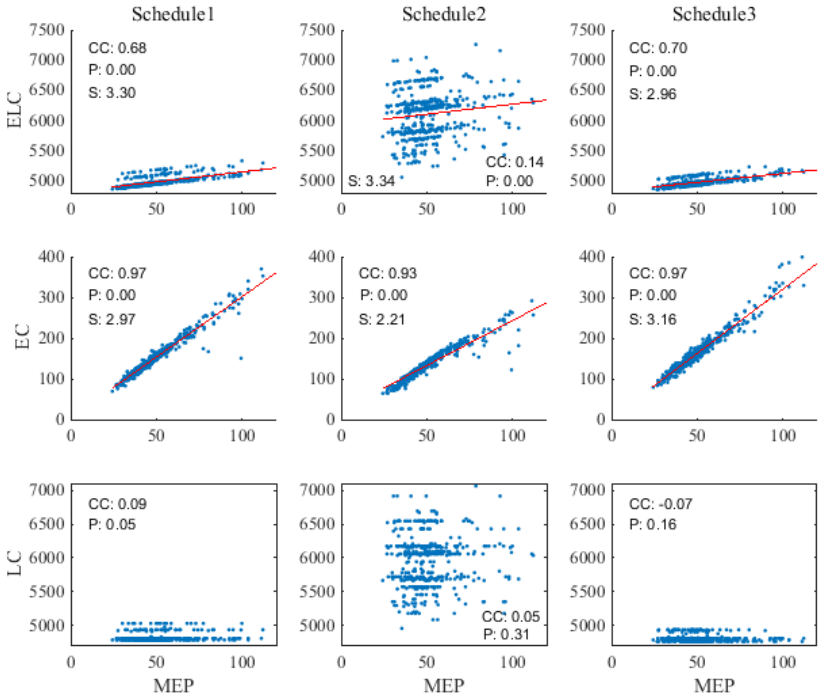


Figure 3.11: Correlation between the mean electricity price (MEP, euro/MWh) per week and the cost (euro) of an optimized schedule. The cost includes joint energy and labor cost, energy cost, and total labor cost. The correlation coefficient (CC), p-value (P) and slope (S) of the fitted line are indicated in the figure.

smallest (S is 2.21), while that of schedule3 is the largest (S is 3.16). The slope of schedule1 (S is 2.97) is closer to that of schedule3, since *TLC* takes the major cost part such that the *ELC* optimization has a larger impact on *TLC* than *TEC*.

For the same reason, the slope order in the first row (Figure 3.11) is inverse: schedule2 has the largest slope (S is 3.34) while schedule3 has the lowest slope (S is 2.96). Again, schedule1 stays at the intermediate level (S is 3.30). This demonstrates that schedule1 reaches moderate sensitivity to the electricity price, in terms of both *ELC* and *TEC*.

3.7.2.2 Economic Saving Potential

Schedule1 was taken as the target schedule, as it integrates both energy and labor awareness and exhibits no extreme sensitivity to an electricity price. Four baseline schedules were used to evaluate its economic performance (Figure 3.12).

Schedule1 is significantly superior to schedule2. It is 25% superior in *ELC*

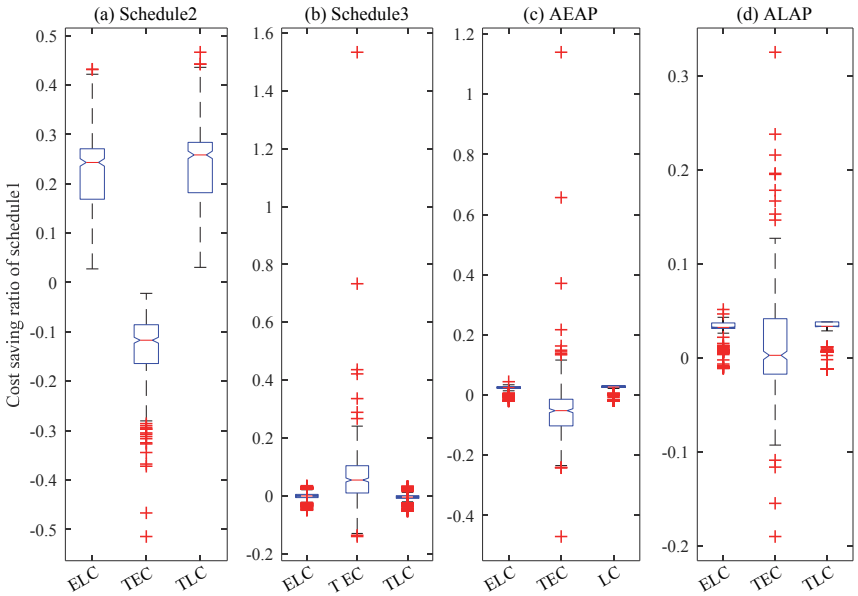


Figure 3.12: Cost saving potential of schedule1 in comparison to schedule2, schedule3, as-early-as-possible (AEAP) schedule, and as-late-as-possible (ALAP) schedule. The cost parts include joint energy and labor cost (ELC), total energy cost (TEC), and total labor cost (TLC). Production is disabled on weekends.

and *TLC*, while 12% inferior in *TEC* (Figure 3.12a). This again hints that it will impede the *TEC* reduction performance by integrating labor and energy awareness for joint optimization. However, the loss in *TEC* minimization is well compensated by the gain in *TLC* minimization.

Schedule1 and schedule3 achieve a similar performance. This is demonstrated by the zero average *ELC* saving ratio of schedule1 in comparison to schedule3 (Figure 3.12b). Schedule1 is 5% superior in *TEC*. But this gain is compensated by its 0.4% inferiority in *TLC*. This implies that the integration of energy and labor awareness will slightly affect the *TLC* optimization, while obtaining some gain in *TEC*.

Schedule1 is slightly superior to the AEAP schedule. It is 2% superior in *ELC* (Figure 3.12c), which is contributed by its 3% gain in *TLC*, despite its 5% inferiority in *TEC*. This demonstrates the effectiveness of schedule1 in *TLC* minimization by shift compression. Schedule1's inferiority in *TEC* is first of all explained by AEAP schedule's compact production without idling time between jobs. Besides, the integration of labor awareness in schedule1 impedes its *TEC* minimization.

Schedule1 is obviously superior to the ALAP schedule. It is 3% superior in *ELC* and *TLC*, with equal performance in *TEC* (Figure 3.12d). This equal perfor-

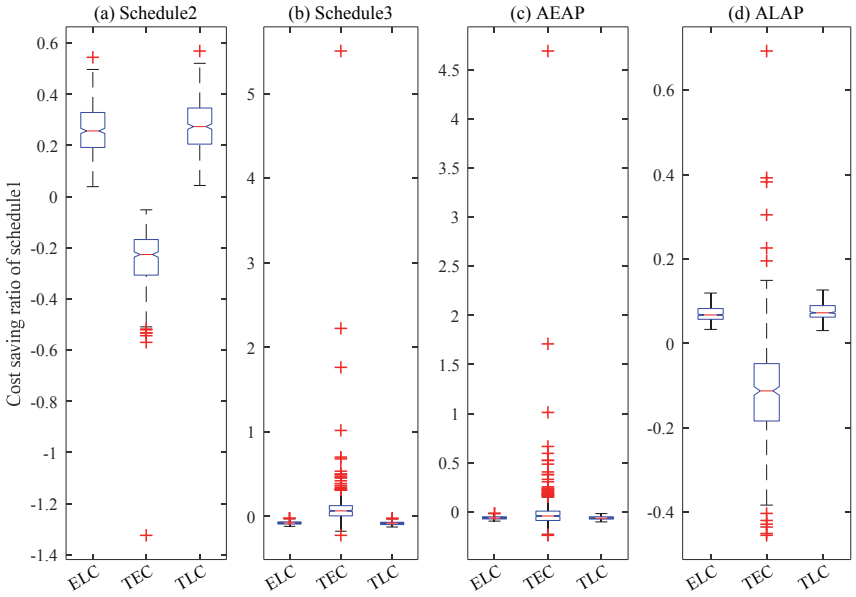


Figure 3.13: Cost saving potential of schedule1 compared to schedule2, schedule3, as-early-as-possible (AEAP) schedule, and as-late-as-possible (ALAP) schedule. The cost parts include joint energy and labor cost (ELC), total energy cost (TEC), and total labor cost (TLC). Production is allowed on weekends.

mance in *TEC* can be explained by the two aforementioned reasons.

3.7.3 Impact of Weekend Production

For the investigated factory, the wage of each personnel type at early and late shifts increases by 36% on weekends, while that at night shifts stays the same. All the other data and configurations remained the same, except that the weekly MEP was calculated from 6h Monday of a week to 6h Monday of the next week, assuming that the production can optionally be enabled on weekends besides the real labor shift of this factory (Table 3.8). The statistics are illustrated in Figure 3.13.

Overall, with enabled weekend production, the cost saving potential of schedule1 evidently rises compared to the ALAP schedule, stays at the same level compared to schedule2, and slightly decreases compared to schedule3 and the AEAP schedule.

Compared to schedule2 (Figure 3.13a), schedule1 achieves 26% less *ELC*, 23% more *TEC*, and 27% less *TLC*. In comparison with the case where production is disabled on weekends (Figure 3.12a), schedule1's *ELC* saving ratio remains the

same, while the inferiority in *TEC* and the superiority in *ELC* are strengthened, respectively. This is explained by the missing labor awareness in schedule2. As the electricity price is lower on weekends, schedule2 is more likely to shift production loads to weekends.

In contrast to schedule3 (Figure 3.13b), schedule1 is nearly 1% higher in *ELC* and *TLC*, and 7% lower in *TEC*. This implies that it is more frequent for schedule1 to shift loads to the daytime (early and late shifts) on weekends. Compared to the case without weekend production (Figure 3.12b), schedule1's superiority in *TEC* and inferiority in *TLC* are enhanced, as it assigns more weekend production to further reduce *TEC* while slightly increasing *TLC*.

In comparison to AEAP schedule (Figure 3.13c), schedule1 is 6% higher in joint cost and *TLC*, and 4% higher in *TEC*. Its inferiority in *TEC* is explained by the two reasons elucidated in Section 3.7.2.2. This inferiority is weakened compared to the peer case (Figure 3.12c), due to the lower electricity price on weekends. Additionally, as AEAP schedule has no weekend shift, schedule1 becomes also inferior in *TLC* and joint cost, compared to this peer case. Overall, this implies that the additional *TEC* gain on weekends is still overwhelmed by the obviously-increasing *TLC*.

Compared with ALAP schedule (Figure 3.13d), schedule1 is 7% lower in *ELC* and *TLC*, and 11% higher in *TEC*. In comparison to the case without production on weekends (Figure 3.12d), schedule1's superiority in *ELC* and *TLC* is enhanced, since weekend shifts are always included in ALAP schedule which increases *TLC* and subsequently *ELC*. However, schedule1 also becomes inferior in *TEC*, as ALAP schedule can make full use of the lower electricity price on weekends.

3.7.4 Impact of Production Loads

3.7.4.1 Economic Sensitivity to Number of Jobs

The number of jobs was varied with the same load (12,500 plastic bottles) to observe the economic performance evolution of schedules 1, 2, & 3. Three one-week periods were selected from the RTP data on the Belgium electricity spot market [55] between 2007 and 2015, such that the cases of lowest (24.32 euro/MWh), highest (163.10 euro/MWh), and median (48.45 euro/MWh) weekly MEP were encompassed.

The sensitivity curves are illustrated in Figure 3.14. Generally, schedule1 exhibits the most stable economic performance.

As shown by the first column in Figure 3.14, schedule1's *ELC* is sensitive to the number of jobs. This sensitivity is contributed by the linear relationship between *TLC* and number of jobs, which is further explained by the increasing shift number. Comparatively, schedule1's *TEC* is insensitive to the number of jobs.

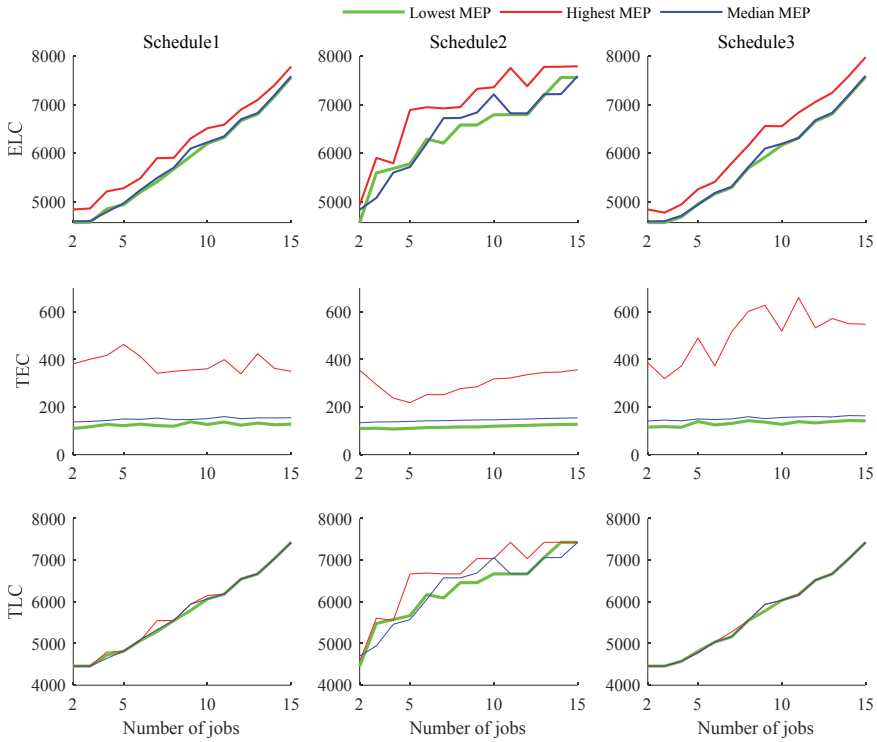


Figure 3.14: Impact of the number of jobs on economic performance of schedule1, schedule2, and schedule3, respectively. The economic performance includes joint energy and labor cost (ELC), total energy cost (TEC), and total labor cost (TLC). Production is disabled on weekends. The MEP denotes the mean electricity price in a week.

This is explained by the evidently lower power consumption during a changeover (9.00 kW), compared to that of bottle production (46.35 kW). Although an increasing number of jobs triggers more machine changeovers, the increased *TEC* for changeovers is minor compared with the *TEC* for bottle production.

The *ELC* of schedule2 (second column in Figure 3.14) increases with the rising number of jobs. This trend is contributed by *TLC*, which is nearly positively-correlated with the number of jobs. The *TLC*'s sensitivity curves under three different MEP exhibit some evident difference with each other, while these almost overlap in schedules 1 & 3. The *TLC* curves of schedule2 exhibit a relatively obvious variation with the rising number of jobs, compared to these of schedules 1 & 3. As the labor awareness is missing in schedule2, this implies that the integration of labor awareness in a schedule can effectively reduce the variations in the *TLC*

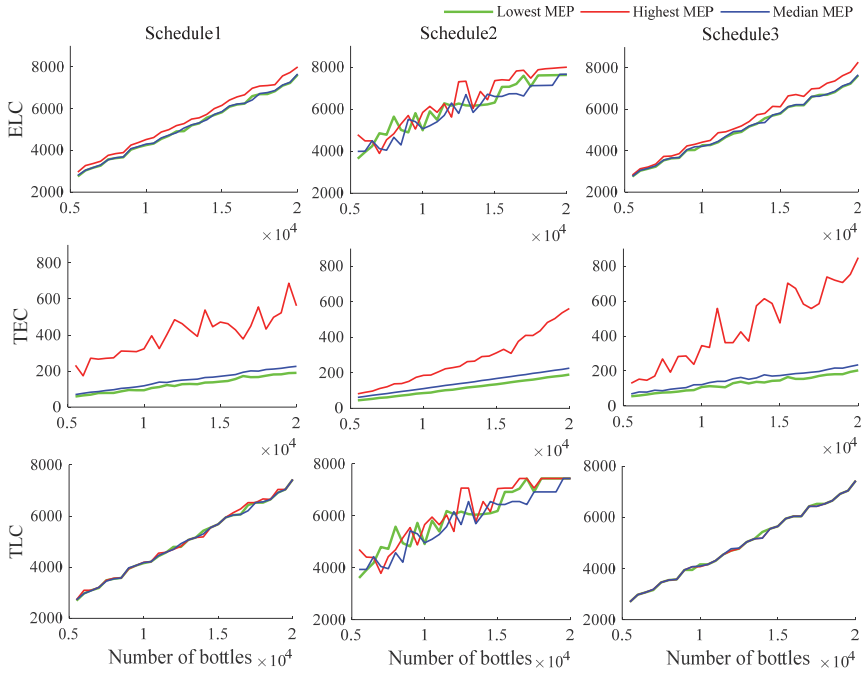


Figure 3.15: Impact of load duration (i.e., number of bottles) on economic performance of schedule1, schedule2, and schedule3, respectively. The economic performance includes joint energy and labor cost (ELC), total energy cost (TEC), and total labor cost (TLC). Production is disabled on weekends. The MEP denotes the mean electricity price in a week.

for a production load despite the potentially varying number of jobs for the same amount of load.

Regarding schedule3 (third column in Figure 3.14), the *ELC* and *TLC* exhibit an analogous sensitivity as that of the other two schedules. Nevertheless, the *TEC* under the highest MEP tends to rise with the increasing number of jobs. This is explained by the missing energy awareness in schedule3. When the electricity price is high, the *TEC* for additional changeovers shows up, compared to the *TEC* for this fixed amount of production (12,500 plastic bottles).

3.7.4.2 Economic Sensitivity to Load Duration

The load duration was varied by changing the number of bottles in each job, while fixing the number of jobs at 5. In the iterative experiment, the size of each job increased by 100 until the required accommodation reaches one week without week-

end production.

The sensitivity curves are depicted in Figure 3.15. Schedule1 demonstrates the most stable and predictable sensitivity to the load duration as well as the highest cost efficiency (e.g., 7792 euro of *ELC* in schedule1 vs. 8279 euro of *ELC* in schedule3 for producing 20,000 plastic bottles under the highest MEP). Each schedule's *ELC* (first row in Figure 3.15) increases with the rising load duration, fundamentally contributed by *TLC*.

Concerning *TEC* (second row in Figure 3.15), the sole energy awareness in schedule2 clearly contributes to the positive linear relationship between the *TEC* and MEP. The integration of energy awareness in schedule1 is also effective, which only creates slight variation under the highest MEP. In comparison, a lack of energy awareness in schedule3 causes significant variation under the highest MEP and an obviously higher *TEC* under a heavy load (number of bottles is above 15,000).

The *TLC* (third row in Figure 3.15) of schedules 1 & 3 steadily increase with the rising number of bottles. The overlap of the three *TLC* sensitivity curves in each of these two schedules demonstrates the effective integration of labor awareness. Comparatively in schedule2, a lack of labor awareness and the frequent load shifting enabled by sole energy awareness cause the evident variation in its *TLC* sensitivity curves.

3.7.5 Additional Test Instances

In addition to the former empirical case where the energy cost is minor compared to the labor cost, 9 test instances were generated by scaling the power consumption (scale: 10, 100, and 1000) and the labor wage (scale: 0.8, 1, and 1.2) in the former case study. Consequently, these test instances encompass small, medium, and large portions of energy cost in the joint energy and labor cost. To demonstrate at a large scale, the number of time slots was 604,800 (scheduling time span of one week with time granularity of one second) and the number of jobs was 300 and 400. For each instance, 10 runs were performed to get the average performance of schedules 1, 2, & 3, respectively. To accommodate the large number of jobs within one week, the number of plastic bottles was set to one, and the changeover time as well as the cycle time of each power state (except *Production* state) was set to one second.

As indicated in Table 3.10, an important trend found in the former case study holds in these test instances: schedule1 can achieve the lowest *ELC*. Several exceptions are observed in the instances where the power scale is 10 and labor wage scale is 1 and 1.2 (bold in Table 3.10). This is explained by the lower ratio of *TEC* over *ELC* (TEC/ELC smaller than 9.2%) compared to other test instances (TEC/ELC above 33.1%). However, these exceptions are minor, since sched-

Table 3.10: Scheduling performance in 10 repetitive runs of 9 different test instances. Schedule1 achieves the lowest ELC^a with exceptions in bold that are nevertheless very close to the minimal level.

Power scale	Labor wage scale	Schedule	Number of jobs = 300					Number of jobs = 400				
			ELC^a (euro)	TEC^a (euro)	TLC^a (euro)	TEC/ELC	Runtime (s)	ELC (euro)	TEC (euro)	TLC (euro)	TEC/ELC	Runtime (s)
10	0.8	1	445	31	414	7.0%	74	455	41	414	9.2%	181
		2	1532	31	1501	2.0%	73	1906	41	1865	2.2%	183
		3	446	58	388	13.0%	72	455	67	388	14.7%	183
	1	1	**548**	31	517	5.7%	73	**559**	42	517	7.5%	182
		2	2034	31	2003	1.5%	75	2103	42	2061	2.0%	182
		3	**542**	58	484	10.7%	74	**551**	67	484	12.2%	183
	1.2	1	**652**	31	621	4.8%	75	**662**	42	621	6.3%	185
		2	2545	31	2514	1.2%	75	2716	41	2675	1.5%	184
		3	**639**	58	581	9.1%	74	**648**	67	581	10.3%	180
100	0.8	1	721	307	414	42.6%	75	830	416	414	50.1%	181
		2	1945	306	1639	15.7%	75	1984	415	1569	20.9%	181
		3	967	579	388	59.9%	73	1055	667	388	63.2%	180
	1	1	824	307	517	37.3%	75	933	416	517	44.6%	181
		2	2275	306	1969	13.5%	74	2597	415	2182	16.0%	183
		3	1064	580	484	54.5%	73	1151	667	484	57.9%	182
	1.2	1	928	307	612	33.1%	74	1037	416	621	40.1%	183
		2	3004	306	2698	9.3%	74	3015	415	2600	13.8%	182
		3	1162	581	581	50.0%	72	1247	666	581	53.4%	183
1000	0.8	1	3483	3069	414	88.1%	74	4570	4156	414	90.9%	184
		2	4602	3064	1538	66.6%	73	5812	4148	1664	71.4%	184
		3	6201	5813	388	93.7%	73	7058	6670	388	94.5%	183
	1	1	3587	3069	517	85.6%	73	4674	4157	517	88.9%	180
		2	4943	3062	1881	61.9%	73	6372	4147	2225	65.1%	183
		3	6295	5811	484	92.3%	73	7147	6663	484	93.2%	184
	1.2	1	3691	3070	621	83.2%	74	4777	4156	621	87.0%	183
		2	5470	3063	2407	56.0%	74	6741	4149	3592	61.5%	182
		3	6382	5801	581	90.9%	72	7239	6658	581	92.0%	183

[a] *ELC*: joint energy and labor cost, *TEC*: total energy cost, *TLC*: total labor cost

ules 1 & 3 have very close *ELC* in all these exceptions, although schedule1 has a slightly higher *ELC* (within 2%).

Similar to the observation in the former case study, schedule1 never has extremely poor economic performance in contrast to the other two schedules. Schedule1 achieves a *TEC* which is equal or close to that of schedule2 in all the test instances (Table 3.10), while schedule3 has an evidently higher *TEC* due to a lack of energy awareness. Besides, schedule1 leads to a *TLC* close to that of schedule3 in all the instances (Table 3.10), while schedule2 causes an extremely high *TLC* (around 3 to 5 times higher compared to schedules 1 & 3) due to the missing labor awareness. Moreover, all these large-scale instances were solved within three minutes (Table 3.10, with Intel i5-3470 CPU @ 3.20 GHz and 8 GB RAM), which is a reasonable time for production scheduling.

3.7.6 Discussions

Based on the research questions in the beginning of Section 3.7 and the former sensitivity analyses, we can advance the understanding of energy and labor awareness integration for sustainable production scheduling, and derive several managerial implications.

3.7.6.1 Research Question 1

Integration of sole energy awareness to a production schedule can reduce the energy cost for production. If the electricity price becomes increasingly volatile, this contribution will be even more important. When the electricity price tends to increase, this integration can effectively slow down the rise of the energy cost. Besides, incorporation of both energy and labor awareness into a production schedule can also reduce the energy cost, although this reduction effect is relatively weaker. In contrast, integration of sole labor awareness in a production schedule will increase the energy cost due to the missing energy awareness.

3.7.6.2 Research Question 2

Integration of sole energy awareness to a production schedule is not effective to decrease the total energy and labor cost. This ineffectiveness is amplified when the energy cost only occupies a minor part. Furthermore, labor cost is quite sensitive to shifting loads to periods with low electricity prices, especially when it takes up an important part in the total cost. The minor energy-saving cost can be easily compensated by the increased labor cost due to additional labor shifts.

Conversely, incorporation of both energy and labor awareness to a production schedule can effectively decrease the total cost. Analogously, this total cost efficiency is gained by sacrificing the energy cost efficiency. The actual trade-off

would depend on the portion of energy cost and labor cost in every specific production case.

Therefore, it is insufficient to only integrate energy awareness to a production schedule, which is the common practice in the existing energy-efficient production scheduling research. Energy and labor awareness are both indispensable for sustainable production. Especially in the cases where labor cost plays the major role, labor awareness is crucial for total cost reduction. In energy-intensive production cases, energy awareness may be vital for total cost saving; but labor awareness is still fundamental for shop floor human worker planning.

3.7.6.3 Research Question 3

For an energy-efficient and labor-aware production schedule, the sensitive factors that influence the joint energy and labor cost include the electricity price, the option for weekend production, the number of jobs, and the number of parts/products. Overall, this schedule demonstrates moderate sensitivity, without any economic performance gap. This robust economic performance facilitates production managers to guarantee stable cost reduction and to perform reliable cost prediction when facing various sensitive production parameters.

3.7.6.4 Comparison with Existing Methods

As identified in Section 3.2.2, the existing studies on energy-efficient production scheduling ignore the consideration of labor, which is directly associated with production load shifting under volatile energy prices. These existing scheduling methods are represented by schedule2 in the former experiments, i.e., solely with energy awareness. It has been demonstrated that schedule2 evidently increases the labor cost as well as joint energy and labor cost, though the energy cost is minimized. Comparatively, the proposed method, which is represented by schedule1, can effectively reduce both energy and labor costs, contributing to robust and competitive economic performance for production execution on the shop floor.

Moreover, compared to the small problem size in most studies, the proposed scheduling method has been proven to effectively work at a larger problem size, regarding the number of jobs and time slots. This competence is increasingly important with the rising needs for highly-mixed and low-volume production.

3.8 Multi-objective Optimization Experiments

The energy and labor data presented in Section 3.6 are used in multi-objective optimization experiments with the AMOMA algorithm proposed in section 3.5.2. More specifically, two-week RTP data was taken from the Belgium electricity spot

market, of which price varies every hour. Note that the purpose of the following experiments is to demonstrate the effectiveness and efficiency of the AMOMA under time-varying electricity prices and labor wage, even with a large number of scheduling time slots. The two-week historical RTP data was thereby taken from the local electricity spot market, though this hourly-dynamic electricity price is known only one day in advance in reality. For real applications, the AMOMA can be used to produce a 24-h schedule under RTP. Artificial neural network (ANN)-based price forecasting may be used to enlarge the scheduling time span [56]. The scheduling time slot δt was set to one second, leading to overall 1,209,600 δt. Overall 10 independent jobs remained to be processed, whose duration varies from 8,960 seconds (500 bottles) to 71,680 seconds (4,000 bottles).

3.8.1 Parameter Tuning of AMOMA

The AMOMA was run on a computer with Intel Core i5-3470 @ 3.2 GHz and 8 GB RAM. The time budget was fixed at 2 minutes. While Δ which is defined by Equation (3.20) was employed to indicate the diversity driven by parameter vector $\vec{p}$, the metric $\gamma(\vec{p})$ was used to measure the convergence steered by $\vec{p}$:

$$\gamma(\vec{p}) = \frac{|NS^*(\vec{p})|}{|NS(\vec{p})|} \tag{3.22}$$

where $NS^*(\vec{p})$ is the global approximation set (domination-based aggregation of all nondominated solutions) contributed by $\vec{p}$, and $NS(\vec{p})$ are nondominated solutions produced by $\vec{p}$. $\vec{p}$ is (population size, crossover rate, mutation rate) in tuning exploration and one of S_{ts}, N_f, and T_{max} in turning exploitation. The AMOMA instance with each $\vec{p}$ was independently run 50 times.

The NSGA-II was tuned for stronger convergence, without local searches. Following the tuning guidance in Section 3.5.2.1, three promising levels for population size, crossover rate, and mutation rate were set to (100, 500, 1000), (0.7, 0.8, 0.9), and (0.1, 0.2, 0.3), respectively. Consequently, (1000, 0.9, 0.2) was selected due to its strongest convergence (γ was 16% compared to others between 0 and 11%).

Both CTS and DTS were tuned for higher convergence and preserved diversity, with the former tuned NSGA-II. As indicated in Figure 3.16a, the variation of S_{ts} within 1 h has little impact on both convergence and diversity. This is because the smallest time granule of electricity prices and labor wages is 1 h, such that a search step smaller than 1 h cannot enable finer exploitation in the solution space. The S_{ts} was thus set to 1 hour to speed up exploitation without affecting the convergence and diversity. As shown in Figure 3.16b, the pure exploration in the first N_f generations moderately influences the diversity, while it has little impact on the convergence. This is explained by the global search characteristic

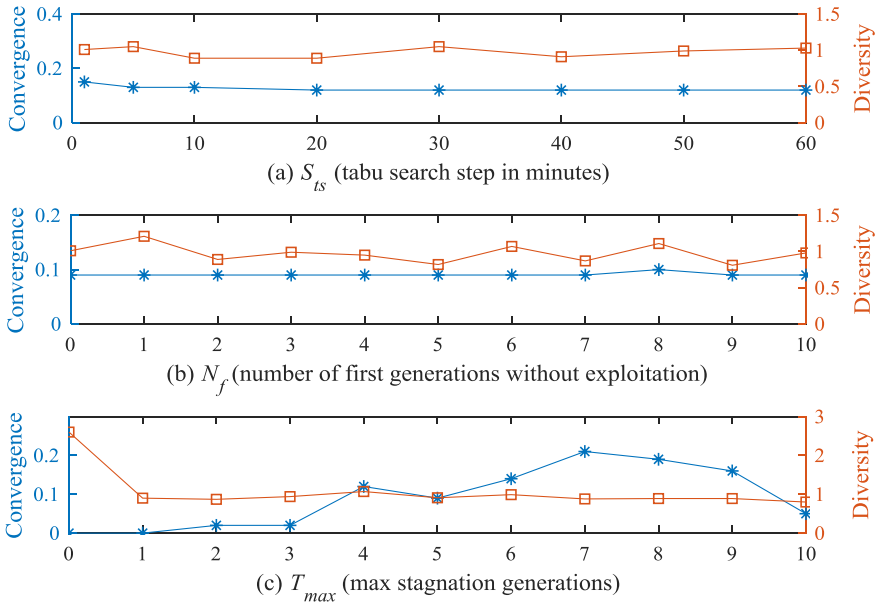


Figure 3.16: Parametric sensitivity of the two proposed tabu searches (CTS and DTS) in convergence and diversity. A higher convergence value and a lower diversity value indicate superior Pareto front approximation.

of NSGA-II, such that it cannot guarantee exploitation in a potential region of a solution space to improve the convergence. N_f were thereby set to 2 to introduce local searches as early as possible while retaining diversity. As implied in Figure 3.16c, the convergence and diversity are sensitive to T_{max} and whether to go on upon stagnation. This reveals that the joint exploration and exploitation of the proposed AMOMA can effectively prevent premature convergence. Therefore, T_{max} was set to 7 to achieve superior levels in both convergence and diversity without having to terminate the search too early or too late.

3.8.2 Scheduling of an Extrusion Blow Molding Process

3.8.2.1 Benchmark

The tuned AMOMA was compared with NSGA-II [22], GRASP (greedy randomized adaptive search procedure) [57], MA-C (hybrid of NSGA-II and CTS), MA-D (hybrid of NSGA-II and DTS), AMOMA-N (AMOMA which only exploits non-dominated solutions), and AMOMA-E (AMOMA which optimizes toward TEC and C_{max}).

NSGA-II, MA-C, MA-D, AMOMA-N, and AMOMA-E remained the corresponding configurations and time budget for AMOMA. In the construction phase

Table 3.11: Performance comparison (mean ± standard deviation) of 7 algorithms in 50 independent runs using the empirical data

Algorithm	$\|NS\|$[a]	γ[b]	Δ[c]
AMOMA	**10.28** ± 5.09	**0.33** ± 0.16	0.88 ± 0.56
NSGA-II	14.04 ± 2.51	0.05 ± 0.02	3.56 ± 0.35
GRASP	6.32 ± 1.82	0.14 ± 0.09	0.60 ± 0.35
MA-C	3.38 ± 0.67	0.06 ± 0.02	0.90 ± 0.49
MA-C	**9.28** ± 4.29	0.18 ± 0.10	0.87 ± 0.65
AMOMA-N	**9.76** ± 5.37	0.18 ± 0.12	0.87 ± 0.82
AMOMA-E	3.08 ± 1.73	0.06 ± 0.03	**0.30** ± 0.22

[a]Number of nondominated solutions.
[b]Convergence.
[c]Diversity.

of GRASP, a solution was constructed by iteratively building a restricted candidate list (RCL) and randomly selecting a job for this solution. In the local search phase of GRASP, CTS and DTS were sequentially used as in AMOMA. The parameter α ('α percent' distant from the nondominated solutions) for building an RCL was tuned at 0:0.2:1. It was set to 1 with the highest γ (0.48), implying that full randomness effectively improved the convergence. Each algorithm went through 50 independent runs.

The performance of an algorithm was evaluated in three dimensions: the number of nondominated solutions ($|NS|$), the convergence (γ) and the diversity Δ. Intermediate $|NS|$ is preferred, because small $|NS|$ provides insufficient trade-off insights and large $|NS|$ causes problems on optimal selection. High γ and low Δ are preferred.

As presented in Table 3.11, the AMOMA is among the best in $|NS|$, the best in convergence, and moderate in diversity. NSGA-II has the worst convergence by achieving the largest $|NS|$ and smallest γ. Its diversity is also the worst, indicating that it is incapable to evenly diversify the convergence introduced by dispatching rules in the initial population. This highlights the need for exploitation. The relatively small $|NS|$ of GRASP indicates its limitation in producing a set of nondominated solutions, compared to the population-based AMOMA which better fits MOPs. The small $|NS|$ of MA-C underlines the need of DTS to diversify the biased convergence introduced by the CTS. Pure CTS in MA-C cannot effectively enhance convergence. It needs the assist of DTS in diversity preservation to achieve comparable convergence of AMOMA. Although diversity preservation of DTS in MAC-D effectively strengthens the convergence compared to NSGA-II and MA-C, a lack of convergence enhancement measures prevents MA-D from achieving comparable convergence of AMOMA. The convergence of AMOMA-N (Table 3.11) is nearly halved by exploiting only nondominated solutions (premium

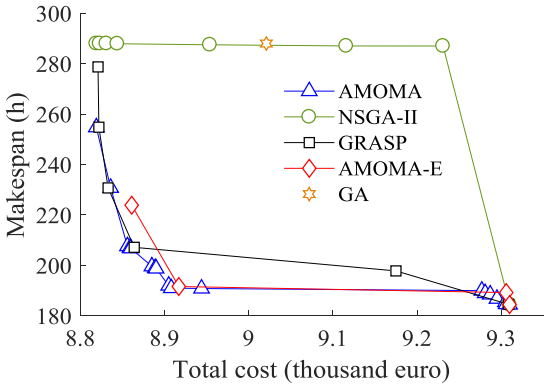


Figure 3.17: Pareto front approximations of 5 different algorithms on the trade-off between the makespan and the total cost (joint energy and labor cost)

group). This emphasizes the contribution of the prioritized grouping strategy in AMOMA for convergence enhancement, where the alternative group introduces randomness for the search to escape a local optimum. The smallest $|NS|$ and low γ of AMOMA-E reveal its incapability in minimizing ELC due to its ignorance of TLC.

3.8.2.2 Trade-Off Analysis

Figure 3.17 illustrates the Pareto front approximations of the best runs of AMOMA, NSGA-II, GRASP, and AMOMA-E regarding convergence, and a solution given by a genetic algorithm (GA) which optimizes toward TEC. Compared to NSGA-II, AMOMA is significantly more effective in reducing C_{max} while maintaining the range of ELC. The nearly constant large C_{max} of the solutions provided by NSGA-II implies its weak convergence for this proposed problem. It attempts to evidently shift production jobs over time to search for the minimization potential of ELC. Consequently, despite its similar ELC range compared to the AMOMA, this is evidently compensated by the prolonged C_{max}.

Although the Pareto front approximations of GRASP and AMOMA-E are partially analogous to that of AMOMA, AMOMA remains superior in the entire range of ELC (8.8 - 9.3 thousand euro) that is achieved by all the algorithms. An observation in the approximation set of AMOMA is that a slight prolongation of a short C_{max} (by 0.5%) can dramatically reduce ELC (by 4%). However, a further decrease in ELC (by 1%) has to be compromised by a significant rise in C_{max} (by 33%). This is because jobs tend to go across weekends to search for more economical periods when C_{max} rises. Nonetheless, the significantly raised time flexibility does not induce corresponding evident reduction in ELC, since the RTP

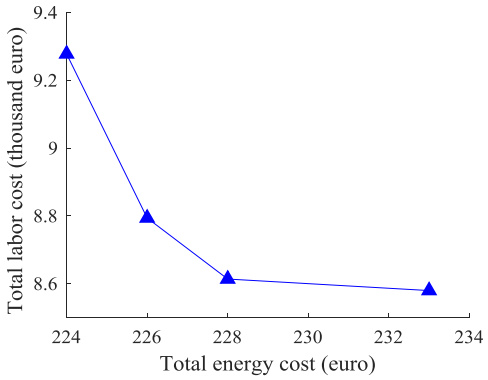


Figure 3.18: Quantified trade-off between the energy cost and the labor cost

data is hourly dynamic but daily similar on weekdays and TLC dominants TEC in this case study.

The single solution provided by the GA is far away from the approximation set of AMOMA (Figure 3.17). This proves that the existing energy-aware scheduling methods cannot optimize ELC and C_{max}, which are important production metrics. Such a poor result is explained by two reasons. Firstly, analogous to the NSGA-II, the GA focuses on the global search without any guarantee on exploitation in the solution space. Secondly, the total labor cost cannot be explicitly minimized in this GA, such that the ELC cannot be efficiently minimized though this GA enlarges the C_{max} to search for more TC minimization potential.

A notable observation on Figure 3.17 is the nearly constant high level of the makespan of the solutions provided by the NSGA-II. This reveals NSGA-II's weak convergence for this proposed scheduling problem and its intrinsic characteristic of global search. It attempts to evidently shift the production jobs over the time horizon to search for the minimization potential of the total cost. Consequently, although it has a similar range of the total cost compared to other benchmark algorithms, this is significantly compromised by a constant high level of makespan.

The trade-off relationship of TEC and TLC is sketched in Figure 3.18, which was obtained by setting objectives as TEC and TLC. A slight decrease in TEC from 228 to 224 euro (2%) can lead to an obvious increase in TLC from 8.6 to 9.3 thousand euro (8%). This could be because TEC is evidently smaller than ELC in this case study. Consequently, ELC is more sensitive to production load shifting over time, compared to TEC. This again highlights the economic necessity to jointly consider energy and labor in production load shifting.

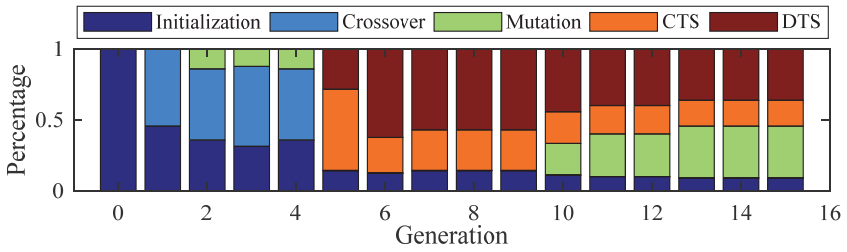


Figure 3.19: Source of nondominated solutions provided by an AMOMA (adaptive multi-objective memetic algorithm) instance, where CTS are DTS are the convergence- and diversity-oriented tabu searches, respectively

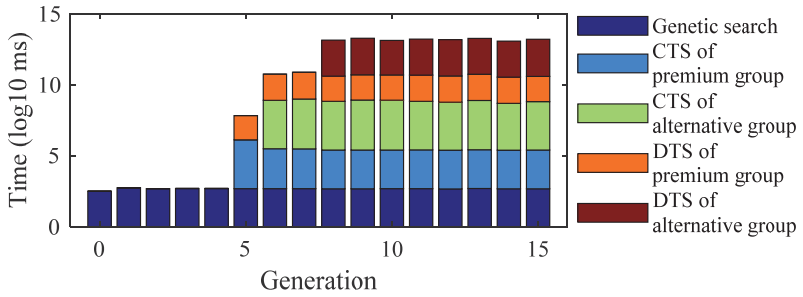


Figure 3.20: CPU time consumed by major components of an AMOMA (adaptive multi-objective memetic algorithm) instance, where CTS are DTS are the convergence- and diversity-oriented tabu searches, respectively

3.8.2.3 Adaptation Behavior

The AMOMA will be proven to have a synergistic and adaptive balance between exploration and exploitation. Figure 3.19 demonstrates the dynamic percentage of AMOMA operators in producing nondominated solutions in an AMOMA search. Genetic search operators (meme-integrated initialization, crossover, and mutation) retain full occupation in the first four generations. Afterward, local search operators (CTS and DTS) dominate while crossover remains a portion of zero. This reveals that exploration and exploitation stay strong and weak in the early stage of a search, respectively, and vice versa in the late stage. In the early stage, potential regions have to be explored as many as possible. In the late stage, these potential regions should be extensively exploited before terminating the entire search. This reasoning is further demonstrated in Figure 3.20. The CPU time is evenly shared among genetic search and two local searches that are applied to two groups, respectively. This exhibits the balanced intensity of different operators. The search

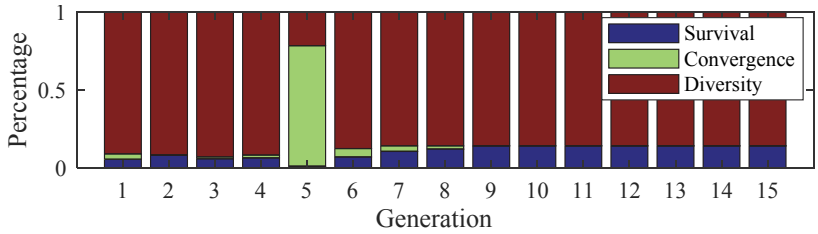


Figure 3.21: Convergence, diversity, and survival rate of an AMOMA (adaptive multi-objective memetic algorithm) instance

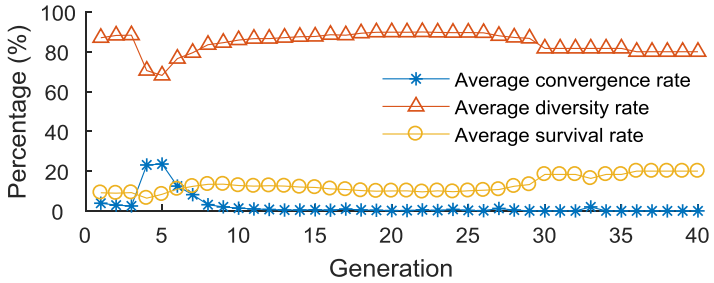


Figure 3.22: Average convergence, diversity, and survival rates over 50 independent runs of the AMOMA.

makes increasing and full use of exploitation in generations 5-7 and 8-15, respectively. This shows the adaptation behavior of AMOMA.

Inspired by the cross-dominance metric λ [53], the cross-nondominance and equality metrics are introduced to measure the diversity and survival of the NS_t compared with the NS_{t-1}. The cross-nondominance indicates the nondominance between two solutions that have at least one different objective value. Equality means that two solutions are equal in all objective values. Figure 3.21 presents a dynamic change of these metrics in an AMOMA instance. Throughout the search, the diversity remains high and the survival rate steadily rises. This implies that premature convergence is avoided in AMOMA. Conversely, the convergence rate varies a lot. It is slightly contributed by the genetic search in generations 1-4 (0-3%), and is significantly contributed by the CTS in generation 5 (77%). It gradually reduces after generation 6 until falling to zero in generation 9 and remaining zero in generations 10-15. The single peak of convergence demonstrates the deepest descent local search property of CTS, which contributes to fast convergence.

Figure 3.22 further presents the average convergence, diversity, and survival rates over these 50 independent AMOMA instances. Note that the number of generations varies between 11 and 40 in these 50 runs, due to the self-adaptive termination criterion of AMOMA. Therefore, the calculation of average rates be-

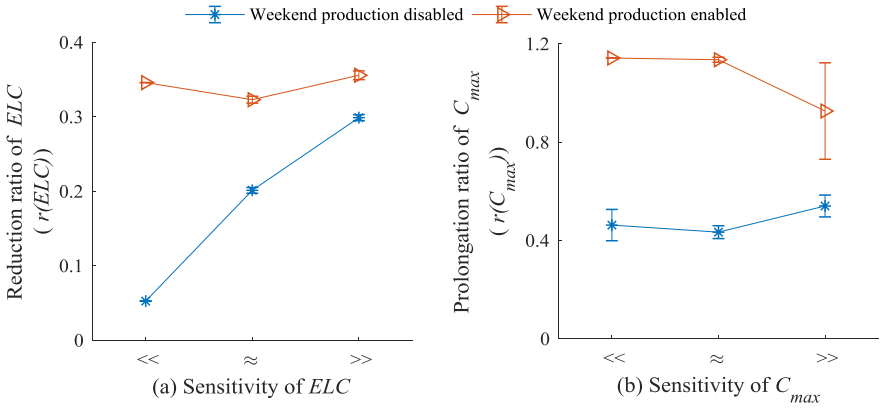


Figure 3.23: Statistical reduction potential of *ELC* (joint energy and labor cost) and prolongation trend of C_{max} (in terms of mean and standard deviation) in three scenarios ($<<$: *TEC* is dominated by *TLC*, $\approx$: *TEC* is comparable to *TLC*, $>>$: *TEC* dominates *TLC*)

tween generations 11 and 40 only considers the runs that has the corresponding generation number. As clearly exhibited by Figure 3.22, the convergence curve nearly gets saturated even for the smallest generation number (11). This indicates the fast convergence of AMOMA within only 2 min, satisfying industrial DR's or production scheduling's requirement of fast and high-quality decision making. Besides, the peak average convergence rate at around generation 5 exhibits that the premature convergence can be generally avoided. The average trends of these three rates comply with these shown in Figure 3.21. This implies that the former analyzed adaptation behavior of AMOMA is representative.

3.8.2.4 Economic Sensitivity

In the former case study, *TEC* is dominated by *TLC* and production is not allowed on weekends. Two other cases were assumed by scaling the power of the EBM machine: *TEC* is comparable to *TLC* (power scale was 50) and *TEC* dominates *TLC* (power scale was 1000). Each case had two scenarios: enabled and disabled weekend production. The AMOMA with the former configuration was repeated 50 times for each scenario. The normalized range of *ELC* and C_{max} of an approximation set was obtained by Equation (3.23) and Equation (3.24), respectively:

$$r(ELC) = \frac{[max(ELC) - min(ELC)]}{max(ELC)} \tag{3.23}$$

$$r(C_{max}) = \frac{[max(C_{max}) - min(C_{max})]}{min(C_{max})} \tag{3.24}$$

where $r(ELC)$ characterizes the maximal ELC reduction rate of AMOMA given a 2-minute time budget, and $r(C_{max})$ measures the maximal prolongation rate of $r(C_{max})$ that has to be compensated to achieve this $r(ELC)$.

As indicated by Figure 3.23a, the option of weekend production impacts $r(TC)$. If it is enabled, $r(ELC)$ remains a relatively high level and is nearly insensitive to the proportion of TEC and TLC. This is because weekends provide the AMOMA more periods with lower electricity prices to optimize ELC considering the trade-off between TEC and TLC. When weekend production is disabled, $r(ELC)$ increases with the rising share of TEC in ELC (Figure 3.23a). This is explained by the higher-priced periods on weekdays, such that the increasing portion of TEC in ELC makes $r(ELC)$ much more sensitive to load shifting over time.

Figure 3.23b demonstrates that if *TEC* does not dominate *TLC*, $r(TC)$ stays constant regardless of weekend production. This implies that if the share of *TEC* is not significant, the C_{max} will increase by over 120% and 40% to achieve the ELC reduction ratio in Figure 3.23a with and without weekend production, respectively. If *TEC* dominates *TLC*, $r(ELC)$ without weekend production slightly grows, as higher time flexibility is needed for more lower-priced periods to reduce the dominant *TEC* and thus *ELC*; $r(ELC)$ with weekend production moderately drops with an obviously larger variation, implying that the major *TEC* portion in *ELC* and the increased lower-pricing periods reduce the prolongation of the C_{max} for *ELC* minimization despite the rising variation.

3.9 Discussions and Conclusions

To finalize this chapter, the strengths, weaknesses, and application of this proposed scheduling method will be first discussed. Then the conclusions will be made.

3.9.1 Discussions

In contrast to the existing energy-aware production scheduling research, the work presented in this chapter pushes forward the knowledge boundary in three dimensions. Firstly, the integrated energy- and labor-aware production scheduling model breaks the conventional research barrier, where production scheduling and personnel planning are performed in a standalone manner. This integrated production system modeling philosophy may trigger more novel modeling and optimization research on integrated decision making for the shop floor production management or even the supply chain management. Secondly, the proposed AMOMA is able to provide high-quality nondominated solutions in less than 2 minutes (with an Intel Core i5-3470 @ 3.2 GHz and 8 GB RAM). This fully complies with the short time budget of production scheduling on a real shop floor. Although conventional dispatching rules (e.g., as-early-as-possible) also require a small time budget, they

are not scalable in the type and complexity of a scheduling problem and they cannot guarantee the quality of the scheduling solutions. Therefore, the presented AMOMA design will trigger more research on novel metaheuristics that provide higher-quality solutions in a shorter time period. Thirdly, an empirical case study is performed in a plastic bottle manufacturer, in comparison to most scheduling research that is based on theoretical data. Empirical economic insights are revealed by extensive sensitivity analyses and Pareto trade-off analyses. They further highlight the economic significance of performing integrated energy-aware production and labor scheduling, which is widely ignored in literature.

On the other hand, the work presented in this chapter exhibits the following weaknesses. Firstly, despite the integrated complexity, the scheduling model is restricted to a single machine. The impact would be greatly amplified if this model extends to a larger scale, such as a flexible job shop. Secondly, the idea of integrated modeling triggers more work that needs to be exploited beyond this presented energy- and labor-aware scheduling model. For instance, AGVs (automated guided vehicles) penetrate in factories for automated transportation. As transportation of material, parts, and semi-products among machines potentially influence the throughput and energy consumption of a production system, it would unlock more production cost reduction potential by further building an integrated scheduling model that considers energy conservation, job and human worker allocations, AGV route planning, as well as the capacity planning of upstream and downstream buffers of each machine. Thirdly, the presented work is limited to numerical experiments. A practical work that makes it more convincing is to apply the proposed schedule on the shop floor and measure and compare the production system performance before and after using the proposed scheduling method.

The introduced work in this chapter can help factories to automatically and fast produce high-quality integrated production and labor schedules that remain cost competitive even under real-time electricity pricing. While both single-objective optimization and multi-objective optimization can be performed according to the preference of human decision makers, multi-objective optimization is recommended, as the outputted nondominated solutions are proven not to be dominated by the single solution provided by the single-objective optimization. Furthermore, the outputted Pareto front approximation can help decision makers to quantify the trade-off between objectives and identify the most beneficial range to play with this trade-off. Last but not least, the sensitivity analysis based on simulation-optimization is able to help factory managers to quantitatively evaluate and predict the performance of a production system by using the proposed scheduling method, and to identify the variables that are most or least related to this performance.

3.9.2 Conclusions

As an industrial demand response (DR) approach to minimize the energy cost under real-time electricity pricing, the existing energy-efficient production scheduling studies only focus on load shifting to lower-priced periods under real-time electricity pricing. They ignore the labor cost which has a trade-off relationship with the energy cost. A lack of labor awareness may significantly increase the labor cost, which compensate the reduced energy cost. To fill this gap and go further beyond Chapter 2, this chapter investigated integrated energy- and labor-aware production scheduling in terms of modeling, simulation, optimization, and empirical quantitative analytics.

The proposed MIP model schedules jobs and human workers on a single machine, while considering energy conservation and energy cost reduction under real time electricity pricing. A continuous-time shift accumulation heuristic is proposed to coordinate the power state evolution and labor shift switch over time in the integrated energy- and labor-aware production simulation. An adaptive multi-objective memetic algorithm (AMOMA) is proposed to fast converge toward the Pareto front without deteriorating diversity. It balances the exploration and exploitation in the search space by synergistically integrating convergence- and diversity-oriented tabu searches (CTS and DTS) in the NSGA-II. The CTS and DTS are reactively launched upon a cross-dominance-based convergence rate of zero. Besides a premium group for local searches, an ordinary group is used to raise the refinement frequency when no qualified local optimum is found from the former group.

A case study was performed in a Belgian plastic bottle manufacturer. While several complete site surveys were carried out, energy measurements were conducted for over one year on an extrusion blow molding (EBM) machine during its production. Through numerical experiments, these hybridization and adaptation measures were proven effective to achieve fast Pareto front convergence without deteriorating diversity. Extensive benchmarking demonstrated the superiority of AMOMA compared to the widely used multi-objective metaheuristics, e.g., NSGA-II and GRASP. Through extensive sensitivity analyses, the electricity price, the weekend production, the number of jobs, and the production quantity turned out to be sensitive factors for the joint energy and labor cost. Compared to a schedule only with energy awareness or labor awareness, an energy-efficient and labor-aware production schedule demonstrated stable and superior economic performance, regarding energy cost, labor cost, and a sum of these two cost parts. While the energy cost is dominated by the labor cost in this case study, this conclusion remains in the other two representative test cases where the energy cost is comparable to or dominates the labor cost, respectively. Therefore, it is recommended to jointly integrate energy and labor awareness in production scheduling in order to unlock more production cost reduction potential on the shop floor.

References

[1] Gökan May, Bojan Stahl, and Marco Taisch. *Energy management in manufacturing: Toward eco-factories of the future – A focus group study*. Applied Energy, 164(Supplement C):628 – 638, 2016.

[2] *Report of the world commission on environment and development: Our common future*. Technical report, Brundtland Commission, 1987.

[3] Cristina Gimenez, Vicenta Sierra, and Juan Rodon. *Sustainable operations: Their impact on the triple bottom line*. International Journal of Production Economics, 140(1):149 – 159, 2012. Sustainable Development of Manufacturing and Services.

[4] Adriana Giret, Damien Trentesaux, and Vittal Prabhu. *Sustainability in manufacturing operations scheduling: A state of the art review*. Journal of Manufacturing Systems, 37(Part 1):126 – 140, 2015.

[5] T. Strasser, F. Andrén, J. Kathan, C. Cecati, C. Buccella, P. Siano, P. Leitão, G. Zhabelova, V. Vyatkin, P. Vrba, and V. Mařík. *A Review of Architectures and Concepts for Intelligence in Future Electric Energy Systems*. IEEE Transactions on Industrial Electronics, 62(4):2424–2438, April 2015.

[6] Y. Liu, C. Yuen, N. Ul Hassan, S. Huang, R. Yu, and S. Xie. *Electricity Cost Minimization for a Microgrid With Distributed Energy Resource Under Different Information Availability*. IEEE Transactions on Industrial Electronics, 62(4):2571–2583, April 2015.

[7] T. V. Theodoropoulos, I. G. Damousis, and A. J. Amditis. *Demand-Side Management ICT for Dynamic Wireless EV Charging*. IEEE Transactions on Industrial Electronics, 63(10):6623–6630, Oct 2016.

[8] Christian Gahm, Florian Denz, Martin Dirr, and Axel Tuma. *Energy-efficient scheduling in manufacturing companies: A review and research framework*. European Journal of Operational Research, 248(3):744 – 757, 2016.

[9] Y. M. Ding, S. H. Hong, and X. H. Li. *A Demand Response Energy Management Scheme for Industrial Facilities in Smart Grid*. IEEE Transactions on Industrial Informatics, 10(4):2257–2269, Nov 2014.

[10] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[11] Lennart Merkert, Iiro Harjunkoski, Alf Isaksson, Simo Säynevirta, Antti Saarela, and Guido Sand. *Scheduling and energy – Industrial challenges and*

opportunities. Computers & Chemical Engineering, 72(Supplement C):183 – 198, 2015. A Tribute to Ignacio E. Grossmann.

[12] Hubert Hadera and Iiro Harjunkoski. *Continuous-time batch scheduling approach for optimizing electricity consumption cost*, volume 32 of *Computer Aided Chemical Engineering*, pages 403 – 408. Elsevier, 2013.

[13] S.T. Newman, A. Nassehi, R. Imani-Asrai, and V. Dhokia. *Energy efficient process planning for CNC machining*. CIRP Journal of Manufacturing Science and Technology, 5(2):127 – 136, 2012.

[14] Ying Liu, Haibo Dong, Niels Lohse, and Sanja Petrovic. *A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance*. International Journal of Production Economics, 179(Supplement C):259 – 272, 2016.

[15] Yong Wang and Lin Li. *Time-of-use based electricity cost of manufacturing systems: Modeling and monotonicity analysis*. International Journal of Production Economics, 156(Supplement C):246 – 259, 2014.

[16] Wen-an Yang, Yu Guo, and Wenhe Liao. *Multi-objective optimization of multi-pass face milling using particle swarm intelligence*. The International Journal of Advanced Manufacturing Technology, 56(5):429–443, Sep 2011.

[17] Yong Wang and Lin Li. *Manufacturing profit maximization under time-varying electricity and labor pricing*. Computers & Industrial Engineering, 104:23 – 34, 2017.

[18] Xu Gong, Marlies Van der Wee, Toon De Pessemier, Sofie Verbrugge, Didier Colle, Luc Martens, and Wout Joseph. *Integrating labor awareness to energy-efficient production scheduling under real-time electricity pricing: An empirical study*. Journal of Cleaner Production, 168(Supplement C):239 – 253, 2017.

[19] Michael L. Pinedo. *Scheduling - Theory, Algorithms, and Systems*. Springer, 5 edition, 2016.

[20] Mitsuo Gen, Wenqiang Zhang, Lin Lin, and YoungSu Yun. *Recent advances in hybrid evolutionary algorithms for multiobjective manufacturing scheduling*. Computers & Industrial Engineering, 2017.

[21] X. D. Xue, K. W. E. Cheng, T. W. Ng, and N. C. Cheung. *Multi-Objective Optimization Design of In-Wheel Switched Reluctance Motors in Electric Vehicles*. IEEE Transactions on Industrial Electronics, 57(9):2980–2987, Sept 2010.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.

[23] L. A. Pereira, S. Haffner, G. Nicol, and T. F. Dias. *Multiobjective Optimization of Five-Phase Induction Machines based on NSGA-II*. IEEE Transactions on Industrial Electronics, PP(99):1–1, 2017.

[24] K. Sindhya, K. Miettinen, and K. Deb. *A Hybrid Framework for Evolutionary Multi-Objective Optimization*. IEEE Transactions on Evolutionary Computation, 17(4):495–511, Aug 2013.

[25] Ferrante Neri and Carlos Cotta. *Memetic algorithms and memetic computing optimization: A literature review*. Swarm and Evolutionary Computation, 2(Supplement C):1 – 14, 2012.

[26] Xu Gong, Toon De Pessemier, Luc Martens, and Wout Joseph. *Energy-Efficient and Labor-Aware Production Scheduling based on Multi-Objective Optimization*. In Antonio Espuña, Moisès Graells, and Luis Puigjaner, editors, 27th European Symposium on Computer Aided Process Engineering, volume 40 of *Computer Aided Chemical Engineering*, pages 1369 – 1374. Elsevier, 2017.

[27] X. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan. *A Multi-Facet Survey on Memetic Computation*. IEEE Transactions on Evolutionary Computation, 15(5):591–607, Oct 2011.

[28] A. Maesani, G. Iacca, and D. Floreano. *Memetic Viability Evolution for Constrained Optimization*. IEEE Transactions on Evolutionary Computation, 20(1):125–144, Feb 2016.

[29] Gilles Mouzon, Mehmet B. Yildirim, and Janet Twomey. *Operational methods for minimization of energy consumption of manufacturing equipment*. International Journal of Production Research, 45(18-19):4247–4271, 2007.

[30] M. B. Yildirim and G. Mouzon. *Single-Machine Sustainable Production Planning to Minimize Total Energy Consumption and Total Completion Time Using a Multiple Objective Genetic Algorithm*. IEEE Transactions on Engineering Management, 59(4):585–597, Nov 2012.

[31] Luoke Hu, Chen Peng, Steve Evans, Tao Peng, Ying Liu, Renzhong Tang, and Ashutosh Tiwari. *Minimising the machining energy consumption of a machine tool by sequencing the features of a part*. Energy, 121(Supplement C):292 – 305, 2017.

[32] Fadi Shrouf, Joaquin Ordieres-Meré, Alvaro García-Sánchez, and Miguel Ortega-Mier. *Optimizing the production scheduling of a single machine to minimize total energy consumption costs*. Journal of Cleaner Production, 67:197 – 207, 2014.

[33] Fadi Shrouf and Giovanni Miragliotta. *Energy management based on Internet of Things: practices and framework for adoption in production management*. Journal of Cleaner Production, 100:235 – 246, 2015.

[34] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *An energy-cost-aware scheduling methodology for sustainable manufacturing*. Procedia CIRP, 29(Supplement C):185 – 190, 2015. The 22nd CIRP Conference on Life Cycle Engineering.

[35] Ada Che, Yizeng Zeng, and Ke Lyu. *An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs*. Journal of Cleaner Production, 129(Supplement C):565 – 577, 2016.

[36] Kan Fang, Nelson A. Uhan, Fu Zhao, and John W. Sutherland. *Scheduling on a single machine under time-of-use electricity tariffs*. Annals of Operations Research, 238(1):199–227, Mar 2016.

[37] Kai Li, Xun Zhang, Joseph Y.-T. Leung, and Shan-Lin Yang. *Parallel machine scheduling problems in green manufacturing industry*. Journal of Manufacturing Systems, 38(Supplement C):98 – 106, 2016.

[38] Hao Zhang, Fu Zhao, Kan Fang, and John W. Sutherland. *Energy-conscious flow shop scheduling under time-of-use electricity tariffs*. CIRP Annals - Manufacturing Technology, 63(1):37 – 40, 2014.

[39] Hao Luo, Bing Du, George Q. Huang, Huaping Chen, and Xiaolin Li. *Hybrid flow shop scheduling considering machine electricity consumption cost*. International Journal of Production Economics, 146(2):423 – 439, 2013.

[40] Hadi Mokhtari and Aliakbar Hasani. *An energy-efficient multi-objective optimization for flexible job-shop scheduling problem*. Computers & Chemical Engineering, 104(Supplement C):339 – 352, 2017.

[41] Yan He, Yufeng Li, Tao Wu, and John W. Sutherland. *An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops*. Journal of Cleaner Production, 87(Supplement C):245 – 254, 2015.

[42] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A stochasticity handling heuristic in energy-cost-aware scheduling for sustainable production*. Procedia CIRP, 48(Supplement C):108 – 113, 2016. The 23rd CIRP Conference on Life Cycle Engineering.

[43] Yingfeng Zhang, Jin Wang, and Yang Liu. *Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact*. Journal of Cleaner Production, 167(Supplement C):665 – 679, 2017.

[44] Xu Gong, Marlies Van der Wee, Toon De Pessemier, Sofie Verbrugge, Didier Colle, Luc Martens, and Wout Joseph. *Energy- and Labor-aware Production Scheduling for Sustainable Manufacturing: A Case Study on Plastic Bottle Manufacturing*. Procedia CIRP, 61(Supplement C):387 – 392, 2017. The 24th CIRP Conference on Life Cycle Engineering.

[45] Niloofar Salahi and Mohsen A. Jafari. *Energy-performance as a driver for optimal production planning*. Applied Energy, 174(Supplement C):88 – 100, 2016.

[46] Abhay Sharma, Fu Zhao, and John W. Sutherland. *Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff*. Journal of Cleaner Production, 108(Part A):256 – 270, 2015.

[47] Eberhard Abele, Niklas Panten, and Benjamin Menz. *Data collection for energy monitoring purposes and energy control of production machines*. Procedia CIRP, 29(Supplement C):299 – 304, 2015. The 22nd CIRP Conference on Life Cycle Engineering.

[48] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G.Rinnooy Kan. *Optimization and approximation in deterministic sequencing and scheduling: a survey*. In P.L. Hammer, E.L. Johnson, and B.H. Korte, editors, Discrete Optimization II, volume 5 of *Annals of Discrete Mathematics*, pages 287 – 326. Elsevier, 1979.

[49] Tarik Aouam, Kobe Geryl, Kunal Kumar, and Nadjib Brahimi. *Production planning with order acceptance and demand uncertainty*. Computers & Operations Research, 91:145 – 159, 2018.

[50] L. Park, Y. Jang, S. Cho, and J. Kim. *Residential Demand Response for Renewable Energy Resources in Smart Grid Systems*. IEEE Transactions on Industrial Informatics, 13(6):3165–3173, Dec 2017.

[51] N. G. Paterakis, O. Erdinç, A. G. Bakirtzis, and J. P. S. Catalão. *Optimal Household Appliances Scheduling Under Day-Ahead Pricing and Load-Shaping Demand Response Strategies*. IEEE Transactions on Industrial Informatics, 11(6):1509–1519, Dec 2015.

[52] F. Qiao, Y. Ma, M. Zhou, and Q. Wu. *A Novel Rescheduling Method for Dynamic Semiconductor Manufacturing Systems*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, PP(99):1–11, 2018.

[53] Andrea Caponio and Ferrante Neri. *Integrating cross-dominance adaptation in multi-objective memetic algorithms*, pages 325–351. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[54] X. Gong, T. De Pessemier, W. Joseph, and L. Martens. *A power data driven energy-cost-aware production scheduling method for sustainable manufacturing at the unit process level*. In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–8, Sept 2016.

[55] Belgium electricity spot market. *Belpex*. http://www.belpex.be/market-results/the-market-today/dashboard/, 2017.

[56] X. Huang, S. H. Hong, and Y. Li. *Hour-Ahead Price Based Energy Management Scheme for Industrial Facilities*. IEEE Transactions on Industrial Informatics, 13(6):2886–2898, Dec 2017.

[57] Gilles Mouzon and Mehmet B. Yildirim. *A framework to minimise total energy consumption and total tardiness on a single machine*. International Journal of Sustainable Engineering, 1(2):105–116, 2008.

4

Energy- and Labor-Aware Flexible Job Shop Scheduling

Beyond the work introduced in Chapter 2 and Chapter 3, this chapter extends the energy- and labor-aware production scheduling method from the previous single-machine level to the flexible job shop level. This is one of the most complex shop floor configurations for production. More specifically, partial flexible job shop, job recirculation, and operation sequence-dependent machine setup times are considered in this shop floor-wide production scheduling model to increase its practical significance. On this modeled shop floor, each machine has a set of power states of which the inter-transitions are performed over time, to mimic the dynamic energy consumption behavior of a machine, as well as the overall energy consumption pattern. The number and type of human workers are matched to the scheduled production loads, with varying labor wage over shifts. The overall production is further framed into a set of labor shifts, to make the scheduling model more practical and increase its economic importance to factories. A discrete-event simulation (DES) framework is used to build this shop floor-wide production model. The whole scheduling problem has five production metrics for simultaneous optimization: makespan, total energy cost, total labor cost, maximal workload, and total

workload. The nondominated sorting genetic algorithm-III (NSGA-III), which has been recently proposed by Deb K. [1], is tailored for this many-objective optimization problem (MaOP). This tailoring includes the encoding and decoding of a scheduling solution, crossover, mutation, and solution evaluation using the DES framework. Through numerical experiments under real-time pricing (RTP) and time-of-use pricing (ToUP), insights are statistically obtained on the relation among these five production metrics. Specifically, the previously-revealed trade-off relations between the energy cost and the makespan (Chapter 2) as well as between the energy cost and the labor cost (Chapter 3), are found to still remain, when the shop floor configuration shifts from the single machine to the flexible job shop floor. Furthermore, the effectiveness and efficiency of NSGA-III in solving a MaOP are demonstrated in this chapter.

4.1 Introduction

As presented in detail by Section 2.1, the adaptation of end users' consumption behaviors to the volatile electricity prices is part of the initiative in smart grids, called price-based or time-based demand response (DR) [2]. Also as pointed out by Section 3.1 and [3], the existing research on DR is much more extensive in residential applications than in the manufacturing industry. This can be explained by the threefold reasons. Firstly, there are more constraints in modeling the production activities on the shop floor due to the interdependency of machines, a common lack of detailed energy data of each machine, the due time which is often a hard constraint, and so on. Secondly, factory managers are less open to production load shifting or reorganization, as this directly affects the actual production organization and the corresponding economic benefits. Thirdly, besides machines, human workers are usually involved in DR, increasing the modeling complexity and economic impact.

Kim et al. [4] mentioned that the implementation of industrial DR by production scheduling should consider the potentially increasing labor cost. The intrinsic trade-off relation between the energy and labor costs was statistically pointed out in chapter 3. An adaptive multi-objective memetic algorithm (AMOMA) was further proposed for fast approximation of this trade-off relation without losing the diversity in an energy- and labor-aware single-machine production scheduling problem. Despite these research efforts, there is still little investigation on integrated energy- and labor-aware (flexible) job shop scheduling, compared to the extensive recent studies on energy-aware production scheduling.

A flexible job-shop is one of the most complex shop floor configurations for production [5]. In a flexible job shop, an operation can be processed by multiple machines, such that two fundamental decision makings need to be simultaneously

done: (1) the specific machine that will process an operation of a job, (2) the operation sequence on each machine while considering the predefined operation sequence of each job. Therefore, jobs have different routes in a flexible job shop, which is a generalization of the flow shop (where all jobs have the same route, though some jobs may bypass a stage in a more complex setting). The ongoing needs for low-volume and high-variety production call for flexible manufacturing systems. This implies that the flexible job shop will get even more prevalent in factories of the future.

A flexible job shop scheduling problem (FJSSP) can be further classified into two categories: (1) full FJSSP, and (2) partial FJSSP. In a full FJSSP, each operation can be performed by all machines; while in a partial FJSSP, each operation has its own set of capable machines, not necessarily all machines. Although both types of problems are non-deterministic polynomial (NP)-complete, a partial FJSSP is more complex than a full FJSSP [6]. The complexity of a partial FJSSP will even increase by considering other practical factors. For instance, if job recirculation is enabled, a job may return to the machine that has processed it in an early stage. This is usually the case in semiconductor manufacturing, where the wafer is fabricated one layer after another [5].

Once a FJSSP is formulated, it has to be solved by an optimization algorithm in a short time, as production scheduling is rather time-sensitive compared to the long-term production planning. An evolutionary algorithm (EA) intrinsically complies with this requirement, since it aims to fast obtain a high-quality optimization solution or solution set, without having to guarantee the exact optimum. On the one hand, an integrated energy- and labor-aware FJSSP calls for multiple objectives to be simultaneously optimized. This is not only because the integration of energy and labor awareness triggers more production objectives, but also due to the fact that production objectives often vary on the shop floor depending on the preference of a specific decision maker or the compromised preferences of multiple decision makers [7]. Therefore, the number of optimization objectives easily exceed three, which is the prevalent maximal number of objectives in the existing multi-objective FJSSP research.

On the other hand, many-objective optimization has become an active research topic in multi-objective evolutionary algorithms (MOEAs). This is due to the new challenges faced by EAs [8]: (1) difficulties in the search for Pareto optimal solutions, (2) difficulties in the approximation of the entire Pareto front, (3) difficulties in the presentation of obtained solutions, (4) difficulties in the choice of a single final solution, and (5) difficulties in the evaluation of search algorithms. Therefore, MaOPs have been specifically developed for the multi-objective optimization problems (MOPs) with more than three objectives [9]. While extensive research on many-objective optimization is focused on designing a generic EA that can be widely used without any deep domain knowledge, there is still little research on

applying or tailoring an EA for a real MaOP, e.g., integrated energy- and labor-aware FJSSP.

This chapter fills these gaps in the following aspects. (1) Compared to the existing research on FJSSP, energy and labor awareness are additionally modeled and integrated; job recirculation, operation sequence-dependent machine setup times, and partial flexible job shop are considered in the proposed model, named EL-FJSSP. Although the EL-FJSSP becomes more complex, it is also more practical and flexible for industrial applications, while being able to help factories to additionally reduce the energy cost and the labor cost under dynamic electricity pricing. (2) A DES framework is used to build a digital twin of the energy- and labor-aware flexible job shop. This not only enables flexible modeling of a complex production problem on the shop floor, but also facilitates fast yet accurate calculation of all the optimization objectives. (3) For the first time, the NSGA-III is tailored for many-objective optimization of this proposed problem, as a representative example to apply a many-objective EA for production scheduling. (4) Through numerical experiments under different dynamic electricity pricing schemes, the underlying relations among the important production objectives are quantitatively revealed, i.e., makespan, total energy cost, total labor cost, maximal workload, and total workload. The effectiveness and efficiency of applying a NSGA-III in solving a many-objective EL-FJSSP are demonstrated by benchmarking with the NSGA-II.

The remainder of this chapter is organized as follows. Section 4.2 gives the literature review and reveals the observed gaps in current research on energy-aware (flexible) job shop scheduling. Section 4.3 describes the proposed EL-FJSSP, with a focus on how to integrate energy and labor awareness in a conventional FJSSP. Section 4.4 introduces the tailored NSGA-III, including scheduling solution encoding and decoding, crossover, mutation, and solution evaluation by DES. Section 4.5 presents the experiments on EL-FJSSP and NSGA-III, with a focus on quantitative characterization of the relation among many objectives in this EL-FJSSP and demonstration of the effectiveness and efficiency of NSGA-III in many-objective optimization.

4.2 Literature Review

Table 4.1 analyzes the recent research on energy-aware (flexible) job shop scheduling in five aspects: energy model, labor model, problem size, number of objectives, and optimization method. Generally, the gaps lie in little investigation of dynamic energy prices and labor in the scheduling model, a small problem size, and a small number of objectives.

Regarding the energy model, only a few studies use complete power states (including startup and shutdown) to mimic the dynamic energy consumption behavior

Table 4.1: Analysis of recent literature on energy-aware job shop scheduling methods

Literature	Energy model		Labor model		Problem size	Number of objectives	Optimization method
	Power state	Electricity price	Shift	Personnel			
[10]	No	No	No	No	Small	Two	Metaheuristic
[11]	No	Yes	No	No	Small	One	MIP + CP [a]
[12]	Partial	No	No	No	Small	Three	Metaheuristic
[13]	Partial	No	No	No	Medium	One	Metaheuristic
[14]	No	No	No	No	Small	Two	Metaheuristic
[15]	Partial	No	No	No	Small	One	LP [a]
[16]	Partial	Yes	No	No	Small	Three	Metaheuristic
[17]	Yes	No	No	No	Small	Two	Metaheuristic
[18]	Yes	No	No	No	Small	Two	Metaheuristic
[19]	Partial	No	No	No	Medium	One	Metaheuristic
[20]	Yes	No	No	No	Small	Two	Metaheuristic
[21]	No	No	No	No	Medium	Two	Metaheuristic
[22]	Partial	No	No	No	Small	Three, five	Metaheuristic
[23]	Partial	No	No	No	Small	Two	Metaheuristic
[24]	No	No	No	No	Small	One	Heuristic
[4]	Partial	Yes	Yes	Yes	Small	One	Metaheuristic
[25]	No	No	No	No	Small	Three	Metaheuristic
[26]	No	No	No	No	Small	Two	Metaheuristic
[27]	No	No	No	No	Small	Three	Metaheuristic
[28]	Yes	No	No	No	Small	One	Metaheuristic + AI [a]
[29]	No	No	No	No	Small	Three	Game theory

[a]MIP: mixed-integer programming, CP: constraint programming, LP: linear programming, AI: artificial intelligence

of machines. May et al. [17] propose a set of power states, which are mapped to machine operation states. These power states include off, idle, standby, setup, and working, while ramp-up and ramp-down are modeled as the transitions between off and idle. In [18] four states are considered: process, idle, standby, and switch on/off. Compared to the idle state, the standby state is "not ready" for operations, because required components stay powered off, e.g., auxiliary systems. Liu et al. [20] consider a power input model for a machine and turn-off/on states. This power input model encompasses idle, switch, and cutting states, where the switch state is a transition to runtime mode. Besides processing and standby states, startup and shutdown states are considered in [28]. The duration of the latter two states enables the switching off/on decision: a machine is switched off if the duration between two adjacent operations is longer than the breakeven duration (i.e., energy consumption of shutdown and startup divided by standby power).

Although studies with partial power states generally consider idle and processing states, they ignore some basic power states, such as shutdown [12, 13] or startup and shutdown states [19]. The investigations that neglect power states often classify the total machine energy consumption into unload and cutting energy [10, 14]. The unload power corresponds to activities that prepare processing, such as loading, unloading, positioning, and changing cutting tools. The cutting power corresponds to actual cutting operations. However, as highlighted in [14], this energy modeling method depends on specific machining processes, hindering its general application.

As presented in Table 4.1, even fewer studies consider dynamic energy prices, despite the economic impact of converting the energy consumption to the energy cost [30]. For production during a Rolling Blackout policy [16], the electricity price is more expensive, due to private power generation which fills the shortage of public power supply. In time-of-use pricing (ToUP) and critical-peaking pricing (CPP), the electricity price in a day may have several levels. The difference between levels is much larger in CPP than in ToUP [11]. Besides ToUP, real-time pricing (RTP) is considered in [4], where the electricity price varies more often than ToUP, e.g., every hour in the next day [30]. A common attempt of these studies is to schedule production loads such that the incurred energy cost is minimized under volatile electricity prices, while satisfying the defined production constraints.

Table 4.1 also reveals a lack of relevant studies on the labor aspect, although machines commonly interact with human workers in practice and an optimized match between these two types of resources is required for economical production. Consideration of human workers requires the fragmentation of a continuous scheduling horizon into discrete labor shifts, which may be further separated by periods of forbidden production, e.g., weekends and holidays. Therefore, this needs new modeling and effective optimization with more constraints. Garcia-Santiago

et al. [13] mentioned that a human operator is required to change the ancillary part of a machine upon a change of product type, but they did not investigate how to economically match machine and human resources. The only relevant literature that considers the labor aspect is [4]. Nevertheless, simple scenario comparisons were performed on whether increasing production and operators in the day or in the night is economical. No hint was given on how to integrate labor in an energy-aware (flexible) job shop scheduling model [4].

Additionally, Table 4.1 indicates the prevalent small problem size in existing research. The size of a JSSP or a FJSSP is characterized by the number of jobs, machines, and time slots ($|J| \times |M| \times |T|$). The number of operations in all jobs also influences the problem size. However, it is not considered in this characterization, as one job contains at least one operation such that this characterization remains reasonable even if the number of operations is taken into account. Thereby, in this chapter, the problem size is small if any of $|J|, |M|$ and $|T|$ is smaller than 20. It is large if these three variables simultaneously surpass 100. Otherwise, it is medium. Following this classification, most studies focused on small-sized problems, though some performed experiments with quite a few different instances [20, 23, 24]. The medium problem size is observed as $117 \times 24 \times 166$ in [13], $20 \times 20 \times |T|$ in [19], and $200 \times 20 \times |T|$ in [21]. Although T in the latter two studies were not explicitly mentioned, the makespan values in their experiments would imply that T is at least no smaller than 20.

Analogous to the problem size, the number of objectives also indicates the complexity of a scheduling problem. As various performance metrics usually exist on the shop floor depending on diverse practical factors, e.g., a specific customer order, preference of a specific plant manager, and real-time machine conditions, the production decision making would be enhanced by considering more than three objectives in a scheduling algorithm. This in return exerts a higher requirement on multi-objective optimization. Table 4.1 clearly demonstrates that all the literature except [22] is limited to three optimization objectives. By exception, up to five objectives were involved in [22]. However, the many-objective optimization was not investigated in this study. A common objective in these investigations is makespan, as it not only directly measures the time to complete a set of jobs, but also is linked to other important production metrics, e.g., earliness/tardiness of finished jobs and workload. The energy-related objectives include minimization of total energy consumption to process a set of jobs [10, 13, 17, 22, 23, 28], minimization of non-processing energy [16, 17], minimization of peak power consumption [18], and minimization of energy cost [11, 16].

In terms of optimization method to solve the JSSP or FJSSP, a majority of studies used various metaheuristics (Table 4.1). A genetic algorithm (GA) [10, 12, 21, 27] as well as its variants NSGA-II [16, 20] and MA [23, 26] are the most prevalent among these metaheuristics. Besides, Xu et al. [22] used a bee algorithm,

Garcia-Santiago et al. [13] utilized a harmony search algorithm, and He et al. [14] employed a nested partitions algorithm. Furthermore, two metaheuristics are often hybridized for enhanced search capability in the solution space. For instance, a GA was hybridized with a simulated annealing algorithm in [19, 25], an artificial immune algorithm was hybridized with a simulated annealing in [18], and NSGA-II was hybridized with SPEA-II in [17]. One important reason for this prevalent usage of metaheuristics is that production scheduling on the shop floor intrinsically requires fast yet high-quality decision making, which is also highlighted in [13]. This also explains why optimization by a standard solver [11, 15] or a heuristic [24] is not extensively used, of which the computation time and tractability are very sensitive to additional constraints and an increasing problem size. An interesting observation is that Zhang et al. [28] employed artificial intelligence to enhance an evolutionary process of a gene expression programming algorithm. Diversified rule mining operations with self-study was designed to enhance the solution quality. Unsupervised learning was utilized to guide the evolution direction by leveraging the global best and current worst. Overall, the identified research trend on solution methods for the JSSP or FJSSP is to leverage artificial intelligence (AI), or more specifically computational intelligence (CI), to achieve fast and high-quality decision making.

4.3 Problem Modeling

The proposed problem is to perform energy- and labor-aware flexible job shop scheduling under dynamic electricity prices, named EL-FJSSP. All jobs have a common release time and must be finished before a common due time (additional constraints may be added to this assumption to get adapted to a specific production case, e.g., jobs with different release time, due time, and priority). Labor shifts are considered, while a job or a changeover may be split into multiple subparts due to production-forbidden periods, e.g., weekends and holidays. Job recirculation is allowed, i.e., a job may revisit the same machine even if it has once been processed by this machine. Five objectives need to be simultaneously optimized: makespan (C_{max}), total energy cost (TEC), total labor cost (TLC), maximal machine workload (MWL), and total machine workload (TWL), as defined in Equation (4.1). The following subsections will describe how to calculate these objectives while modeling this production problem. Table 4.2 summarizes the symbols that are used in the formulation of the EL-FJSSP.

$$\min(C_{max}, TEC, TLC, MWL, TWL) \tag{4.1}$$

Table 4.2: Nomenclature of the proposed energy- and labor-aware production scheduling model (italic: output variables or variables to be determined by the model; non-italic: input variables; operation (i, j, k, l) means the *l*-th assigned operation on machine *i* while this operation is of type *k* and in job *j*)

Parameter	Notation
C_j	Makespan of job *j*
C_{max}	Makespan of the entire production
d_{idle}^{ijkl}	Time duration to remain the idle mode before the operation (i, j, k, l)
d_{off}^{ijkl}	Time duration to remain the idle mode before the operation (i, j, k, l)
D	Duration of the electricity pricing slot
D_{poff}^{i}	Duration to power off the *i*-th machine
DO_{ijkl}	Duration of operation (i, j, k, l)
D_s	Duration of machine power state *s*
DT	Common due time of all jobs
ECC_i	Electricity cost for performing changeovers on the *i*-th machine
ECI_i	Electricity cost for performing idling on the *i*-th machine
ECO_i	Electricity cost for processing operations on the *i*-th machine
EP_{ts}	Electricity price on the *ts*-th time slot
ETC_{ijkl}^{n}	End time in δt of the *n*-th subpart of the *i*-th machine changeover before operation (i, j, k, l)
ETI_{ijkl}	End time in δt of the machine idling before the operation (i, j, k, l)
ETO_{ijkl}^{n}	End time in δt of the *n*-th subpart of the operation (i, j, k, l)
$ETSC_{ijkl}^{n}$	End time in electricity pricing slots of the *n*-th subpart of the machine changeover before the operation (i, j, k, l)
$ETSI_{ijkl}$	End time in electricity pricing slots of the machine idling before the operation (i, j, k, l)
$ETSO_{ijkl}^{n}$	End time in electricity pricing slots of the *n*-th subpart of the operation (i, j, k, l)
$f(\cdot, \cdot)$	Mapping from a pair of operations on a machine to the machine setup time
J	Set of jobs to be processed
K	Set of operation types
M	Set of machines
MWL	Maximal workload
NSC_{ijkl}	Number of subparts of the changeover before the operation (i, j, k, l)
$N_{sh_i}^{pt}$	Number of human workers of personnel type *pt* and in the shift sh_i
NSO_{ijkl}	Number of subparts of the operation (i, j, k, l)
P_{p}	Power consumption of the machine power state *Production*
P_s^t	Power consumption of the machine power state *s* at time *t* in δt
PT	Set of personnel types (e.g., operator and quality checker)
RT	Common release time of all jobs to be processed
pt	Type of personnel required in shift *sh*
S_{c}	Sequenced power states for performing a machine changeover

Table 4.2: Continuation of Table 4.2 on the previous page

Parameter	Notation
SH	Set of labor shifts in the scheduling time span without considering the match for production loads
sh_i	Assigned labor shift for the *i*-th machine
S_I	Possible sequenced machine power states for idling
S_I^{ijkl}	S_I that follows the operation (i, j, k, l)
S_{idle}	Sequenced power states for a machine to switch to and recover from the idle mode
S_{off}	Sequenced power states for a machine to switch to and recover from the off mode
ST	Set of labor shift types on a weekday or a day on weekends
ST_i	Operation sequence-dependent setup time on the *i*-th machine
STC^n_{ijkl}	Start time in δt of the *n*-th subpart of the changeover before the operation (i, j, k, l)
STI_{ijkl}	Start time in δt of the machine idling before the operation (i, j, k, l)
STO^n_{ijkl}	Start time in δt of the *n*-th subpart of operation (i, j, k, l)
$STSC^n_{ijkl}$	Start time in electricity pricing slots of the *n*-th subpart of the changeover before the operation (i, j, k, l)
$STSI_{ijkl}$	Start time in electricity pricing slots of the machine idling before the operation (i, j, k, l)
$STSO^n_{ijkl}$	Start time in electricity pricing slots of the *n*-th subpart of the operation (i, j, k, l)
t	Absolute time or clock time
TEC	Total energy cost for processing the jobs
TLC	Total labor cost for processing the jobs
ts	Time in electricity pricing slots
TWL	Total workload for processing the jobs
W_i	Work load of the i-th machine
$W^{pt}_{sh_i}$	Labor wage of the personnel type *pt* in the shift sh_i
α_{ijkl}	Boolean machine idling mode indicator before the operation (i, j, k, l)
β_{ts}	Boolean electricity pricing time slot indicator
δsh	Duration of one labor shift
δt	Basic scheduling time slot
λ_p	Boolean production-forbidden period indicator
π_i	Assigned operation sequence on the *i*-th machine
$\theta^{pt}_{sh_i}$	Boolean indicator for the personnel type *pt* in the shift *sh*

4.3.1 Objectives

C_{max} is defined by Equation (4.2) as the time when the last job is completed and leaves the production system:

$$C_{max} = \max_{j \in \mathrm{J}}(C_j) \tag{4.2}$$

where C_j is the completion time of job j. C_{max} is closely related to the throughput

of a production system, while throughput maximization is of the utmost importance and managers are often measured how well they do so. Therefore, minimizing C_{max} of a production system with a finite number of jobs tends to maximize the throughput [5].

TEC is composed of the energy cost for operation processing of all the jobs ($\sum_{i\in \mathrm{M}} ECO_i$), machine changeover/setup ($\sum_{i\in \mathrm{M}} ECC_i$) and idling between adjacent operations ($\sum_{i\in \mathrm{M}} ECI_i$). While Equation (4.3) gives a general description, the detailed methods of calculating each energy cost part will be introduced in Section 4.3.2.

$$TEC = \sum_{i\in \mathrm{M}} (ECO_i + ECC_i + ECI_i) \tag{4.3}$$

TLC depends on the assigned labor shifts for all machines and type of human workers in each shift. It comprises the labor cost of all human workers who are involved in these shifts. It is calculated by Equation (4.4). The binary variable $\theta^{pt}_{sh_i}$ indicates whether a personnel type (pt) is included in a shift for machine *i* (sh_i) according to the sequenced power states assigned to this shift. A pt is identified by the skill that is linked to the worker. $W^{pt}_{sh_i}$ is the labor wage of the personnel type (pt) in the labor shift (sh_i) for machine i. $N^{pt}_{sh_i}$ indicates the number of workers of personnel type pt during the shift for machine *i* (sh_i).

$$TLC = \sum_{i\in \mathrm{M}} \sum_{sh_i \in \mathrm{SH}} \sum_{pt\in \mathrm{PT}} \left(\theta^{pt}_{sh_i} \cdot W^{pt}_{sh_i} \cdot N^{pt}_{sh_i}\right) \tag{4.4}$$

MWL and TWL are two workload-related objectives for a (flexible) job shop [12, 31]. MWL indicates the maximum working time spent on any machine, as formulated in Equation (4.5). As described in Equation (4.6), TWL measures the total workload of machines, where W_i is the work load of machine *i*. It represents the total working time of all machines.

$$MWL = \max_{i\in \mathrm{M}} (W_i) \tag{4.5}$$

$$TWL = \sum_{i\in \mathrm{M}} W_i \tag{4.6}$$

4.3.2 Total Energy Cost

The energy cost for all operations on machine *i* (ECO_i) is accumulated according to the operation sequence of machine i (π_i), where π_i contains a sequence of indexes (i, j, k, l) of these operations. As defined in Equation (4.7), the energy consumption for processing operation (i, j, k, l) is mapped to the corresponding electricity pricing slots via the time slot indicator β_{ts}, such that the energy cost for

processing operation (i, j, k, l) is accumulated over these pricing time slots. The option for a period of forbidden production (e.g., weekend and holiday) is integrated due to the consideration of labor. If the production is prohibited during a period ($\lambda_p = 1$), an operation may be split by this period into multiple subparts with machine power-off&on in-between adjacent subparts. A set of sequential power states ($\mathrm{S_{off}}$) is involved in powering off a machine and powering it on later. Each power state (s) is characterized by an averaged power level (P_s) and an averaged time duration (T_s), i.e., power profile. In Equation (4.8) and Equation (4.9), the current time is mapped to the corresponding electricity pricing slot.

$$\begin{aligned} ECO_i = \sum_{(i,j,k,l)\in\pi_i} \sum_{ts=STSO^n_{ijkl}}^{ETSO^n_{ijkl}} \mathrm{EP_{ts}} \cdot \Bigg(\sum_{n=1}^{NSO_{ijkl}} \sum_{t=STO^n_{ijkl}}^{ETO^n_{ijkl}} \beta_{ts} \cdot \mathrm{P_p} \cdot t \\ + \lambda_p \cdot \sum_{n=2}^{NSO_{ijkl}} \sum_{t=STO^n_{ijkl}}^{ETO^n_{ijkl}} \sum_{s\in\mathrm{S_{off}}} \beta_{ts} \cdot P_s^t \cdot t \Bigg) \end{aligned} \tag{4.7}$$

$$\beta_{ts} = \begin{cases} 1, \text{ if } t \in [ts \cdot \mathrm{D}, (ts+1) \cdot \mathrm{D}) \\ 0, \text{ otherwise} \end{cases} \tag{4.8}$$

$$ts = \lfloor (t - \mathrm{RT})/\mathrm{D} \rfloor, t \in [\mathrm{RT}, \mathrm{RT} + \delta t, ..., \mathrm{DT} - \delta t, \mathrm{DT}] \tag{4.9}$$

The energy cost for all changeovers on machine i (ECC_i) is described in Equation (4.10). Analogous to Equation (4.7), it is accumulated over all changeovers on machine i by mapping the energy consumption of each changeover to the energy cost under dynamic electricity prices. The split of one changeover into multiple subparts due to a period of forbidden production (e.g., prohibited production on weekends) is also considered, such that the additional energy cost for powering off&on machines before and after this special period is calculated. A set of sequential power states ($\mathrm{S_c}$) may be involved in a changeover. Note that a machine does not have to perform a changeover if the changeover duration is zero, i.e., $STC_{ijkl} = ETC_{ijkl}$; otherwise, this machine performs a changeover right before the start of the upcoming operation (i, j, k, l).

$$\begin{aligned} ECC_i = \sum_{(i,j,k,l)\in\pi_i} \sum_{STSC_{ijkl}}^{ETSC_{ijkl}} \mathrm{EP_{ts}} \cdot \Bigg(\sum_{n=1}^{NSC_{ijkl}} \sum_{t=STC_{ijkl}}^{ETC_{ijkl}} \sum_{s\in\mathrm{S_c}} \beta_{ts} \cdot P_s^t \cdot t \\ + \lambda_p \cdot \sum_{n=2}^{NSC_{ijkl}} \sum_{t=STC_{ijkl}}^{ETC_{ijkl}} \sum_{s\in\mathrm{S_{off}}} \beta_{ts} \cdot P_s^t \cdot t \Bigg) \end{aligned} \tag{4.10}$$

The energy cost for the idling of machine i (ECI_i) is accumulated over all the operations and over time in Equation (4.11), which is similar to Equation (4.7)

and Equation (4.10). The machine idling encompasses three cases, as described in Equations (4.12) - (4.14). (1) $\alpha_{ijkl} = 0$: no idling mode follows operation (i, j, k, l), i.e., the set of idling power state is empty ($S_I = \oslash$). Consequently, a potential changeover immediately starts upon the end of operation (i, j, k, l), which is followed by the next operation $(i, j', k', l + 1)$. (2) $\alpha_{ijkl} = 1$: an idle mode is set between operations (i, j, k, l) and $(i, j', k', l + 1)$, i.e., $S_I = SI$. (3) $\alpha_{ijkl} = 2$: the off mode with an off duration (d_{off}) is set between operations (i, j, k, l) and $(i, j', k', l + 1)$, i.e., $S_I = SO$.

$$ECI_i = \sum_{(i,j,k,l)\in\pi_i} \sum_{ts=STSI_{ijkl}}^{ETSI_{ijkl}} \mathrm{EP_{ts}} \cdot \left(\beta_{ts} \cdot \sum_{t=STI_{ijkl}}^{ETI_{ijkl}} \sum_{s\in S_I} P_s^t \cdot t \right) \tag{4.11}$$

$$S_I = \begin{cases} \oslash, \text{ if } \alpha_{ijkl} = 0 \\ \mathrm{S_{idle}}, \text{ if } \alpha_{ijkl} = 1 \\ \mathrm{S_{off}}, \text{ if } \alpha_{ijkl} = 2 \end{cases} \tag{4.12}$$

$$STI_{ijkl} = ETO_{ijkl} \begin{cases} = ETI_{ijkl}, \text{ if } \alpha_{ijkl} = 0 \\ \neq ETI_{ijkl}, \text{ otherwise} \end{cases} \tag{4.13}$$

$$ETI_{ijkl} = \begin{cases} ETO_{ijkl} = STI_{ijkl}, \text{ if } \alpha_{ijkl} = 0 \\ STI_{ijkl} + \sum_{s\in \mathrm{S_{idle}}} D_s + d_{idle}^{ijkl}, \text{ if } \alpha_{ijkl} = 1 \\ STI_{ijkl} + \sum_{s\in \mathrm{S_{off}}} D_s + d_{off}^{ijkl}, \text{ if } \alpha_{ijkl} = 2 \end{cases} \tag{4.14}$$

4.3.3 Total Labor Cost

The TLC is a sum of the labor cost of all human workers in all involved shifts over all machines, as formulated in Equation (4.4). As defined in Equation (4.15), within one shift (sh_i), once a personnel type (pt) is required by an involved power state (s), this pt will be included in this sh_i ($\theta_{sh_i}^{pt} = 1$). Otherwise, the binary personnel occupation indicator $\theta_{sh_i}^{pt}$ is zero. In other words, once a person is needed some time in a shift, this person will work and be paid for the whole shift, regardless of the actual workload.

$$\theta_{sh_i}^{pt} = \begin{cases} 1, \text{ if } pt \text{ is required in } sh_i \\ 0, \text{ otherwise} \end{cases} \tag{4.15}$$

Workers, machine production, and machine power consumption are coupled by the link between the skills required by different machine operations. In this

Table 4.3: Mapping between generic machine power states and the required type of personnel

Current state	Trigger event	Destination power state	Next power state	Type of personnel
Off	Power-on	*Ready*	*Startup*	None
Startup	Automatic	*Ready*	*Startup*	Operator
Ready	Idling Produce	*Ready* *Production*	*Ready* *Production*	Operator
Production	Automatic (when current job is completed)	*Ready*	*Ready*	Operator, quality checker (for jobs whose last operation is performed on this machine)
Shutdown	Automatic	*Off*	*Off*	Operator

manner, these three decision variables are fully integrated, compared to the existing literature which performs independent decision making on these decision variables. Table 4.3 presents the mapping between generic power states and the type of personnel, while Figure 4.5 presents the generic power state-based energy model that accommodates these three decision variables. The generic energy model and personnel type in Table 4.3 and Figure 4.5 can be extended to specific cases.

The duration of a shift is defined by Equation (4.16), i.e., 24 hours divided by the number of shifts in a day. |ST| is the set of shifts or shift types in a day which may be a weekday and a weekend day. If weekend production is allowed, the labor wage in the same shift is higher on a weekend day than on a weekday. The labor wage also varies in a day, increasing in the night shift than in the other shifts.

$$\delta sh = 24/|\mathrm{ST}| \tag{4.16}$$

4.3.4 Operation, Job, and Changeover

A job *j* has a fixed sequence of operations. To complete a job, each of these operations must be performed by one qualified machine. Operations of distinct jobs may follow an arbitrary sequence on a machine but with a common release time (RT) and due time (DT). Recirculation is allowed. From a job perspective, this means that a job may contain multiple identical operations and return to a previous machine it passed through. Due to the constraint of production-forbidden periods introduced by the labor model, an operation (i, j, k, l) contains one or more sub-durations, as indicated in Equation (4.17).

$$\mathrm{DO_{ijkl}} = \sum_{n=1}^{NSO_{ijkl}} \left(ETO_{ijkl}^{n} - STO_{ijkl}^{n}\right), i \in \mathrm{M}, j \in \mathrm{J}, k \in \mathrm{K}, l \in \pi_i \tag{4.17}$$

The last operation on a machine must be completed before DT, considering the duration to power off this machine, as described in Equation (4.18).

$$ETO_{ijkl} + \mathrm{D}^{i}_{\text{poff}} \leq \mathrm{DT}, i \in \mathrm{M}, j \in \mathrm{J}, k \in \mathrm{K}, l = \pi_i(|\pi_i|) \tag{4.18}$$

As defined by Equation (4.19), a machine changeover is required between adjacent operations of different types. It starts right before the upcoming operation (i, j, k, l), such that the end time of the last subpart of such a changeover ($ETC^{NSC_{ijkl}}_{ijkl}$) is equal to the start time of the first subpart of this upcoming operation ($STO^{1}_{ijk'l}$).

$$ETC^{NSC_{ijkl}}_{ijkl} = STO^{1}_{ijk'l}, i \in \mathrm{M}, j \in \mathrm{J}, k, k' \in \mathrm{K}, l \in \pi_i(2 : |\pi_i|) \tag{4.19}$$

Analogously, the duration of a changeover is the sum of one or multiple subdurations, as described in Equation (4.20).

$$DC_{ijkl} = \sum_{n=1}^{NSC_{ijkl}} \left(ETC^{n}_{ijkl} - STC^{n}_{ijkl}\right), i \in \mathrm{M}, j \in \mathrm{J}, k \in \mathrm{K}, l \in \pi_i(2 : |\pi_i|) \tag{4.20}$$

4.3.5 Machine

A machine is assumed to have sufficient material supply and have no breakdown. It cannot simultaneously perform multiple operations and does not allow preemption (Equation (4.21)). An idle mode is only applicable to a period that can accommodate it (Equation (4.22)). If the production is prohibited during a period (e.g., weekend and holiday), the machine must be powered off before this period, as indicated in Equation (4.23). The changeover/setup time depends on the sequence of adjacent operations on a machine, as described in Equation (4.24).

$$ETO^{NSC_{ijkl}}_{ijkl} < STO^{1}_{ij'k'(l+1)}, i \in \mathrm{M}, j, j' \in \mathrm{J}, k, k' \in \mathrm{K}, l \in \pi_i(1 : |\pi_i| - 1) \tag{4.21}$$

$$\sum_{s \in SI_{ij}} D_s \leq STC^{1}_{ijk(l+1)} - ETO^{NSC_{ijkl}}_{ijkl}, i \in \mathrm{M}, j \in \mathrm{J}, k \in \mathrm{K}, l \in \pi_i(1 : |\pi_i| - 1) \tag{4.22}$$

$$P^{t}_{s} = 0, \text{ if } (\lambda = 1) \ \& \ (\forall t \in \text{weekends \& holidays}) \tag{4.23}$$

$$ST_i = f((i, j, k, l), (i, j', k', l + 1)), i \in \mathrm{M}, j, j' \in \mathrm{J}, k, k' \in \mathrm{K}, l \in \pi_i(1 : |\pi_i| - 1) \tag{4.24}$$

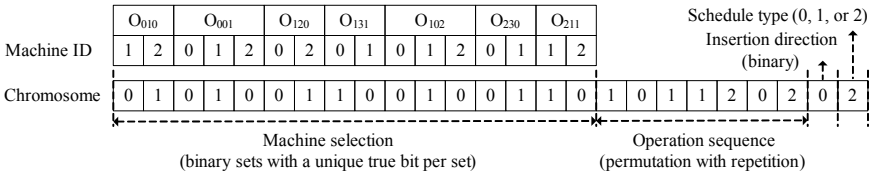


Figure 4.1: Heterogeneous chromosome representation. For an operation O_{jkl}, j represents a job, k represents a type of operation, and l represents the sequence of operation k in job j.

4.4 Solution Algorithm: Tailored NSGA-III

This section is the first attempt to tailor the NSGA-III [1] in solving a many-objective FJSSP. It specifically tackles the following fourfold issues in this attempt. (1) How to represent an EL-FJSSP scheduling solution in an evolutionary search? This solution representation not only aims for a conventional flexible job shop, but also should allow for job recirculation on the same machine in the problem domain and promote diversity in a population during an evolutionary search. (2) How to effectively decode a potentially complex chromosome to a corresponding scheduling solution? (3) How to define crossover and mutation operators? (4) How to efficiently evaluate a scheduling solution given the many-objective and highly-constrained FJSSP?

4.4.1 Scheduling Solution Encoding

A canonical evolutionary algorithm usually produces some infeasible solutions after recombination by crossover and mutation. Consequently, a repair mechanism is often required for remedy, although this slows down the evolutionary search. In this chapter, the solution is encoded by four heterogeneous chromosome segments, such that these repair efforts are removed while guaranteeing the feasibility of all solutions.

In the example of Figure 4.1, job0 has two operations (O_{010}, O_{001}), job1 has three operations (O_{120}, O_{131}, O_{102}), and job2 has two operations (O_{230}, O_{211}). These operations are aligned in a lexicographical order of the job index and the operation sequence of each job. The first chromosome segment in Figure 4.1 refers to machine selection and is encoded as binary sets. A true bit (i.e., one) means that the machine is selected, and vice versa for a false bit (i.e., zero). Therefore, the machines selected for (O_{010}, O_{001}, O_{131}, O_{102}, O_{230}, O_{012}, O_{211}) are (2, 0, 2, 0, 1, 1, 2). The second segment represents the sequence in which operations of all jobs are placed on the time horizon of the corresponding machine. It is encoded as permutation with repetition [32], which naturally maintains the relative order among operations of a job and further facilitates the definition of crossover

(Section 4.4.3) and mutation (Section 4.4.4). In the example of Figure 4.1, the operation sequence is (O_{120}, O_{010}, O_{131}, O_{102}, O_{230}, O_{001}, O_{211}). The third segment indicates the operation insertion direction. It is encoded as a binary bit. A true bit (i.e., one) indicates the forward insertion of operations on the time horizon of a schedule (Section 4.4.2.1), while a false bit (i.e., zero) implies the backward insertion of operations (Section 4.4.2.2).

The fourth chromosome segment in Figure 4.1 indicates the schedule type, which is a decimal number with a range from zero to two (0: original schedule, 1: semi-active schedule, 2: active schedule). Based on the primary classification of forward schedules in [5], this chapter further classifies feasible non-preemptive forward and backward schedules in the following three categories: (1) original schedule (operations are placed on the time horizons of machines one after another by strictly following the assigned forward or backward operation sequence), (2) semi-active schedule (if no operation can be completed earlier or later for a forward or backward schedule, respectively, without changing the order of processing on any one of the machines), (3) active schedule (if no operation can start earlier without delaying at least one other operation in a forward schedule, or if no operation can end later without advancing at least one other operation in a backward schedule). Therefore, the schedule type determines the location of time horizon of the corresponding machine where an operation is placed and even the operation sequence on a machine for an active schedule. Despite having a longer makespan compared to active schedules, original and semi-active schedules are equally important. In the context of industrial DR, more free durations in the latter two types of schedules allow for machine powering off and idling during peak electricity pricing periods. The combination of insertion direction and schedule type leads to overall six schedule types. Consequently, this diversifies the population to avoid premature convergence.

4.4.2 Scheduling Solution Decoding

Figure 4.2 demonstrates how to decode a scheduling solution from the heterogeneous chromosome illustrated in Figure 4.1. Operations are assigned one after another to corresponding machines in the indicated direction and following the assigned sequence, while the final operation sequence on a machine may be altered in an active schedule. In an original schedule, an operation is simply inserted at the end (forward decoding in Figure 4.2a) or the start (backward decoding in Figure 4.2b) of its precedent, which is indicated by the chromosome. In a semi-active schedule, an operation starts as early as possible (forward decoding in Figure 4.2c) or end as late as possible (backward decoding in Figure 4.2d) without changing the assigned operation sequence. In an active schedule, the operation sequence can be broken, and no left-shift can be further performed without right-shifting another

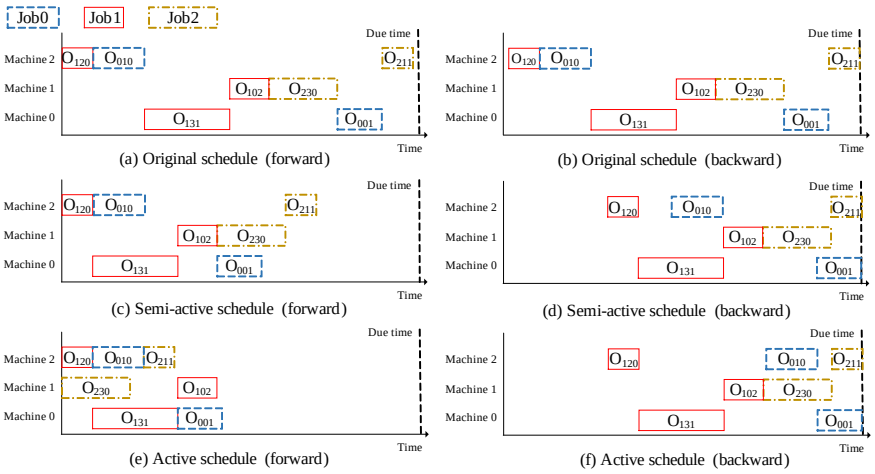


Figure 4.2: Example of forward and backward decoding an original, semi-active and active schedule from a chromosome with the same machine selection and operation sequence segments illustrated in Figure 4.1.

operation (forward decoding in Figure 4.2e), or no right-shift can be further performed without left-shifting another operation (backward decoding in Figure 4.2f).

Figure 4.2 does not illustrate job or changeover split for brevity. However, when actually inserting an operation, a weekend and a holiday should be excluded from time allocation if weekend production is not allowed. Consequently, a job or a changeover may have multiple start and end times, indicating that it is split by at least one weekend. For the concern of job or changeover split, the following two definitions are provided.

Definition 1: *first start time* is the start time of the first subpart of a job or a changeover, considering the possibility that this job or this changeover may be split by one or more prohibited periods which can be planned in advance, e.g., a weekend without production and a maintenance period for a machine.

Definition 2: *last end time* is the end time of the last subpart of a job or a changeover, considering the same job split possibility in Definition 1.

If a job or a changeover is not split into multiple subparts, these two definitions still hold by considering that it has only one subpart. Moreover, the number and type of personnel in each labor shift should be considered when inserting an operation, to enable labor awareness in the schedule.

Note that although the assigned operation sequence may be altered in decoding an active schedule, it rather refers to the operation sequence on a machine. The operation sequence of a job remains fixed, as a constant input for a scheduling algorithm, to ensure that a job is processed in the designed technological sequence. Analogously, even when backward inserting operations in a schedule, the opera-

Algorithm 2 Forward decoding an active flexible job shop production schedule

Input: *sequencedOperations* (with assigned machines)
Output: a production schedule π which ends as early as possible

1: **for** *index* $\leftarrow$ 1 : size(*sequencedOperations*) **do**
2: $operation \leftarrow sequencedOperations(index)$
3: **if** $index$ == 0 **then**
4: $lowerBoundInAJob \leftarrow startTimeSchedule + durationStartup$
5: **else**
6: $lowerBoundInAJob \leftarrow operation.job.previousOperation.lastEndTime$
7: **end if**
8: π.insertAnOperationOnAMachine($lowerBoundInAJob$, $operation$, **true**) using Algorithm 4
9: **end for**

Algorithm 3 Backward decoding an active flexible job shop production schedule

Input: *sequencedOperations* (with assigned machines)
Output: a production schedule π which ends as late as possible

1: **for** *index* $\leftarrow$ size(*sequencedOperations*) : 1 **do**
2: $operation \leftarrow sequencedOperations(index)$
3: **if** $index$ == size($sequencedOperations$) **then**
4: $upperBoundInAJob \leftarrow dueTime - durationShutdown$
5: **else**
6: $upperBoundInAJob \leftarrow operation.job.nextOperation.firstStartTime$
7: **end if**
8: π.insertAnOperationOnAMachine($upperBoundInAJob$, $operation$, **false**) using Algorithm 4
9: **end for**

tion sequence of a job should be fully respected by backward following the fixed operation sequence of a job.

The complexity of decoding an original or semi-active schedule is O(n). Comparatively, the complexity of decoding active schedule is O(n^2). While it is apparent to decode an original or semi-active schedule, Algorithm 2 and Algorithm 3 are proposed as follows to forward and backward decode an active schedule, respectively.

4.4.2.1 Forward Decoding an Active Schedule

Algorithm 2 describes how to forward insert operations to build an active schedule (Figure 4.2e). Operations are inserted to the time horizon of the corresponding machine following the forward operation sequence (lines 1-2, Algorithm 2). Each insertion should be performed as early as possible, while respecting the operation sequence of a job (lines 3-7, Algorithm 2) and machine availability (line 8,

Algorithm 4 Operation insertion on a machine of a flexible job shop

Input: *boundInAJob*, *operationToBeInserted*, *isForwardInsertion*
Output: **true** if this insertion is successful, **false** otherwise

```
 1: mach ← operation.assignedMachine
 2: if isForwardInsertion==true then
 3:     freePeriods ← mach.getFreePeriodsLaterThan(boundInAJob)
 4:     indexSet ← 1 : size(freePeriods)
 5: else
 6:     freePeriods ← mach.getFreePeriodsEarlierThan(boundInAJob)
 7:     indexSet ←size(freePeriods) : 1
 8: end if
 9: for index ∈ indexSet do
10:     freePeriod ← freePeriods(index)
11:     if mach has no assigned operation then
12:         requiredDuration ← operation.duration
13:     else
14:         if period is before existing operations on mach then
15:             requiredDuration ← operation.duration +
                    mach.getSetupTime(operation, mach.nextOperation)
16:         else if period is after existing operations on mach then
17:             requiredDuration ← operation.duration +
                    mach.getSetupTime(mach.previousOperation, operation)
18:         else
19:             requiredDuration ← operation.duration +
                    mach.getSetupTime(mach.previousOperation, operation) +
                    mach.getSetupTime(operation, mach.nextOperation)
20:         end if
21:     end if
22:     if period ≥ requiredDuration then
23:         mach.insert(operation, period)
24:         return true
25:     end if
26: end for
```

Algorithm 2). If an operation is the first one in a job, the operation sequential constraint does not exist, such that the $lowerBoundInAJob$ is set as the start of the scheduling span plus the time to start up this machine (lines 3-4). Otherwise, the $lowerBoundInAJob$ is the last end time of the previous operation in this job (lines 5-7, Algorithm 2).

The function insertAnOperationOnAMachine($lowerBoundInAJob$, $operation$, true) in line 8 of Algorithm 2 is described in detail by Algorithm 4. It starts from the input $lowerBoundInAJob$, and forward looks for the earliest feasible free period on the target machine to accommodate the input $operationToBeInserted$ (lines 2-4 & 9-26, Algorithm 4). If the target machine is not yet assigned any operations, a feasible free period should accommodate the duration of $operation$ (lines 11-12, Algorithm 4). Otherwise, the sequence-dependent setup times should

be additionally considered for accommodation (lines 14-21, Algorithm 4).

Generally, feasible free periods ($freePeriod(s)$, Algorithm 4) are categorized in three types, regarding the operation insertion position relative to the existing operations on the target machine: (1) leftmost of existing operations (lines 14-15, Algorithm 4), (2) between existing operations (lines 16-17, Algorithm 4), (3) rightmost of existing operations (lines 18-19, Algorithm 4). For the first type of $freePeriod$, a machine changeover occurs between the $operationToBeInserted$ and the first existing operation (i.e., with the earliest start time among all existing operations). For the second type of $freePeriod$, two changeovers should be considered between the previous existing operation (relative to $operation$) and $operation$, as well as between $operation$ and the next existing operation (relative to $operation$). For the third type of $freePeriod$, a changeover between the last existing operation (i.e., with the latest start time among all existing operations) and $operation$ should be performed.

4.4.2.2 Backward Decoding an Active Schedule

The existing FJSSP research focuses on minimizing the makespan (Table 4.1). This chapter additionally employs the as-late-as-possible philosophy to diversify the population. Although the makespan increases, this aims to create more trade-off scheduling solutions along the Pareto approximation frontier.

Algorithm 3 presents the method to backward insert operations to build an active schedule from a chromosome. Operations are iterated from the last operation toward the first one (lines 1-2, Algorithm 3). Each insertion should be performed as late as possible, while simultaneously satisfying the operation sequence in a job (lines 3-7, Algorithm 3) and machine availability (line 8, Algorithm 3). If an operation is the last one in a job, the *upperBoundInAJob* equals the due time minus the time to shut down the target machine (lines 3-4, Algorithm 3). Otherwise, the *upperBoundInAJob* is the first start time of the next operation in this job (lines 5-7, Algorithm 4).

Analogously, the function insertAnOperationOnAMachine(*upperBoundInAJob*, operation, false) is introduced in Algorithm 4. It starts from the input *upperBoundInAJob*, and backward looks for the latest feasible free period on the target machine to accommodate the input *operationToBeInserted* (lines 5-8 & 9-26, Algorithm 4).

4.4.2.3 Timing Policy

When using Algorithms 2-4, the exact start time of each operation still needs to be assigned. This chapter employs the directional timing policy while other timing policies may be alternatively used. In a directional timing policy, if the decoding is forward, the first start time of an operation to be inserted is the start time of the earliest feasible free period, and this operation is forward inserted on the time

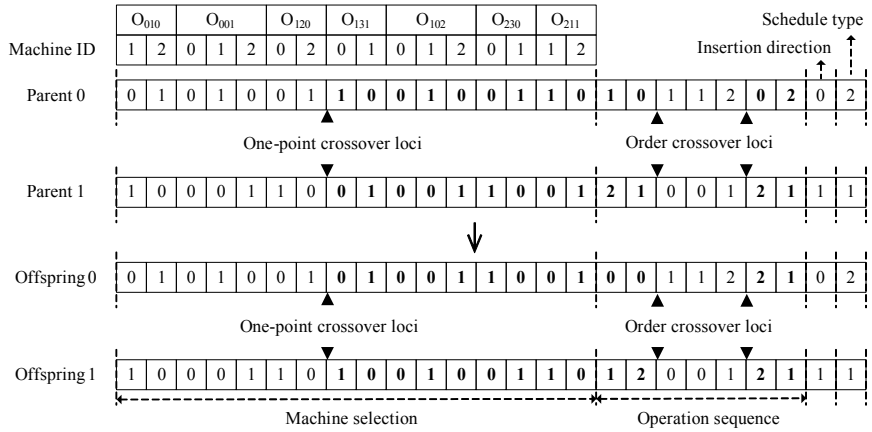


Figure 4.3: Example of one-point and order crossovers applied for the chromosome segments of machine selection and operation sequence, respectively. The genes in bold are exchanged from the other parent.

horizon; otherwise, the last end time of an operation to be inserted is the end time of the latest feasible free period, and this operation is backward inserted.

4.4.3 Crossover

The one-point crossover (also used in Section 2.6) and order crossover are applied for the chromosome segments of machine selection and operation sequence, respectively. Figure 4.3 illustrates based on the chromosome in Figure 4.1. For the one-point crossover, the crossover locus is randomly generated between two adjacent operations. Then two parents swap their chromosome segments after this crossover locus, while keeping their chromosome segments before this crossover locus. The indicated crossover locus is randomly generated among six optional crossover loci. For the order crossover [33], two cut points are randomly selected and applied to both parents. Genes between these two cut points (inclusive) remain to the corresponding offspring. Starting from the second cut point, the rest genes are copied one by one from the other parent if the number of a certain job ID is not larger than the original number; otherwise, this copy operation is omitted. The former copy operation returns to the gene position of zero if it reaches the end of a chromosome. It continues until reaching the first cut point.

4.4.4 Mutation

The swap-based mutation is applied to both chromosome segments of machine selection and operation sequence. Figure 4.4 demonstrates an example. For the

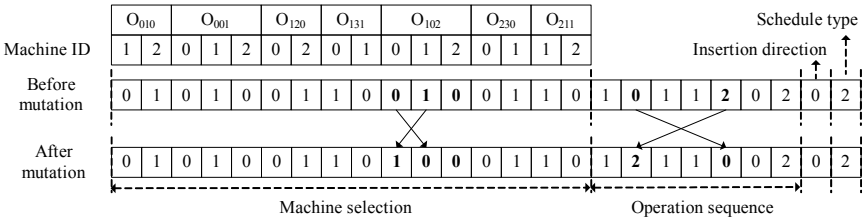


Figure 4.4: Example of swap-based mutation of the chromosome parts of machine selection and operation sequence. The genes in bold numbers are exchanged.

former segment, an operation is randomly selected from all operations each of which can be processed on multiple machines. Then the unique true bit swaps with another randomly selected false bit of the same operation. For the latter chromosome segment, two loci with different job IDs are randomly generated and then swap their genes, i.e., these two job IDs.

4.4.5 Solution Evaluation based on Discrete-Event Simulation

A scheduling solution is evaluated in terms of objectives that are defined in Equations (4.2)-(4.6). As solution evaluation is frequently performed in population initialization and recombination, the model formulated in Section 4.3 is extensively calculated in an evolutionary search. DES is used to reduce this calculation overhead, since its runtime depends on the number of discrete events in a simulation instead of the number of time slots. It thus facilitates large-scale optimization, in contrast to the commercial off-the-shelf solver (e.g., IBM CPLEX) which is very sensitive to the size of an optimization problem.

Figure 4.5 presents the multi-layer framework for energy- and labor-aware discrete-event production simulation on the shop floor. The shop floor level comprises multiple machines and human workers. Machines of different types have distinct sets of operations that they can perform, e.g., grinding, milling, and cutting. Human workers of different types have distinct skill sets, such as operation assisting, quality checking, and packaging. A machine may be accompanied by a number of human workers, depending on the production configuration. Machines do not have any explicit connection among each other in a flexible job shop, as jobs may have different routes.

The machine level in Figure 4.5 consists of five generic power states that mimic both energy consumption and production behaviors of a machine. The *Off* state is the location to start and end a simulation. It has a self-transition triggered by the command event "stay off" which indicates the duration for a machine to stay off. It transitions to *Startup* upon the command event "power on". A machine auto-

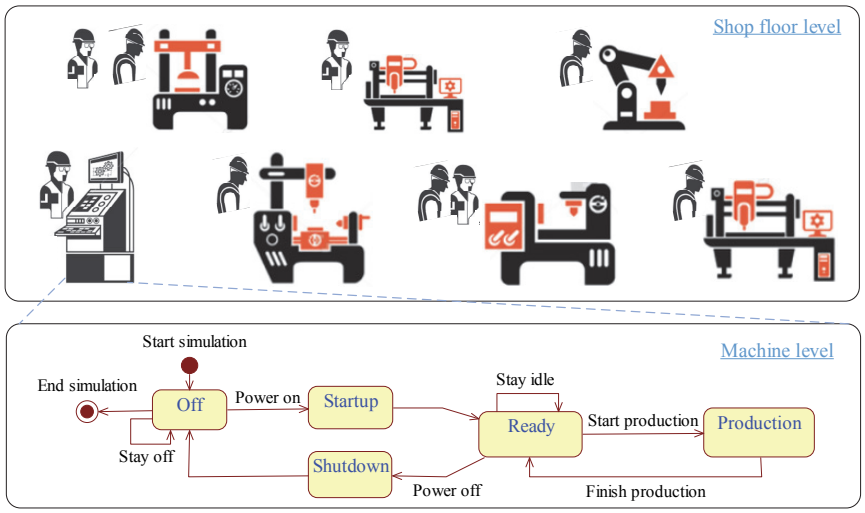


Figure 4.5: Multi-layer energy- and labor-aware shop floor production simulation framework

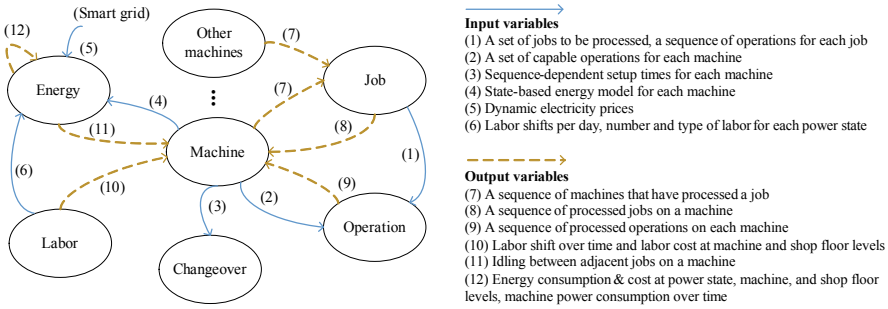


Figure 4.6: Major input and output variables of energy- and labor-aware shop floor production simulation, as well as the interdependency of sustainable production aspects including machines, operations, changeovers, jobs, energy, and labor.

matically transitions from *Startup* to *Ready* when it completes the startup process. *Ready* state has three types of events: (1) "stay idle" upon which a machine stays idle for an assigned duration, (2) "start production" upon which a machine transitions to the *Production* state, (3) "power off" upon which a machine transitions to the *Shutdown* state for powering off. A machine returns from *Production* to *Ready* when it finishes processing the assigned job. When entering into the *Shutdown* state, it automatically transitions to *Off* after completing the shutdown procedure. Figure 4.6 further introduces the key input and output variables for such a simu-

lation, as well as the interdependency of machines, operations, changeovers, jobs, energy, and labor.

4.4.6 NSGA-III Framework

The NSGA-III [1] is briefly presented below regarding its enhancement compared to NSGA-II in order to effectively solve an MaOP. It follows the NSGA-II framework that emphasizes nondominated solutions in a population. However, unlike in NSGA-II, it also emphasizes population members that are in some sense associated with each of the well-spread reference points, which are supplied upon the start of a NSGA-III instance and adaptively updated with the population evolution. To this end, the crowding distance operator in NSGA-II is replaced by the following sequential approach.

(1) Determination of reference points on a hyper-plane: the chosen reference points can either be predefined in a structured manner (e.g., the systematic approach of Das and Dennis [34]) or supplied preferentially by the user. Since these reference points are widely distributed on the entire normalized hyperplane, the obtained solutions are also likely to be widely distributed on or near the Pareto-optimal font. (2) Adaptive normalization of population members: the normalization procedure and the creation of the hyper-plane is done at each generation using extreme points ever found from the start. (3) Association operation: a reference line corresponding to each reference point on the hyper-plane is defined, and a population member is associated with the reference point whose reference line is closest to this population member in the normalized objective space. (4) Niche-preservation operation: a new niche-preserving operation is devised based on the niched count, which indicates the number of population members that are associated with a reference point.

4.5 Numerical Experiments

4.5.1 Configurations

The numerical experiments were performed on a computer with Intel i5-3470 CPU @ 3.20 GHz and 8 GB RAM. The $8 \times 8 \times 27$ (number of machines $\times$ number of jobs $\times$ number of operations) partial JFSSP instance was taken from the commonly-used benchmarking instance set [35]. While the machine processing times were indicated in the original instance, the machine processing power of an instance was randomly generated from the range [5, 26] kW based on the general machine processing power which is either empirically measured or reported in literature. Each job in this instance was set to contain 5000 workpieces. The operation-sequence dependent machine setup times were randomly generated from

the discrete set $\{10\text{ s}, 20\text{ s}, 30\text{ m}, 1\text{ h}, 1\text{ h }30\text{ m}, 2\text{ h}\}$, which was statistically extracted from the changeover data of a Singaporean manufacturer in the precision engineering domain. The setup time is zero for two contiguous operations of the same type.

A 24-hour horizon is divided into three labor shifts, each lasting 8 hours, i.e., the morning shift (6 a.m. - 2 p.m.), the late shift (2 p.m. - 10 p.m.), and the night shift (10 p.m. - 6 a.m. on the next day). As the general energy model shown in Figure 4.5 was used, the mapping between power states and type of personnel follows that of Table 4.3, where the number of workers per type of personnel was set to 1. The labor wage is the same in morning and late shifts, and increases by 10% in a night shift. Compared to a shift on a working day, the labor wage in the same shift but on a weekend day rises by 36%, if weekend production is allowed on the shop floor.

The NSGA-II [36], which is widely used for multi-objective optimization, was taken as a benchmark algorithm of NSGA-III for the many-objective EL-FJSSP. The hypervolume (HV) [37] and inverse generational distance (IGD) [38] were chosen as the performance metric for a many-objective evolutionary search, respectively, as each of them can simultaneously measure the convergence and diversity of the obtained nondominated solutions. A larger HV and a smaller IGD indicate an approximation set with better convergence and diversity. To enable the calculation of both metrics, the reference Pareto front approximation set was obtained by running NSGA-II and NSGA-III twice, respectively, and aggregating the obtained nondominated solutions into a global approximation set. The stop criteria was maximum 3 minutes. The crossover and mutation rates were 0.9 and 0.1, respectively. The population size was 212, which was approximately the number of reference points (6 outer divisions and 0 inner division) using the Das and Dennis's approach [34].

4.5.2 Scheduling under Real-Time Pricing (RTP)

The RTP or day ahead pricing data was taken from Belpex, the Belgium electricity spot market [39]. Under this pricing scheme, electricity prices vary in each hour; the prices for the next day are known after 3 p.m. on the current day. To get adapted to this pricing scheme and the labor shift configured in Section 4.5.1, the scheduling time span was set to 24 hours, lasting from 10 p.m. (included) on the current day to 10 p.m. (excluded) on the next day. NSGA-II and NSGA-III were run for 30 independent times, respectively, on the small-sized problem instance (Section 4.5.1).

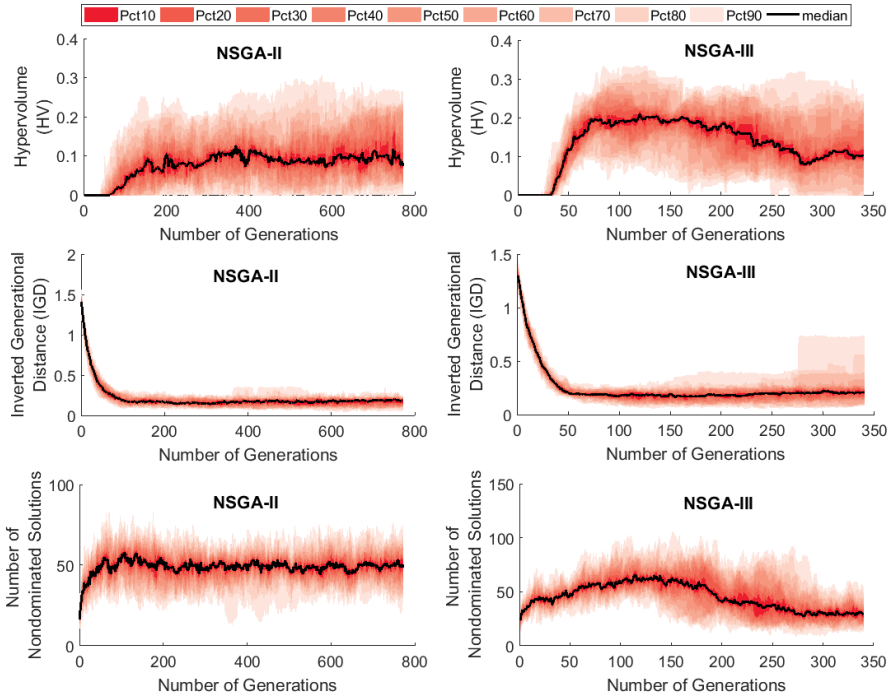


Figure 4.7: Statistical convergence curves of NSGA-II and NSGA-III for the $8 \times 8 \times 27$ energy- and labor-aware flexible job shop scheduling problem (EL-FJSSP) under real-time pricing (RTP), in terms of hypervolume (HV), inverted generational distance (IGD), and number of nondominated solutions. These curves were obtained from 30 independent runs of NSGA-II and NSGA-III, respectively.

4.5.2.1 Convergence of NSGA-II and NSGA-III

Figure 4.7 depicts the statistical convergence curves of NSGA-II and NSGA-III, in terms of HV, IGD, and number of nondominated solutions. The number of generations in each algorithm is the minimal among 30 runs to ensure the statistical meaning of all these stochastic evolutionary searches. All the 6 curves in Figure 4.7 are converged within the given time budget, except that the HV curve of NSGA-II still vibrates a bit near the end of a run. In contrast, the HV curve of NSGA-III remains stable after the 250th generation. The final median HV of NSGA-III (0.060) is higher than that of NSGA-II (0.045). The IGD curves of both algorithms rapidly converge at around the 100th generation. Compared to HV curves, IGD curves of both algorithms have much smaller variations in each generation. The final median IGD of NSGA-III (0.200) is comparable to that of NSGA-II (0.155). The number of nondominated solutions of NSGA-II vibrates around 40 early starting

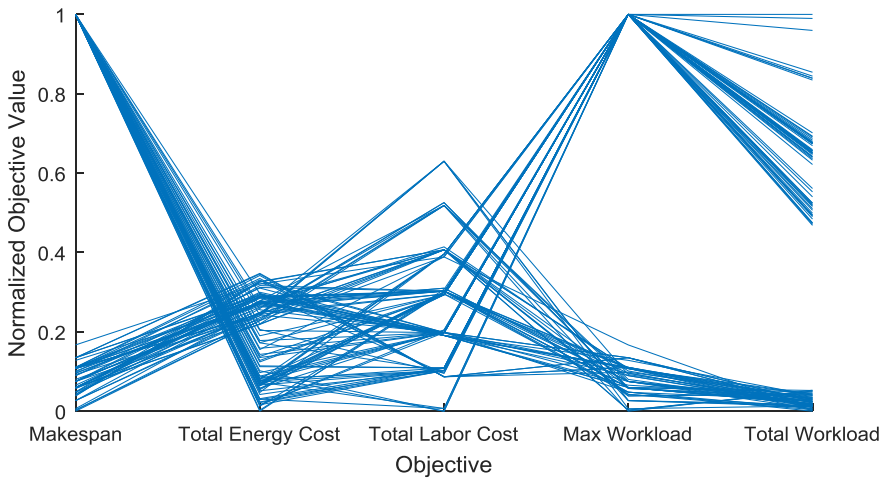


Figure 4.8: Parallel coordinates plot of an approximation set obtained by NSGA-III for an $8 \times 8 \times 27$ energy- and labor-aware flexible job shop scheduling problem (EL-FJSSP) under real-time pricing (RTP). The scheduling time span is 24 hours. The values of each objective (makespan, total energy cost, total labor cost, maximum workload, and total workload) are normalized, respectively

from the 100th generation, while that of NSGA-III steadily rises to 55 in the first 140 generations, drops till the 250th generation, and remains around 30 afterward. The variations in the number of nondominated solutions of both algorithms are notable, which are up to 100% compared to the median. Another observed phenomenon is that the number of generations in NSGA-II is as about 5 times larger as that in NSGA-III.

Overall, NSGA-III is more effective than NSGA-II in solving a many-objective EL-FJSSP, as the former has a higher median HV and comparable IGD, and produces less nondominated solutions, which decreases the difficulty in solution selection for a production manager or a decision maker.

4.5.2.2 Relation among Five Production Objectives

Figure 4.8 illustrates the parallel coordinates plot of an approximation set provided by NSGA-III, where the values of each objective were normalized. In such a plot, a curve across the 5 objectives (Equations (4.2)-(4.6), i.e., makespan, total energy cost, total labor cost, maximum workload, and total workload) represents a nondominated solution. The superposition of all curves reveals the intrinsic relation among these 5 objectives. As clearly exhibited in Figure 4.8, the makespan objective strongly conflicts with the total energy cost objective. This is explained by the

fact that more and longer free durations will be created by postponing production jobs (thus raising the makespan). These free durations provide the evolutionary search with more opportunities to make use of lower-priced periods. The situation is vice versa if production jobs are fast processed: the total energy cost has to increase as a trade-off.

The total energy cost shows a relatively weak trade-off relation with the total labor cost (Figure 4.8). A lower total energy cost is slightly compromised by a higher total labor cost, and vice versa. This is explained by the contradiction of the electricity price and the labor wage: the former is usually higher during the day and lower in the night, and the other way around for the latter. Consequently, if an optimization process shifts the whole production load from the day to the night for energy cost reduction, this has to be compensated by an increasing labor cost. However, this contradiction may decrease if part of the production loads are shifted out of the night to the day, such that the rising part of total labor cost is insignificant. This is why some curves have lower absolute slope values between the total energy cost and the total labor cost, though these absolute slope values are not zero.

The total labor cost has a notable trade-off relation with the maximum workload (Figure 4.8). If the maximum workload is reduced, production jobs are more evenly distributed on machines in a flexible job shop. A rising number of labor shifts are thus more likely to be triggered, since even one job on a machine will require at least one operator to work during the whole shift on that machine. On the contrary, if the maximum workload rises, production is more concentrated on a limited number of machines. As a result, the production in each labor shift is more compact, leading to a lower total labor cost. Besides, the maximal workload has two clusters of values, which are at the two extremes of the whole range. This implies that it is a sensitive variable for an EL-FJSSP. A machine has either a very high workload or a very low workload, which is an important consideration factor when a production manager or a decision maker selects a final scheduling solution from an approximation set.

Nevertheless, the relation between the maximal workload and the total workload is nearly harmonious (Figure 4.8). The curves between these two objectives do not cross each other. This indicates that there is no trade-off relation between the maximal workload and the total workload. A low and a high maximal workload corresponds to a low and a high total workload, respectively. However, the corresponding range of the total workload is large (upper half of the whole range). This is due to the high maximal workload. In such a scenario, a limited number of machines are highly used and could be the bottleneck of the whole production, as other jobs are more likely to wait before these machines until the jobs under execution are completed. Consequently, this delays the whole production and increases the total workload. On the contrary, if jobs are distributed on more machines, the

maximal workload will significantly decrease by reducing the number of bottleneck machines, and the total workload will thus reduce as jobs are less likely to wait before bottleneck machines.

Both the total energy cost and the total labor cost remain in a limited ranges of small values (Figure 4.8). This proves that the integration of energy and labor awareness in a conventional flexible job shop scheduling model is effective, and the tailored NSGA-III is efficient in simultaneous minimization of both cost parts, despite the trade-off between them and the other three objectives that also need to be optimized.

4.5.3 Schedule Visualization

A backward active scheduling solution from the approximation set provided by NSGA-III in Section 4.5.2 is visualized in Figure 4.9. Evidently, the operations of job0-job7 are backward placed on the time horizons of the corresponding machines, such that no operation can be further postponed without advancing other operations. The machine setup time exhibits its undeniable impact on the makespan of processing the 8 jobs. Machine3 turns out to be a bottleneck of the whole production, in the sense that it is under full utilization since its relatively early startup at 13:17:39 on 15 November 2016 until the due time. This high utilization could be explained by two reasons. Firstly, both job1 and job4 recirculate on machine3 such that machine3 has relatively more operations to process. Secondly, the makespan of many other machines is explicitly or implicitly affected by the full occupation of machine3, e.g., machine1, machine2, and machine5. Specifically, the job processing on machine5 highly depends on the operation processing of the same job on other machines. Consequently, it leads to the earliest start time in the schedule, directly influencing the makespan of the whole production.

Besides jobs and changeovers, the labor and the shift are also assigned in the schedule shown in Figure 4.9. Both operator and quality checker are scheduled based on shifts. As the whole production are concentrated on the second half of the time span, all machines except machine0, machine4, and machine6 are assigned a morning shift and a late shift after 6 a.m. on 15 November 2016, during which an operator is required on each of these machine. Despite the light production load on these machines, the morning shift is triggered besides the late shift. This gives an illustration on the potential of total labor cost reduction by framing more compact production within less shifts. Machine4 is assigned a late shift, as it only has production close to the due time (10 p.m. on 15 November 2016 in Figure 4.9). Less quality checker shifts are required compared to the operator shifts (4 quality checkers vs. 11 operators in Figure 4.9), as a quality checker only needs to inspect the production quality at the last processing stage of a job.

Furthermore, the overall power consumption of all machines is predicted by

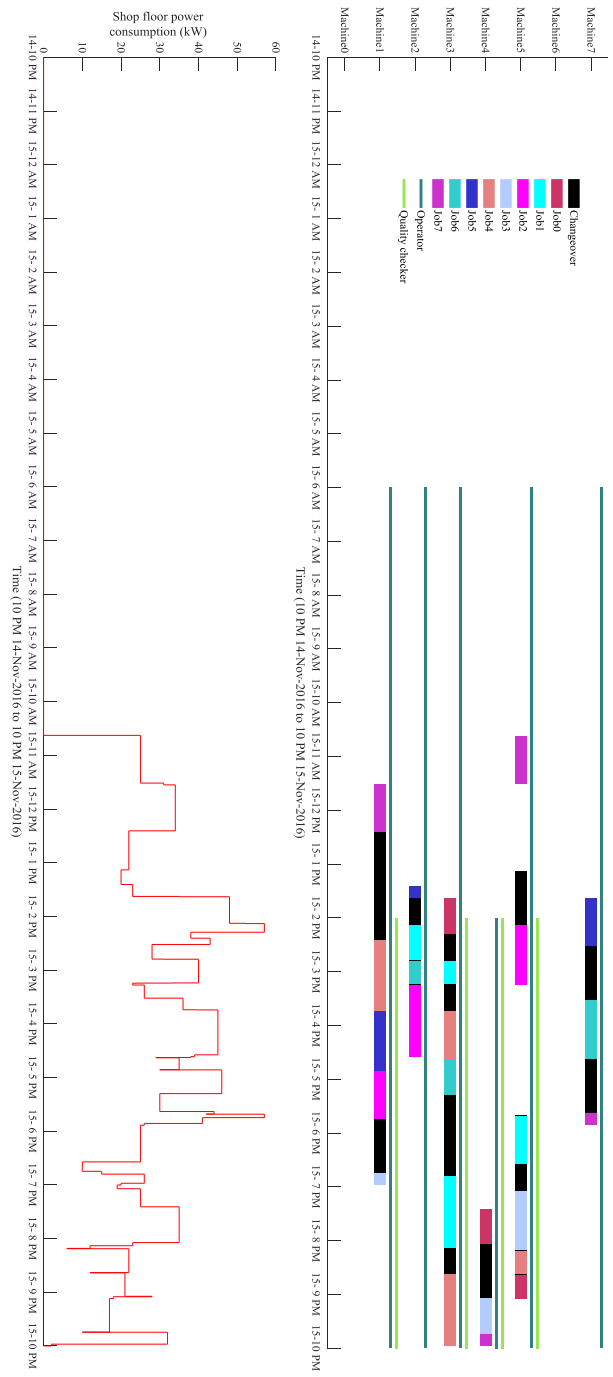


Figure 4.9: Visualization of an active backward production and labor schedule for an $8 \times 8 \times 27$ energy- and labor-aware flexible job shop and the predicted overall power consumption over time.

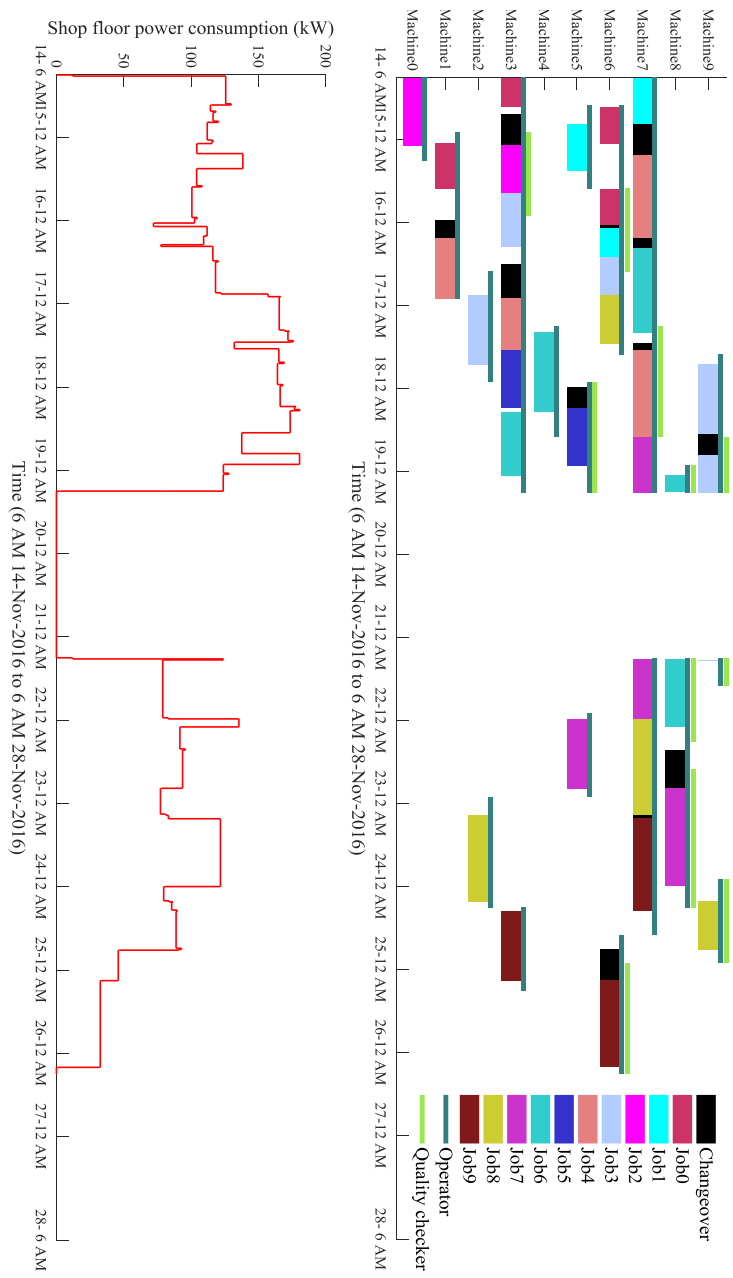


Figure 4.10: Visualization of an original forward production and labor schedule for a $10 \times 10 \times 34$ energy- and labor-aware flexible job shop and the predicted overall power consumption over time. The scheduling time span is two weeks, where weekend production is not allowed.

the DES framework (Section 4.4.5) and visualized in Figure 4.9. The power stays zero from the start of the scheduling time span until the earliest production (job7 on machine5). It then becomes a dynamic curve with different production loads on all machines over time. The highest power consumption occurs at about 2 p.m. and 6 p.m. on 15 November 2016, respectively. This is caused by the simultaneous job processing or setup of most machines during these two periods. In this way, the shop floor simulation framework (Figure 4.5) provides production or factory managers the capability to notice the peak consumption before the actual production and proactively take the counter-measures, e.g., adjusting the production schedule to lower the power consumption. The predicted power consumption over time can also help a factory to achieve a better negotiation with power plants for a more economical energy contract.

Figure 4.10 further demonstrates an original forward schedule for a $10\times10\times34$ flexible job shop over two weeks, where the weekend production is prohibited. Compared to the schedule in Figure 4.9, a job or a changeover split is additionally demonstrated in Figure 4.10. Such a split is illustrated by job7 on machine7, job6 on machine8, and job3 on machine9. Correspondingly, the power of the whole shop floor drops to and remains zero during the whole weekend while considering the labor shift, i.e., from 6 a.m. on 19 November to 6 a.m. on 21 November 2016.

4.5.4 Scheduling under Time-of-Use Pricing (ToUP)

The proposed scheduling method was then demonstrated under ToUP, where the electricity pricing data was taken from a Belgium plastic bottle manufacturer. To ensure statistical significance, NSGA-II and NSGA-III were independently run 30 times, respectively. Figure 4.11 presents the statistical convergence curves of both algorithms. All the 6 curves indicate the saturation of both algorithms within the given time budget, with two exceptions. The first gentle exception is the HV of NSGA-II. It steadily rises since the 50th generation and reaches 0.149 at the end (top left of Figure 4.11). Although this HV remains close to 0.149 after the 420th generation, the saturation trend is not evident enough. The second slight exception is the number of nondominated solutions of NSGA-II, which gently increases to 40 at the end of an evolutionary search (bottom left of Figure 4.11). In contrast, the number of nondominated solution of NSGA-III has a rise to 38 in the first one third of generations, and steadily decreases in the latter two thirds of generations with a final value of 23. Analogous to the observations in Section 4.5.2.1, NSGA-III has a higher median HV (0.263). Its IGD (0.345) is comparable to that of NSGA-II (0.365).

Overall, two conclusion points can be drawn from Figure 4.11, while taking into account the observations in Section 4.5.2.1. Firstly, NSGA-III is superior to NSGA-II in solving a many-objective EL-FJSSP, due to its higher and more

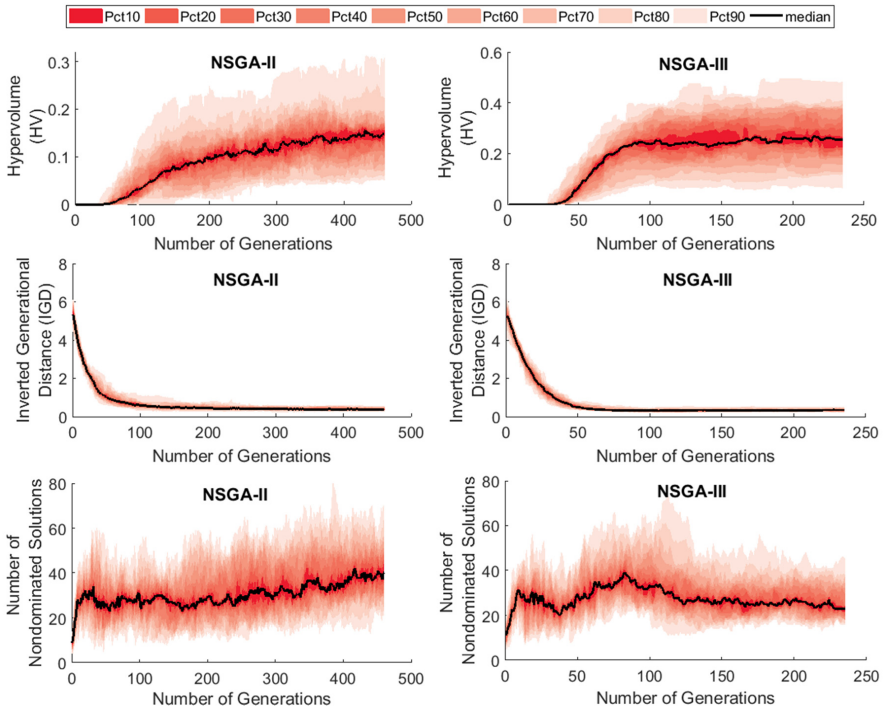


Figure 4.11: Statistical convergence curves of NSGA-II and NSGA-III for the $8 \times 8 \times 27$ energy- and labor-aware flexible job shop scheduling problem (EL-FJSSP) under time-of-use pricing (ToUP), in terms of hypervolume (HV), inverted generational distance (IGD), and number of nondominated solutions. These curves were obtained from 30 independent runs of NSGA-II and NSGA-III, respectively.

stable HV as well as comparable IGD on the one hand, and evidently less nondominated solutions that are outputted on the other hand. Secondly, HV is a more suitable metric in an EL-FJSSP. Although IGD provides performance evaluation with smaller variations, it cannot evidently differentiate the many-objective optimization performance of different evolutionary algorithms.

The relation among the 5 objectives (Equations (4.2)-(4.6), i.e., makespan, total energy cost, total labor cost, maximal workload, and total workload) is demonstrated by the parallel coordinates in Figure 4.12. Compared to the observations in Section 4.5.2.2, the trade-off between the makespan and the total energy cost still exists, though the conflict strength decreases a bit. The contradiction between the total energy cost and the total labor cost also holds. The range of total energy cost is nearly doubled. This is explained by the electricity price structure. The ToUP has only two pricing levels: on-peak and off-peak. Besides, the on-peak period is

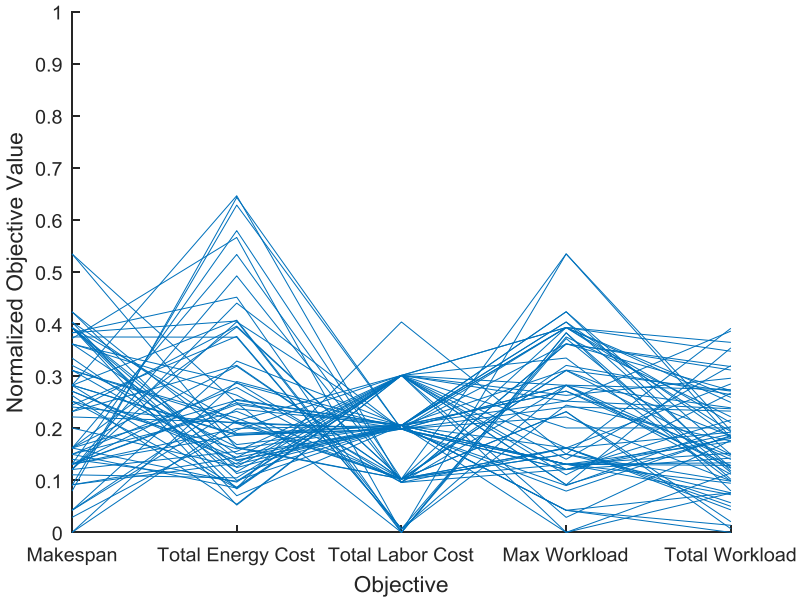


Figure 4.12: Parallel coordinates plot of an approximation set obtained by NSGA-III for an $8 \times 8 \times 27$ energy- and labor-aware flexible job shop scheduling problem (EL-FJSSP) under time-of-use pricing (ToUP). The scheduling time span is two weeks while production is not allowed on weekends. The values of each objective (makespan, total energy cost, total labor cost, maximum workload, and total workload) are normalized, respectively.

longer than the off-peak period. Both factors lead to less opportunities in reducing the total energy cost, compared to RTP. As a result, it is easier for the total energy cost to rise and have a larger range under ToUP than under RTP.

However, the range of total labor cost reduces by nearly 50% in Figure 4.12, compared to that in Figure 4.8. As there is less production load shifting under ToUP, the assignment of labor shifts varies in a weaker manner. This consequently leads to the smaller variation in the total labor cost under ToUP. The conflict between the total labor cost and the maximal workload still remains in 4.12, though the strength decreases compared to that in Figure 4.8. The range of maximal workload decreases by almost 50%. This is also because of the less production load shifting under ToUP, which facilitates the NSGA-III to minimize in a general sense. The significantly reduced range is analogously observed in the total workload (Figure 4.12) due to the same reason. Nonetheless, the nearly harmonious relation between the maximal workload and the total workload remains as that in Figure 4.8.

Again, both the total energy cost and the total labor cost still remain in ranges with small values (Figure 4.12). This also demonstrates the effectiveness and efficiency of the proposed model and tailored NSGA-III in simultaneous minimization of these two cost parts, despite the trade-off between them and the three other objectives that are minimized at the same time.

4.6 Conclusions and Future Work

This chapter proposes an energy- and labor-aware flexible job shop scheduling problem (EL-FJSSP) under dynamic electricity pricing and tailors a NSGA-III for many-objective optimization of this problem. The contributions are threefold, covering problem modeling, solution algorithm, and analytics. (1) Machine energy consumption, human worker, and labor shift are jointly modeled and integrated in the EL-FJSSP. In this way, the intrinsic trade-off relation between the energy cost and the labor cost under dynamic electricity pricing is captured, instead of being ignored in the existing research on energy-aware FJSSP, and human workers can be matched to the scheduled production, instead of isolated production scheduling and human worker planning in literature. (2) For the first time, the NSGA-III is tailored for many-objective optimization of a FJSSP. Discrete-event simulation (DES) is used to build a digital twin of the energy- and labor-aware flexible job shop. DES is coupled with the tailored NSGA-III for efficient shop floor simulation and scheduling solution evaluation. (3) Through numerical experiments under different dynamic electricity pricing schemes, the underlying relations among the important production objectives are quantitatively revealed, i.e., makespan, total energy cost, total labor cost, maximal workload, and total workload. The effectiveness and efficiency of applying a NSGA-III in solving a many-objective EL-FJSSP are demonstrated by benchmarking with NSGA-II.

Some important conclusions have been drawn. (1) It is of economic importance to model the labor aspect in an energy-aware FJSSP under dynamic electricity pricing, due to the demonstrated conflict between the energy cost and the labor cost. Although the portion of both cost parts in the overall production cost varies, the consideration of both energy and labor aspects in a FJSSP increases the flexibility of the model, when facing diverse industrial application cases. (2) In contrast to integer programming in a commercial off-the-shelf solver (e.g., IBM CPLEX), which has intrinsic problems of intractability and scalability, the DES is an effective and efficient approach to implement a complex model, e.g., the proposed EL-FJSSP. (3) NSGA-III is more effective and efficient than NSGA-II in solving a many-objective EL-FJSSP, due to a higher hypervolume (HV) and a lower number of nondominated solutions in all experiments. This conclusion may still hold for other variants of FJSSP or even other scheduling problems. (4) The

HV is a more suitable metric in an EL-FJSSP, compared to the inverted generational distance (IGD), as the IGD has difficulty in differentiating multi-objective optimization performance of different algorithms. This conclusion may hold for other scheduling problems.

Future research can be executed in the following aspects: (1) investigation on whether conclusions 3 & 4 still hold for other production scheduling problems or other types of scheduling problems, (2) design of a novel evolutionary algorithm for many-objective optimization in the context of fast yet high-quality decision making, (3) integration of intra-factory transportation in the current EL-FJSSP and its implementation in a DES environment, (4) application of the proposed energy- and labor-aware production scheduling method to the real production on the shop floor and benchmarking by empirical measurements on the production lines.

References

[1] K. Deb and H. Jain. *An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints*. IEEE Transactions on Evolutionary Computation, 18(4):577–601, Aug 2014.

[2] R. Deng, Z. Yang, M. Y. Chow, and J. Chen. *A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches*. IEEE Transactions on Industrial Informatics, 11(3):570–582, June 2015.

[3] Yong Wang and Lin Li. *Critical peak electricity pricing for sustainable manufacturing: Modeling and case studies*. Applied Energy, 175(Supplement C):40 – 53, 2016.

[4] Sojung Kim, Chao Meng, and Young-Jun Son. *Simulation-based machine shop operations scheduling system for energy cost reduction*. Simulation Modelling Practice and Theory, 77(Supplement C):68 – 83, 2017.

[5] Michael L. Pinedo. *Scheduling - Theory, Algorithms, and Systems*. Springer, 5 edition, 2016.

[6] N. B. Ho and J. C. Tay. *GENACE: an efficient cultural algorithm for solving the flexible job-shop problem*. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), volume 2, pages 1759–1766 Vol.2, June 2004.

[7] Jose M. Framinan, Rainer Leisten, and Rubén Ruiz García. *Manufacturing Scheduling Systems*. Springer-Verlag London, 2014.

[8] H. Ishibuchi, N. Akedo, and Y. Nojima. *Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems*. IEEE Transactions on Evolutionary Computation, 19(2):264–283, April 2015.

[9] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. *Many-Objective Evolutionary Algorithms: A Survey*. ACM Comput. Surv., 48(1):13:1–13:35, September 2015.

[10] Min Dai, Dunbing Tang, Yuchun Xu, and Weidong Li. *Energy-aware integrated process planning and scheduling for job shops*. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 229(1_suppl):13–26, 2015.

[11] Joon-Yung Moon and Jinwoo Park. *Smart production scheduling with time-dependent and machine-dependent electricity cost by considering distributed*

energy resources and energy storage. International Journal of Production Research, 52(13):3922–3939, 2014.

[12] AL-QASEER Firas and GIEN Denis. *A multi-objective genetic method minimizing tardiness and energy consumption during idle times*. IFAC-PapersOnLine, 48(3):1216 – 1223, 2015. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[13] C.A. Garcia-Santiago, J. Del Ser, C. Upton, F. Quilligan, S. Gil-Lopez, and S. Salcedo-Sanz. *A random-key encoded harmony search approach for energy-efficient production scheduling with shared resources*. Engineering Optimization, 47(11):1481–1496, 2015.

[14] Yan He, Yufeng Li, Tao Wu, and John W. Sutherland. *An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops*. Journal of Cleaner Production, 87(Supplement C):245 – 254, 2015.

[15] S. Kemmoé, D. Lamy, and N. Tchernev. *A Job-shop with an Energy Threshold Issue Considering Operations with Consumption Peaks*. IFAC-PapersOnLine, 48(3):788 – 793, 2015. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[16] Ying Liu, Haibo Dong, Niels Lohse, and Sanja Petrovic. *Reducing environmental impact of production during a Rolling Blackout policy – A multi-objective schedule optimisation approach*. Journal of Cleaner Production, 102:418 – 427, 2015.

[17] Gökan May, Bojan Stahl, Marco Taisch, and Vittal Prabhu. *Multi-objective genetic algorithm for energy-efficient job shop scheduling*. International Journal of Production Research, 53(23):7071–7089, 2015.

[18] T. Stock and G. Seliger. *Multi-objective Shop Floor Scheduling Using Monitored Energy Data*. Procedia CIRP, 26(Supplement C):510 – 515, 2015. 12th Global Conference on Sustainable Manufacturing – Emerging Potentials.

[19] Dunbing Tang and Min Dai. *Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem*. Chinese Journal of Mechanical Engineering, 28(5):1048–1055, Sep 2015.

[20] Ying Liu, Haibo Dong, Niels Lohse, and Sanja Petrovic. *A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance*. International Journal of Production Economics, 179(Supplement C):259 – 272, 2016.

[21] Miguel A. Salido, Joan Escamilla, Adriana Giret, and Federico Barber. *A genetic algorithm for energy-efficiency in job-shop scheduling*. The International Journal of Advanced Manufacturing Technology, 85(5):1303–1314, Jul 2016.

[22] Wenjun Xu, Luyang Shao, Bitao Yao, Zude Zhou, and Duc Truong Pham. *Perception data-driven optimization of manufacturing equipment service scheduling in sustainable manufacturing*. Journal of Manufacturing Systems, 41(Supplement C):86 – 101, 2016.

[23] Rui Zhang and Raymond Chiong. *Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption*. Journal of Cleaner Production, 112(Part 4):3361 – 3375, 2016.

[24] Davide Giglio, Massimo Paolucci, and Abdolreza Roshani. *Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems*. Journal of Cleaner Production, 148(Supplement C):624 – 641, 2017.

[25] Hadi Mokhtari and Aliakbar Hasani. *An energy-efficient multi-objective optimization for flexible job-shop scheduling problem*. Computers & Chemical Engineering, 104(Supplement C):339 – 352, 2017.

[26] Miguel A. Salido, Joan Escamilla, Federico Barber, and Adriana Giret. *Rescheduling in job-shop problems for sustainable manufacturing systems*. Journal of Cleaner Production, 162(Supplement):S121 – S132, 2017.

[27] Lvjiang Yin, Xinyu Li, Liang Gao, Chao Lu, and Zhao Zhang. *A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem*. Sustainable Computing: Informatics and Systems, 13(Supplement C):15 – 30, 2017.

[28] Liping Zhang, Qiuhua Tang, Zhengjia Wu, and Fang Wang. *Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops*. Energy, 138(Supplement C):210 – 227, 2017.

[29] Yingfeng Zhang, Jin Wang, and Yang Liu. *Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact*. Journal of Cleaner Production, 167(Supplement C):665 – 679, 2017.

[30] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[31] Guohui Zhang, Xinyu Shao, Peigen Li, and Liang Gao. *An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem*. Computers & Industrial Engineering, 56(4):1309 – 1318, 2009.

[32] Christian Bierwirth. *A generalized permutation approach to job shop scheduling with genetic algorithms*. Operations-Research-Spektrum, 17(2):87–92, Jun 1995.

[33] P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, I. Inza, and S. Dizdarevic. *Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators*. Artificial Intelligence Review, 13(2):129–170, Apr 1999.

[34] Indraneel Das and J. E. Dennis. *Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems*. SIAM Journal on Optimization, 8(3):631–657, 1998.

[35] I. Kacem, S. Hammadi, and P. Borne. *Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(1):1–13, Feb 2002.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.

[37] E. Zitzler and L. Thiele. *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach*. IEEE Transactions on Evolutionary Computation, 3(4):257–271, Nov 1999.

[38] Q. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. *Multiobjective optimization test instances for the CEC-2009 special session and competition*. Technical report, Nanyang Technological University, 2008.

[39] Belgium electricity spot market. *Belpex*. http://www.belpex.be/market-results/the-market-today/dashboard/, 2017.

Part II

Planning and Reconfiguration of Wireless Communication Systems

5

Planning of Dense and Robust Industrial Wireless Networks

The manufacturing industry is currently undergoing an emerging evolution closely associated with Internet of Things (IoT) technologies. In the Industry 4.0 initiative [1], field devices, machines, plants and factories are envisioned to be connected to a network (e.g., the Internet or a private factory network) by means of IoT infrastructures. This will further allow storing, processing, and analyzing in the cloud or edge all the information and knowledge about the physical objects (e.g., 3-D model, topology, process data, etc.), and will eventually stimulate the creation of new business models to fully make use of the available bulk IoT data. For instance, many traditionally high-cost and low-yield industrial applications are envisioned to be unlocked, e.g., predictive maintenance, compliance with regulations, field rounds, safety, and asset management [2].

However, it is still a widely recognized challenge to deploy reliable IoT networks in harsh industrial environments. Compared to the office and commercial environments which are already deployed with diverse wireless networks, a typical industrial environment is full of metal obstacles, such as racks, machines, robot arms, automated guided vehicles (AGVs), and forklifts. While these metal obsta-

cles easily cause wireless coverage holes on the shop floor or in the warehouse, the conventional wireless planning tools have not fully considered this harshness.

To fill this gap, this chapter proposes an over-dimensioning (OD) model, which automates the decision making on deploying a robust industrial wireless local area network (IWLAN). This model creates two full coverage layers while minimizing the deployment cost, and guaranteeing a minimal separation distance between two access points (APs) to prevent APs that cover the same region from being simultaneously shadowed by an obstacle. Moreover, an empirical one-slope path loss model, which considers three-dimensional obstacle shadowing effects, is proposed for simple yet precise coverage calculations. To solve this OD problem even at a large size, an efficient genetic algorithm based over-dimensioning (GAOD) algorithm is designed. Genetic operators, parallelism, and speedup measures are tailored to enable large-scale optimization. A greedy heuristic based over-dimensioning (GHOD) algorithm is further proposed, as a state-of-the-art heuristic benchmark algorithm. In small- and large-size OD problems based on industrial data, the GAOD was demonstrated to be 20%-25% more economical than benchmark algorithms for OD in the same environment. The effectiveness of GAOD was further experimentally validated with a real deployment system. Although this chapter focuses on an IWLAN, the proposed GAOD can serve as an automated decision making tool for deploying robust industrial wireless networks of many other types, such as wireless sensor networks and RFID (radio-frequency identification) networks.

5.1 Introduction

While IoT penetrates in manufacturing industry, wireless technologies gain dominant popularity over cabled technologies in the various industrial upgrades. According to the analysis of Cisco and Rockwell Automation, the advantages that wireless networks can bring to industry include [3]: 1) lower installation costs due to cabling and hardware reduction, 2) lower operational costs by eliminating cable failures, 3) ability to connect hard-to-reach and remote areas, 4) gains in productivity and efficiency due to equipment mobility, 5) higher productivity and less downtime due to personnel mobility. In the concept of Industry 4.0, ubiquitous industrial modular structures are proposed to be interconnected by a wireless network via various wireless technologies, e.g., WiFi, ZigBee, Bluetooth, 6LoWPAN, cellular networks, WirelessHART, ISA100, and VHF/UHF radios [4]. These various wireless technologies can be run at a range of industrial end-entities such as computer hosts, robots, sensor-mounted autonomous mobile units for industrial use (e.g., AGV), data-support systems (e.g., servers storing inventory details), and roaming workers with personal digital assistants (PDAs).

To deploy a wireless network in a target environment, a human network planner may perform a site survey to empirically determine the number and location of wireless nodes [5]. Evidently, this is rather a trial-and-error approach, which is inefficient. Fortunately, a wireless network planning tool automates and enhances this decision making, by recommending an optimal or nearly-optimal deployment solution for a target environment. In this way, a human network planner only needs to place each wireless node according to this obtained solution. Both the deployment time and the deployment cost are thereby reduced. The contribution of such an automated wireless network planning tool is even strengthened when applying to an industrial indoor environment, which can be very large and otherwise requires extensive manual site survey work.

Nevertheless, an industrial indoor environment is harsh for radio propagation [6, 7], compared to office environments which most wireless network planning tools focus on. It is dominated by various metal or steel objects, such as production machines, storage racks, materials (e.g., steel bars, metal plates, etc.), and vehicles (e.g., AGVs, cranes and forklifts). These obstacles shadow the radio propagation and cause coverage holes on desired areas. According to [8], the steel, metal, and rotating machinery often cause an additional path loss as high as 30 - 40 dB. This jeopardizes stable wireless connection of personnel and machines on the shop floor or in the warehouse. Consequently, only one coverage layer provided by the existing wireless network planners [9, 10] is vulnerable to these shadowing effects.

Furthermore, large-scale industrial WLAN (IWLAN) deployment has rarely been investigated in literature. For instance, the warehouse of a typical car manufacturer in Belgium measures 83,000 m^2. If the grid cell size is one meter, there are then 83,000 candidate locations for placing an AP. Most of the wireless network planning research is only limited to a small- or medium-scale environment, varying from several hundreds of square meters to several thousands of square meters [11]. A large building floor of 12,600 m^2 was considered in [12] for WLAN planning. But only 258 candidate AP locations and a dozen of APs were involved, which significantly reduces the complexity of the wireless planning problem. A similar simplification can be observed in [13–15], which only enables optimization at a small or medium scale. It is challenging to perform large-scale optimization because of the significantly increased computational resources (memory and CPU time) and the stricter requirement for efficient algorithm design.

Additionally, recent studies focus on wireless sensor network (WSN) planning rather than on WLAN planning, though the deployment cost of a dense WLAN is far more than ignorable and needs dedicated planning or optimization. For example, a WSN often contains 10 - 1,000 cheap sensor nodes [15]. If one sensor costs 10 euro, the deployment cost can surpass 10,000 euro. As a result, the large-scale property of a WSN makes it still of economic importance to perform WSN planning [14, 16, 17]. Analogously, the deployment cost of a dense IWLAN cannot

be ignored due to the much higher price of an industrial AP and the large size of an industrial environment, though the number of APs would be smaller due to the larger coverage radius of an AP. For instance, the total cost of one Siemens Scalance W788-2 M12 AP is more than 1,550 euro, including the necessary accessories such as six antennas, one power cable, one power supply box, one connector, etc. Then the deployment of 100 APs of this type will cost more than 155,000 euro, without even considering the labor cost and other engineering costs. Therefore, it is also of economic significance to minimize the IWLAN deployment cost.

This significance is even enhanced when redundant APs are deployed for enhancing robustness, which is a prevalent WSN deployment strategy [18]. The idea of creating redundancy for reliable communications can be found in many other existing communication and network technologies, such as redundant radio [19], multiple channels [20], and multiple network paths [21]. A number of recent studies on WSN planning also create at least one redundant coverage layer for reliability [16, 17]. However, this idea is rarely observed in deploying redundant APs for a robust WLAN. Compared to most of the existing work, the robustness enhancement approach proposed in this chapter does not require any change in existing protocols, making it cost effective by using commercial off-the-shelf devices (i.e., APs).

To fill these gaps, this chapter makes fourfold contributions. (1) An industrial over-dimensioning (OD) problem is investigated. In this novel problem, two full WLAN coverage layers are planned in a large harsh industrial indoor environment, while the deployed cost is minimized. An empirical one-slope path loss model, which considers the shadowing effects of three-dimensional (3D) obstacles, is utilized for precise yet simple coverage calculation. (2) An efficient genetic algorithm based OD (GAOD) algorithm is proposed for solving this OD problem even at a large size. To enable large-scale optimization, the solution representation, population initialization, crossover, and mutation of GAOD are designed, and parallel genetic search framework and problem-dependent speedup measures are proposed. (3) A greedy heuristic based OD (GHOD) is also introduced, which represents a state-of-the-art OD heuristic and serves as a benchmark algorithm for the GAOD. (4) The effectiveness and superiority of this GAOD is both experimentally validated and numerically demonstrated, in contrast to most wireless planning literature that only has numerical experiments without any real system deployment.

The rest of this chapter is organized as follows. Section 5.2 provides the literature review. Section 5.3 presents an overview of the proposed method to enhance the robustness of industrial wireless coverage. Section 5.4 describes the mathematical formulation the OD problem. Section 5.5 and Section 5.6 present the GHOD and the GAOD, respectively. Section 5.7 experimentally validates the proposed GAOD. Section 5.8 numerically demonstrates the effectiveness and efficiency of GAOD in two vehicle manufacturers' indoor environments, standing for a small

and large industrial indoor environment, respectively. Section 5.9 draws conclusions.

5.2 Literature Review

5.2.1 Wireless Network Planning

Table 5.1 summarizes and compares the recent studies on wireless network planning, in terms of network type, number of coverage layers, spatial dimensionality, problem size/scale, coverage/path loss model, consideration of obstacles, and solution algorithms. A fundamental issue for all these studies is coverage maximization, although the investigated problem differs in every study. Moreover, metaheuristics gain more popularity compared to heuristics that are very problem-dependent. As clearly presented gaps, most studies are limited within WSN planning, one coverage layer, two-dimensional (2D) space, a small or medium problem size, a Boolean disk model, and ignorance of obstacles in the target environment. Although obstacles are considered in [22, 23], the obstacle shadowing effect is simply modeled: a grid point is not considered to be covered by a sensor node if an obstacle is located between this sensor node and this grid point. Two more advanced path loss models are used in [24, 25]. But they neglect the shadowing of obstacles, due to their focus on office environments which are less harsh than industrial environments.

5.2.2 Measurement-based Techniques for Robustness

In addition to wireless planning, recent research investigates the robust industrial wireless communications at the network usage stage, during which the deployed network is in use by clients. Extensive measurements are performed to enable a better insight into the robustness performance of various countermeasures against the harshness of industrial environments.

Through live tests in a mineral processing factory, the proposed lightweight packet error discriminator (LPED) was demonstrated to enable quick recovery from link outage [34]. Forward error correction (FEC) is used in [35] to determine the error pattern in corrupted packets. This contributes to shorter detection time and boosts the reliability. The measurements in a paper mill and a paper roll warehouse demonstrated that the LPED accelerates link diagnostics by at least 190%, compared to the state-of-the-art approaches.

The bit- and symbol-error properties of IWSN were extracted and scrutinized in harsh industrial environments [36]. The measurement campaign was conducted at two paper mills and a paper warehouse during 14 days. The diversity of environments (highly reflective and absorbent) and setups (large and small separations,

Table 5.1: Review of recent wireless network planning studies

Literature	Network	Coverage layer	Space	Scale	Path loss model	Obstacle	Solution algorithm
[25]	WLAN[a]	1	2D	Small	One-slope	No	Metaheuristic
[24]	WLAN+LTE[b]	1	2D	Small	Two-slope	No	Metaheuristic
[26]	RFID[c]	1	2D	Medium	Boolean disk	No	Heuristic
[27]	RFID	1	2D	Small	Boolean disk	No	Metaheuristic
[28]	RFID	1	2D	Medium	Boolean disk	No	Metaheuristic
[16]	WSN[d]	$k\ (k \geq 1)$	3D	Small	Boolean disk	No	Metaheuristic
[23]	WSN	1	2D	Medium	Boolean disk + probabilistic	Yes	Metaheuristic
[14]	WSN	1	2D	Small	Boolean disk	No	Heuristic
[29]	WSN	1	2D	Large	Boolean disk	No	Heuristic
[30]	WSN	1	2D	Large	Boolean disk	No	Heuristic
[31]	WSN	$k\ (k \geq 1)$	2D	Large	Boolean disk	No	Heuristic
[32]	WSN	1	3D	Medium	Sphere	No	Heuristic
[33]	WSN	1	2D	Small	Boolean disk	No	Metaheuristic
[17]	WSN	$k\ (k \geq 1)$	2D	Small	Boolean disk	No	Metaheuristic
[22]	WSN	1	2D	Medium	Boolean disk	Yes	Metaheuristic

[a] Wireless local area network.
[b] Long term evolution.
[c] Radio frequency identification.
[d] Wireless sensor network.

moving and static clutters, interfered and non-interfered links) provide a high degree of generality to the measurements. Symbol-interleaving was proven to outperform its bit counterpart. In [37], the proposed methods predict radio signal coverage by considering typical industrial environments characterized by highly dense building blockage. They are corroborated by measurements in an oil refinery site.

Although these investigations perform extensive measurements, they have an assumption that the wireless connections are well maintained. There is very limited literature focusing on the case where a wireless link may become disconnected due to shadow fading on the shop floor.

5.2.3 Wireless Standard for Industry

Although IEEE 802.11/WiFi is a dominant technology with a common familiarity, it is currently deemed unsuitable for industrial applications, e.g., real-time control and localization. Various research is dedicated to the enhancement of industrial WiFi performance. A QoS (quality of service)-enabled 802.11 network is presented in [38], which considers real-time constraints for connecting industrial intelligent devices and controllers. In [39], redundant wireless adapters share information about the outcome of acknowledged transmissions, in order to have an enhanced MAC for reliable WiFi networks. In [40], two alternative rate adaptation (RA) techniques for IEEE 802.11 are designed to meet the specific requirements of industrial communication systems. However, they all focus on modifying WiFi standards, where commercial off-the-shelf devices cannot be used anymore.

Other wireless technologies also have limitations in satisfying the critical industrial requirements. Specifically, WirelessHART is an emerging wireless technology dedicated to industrial systems due to its high reliability and robustness. Its MAC layer is based on globally synchronized multi-channel TDMA which performs channel hopping at each time slot. Its network layer supports multi-path multi-hop routing to provide robust routing [41]. It is used by [42] to enable real-time mixed-criticality communication in wireless sensor-actuator networks. However, its high latency remains a challenge for real-time industrial control.

5.3 Method Overview

The proposed method for robust industrial indoor wireless coverage includes three sequential components, as illustrated in Figure 5.1 where the mapping to the physical or cyber space of a cyber-physical system (CPS) is also indicated. In a general sense, the sequence to apply this solution is 1) mobile measurements on the target shop floor (in the physical space), 2) over-dimensioning (in the cyber space), 3) network reconfiguration based on mobile measurements and over-dimensioning

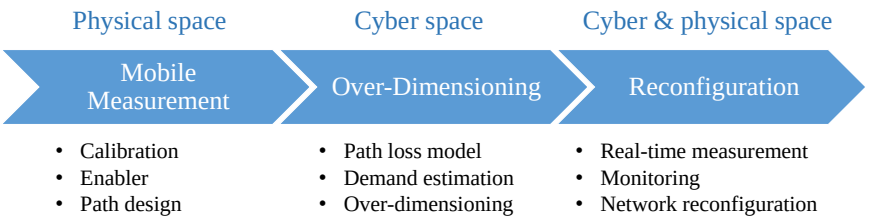


Figure 5.1: Overview of the proposed method for robust wireless coverage in harsh industrial indoor environments from the perspective of cyber-physical systems

(in both cyber and physical space). This composite solution can be either fully or partially applied to the industrial cases, implemented on a computer as a central controller or an integrated decision support system. While each component will be introduced in the following subsections, this chapter will focus on OD modeling and solution methods in the cyber space; Chapter 6 will further investigate modeling and solution methods for network reconfiguration.

5.3.1 Mobile Measurement

The proposed mobile measurement is different from conventional manual wireless measurements. It aims to improve the measurement efficiency by automating the entire wireless measurement procedure without affecting the measurement precision.

5.3.1.1 Calibration

The received signal strength indicator (RSSI) is a good indicator of radio channel performance since it is vendor-independent. However, there exists a discrepancy between the RSSI and the RF power in practice. The mapping is not standardized. It is done by each wireless chipset manufacturer separately and is locked to the public, which may therefore be subject to inaccuracies. Furthermore, commercial off-the-shelf wireless products based on these chipsets contain additional circuitry, PCB lanes, and soldered connectors around the chipset, which causes an additional deviation between the two parameters.

To this end, two types of measurements are designed to be performed simultaneously: 1) RSSI measurements by a client PC equipped with commercial off-the-shelf antennas, and 2) the actual RF power measurements by a spectrum analyser, or the actual received signal strength (RSS). The same antennas and antenna feeder cables are used for both measurements. In this way, the second type of measurements serves as a mapping of the RSSI values, to correct them to actual power.

5.3.1.2 Automated Measurement Enablers

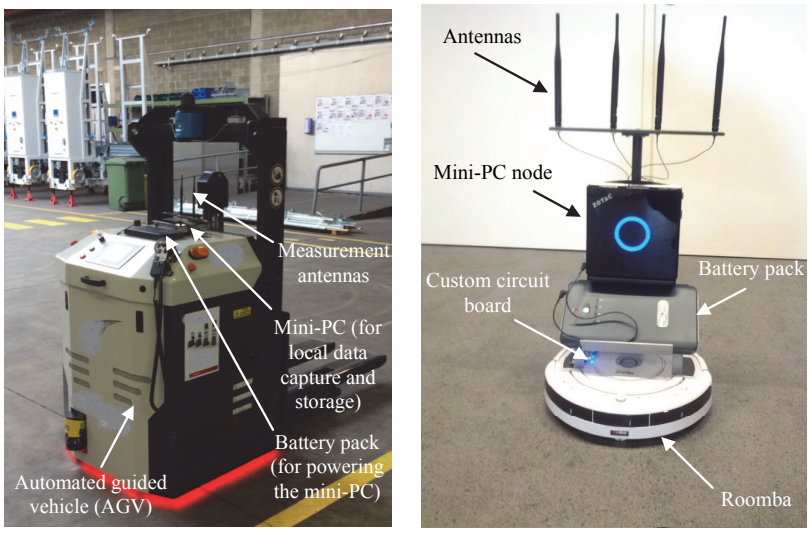
In the mobile measurement, the measurement campaigns are automatically carried out through the central controller, so as to characterize the target industrial wireless environment. The relevant advantages can be threefold: 1) enhancement of measurement efficiency compared to conventional manual measurements, 2) feasibility for dangerous or hard-to-reach measurement locations, 3) improved measurement reproducibility. The mobility of measurements can be enabled by integration of measurement setups onto various existing industrial movable infrastructures, e.g., AGVs, mobile robots, cranes, forklifts, etc. The following sub-sections give two illustrations: AGVs and mobile robots.

In flexible manufacturing systems (FMSs), rapid machine configuration can be loaded for producing different types of products depending on personalized customer demands. Flexible material handling systems (MHSs) are a necessary part of a FMS. Among heterogeneous MHSs, AGVs are especially suitable for applications where space is at a premium, flexibility is critical [43], and the intra-factory transportation is highly scheduled [44]. As a result, AGVs have been widely chosen by manufacturers to implement truly flexible MHSs [43]. The easy-to-install and cost-effective integration of measurement setups with AGVs will facilitate automatic real-time monitoring of an industrial radio environment, while AGVs are performing their scheduled tasks. This is feasible, since AGVs have their own navigation systems to iteratively follow some pre-designed paths either on a shop floor or in a warehouse. Figure 5.2a illustrates such an integration. The measured data will be wirelessly transmitted from one or multiple AGVs to the central controller. Synchronization and aggregation are then performed on the data in order to ensure that the collected data reveal the objective radio environment at the right location and the right time for the right radio resource.

Currently, there are various mobile robots which are remotely controllable and available on the market, e.g., the iRobot Roomba vacuum cleaning robots that were made fully configurable by the wireless experimenter in the w-iLab.t of the Information Technology Department at Ghent University [45]. Figure 5.2b demonstrates such an integration. The robots can be further driven by the surrogate modeling Matlab toolbox [46], which is installed in the central controller. Advanced measurement path design algorithms can be written in the surrogate modeling toolbox. Therefore, the robots can be controlled to conduct optimized on-site measurements, e.g., to cover the target area as much as possible, to have a shortest measurement path, etc.

5.3.1.3 Measurement Path Design

AGVs have their own paths for moving materials around a factory. Therefore, semi-flexible mobile measurements are enabled by integrating measurement setups



(a) Integration with an AGV (b) Integration with a mobile robot

Figure 5.2: Illustration of measurement setups with various mobile infrastructures which are available for manufacturing industry (AGV: automated guided vehicle)

with AGVs. In comparison, fully-flexible mobile measurements are facilitated by the integration with mobile robots, as mobile robots are remotely controllable.

A hybrid sequential design algorithm is proposed and integrated in the surrogate modeling Matlab toolbox in the central controller. Sequential sampling strategies have been used to reduce the amount of measurements to identify an electromagnetic pattern in a given environment [47, 48]. This algorithm path-efficiently and automatically conducts the measurement campaign with the mobile robots. The measured data helps to precisely assess and model the spatial changes of radio QoS metrics (the received signal strength or RSS is solely considered in this chapter). Furthermore, this contributes to radio environment maps (REMs), which predict the values of radio QoS metrics all over the target area.

This hybrid sequential design algorithm starts from the initial design, where a small set of measurements are chosen in a space-filling manner. Then it sequentially specifies additional optimized measurement locations in the environment until the overall electromagnetic pattern is characterized. This is named sequential design. In both initial and sequential designs, a limited number of maximally informative measurements are given by this algorithm, while all 2D location- and time-dependent variations in these measured values are characterized. The total distance traveled by the mobile measurement setup is also minimized under the constraint of physical obstacles on the shop floor. These measured values are then,

in turn, used to build a surrogate model for the RSS. This eventually leads to an intelligent monitoring sub-system (of the entire decision support system introduced in the beginning of Section 5.3). While this chapter does not focus on surrogate modeling, the sequential design procedure is presented in detail in [6].

5.3.2 Over-Dimensioning

Based on the former automatic measurements, the over-dimensioning approach is designed for AP planning and deployment. It guarantees that each target location on a shop floor is able to be covered by at least two APs. It includes the following three steps.

5.3.2.1 Path Loss Model Establishment

The data collected from automatic measurements can be used to characterize the radio propagation in indoor factories. By considering exclusively the large-scale fading, PL (in dB) is calculated as:

$$PL = P_{T_x} + G_{T_x} + G_{R_x} - \widehat{p} \tag{5.1}$$

where P_{T_x} is the transmit power, G_{T_x} and G_{R_x} are the T_x and R_x antenna gains, and $\widehat{p}$ is the mean RF power samples or RSS (in dBm) over a distance of several lambdas to eliminate the small-scale fading with approximately the same path loss.

Furthermore, PL samples can be fitted in the following form:

$$PL(d) = PL0 + 10 \cdot \log_{10}(d) + \xi \tag{5.2}$$

where $PL0$ is the PL at the reference distance of 1 m (in dB), *n* is the PL exponent which is a dimensionless parameter indicating the increase in PL with distance, and ξ is the deviation between measurement and model (in dB).

5.3.2.2 Demand Estimation

A clear view should be gained on the maximum throughput demand of the target industrial region, in order to accordingly determine the capacity of each AP. This means that it is necessary to identify and forecast the wireless application per client, per time unit, and per room or hall in the target factory. Each application has its corresponding bit rate range. Table 5.2 presents some common industrial wireless applications and their reference bit rates. The required throughput can be further converted into the corresponding minimum received RF power or RSS according to existing literature. An illustrative mapping for 802.11g at 2.4 GHz is indicated by Table 5.2. Whether this required throughput is actually achieved depends not only on the received power, but also on the number of simultaneous

users, the interference, protocols on higher layers, etc. Nevertheless, this is out of scope of this section.

5.3.2.3 Over-Dimensioning (OD)

In the OD philosophy, a redundant coverage is created on the target industrial environment. This is similar to the philosophy of mesh networks, where redundant paths exist for robustness. However, the distinction is that the OD is rather a physical approach, while the mesh is on the network layer. A redundant coverage is essential for robust industrial wireless communications, where the radio propagation easily becomes susceptible to the dynamic disturbance, e.g., moving objects and human operators which cause short-term shadowing, and reorganized production lines which cause long-term shadowing. As a consequence, an obvious QoS degradation may occur randomly and frequently. In case that the QoS provided by a link is or is predicted to be threatened, the network will start the reconfiguration to maintain an acceptable QoS. Therefore, OD serves as a network design strategy for robustness by increasing AP/link redundancy against temporal changes in AP/link states or properties. Instead of requiring a long-term change to the communication protocol stacks, commercial off-the-shelf hardware can be used in OD, indicating a competitive advantage of OD in deployment speed and cost.

5.3.3 Reconfiguration

When the deployed network is in use, automated measurements increase the network's diagnostic capability, and the network reconfigurability aims at dealing with shadowing problems which occur in a short-term and quite randomly on the target shop floor.

5.3.3.1 Real-Time Measurement

The mobile measurement setups are remotely, centrally, and continuously controlled during the network usage. The distributed mobile measurement setups automatically conduct measurements around the shop floor and return real-time feedback. The sequential design [6] can be applied to enhance the measurement efficiency. In this way, only a limited number of location points are automatically selected for additional measurements. Furthermore, the fixed APs and wireless clients can monitor the radio environment themselves, and thus provide additional timely feedback to the central controller. Consequently, the feedback effectively provides rich and objective information on the radio environment to the central controller to update the REM.

Table 5.2: Industrial wireless application, performance requirement, and mapping between physical throughput and sensitivity (received RF power)

Industrial wireless application	Reference PHY throughput	Sensitivity[b] (Reference received RF power)	Latency	Packet loss rate	Sensitivity to jitter	High availability
Supervisory control	256 kbps	-88 dBm	$<$ 100 ms	$< 10^{-9}$	Yes	Yes
Distributed I/O[a] control	512 kbps	-88 dBm	$\approx$ 100 ms	$< 10^{-9}$	Yes	Yes
Peer-to-peer control	1 Mbps	-88 dBm	$<$ 100 ms	$< 10^{-9}$	Yes	Yes
Mobile HMI[a] or mobile operator	1 Mbps	-88 dBm	$>$ 100 ms	$< 10^{-4}$	No	Yes
Long haul SCADA[a]	256 kbps	-88 dBm	$>$ 100 ms	$< 10^{-4}$	Yes	Yes
Asset tracking and RFID[a]	128 kbps	-88 dBm	$>$ 100 ms	$< 10^{-4}$	No	Yes
Condition based monitoring	128 kbps	-88 dBm	$>$ 100 ms	$< 10^{-4}$	Yes	Yes
Remote video monitoring	24 Mbps	-79 dBm	$>$ 100 ms	$< 10^{-4}$	Yes	Yes

[a] I/O: input/output, HMI: human-machine interface, SCADA: supervisory control and data acquisition, RFID: radio-frequency identification.
[b] The mapping from PHY throughput to sensitivity is based on the reference conversion for a 802.11g chipset at 2.4 GHz [49].

5.3.3.2 Monitoring

The real-time measurement provides information on the radio environment to the central controller. Based on measured QoS values, a surrogate model can be constructed by using the Kriging interpolation algorithm in the surrogate modeling toolbox [46]. This model contributes to an REM, which presents the spatial distribution of QoS values over the target environment. Triggered by predefined events, the central controller updates the REM. The trigger events can be various, e.g., receiving a predefined amount of latest feedback measurements, a certain time period, a change of industrial indoor topography, the model uncertainty surpassing an interpolation error threshold, etc. Therefore, the latest surrogate model characterizes the current environment, and facilitates the network reconfiguration for staying robust.

5.3.3.3 Network Reconfiguration

Once a weak region is indicated by the REM and stays unimproved during a defined judging period, a proper network reconfiguration will be cognitively triggered to achieve the 'best coverage'. The network reconfiguration can be illustrated as powering on/off APs, switching power levels of APs, switching radio channels, etc. The judging period, trigger events, and criteria for finding the 'best coverage' can all be varied, and depend on a specific industrial case. For instance, the defined period can be 2% of the scheduled production period for a certain production line. During this defined period, the indicated region is continuously focused on and evaluated with the REM. An example of a trigger event is that less than 95% of the whole shop floor is covered. The 'best coverage' can be the full coverage on the shop floor, an always-guaranteed coverage on some emphasized regions where there is a prioritized requirement for continuous throughput provision, etc.

5.4 Over-Dimensioning Model

The OD problem is to minimize the number of deployed industrial APs, under the constraints of two full coverage layers in a target industrial indoor environment and an inter-AP separation larger than a limit distance. Signal attenuation caused by dominant three-dimensional (3D) obstacles are considered in the path loss calculation. APs are assumed to be of the same type, remaining to be deployed in a 2D environment. A solution to the OD problem is denoted by $\vec{l}$, a vector of AP's 2D locations.

The second coverage layer serves as redundancy against coverage holes on the first layer that are caused by dominant obstacles. If two or more APs are placed

very close to each other, they are likely to be simultaneously shadowed by the same obstacle. To make the OD solution $\vec{l}$ robust against shadowing effects of obstacles, a minimal inter-AP separation (d_{APmin}), i.e., the minimal distance between any two APs, is thus a necessary constraint for this OD problem.

In the following subsections, this OD problem is mathematically modeled in Section 5.4.1. A concise illustration is demonstrated in Section 5.4.2 for enhancing the ease of understanding of this OD model. Then, the complexity analysis is performed in Section 5.4.3, in terms of time complexity of this model and its variables that influence the difficulty of large-scale optimization of such an OD problem. Finally, the concepts that are frequently used in the OD solution algorithm design are defined.

5.4.1 Model Formulation

Table 5.3 lists the notations used for the proposed OD model. In this model, a target rectangular environment is 2D, i.e., horizontal and vertical on the plan (corresponding to the length and width of a real environment). It is represented by two extreme 2D points, i.e., the bottom left point $(xMin, yMin)$ and the top right $(xMax, yMax)$ of the plan. It is discretized into $gs \times gs$ small grid cells, where gs is the grid cell size that is preset as an input of the model. Each grid point (GP) is represented by the upper-left vertex, and denoted as gp_g, where g is a unique index for each GP. A lexicographical order is applied to all the GPs:

$$(x0, y0) < (x1, y1) \Longleftrightarrow x0 < x1 \vee (x0 = x1 \wedge y0 < y1) \tag{5.3}$$

where $(x0, y0)$ and $(x1, y1)$ are two distinct GPs in the environment.

A target environment is thus described by a set of ordered GPs, which is denoted as Ω. The GP index *i* within Ω starts from one, corresponding to the extreme point $(xMin, yMin)$ of the rectangular environment. It increases one by one following the lexicographical order, until reaching Ω, the total number of GPs. Then the set of GPs is denoted by $I = 1, 2, ..., |\Omega|$. The following formula is used to determine the size of Ω:

$$|\Omega| = ceil\left[(xMax - xMin)/gs\right] \times ceil\left[(yMax - yMin)/gs\right] \tag{5.4}$$

A receiver (Rx) is placed on each GP except the GPs where APs are placed. The received power on the downlink is considered in coverage calculation. Different physical bitrates of an Rx require distinct minimum received power levels, named threshold (*THLD*) hereafter. The quantified relation can be found in [6]. A GP is covered by an AP if the received power of the Rx on that GP is higher than or equal to the threshold. The coverage of an AP is hence represented as the GPs that are covered by this AP.

Table 5.3: Nomenclature of industrial wireless network over-dimensioning model

Symbol	Meaning
AP	Access point
d_{APmin}	Minimal separation distance of two arbitrary access points
d_{ij}	Distance between the *i*-th grid point and the *j*-th access point
$d_{jj'}$	Distance between two access points
d_{max}	Maximal coverage radius of an access point
G	Total gain of a pair of transmitter and receiver
GP	Grid point
gp_g	The g-th grid point in an environment
gs	Grid cell size in a discretized environment
I	Set of indices of grid points
J	Set of indices of access points
$\vec{l}$	Location vector of over-dimensioned access points
M	Margin (dB) considering shadowing, fading, and interference
n	Path loss exponent
N_{AP}	Total number of over-dimensioned access points
N_o	Total number of dominant obstacles in an environment
OL_{ij}	Total obstacle loss (dB) between the *i*-th grid point and the *j*-th access point
OL_k	Obstacle loss (dB) of the *k*-th dominant obstacle in the environment
P_{ij}	Stable power received by the *i*-th grid point from the *j*-th access point at 99% of the time
$PL0$	Path loss (dB) at the location that is one meter away from a target access point
$PL(d_{ij})$	Path loss (dB) between the *i*-th grid point and the *j*-th access point
Rx	Wireless signal receiver
Tx	Wireless signal transmitter
$THLD$	Threshold received power (dBm) of a client receiver
TP_{max}	Maximal transmit power (dBm) of an access point
x_j	Horizontal coordinate of an access point
$xMax$	Maximal horizontal coordinate of an environment
$xMin$	Minimal horizontal coordinate of an environment
y_j	Vertical coordinate of an access point
$yMax$	Maximal vertical coordinate of an environment
$yMin$	Minimal vertical coordinate of an environment
Ω	Set of all grid points in an environment
ξ	Deviation between measurement and a path loss model, with zero mean and a standard deviation σ
α_{ij}	Logical coverage variable for the *i*-th grid point and the *j*-th access point
β_{ij}^k	Logical signal blockage variable for the *i*-th grid point, the *j*-th access pint, and the *k*-th dominant obstacle

The maximal transmit power TP_{max} of an AP is selected, enabling to minimize the number of over-dimensioned APs. All APs are of the same type. They can be placed anywhere within the target environment. The environment is assumed to have no previously-installed APs. The shadowing effects of dominant obstacles are considered in the path loss calculation. There are $\left|\vec{l}\right|$ APs in an OD solution $\vec{l}$. The AP set is denoted as $J = 1, 2, ..., \left|\vec{l}\right|$, also following a lexicographical order.

Without loss of generality, a one-slope path loss model [50] is used to calculate power loss between the AP transmit power and the received power of an Rx, additionally considering the signal attenuation caused by dominant obstacles:

$$PL(d_{ij}) = PL0 + 10 \cdot n \cdot \log_{10}(d_{ij}) + OL_{ij} + \xi \tag{5.5}$$

where $PL0$ (in dB) is the path loss at the distance of one meter, n is the path loss exponent which is a dimensionless parameter indicating that the path loss increases with the distance, d_{ij} is the distance (in m) between the Rx placed on the *i*-th GP and the *j*-th AP, OL_{ij} is the total obstacle loss (in dB) caused by the dominant obstacles that block the line between the Rx placed on the *i*-th GP and the *j*-th AP, and ξ (in dB) is the deviation between the measurement and model.

Obstacle locations are assumed to be fixed in an environment. The deviation ξ in Equation (5.5) follows a Gaussian distribution, with a mean of zero and a standard deviation σ. The gain and margin are considered in the link budget calculation to be more realistic, which is not taken into account in [51]. The total gain *G* (in dB) is the sum of the AP transmitter's gain and the Rx's gain. The margin *M* (in dB) is the sum of shadowing margin, fading margin and interference margin.

The OD model is formulated from Equation (5.6) to Equation (5.15). Equation (5.6) is the objective function for OD, while the rest equations define the constraints. The objective is to minimize the number of APs ($|J|$) that are over-dimensioned. In homogeneous network planning, this equals minimization of deployment cost. The variable is the 2D location of each AP. The output of this objective function is a vector of APs that are over-dimensioned in a target industrial indoor environment. The number of APs is unknown and can vary between different solutions for the same OD problem. The rest of the formulations define the constraints of this OD problem.

$$\min_{(x_i,y_i)\in\Omega, j\in J} (\left|\vec{l}\right|) \tag{5.6}$$

subject to:

$$\sum_{j=1}^{N_{AP}} \alpha_{ij} \geq 2, \forall i \in I \tag{5.7}$$

$$d_{max} = 10^{(\frac{TP_{max}+G-M-THLD-PL0}{10\cdot n})} \tag{5.8}$$

$$OL_{ij} = \sum_{k=1}^{N_o} \beta_{ij}^{k} \cdot OL_k, \forall i \in I, \forall j \in J \tag{5.9}$$

$$P_{ij} = TP_{max} + G - M - PL(d_{ij}), \forall i \in I, \forall j \in J \tag{5.10}$$

$$\alpha_{ij} = \begin{cases} 1, \text{if } P_{ij} \geq THLD \\ 0, \text{otherwise} \end{cases}, \forall i \in I, \forall j \in J \tag{5.11}$$

$$\beta_{ij} = \begin{cases} 1, \text{if the } k\text{-th obstacle blocks between the } i\text{-th GP and the } j\text{-th AP} \\ 0, \text{otherwise} \end{cases}, \\ \forall i \in I, \forall j \in J, \forall k \in [0, 1, ..., N_o] \tag{5.12}$$

$$d_{jj'} \geq d_{AP_{min}}, \forall j, j' \in J, j \neq j' \tag{5.13}$$

$$0 < d_{AP_{min}} < d_{max}/2 \tag{5.14}$$

$$xMin \leq x_j \leq xMax, yMin \leq y_j \leq yMax, \forall j \in J \tag{5.15}$$

Equation (5.7) requires that each GP in the environment should be covered by at least two APs. Equation (5.8) calculates the maximal distance d_{max} (in m) that an AP can potentially cover, by considering the maximal transmit power on the AP side and the minimal received power (or *THLD*) on the Rx side, and without any obstacle blocking the line of this Tx-Rx pair. Equation (5.9) calculates the total obstacle loss (in dB) for the pair of *i*-th GP and *j*-th AP.

Equation (5.10) calculates the power that an Rx on the *j*-th GP receives from the *i*-th AP. Equation (5.11) defines the logical coverage variable α_{ij} for all pairs of GP-AP. It is one, if the power that an Rx on the *i*-th GP receives from the *j*-th AP is not lower than *THLD*. Otherwise, it is zero. Equation (5.12) defines the logical blockage variable β_{ij}^{k} for the *i*-th GP, *j*-th AP, and *k*-th dominant obstacle. It equals one, if the *k*-th dominant obstacle blocks the line-of-sight propagation between the *j*-th AP and *i*-th GP. Otherwise, it equals zero.

Equation (5.13) forces that any intra-AP separation should not be shorter than the preset limit distance $d_{AP_{min}}$. Equation (5.14) sets the lower and upper bounds of $d_{AP_{min}}$. Equation (5.15) indicates where APs can be placed: inside the rectangle target environment or exactly on the boundaries (i.e., side walls of an industrial indoor environment).

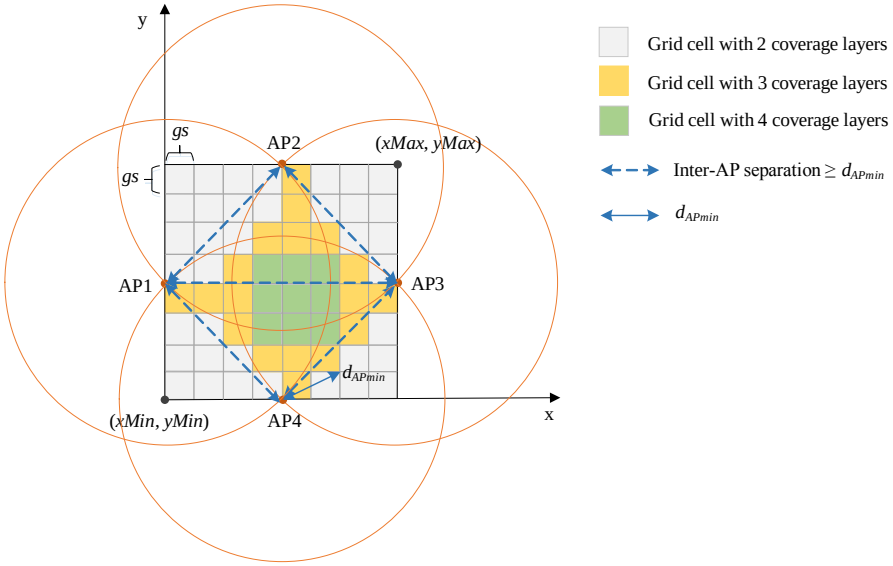
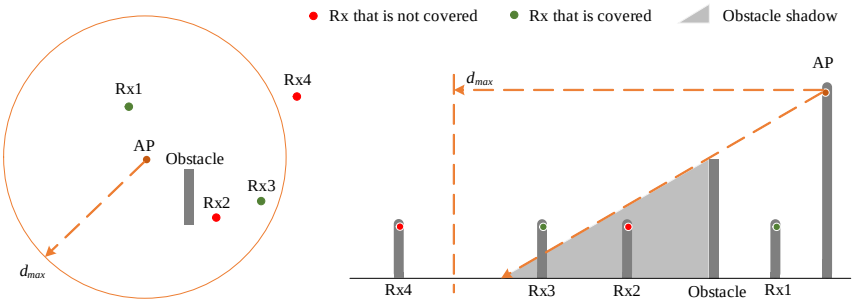


Figure 5.3: Illustration of an over-dimensioning (OD) model: 4 access points (APs) are placed in the target environment such that each $gs \times gs$ grid cell is covered at least twice and any two APs should not be placed within a minimal spatial distance d_{APmin}. A smaller number of APs indicates high quality of an OD solution.

5.4.2 Illustration of the proposed Over-Dimensioning Model

The formulated OD model is illustrated by a simple example. As depicted in Figure 5.3, a target rectangular environment is characterized by its bottom-left vertex $(xMin, yMin)$ and top-right vertex $(xMax, yMax)$. It is discretized into 8×8 grid cells. Each grid is sized by $gs \times gs$. If the top-left vertex of a grid cell is within the coverage of an AP, this grid cell is considered covered by this AP. The demonstrated OD solution in Figure 5.3 is qualified, in the sense that all grid cells are covered at least twice, and any pair of APs are placed beyond the required minimal separation distance d_{APmin}. Evidently, a new OD solution is also qualified if additional APs are placed in Figure 5.3 while satisfying the inter-AP separation of d_{APmin}. However, this new OD solution is inferior, as it outputs more APs than the OD solution shown in Figure 5.3.

Figure 5.4 further illustrates how to determine whether an Rx, which is placed on a grid, is covered by an AP. For this example illustration, i.e., Figure 5.4a and Figure 5.4b, diffraction is ignored for Figure 5.4a presents the top view. Rx1 and Rx3 are covered by an AP, as they are within the maximal coverage radius (d_{max}) of this AP. Moreover, Rx1 receives a stronger signal than Rx3, as Rx1 is closer to the AP. Rx2 is not covered due to the significant shadowing effect of the obstacle



(a) Top view of the coverage of an AP

(b) Side view of the coverage of an AP

Figure 5.4: Illustration of the coverage of an access point (AP) which is attenuated by three-dimensional (3D) obstacles. The maximal coverage radius of this AP is d_{max} in the free space.

in-between the AP and Rx2. Rx4 is not covered by the AP, either, as it is located beyond (d_{max}). Figure 5.4b shows the side view of the same coverage. Although both Rx2 and Rx3 are behind the obstacle in the two horizontal dimensions in Figure 5.4a, Rx3 is not vertically within the shadow of this obstacle. Consequently, Rx3 is still covered by the AP, while Rx2 is shadowed by the obstacle whose signal loss is strong enough to make Rx2 uncovered by the AP.

5.4.3 Complexity Analysis

As shown in [52, 53], it is non-deterministic polynomial complete (NP-complete) to achieve k-cover with a minimum number of nodes in grid-based networks. Complying with this condition, the above OD problem has additional constraints of 3D obstacle shadowing and AP separation. Therefore, this OD problem is NP-complete.

According to the classification of difficult factors in large-scale optimization [54], the complexity of the proposed OD problem is influenced by the following variables: (1) the size of an industrial indoor environment (input variable of this OD problem), (2) the number of APs (output variable of this OD problem), (3) the spatial resolution (grid cell size or *gs*) (input variable), (4) the number of coverage layers (input variable), (5) the spatial separation of APs or interdependency of AP placements (input variable), (6) the path loss model that is used (input variable). The first three variables determine the size of a search space, i.e., the total number of possible solutions. On the one hand, the fourth and fifth variables add additional yet practical constraints, which change the property of a search space. On the other hand, they introduce the active interaction between AP locations, which means that the location of each AP cannot be individually determined to find the global

optimal solution. The last variable impacts the expense of evaluating a solution during which the path loss between AP and an Rx is extensively calculated.

Given the constraint of double full coverage and minimal AP separation and using the one-slope path loss model, the outputted number of APs highly depends on the size of an industrial indoor environment. The optimization runtime depends on both the number of APs (and thus essentially the size of an environment) and the number of candidate locations which are contributed by the spatial resolution (gs). Therefore, an OD problem is considered as large-scale if the target industrial indoor environment has a large size ($> 10,000\ m^2$) and the gs is small (within several meters). Otherwise, it is considered as small-scale.

5.4.4 Concept Definitions

Several frequently-used concepts and properties are defined to facilitate the design of the OD solution algorithm in Section 5.5 and Section 5.6, respectively.

Definition 1: *covered GPs* refer to the set of GPs that are covered by at least two APs at the maximal transmit power level.

Definition 2: *once-covered GPs* stand for the set of GPs that are covered by only one AP.

Definition 3: *new once-covered GPs* represent the set of GPs that are not yet covered by any AP, but can be covered by a given AP at the maximal transmit power level.

Definition 4: *new twice-covered GPs* denote the set of GPs that are covered only once, and can be covered twice by a given AP at the maximal transmit power level.

Definition 5: *candidate GPs* indicate the set of GPs that are available for placing APs.

Definition 6: *uncovered GPs* refer to the set of GPs that are covered less than twice by all placed APs at the maximal transmit power level.

Definition 7: *valid GPs* refer to the set of GPs that are located beyond the minimal AP separation (d_{APmin}) of all the APs that are already placed in the environment.

Theorem 1: by placing APs of the same type (i.e., equal d_{max}) one by one, all the GPs within the minimal AP separation distance (d_{APmin}) of all placed APs can be covered at least twice.

Proof: an extreme case is assumed (Figure 5.5). The purpose is to find in such an extreme case a feasible region for accommodating a new AP to twice-cover the once-covered GP $(x1, y1)$. As $(x1, y1)$ is covered by an AP on $(x0, y0)$, circle0 (C0) can be obtained: $(x - x0)^2 + (y - y0)^2 = (d_{APmin})^2$. Given the constraint of minimal AP separation (d_{APmin}) defined by Equation (5.13), no additional AP can thereby be placed within C0. Besides, existing APs are already placed

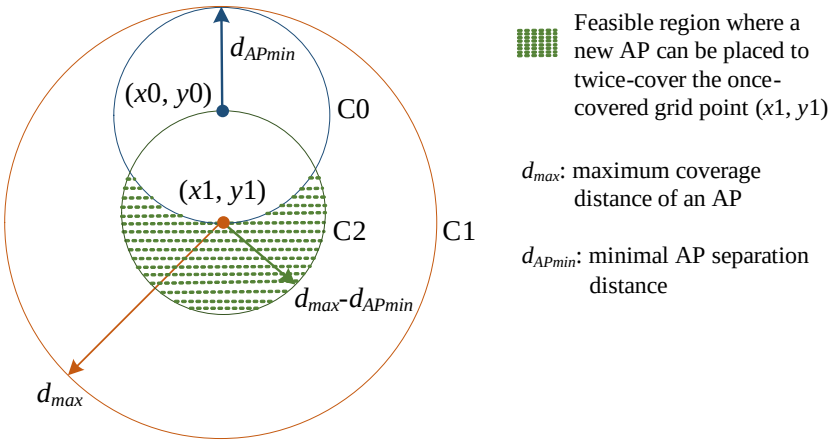
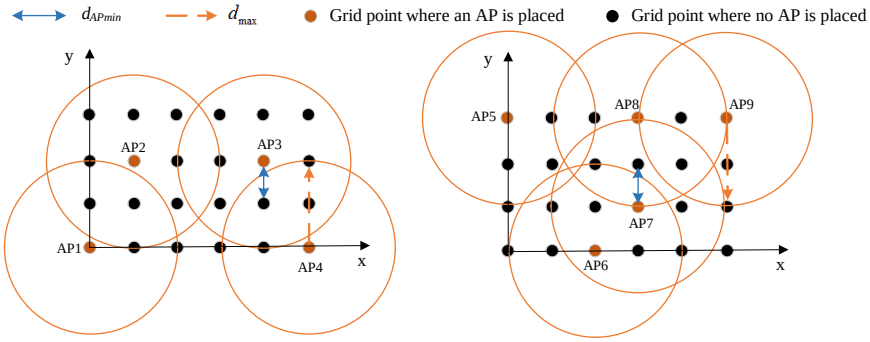


Figure 5.5: Demonstration of double full coverage in an extreme case: $(x0, y0)$ and the outer edge of circle C1 are placed with access points (APs), such that $(x1, y1)$ is only once-covered and only the shadowed region, i.e., $C2 - C0 \cap C2$, is feasible for accommodating an additional AP to twice cover $(x1, y1)$.

on the outer edge of circle1 (C1): $(x - x1)^2 + (y - y1)^2 = (d_{max})^2$, exerting extra constraints of d_{APmin} on the feasible region. Consequently, the only feasible region is $C2 - C0 \cap C2$, i.e., the shadowed region in Figure 5.5, where circle2 (C2) is $(x - x1)^2 + (y - y1)^2 = (d_{max} - d_{APmin})^2$. This feasible region always exists due to the lower and upper bounds of d_{APmin} defined by Equation (5.14), such that $d_{max} - d_{APmin} > 0$. As a result, the once-covered GP on $(x1, y1)$ can always be twice-covered, by placing a new AP in this feasible region. Theorem 1 is thus true in this extreme case. In addition, many less stringent cases evidently exist which fit theorem 1. For instance, if not all the GPs on the outer edge of C1 are placed with APs in Figure 5.5, the feasible region to place a new AP obviously becomes larger. Furthermore, if multiple *once-covered GPs* exist regardless of the location within C0, the aforementioned process can be iterated. Therefore, theorem 1 remains true.

Theorem 2: by placing APs of the same type (i.e., equal d_{max}) one by one, all the GPs beyond the minimal AP separation distance (d_{APmin}) of all placed APs (i.e., *valid GPs*) can be covered at least twice.

Proof: given the constraint of Equation (5.14), this property can be bounded by two extreme cases, i.e., $d_{APmin} = 0$ and $d_{APmin} = d_{max}/2$. If $d_{APmin} = 0$, then two APs can be placed on the same GP for double full coverage of the same area. Theorem 2 is thus true. If $d_{APmin} = d_{max}/2$, the environment is sure to have the first coverage layer, as illustrated in Figure 5.6a where $d_{max} = 2gs$ and the first full coverage layer can be created by AP1, AP2, AP3, and AP4. Then, the



(a) First full coverage layer

(b) Second full coverage layer

Figure 5.6: Full double coverage over the area beyond the minimal AP separation distance for the case where $d_{APmin} = d_{max}/2 = gs$ (grid cell size). Large orange circle denotes the coverage area of an AP with the maximal coverage radius d_{max}.

environment can be fully covered for the second time by five additional APs, as illustrated in Figure 5.6b. Obviously, all the placed APs are beyond $d_{APmin}(gs)$. Theorem 2 is thus true. There are many more cases between the two bounds that can meet theorem 2. Therefore, theorem 2 remains true.

5.5 Greedy Heuristic based Over-Dimensioning

The greedy heuristic based over-dimensioning (GHOD) algorithm is proposed in this section. It represents a heuristic to solve the OD problem and can serve as a benchmark algorithm for the GAOD proposed later in Section 5.6. It is inspired from the recently proposed wireless planning algorithm in [11]. This original algorithm determines the minimal number of APs and their locations while satisfying a specified physical bitrate in an office environment. The same idea is used in the GHOD, by placing APs one after another such that a target environment is gradually covered toward the required coverage rate. Therefore, the GHOD represents a specific heuristic for this OD problem (Section 5.4).

However, two enhancements have been made to adapt the GHOD to this OD problem. First, it creates two coverage layers instead of one and under the additional constraint of a minimal inter-AP separation. Second, it achieves the linear-time calculation in setting up the first coverage layer, by reducing the time complexity from $O(n^3)$ to $O(n)$, where n is the size of a 2D environment and the grid cell size remains the same in the orginal algorithm and the GHOD.

The time complexity $O(n^3)$ in the original algorithm is introduced by the d_{avg} criterion in judging the best AP location when placing each AP. This criterion cal-

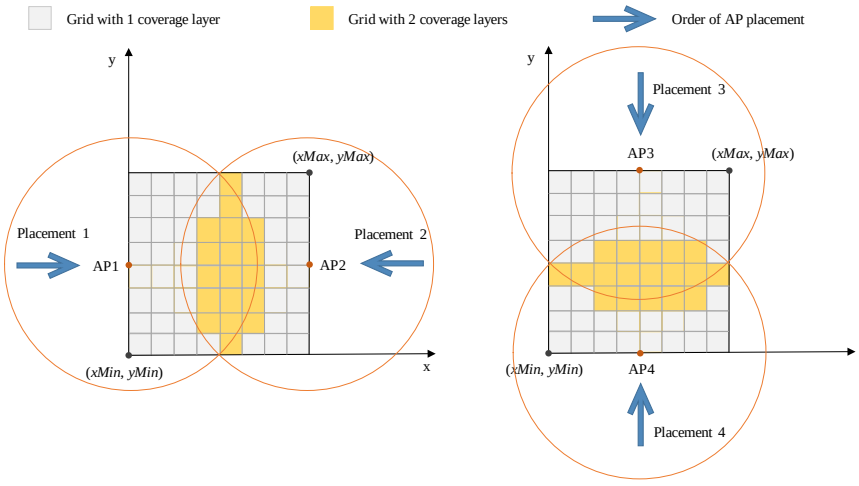
Algorithm 5 A greedy heuristic based over-dimensioning (GHOD)

Input: none
Output: an over-dimensioning solution $\vec{l}$

1: $candidateGPs \leftarrow$uniformly pick out GPs from Ω
2: **while** $|once\text{-}coveredGPs| \neq |\Omega|$ **do**
3: $bestLocation \leftarrow$GP $\in candidateGPs$ on which an AP is placed having the most *new once-covered GPs*
4: add *bestLocation* to $\vec{l}$
5: remove from *candidateGPs* all GPs within d_{APmin} of *bestLocation*, including *bestLocation*
6: **end while**
7: $candidateGPs \leftarrow \Omega$
8: remove from $candidateGPs$ all GPs within d_{APmin} of all APs that are placed, including GPs on which the APs are placed
9: **while** $|covered\text{-}GPs| \neq |\Omega|$ **do**
10: $bestLocation \leftarrow$ GP $\in candidateGPs$ on which an AP has the most *new twice-covered GPs*
11: add *bestLocation* to $\vec{l}$
12: remove from *candidateGPs* all GPs within d_{APmin} of *bestLocation*, including *bestLocation*
13: **end while**
14: Apply the lexicographical order to $\vec{l}$

culates the average distance among all uncovered GPs. It is effective to offer a minimized number of APs for a full coverage layer. Nevertheless, two full coverage layers should be set up in OD, which changes the context of the d_{avg} criterion. Besides, $O(n^3)$ is inefficient for large-scale network planning. For instance, it took 1093 seconds for creating one full coverage layer over an environment of 200 $m \times 50$ m using a PC with an Intel i5-3470 CPU and 8G RAM. It will then take about 6.25E5 seconds (about 173.6 hours) for an industrial hall of 415 $m \times 200$ m, which is very time-consuming for network planning. The original algorithm was run for the latter large-sized problem. As expected, no result was obtained after 40 hours, as this algorithm was still running.

To reduce the time complexity, the criterion of determining the best AP in [11] is altered in the proposed GHOD. As described in Algorithm 5, the GHOD establishes the two coverage layers one by one. Candidate AP locations are first established (line 1), and then iterated for picking out the best location (lines 2-6). A new AP is placed on the best location, and is powered on with the maximal transmit power to cover as many GPs as possible (line 3). The best location is the one on which a newly-placed AP contributes to the highest number of *new once-covered GPs*. It is then added to the OD solution vector $\vec{l}$ (line 4). After that, the candidate AP locations are updated by removing once-covered GPs as well as the new AP's location (line 5). The former process of AP placement (lines 3-5) is iterated until the environment is fully once-covered (line 6). Then a new



(a) AP1 and AP2 achieve the first full coverage layer

(b) AP3 and AP4 achieve the second full coverage layer

Figure 5.7: Illustrative example of the greedy heuristic based over-dimensioning (GHOD). Access points (APs) are sequentially placed such that AP1 and AP2 ensure the first coverage layer, while AP3 and AP4 guarantee the second coverage layer. Therefore, these 4 APs achieve double full coverage layers in the target environment.

set of candidate AP locations are created by removing all GPs that are within d_{APmin} of existing APs including AP locations (lines 7-8). In this way, these new candidate AP locations can fully comply with the constraint of minimal AP separation distance, i.e. Equation (5.13). Then an analog iteration is performed to ensure the environment is fully twice-covered (lines 9-13). APs in the final OD solution are reordered by following the lexicographical order (line 14), i.e., Equation (5.3). Due to theorems 1 & 2, the two iterations (lines 2-6 & 9-13, Algorithm 5) cannot be endless loops. Figure 5.7 further gives an example to perform the GHOD.

5.6 Genetic Algorithm based Over-Dimensioning

The GHOD algorithm treats an OD solution as sequential steps, and makes the local optimal decision at each step (Section 5.5). Although it has a simple time complexity of $O(n)$, it cannot guarantee a global optimal solution.

Comparatively, a genetic algorithm (GA) is one of the best-known metaheuristics in the family of evolutionary algorithms. It can give a global optimal or near-optimal solution within a reasonably short period. It has been successfully applied to solve planning and optimization problems for the manufacturing industry, such

as energy-cost-aware production scheduling [55]. Therefore, the genetic algorithm based over-dimensioning (GAOD) is proposed for the formulated OD problem (Section 5.4). To facilitate large-scale optimization, three aspects are specifically focused on in the design of GAOD, including enhancement of search efficiency, reduction of computational time, and decrease of temporary memory usage.

5.6.1 Solution Encoding

It requires special concern on the solution encoding for an efficient genetic search. Authors in [16] encode a wireless sensor placing solution as a vector, which contains all the candidate GPs for placing APs in a target environment. If a location is placed with a sensor, the value with the corresponding index in the vector is one. Otherwise, the value is zero. Unfortunately, enormous redundancy exists in this encoding space. It is natural that the number of candidate locations is larger than or equal to the number of wireless nodes to be placed. Therefore, the candidate locations, on which no AP is finally placed, contribute nothing to the final wireless planning solution. Consequently, redundancy exists in solution encoding space, which impedes the efficiency of a genetic search. A similar concern is described in [56], where a normalization method is employed to map between the genotype space and the phenotype space.

To minimize memory usage and remove encoding redundancy, the proposed GAOD encodes an OD solution as a vector that only contains the 2D locations of over-dimensioned APs, i.e., the APs that are determined to be placed on a specified location in a target environment. This vector follows the lexicographical order in Equation (5.3). Figure 5.8 illustrates an example of such an encoding/decoding scheme: although APs are placed in the order of their indices (i.e., AP1, AP2, AP3, and AP4), they are reordered in the solution vector $\vec{l}$ by following the lexicographical order (i.e., AP1, AP4, AP2, and AP3). Each element in $\vec{l}$ denotes a 2D location in the target environment for the decoding process, and vice versa for the encoding process. Compared to conventional encoding schemes which require the length of a chromosome to be fixed, the proposed encoding scheme enables chromosomes to have varying lengths and thus varying dimensionality of a search space. Although this leads to light-weighted OD solution representation, the key genetic operators (initialization, crossover, and mutation) need to be specifically designed, which will be introduced from Section 5.6.2 to Section 5.6.5.

5.6.2 Population Initialization

The initial population contains *popSize* qualified individuals which are randomly generated. The purpose is to guarantee that each initial individual can satisfy all the constraints of the OD model, and consequently to ensure an effective large-

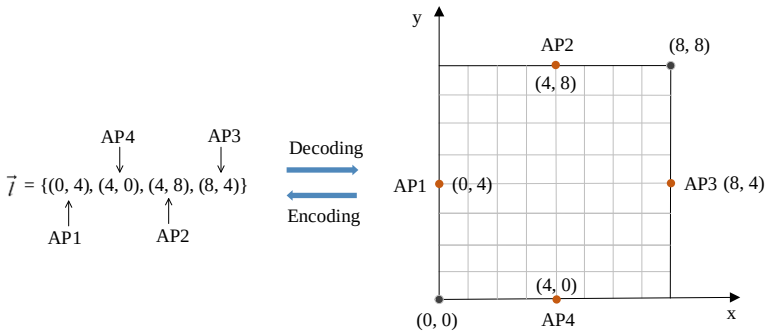


Figure 5.8: Illustrative example of the over-dimensioning solution encoding

scale genetic research.

Generally, it is not a prerequisite to initialize individuals that fully satisfy the constraints. Individuals that cannot comply with all the constraints can be assigned with the worst fitness, and are likely to be eliminated by elitism and roulette wheel selection [57]. Besides, these worst individuals have the opportunity to be improved through crossover and mutation.

Despite the feasibility of the above disqualification handling method, it obviously wastes computational resources, by possibly having unqualified individuals as candidate solutions, and evolving based on a mix of qualified and unqualified individuals. As the efficiency of a genetic search is a sensitive factor for large-scale optimization, it is of crucial importance to remove an unqualified solution upon its appearance, such that no additional computational resources are needed to propagate this disqualification. To demonstrate that the propagation of disqualification significantly affects the efficiency of a large-scale genetic search, a computational experiment was conducted. In this experiment, more than 1,000 random individuals were independently generated for a large-scale OD model (a warehouse of $415m \times 200m$). These individuals do not necessarily satisfy the constraint of two full coverage layers that are defined in Equation (5.7). As a result, no qualified OD solution was obtained from this experiment. This thus highlights the necessity of a method that fully generates qualified solutions. Such a method is described in Algorithm 6 which produces a qualified initial individual.

Algorithm 6 comprises two sequential procedures after the population initialization (lines 1-3): iteration 1 (lines 4-11) and iteration 2 (lines 12-26). The initialization allows the accommodating PC to temporally allocate memory to the three variables (i.e., *validGPs*, *uncoveredGPs* and *candidateGPs*), instead of storing them locally at each individual. This is because the number of GPs in an industrial indoor environment can be huge. It may take up a significant amount of memory to represent all the GPs. The genetic search may thereby suffer from a lack of memory.

Algorithm 6 Individual initialization of the genetic algorithm-based over-dimensioning (GAOD)

Input: none
Output: an over-dimensioning solution $\vec{l}$

1: $validGPs \leftarrow \Omega$
2: $uncoveredGPs \leftarrow \Omega$
3: $candidateGPs \leftarrow \Omega$
4: **while** $candidateGPs \neq \oslash$ **do**
5: add to $\vec{l}$ a random GP of $candidateGPs$
6: place a new AP on this GP and power it on with TP_{max}
7: remove from $validGPs$ all GPs within d_{APmin} of this GP
8: increase by one the coverage layer number of each GP that is within the coverage of this AP
9: remove the new *covered GPs* from *unceoverGPs*
10: $candidateGPs \leftarrow validGPs \cap uncoveredGPs$
11: **end while**
12: **while** $uncoveredGPs \neq \oslash$ **do**
13: $centerGP \leftarrow$ first GP in $uncoveredGPs$
14: $poolGPs \leftarrow$ all GPs within the $d_{APmin} \times d_{APmin}$ square which is centered at $centerGP$
15: $candidateGPs \leftarrow poolGPs \cap validGPs$
16: **if** $candidateGPs = \oslash$ **then**
17: $poolGPs \leftarrow$ all GPs within the $d_{max} \times d_{max}$ square which is centered at $centerGP$
18: $candidateGPs \leftarrow poolGPs \cap validGPs$
19: **end if**
20: add to $\vec{l}$ a random GP of $candidateGPs$
21: remove from $validGPs$ all GPs within d_{APmin} of this GP
22: place a new AP on this GP and power it on with TP_{max}
23: increase by one the coverage layer number of each GP that is within the coverage of this AP
24: remove the new *covered GPs* from *uncoveredGPs*
25: **end while**
26: apply the lexicographical order to $\vec{l}$

Iteration 1 places APs one by one on GPs that are covered less than twice and beyond d_{APmin} of all APs that are already placed. It is greedy in the sense that the number of *uncovered GPs* decreases one loop after another. But it does not have to be strictly greedy as the GHOD, i.e., *uncovered GPs* do not have to reduce to the maximal degree in each loop of iteration 1. This is because this strict greedy property will not guarantee the global optimum while raising the computation burden. Furthermore, it cannot be endless owing to theorem 2 (Section 5.4.4).

Iteration 2 (lines 12-26, Algorithm 6) follows and complements iteration 1. It intends to twice cover all the *uncovered GPs*, which are located within d_{APmin} of the placed APs, in a number of loops. In each loop, *candidate GPs* are selected from a specified rectangular area, which is centered at the first GP of *uncovered*

GPs (line 13). First, it is a small $d_{APmin} \times d_{APmin}$ rectangle (lines 14-15), such that a new AP can be placed in the vicinity of the first *uncovered GP*, while staying beyond d_{APmin} of all placed APs. If this small area has no qualified *candidate GPs*, a large $d_{max} \times d_{max}$ area is then created (lines 16-19). Given theorem 1 (Section 5.4.4), there must exist GPs within this large area that can accommodate new APs to cover the first *uncovered GP* for at least twice. Therefore, iteration 2 cannot be endless, either.

Moreover, as Algorithm 6 generates a random OD solution, it represents an ad hoc manual AP placement in practice and can be used as a benchmark for GHOD and GAOD. An individual is an OD solution. The number of APs in different individuals may differ due to the randomness of Algorithm 6. The minimization of the number of APs will depend on the population evolution, which is driven by the crossover, the mutation, and the elitism.

5.6.3 Crossover

The one-point crossover operation, which guarantees the qualification of offspring for the OD problem, is defined by Algorithm 7. The input is two qualified individuals (i.e., *indiv1* and *indiv2* in Algorithm 7), which are selected by the roulette wheel selection mechanism [57].

The crossover point is defined as a vertical line, named *xCrossover*. The horizontal coordinate of *xCrossover* is randomly selected (line 3) from the effective range calculated by lines 1-2 in Algorithm 7. This vertical line splits the rectangular environment into two rectangular subparts, i.e., the parts of which all the involved horizontal coordinates are smaller (part 1) and larger (part 2) than the randomly selected one, respectively. Then the two parts on the two individuals are swapped to get two children solutions (lines 4-6).

The constraint of minimal AP separation may be broken after the swap. However, it is unnecessary to check over the whole environment, since this can only occur within the small rectangular area around the vertical split line, i.e., $xCrossover - d_{APmin} \leq x \leq xCrossover + d_{APmin}$. Therefore, for speedup within this small rectangular area, if an AP is within d_{APmin} of another AP, this AP is removed from the OD solution represented by the current child (line 8).

The two children solutions are then checked (lines 9-13) whether they satisfy the constraint of two full coverage layers defined by Equation (5.7). If this constraint cannot be satisfied, iterations 1 and 2 in Algorithm 6 will be performed (lines 14-22, Algorithm 7). This is not costly in terms of time and memory, since after a swap, this constraint can be broken only in the small area $xCrossover - d_{APmin} \leq x \leq xCrossover + d_{APmin}$.

To remain memory efficient, memory-consuming variables (such as *uncoveredGPs* and *validGPs* in Algorithm 7) do not have to be locally stored in the

Algorithm 7 Crossover of the genetic algorithm-based over-dimensioning (GAOD)

Input: *indiv1* and *indiv2*
Output: *newIndiv1* and *newIndiv2*

1: $xMin \leftarrow \max$(minimal horizontal coordinates of all APs in *indiv1* and *indiv2*)
2: $xMax \leftarrow \min$(maximal horizontal coordinates of all APs in *indiv1* and *indiv2*)
3: $xCrossover \leftarrow$ a random coordinate $\in [xMin, xMax)$
4: chop graphically indiv1 and indiv2 into two parts along the same vertical line $xCrossover$, respectively
5: $newIndiv1 \leftarrow$ 1st part of *indiv1* + 2nd part of *indiv2*
6: $newIndiv2 \leftarrow$ 1st part of *indiv2* + 2nd part of *indiv1*
7: **for** $indiv \in \{newIndiv1,\ newIndiv2\}$ **do**
8: remove APs that are within d_{APmin} of any AP $\in \{$APs in *indiv*$\}$ in the rectangular area $xCrossover - d_{APmin} \leq x \leq xCrossover + d_{APmin}$
9: $uncoveredGPs \leftarrow \Omega$
10: **for** $AP \in \{$APs that remain in *indiv*$\}$ **do**
11: increase by one the coverage layer number of each GP within the coverage of the new AP
12: remove all new *covered GPs* from *uncoveredGPs*
13: **end for**
14: **if** $uncoveredGPs \neq \oslash$ **then**
15: $valideGPs \leftarrow \Omega$
16: **for** AP $\in \{$APs that remain in *indiv*$\}$ **do**
17: remove from *validGPs* all GPs within d_{APmin} of this AP
18: **end for**
19: $candidateGPs \leftarrow validGPs \cap uncoveredGPs$
20: iteration 1 (lines 4-11) in Algorithm 6
21: iteration 2 (lines 12-25) in Algorithm 6
22: **end if**
23: **end for**
24: apply the lexicographical order to *newIndiv1* and *newIndiv2*

individuals and population. Instead, they are locally generated, meaning that the occupied huge memory will be immediately freed up at the end of Algorithm 7.

5.6.4 Mutation

A mutation operation must produce a qualified individual. This new individual should be different from all existing individuals as much as possible, because in concept mutation adds diversity to a generation and avoids a GA search to quickly converge in a single direction in the solution space.

To this end, Algorithm 8 is designed for mutation in the GAOD. It mainly consists of two steps. At step 1 (lines 1-10), additional APs of the same type are added to the target environment, while respecting the constraint of minimal AP separation. At step 2 (lines 11-18), each AP in the original OD solution is

Algorithm 8 Mutation of the genetic algorithm-based over-dimensioning (GAOD)

Input: an individual $\vec{l}$
Output: new individual (updated $\vec{l}$)

1: $validGPs \leftarrow \Omega$
2: **for** AP$\in\{$original APs placed on $\vec{l}\}$ **do**
3: remove from $validGPs$ all GPs within d_{APmin} of this AP
4: **end for**
5: $numAdditionalAPs =$ ceil($rateMutation \cdot numAllAPs \cdot$ 0.5)
6: **for do**$i = 1 : numAdditionalAPs$
7: place a new AP on a random GP $\in$ $validGPs$
8: add the new AP to the set $additionalAPs$
9: remove from $validGPs$ all GPs within d_{APmin} of this AP
10: **end for**
11: **for** AP $\in$ $additionalAPs$ **do**
12: add GPs that are covered by this AP to the set $newCovGPs$
13: **end for**
14: **for** AP $\in$ $\{$original APs placed on $\vec{l}\}$ **do**
15: **if** all GPs covered by this AP $\subset$ $newCovGPs$ **then**
16: remove the location of this AP from $\vec{l}$
17: **end if**
18: **end for**
19: apply the lexicographical order to the new individual $\vec{l}$

checked whether it can be removed, while still satisfying the constraint of full double coverage defined by Equation (5.7). The final new solution is reordered by following the lexicographical order (line 19).

Analogously, for memory efficiency, memory-consuming variables (such as *validGPs* in Algorithm 8) are not locally stored in each individual. Instead, they are locally generated. The huge memory taken by these variables will consequently be freed up at the end of Algorithm 8.

5.6.5 Parallel Genetic Algorithm

A GA exhibits an intrinsic characteristic of parallelism, because it does not evaluate and improve a single solution but analyzes and modifies a set of solutions simultaneously [58]. Therefore, instead of being viewed as a mono-thread algorithm, it can be seen as a "divide and conquer" algorithm, also referring to as "map and reduce". As to the "map" phase, the data space is split into smaller and independent chunks to be processed. Once the chunks are processed, partial results are collected to form up the final result, which is the "reduce" phase.

Accordingly, parallel computing (e.g., multithreading) [59] is used to shorten the runtime of large-scale optimization. The substructures of a classical GA where the parallel computing can be applied include population initialization, crossover and mutation of two individuals, and fitness calculation.

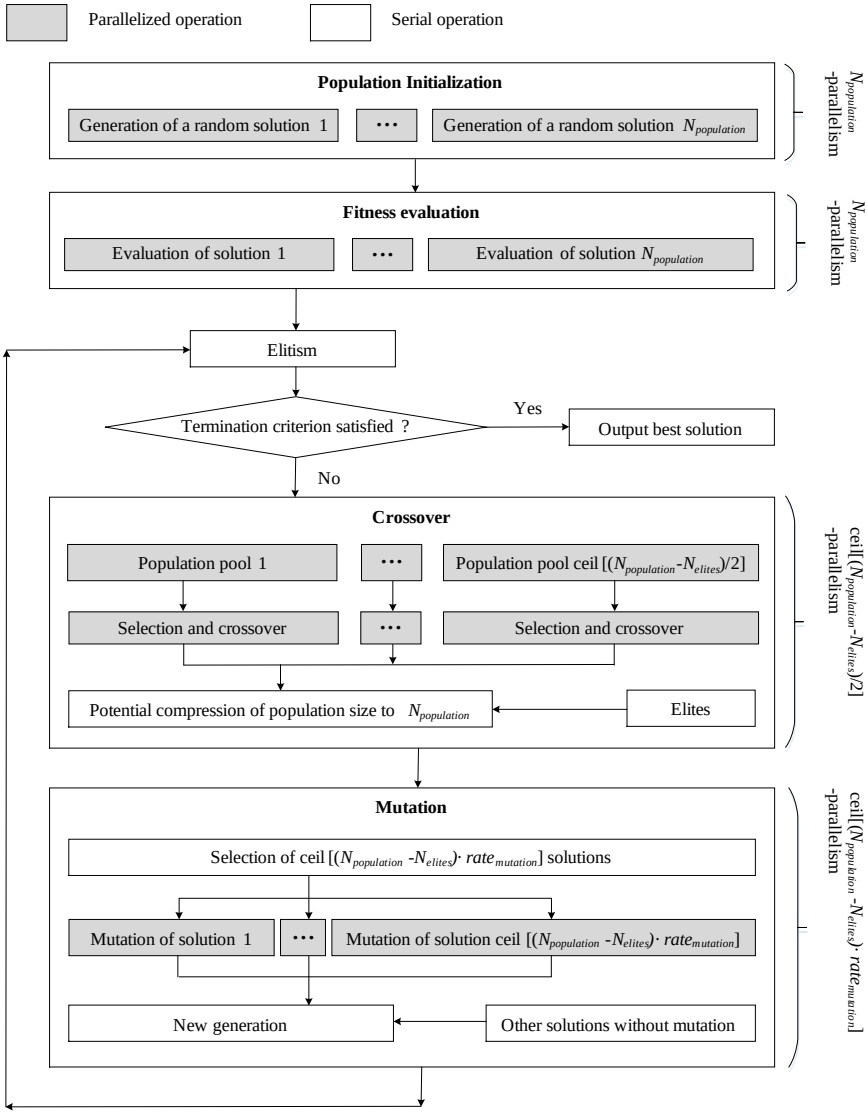


Figure 5.9: Flowchart of the proposed parallel genetic algorithm (GA)

Figure 5.9 presents a flow graph of the structure of the parallel GA for solving a large-scale OD problem. The number of parallelism depends on each genetic operator. In population initialization, the random generation of $N_{population}$ individuals is completely standalone, such that $N_{population}$ instances of Algorithm 2 can be parallelized. Similarly, $N_{population}$ fitness evaluations are parallelized at the end of a generation. To unlock the parallelism potential of crossover and mutation,

the two operators are performed based on a population, instead of two individuals and one individual, respectively, in a decent GA. For a generation, the numbers of parallel crossover and mutation operations are ceil[$(N_{population} - N_{elites})$/2] and ceil[$(N_{population} - N_{elites}) \cdot rate_{mutation}$], respectively. As all the parallelized crossover operations should simultaneously have full access to the entire generation, ceil[$(Npopulation - Nelites)$/2] population pools are created at the start of parallel crossover. Each population pool is a copy of an entire population.

5.6.6 Additional Speedup Measures

As described in the former subsections, the design of GAOD follows the idea of enhancing search efficiency and reducing the memory and the computation time as much as possible, to facilitate large-scale optimization. Next to this, additional specific speedup measures are taken on two types of calculations which are extensively employed in GAOD.

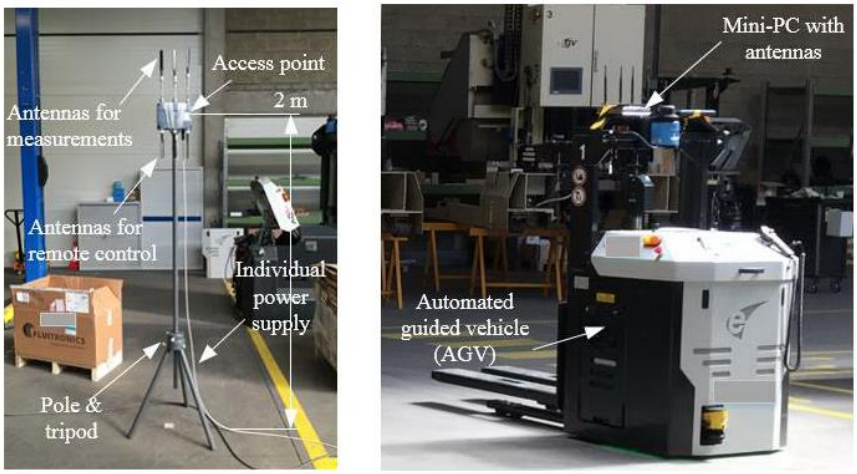
First, d_{max} is calculated by the path loss model in advance and stored as a constant, instead of repeating the same path loss calculation for millions of times during a genetic search. Second, Algorithms 6-8 extensively search the area that an AP can cover with TP_{max} as well as the area which is within d_{APmin} of an AP. Instead of a rude iteration of all GPs in the environment to find the qualified GPs, such a search is only restricted within the $d_{max} \times d_{max}$ and $d_{APmin} \times d_{APmin}$ rectangular areas which are centered at the investigated AP.

5.7 Experimental Validation

The proposed GAOD was experimentally validated in a small open environment ($\approx 10\ m \times 10\ m$) in the factory hall of an automated guided vehicle (AGV) manufacturer, in Flanders, Belgium.

5.7.1 Facilities, Configurations, and Measurements

The WLAN coverage measurement facilities include (1) 4 Siemens industrial APs (Scalance W788-2) with individual power supply, (2) a Zotac mini-PC as a wireless client that receives signals from the APs, (3) an AGV as the controllable mobile vehicle which carries the client on the top such that the shadowing effects of AGV itself is prevented, (4) 4 poles with tripods to mount the APs, and (5) a central software system that controls measurements in terms of AGV location recording, synchronization with the measurement database in the mini-PC, and design of measurement experiments, e.g., assignment of wireless signal metrics to be measured and duration of an experiment. While the fifth measurement facility was a laptop, Figure 5.10 demonstrates the first four measurement facilities.



(a) Transmiter: one of the four access points deployed in the environment

(b) Receiver: mobile client composed of an automated guided vehicle and a mini-PC with antennas

Figure 5.10: Wireless signal transmitter and receiver for the over-dimensioning experiment

As placing APs within the environment would impede the AGV's movement, the 4 APs were constrained to be placed only on the four boundaries of the environment. The minimal AP separation distance d_{APmin} was set to 5 m, such that they cannot easily be shadowed by an obstacle and that the interference is reduced between adjacent APs. Besides, 44-dB attenuators were installed to all transmit radio interfaces of the four APs, so as to mimic a larger environment that needs four APs for double full coverage. The proposed GAOD was implemented in the central measurement control system to provide an OD solution for this environment. Table 5.4 further lists the configurations for the APs, the mobile client, and the GAOD.

Upon the start of an experiment, the measurement task was remotely sent from the measurement control system to the mini-PC-based client. This client then connected to the specified AP and started to record the received signal strength and the corresponding time for each record. During an experiment, the AGV was controlled to drive around in the environment at 20 cm/s such that the AGV path swept the environment as evenly as possible. Meanwhile, the measurement control system recorded the real-time AGV 2D locations by communicating to the backend server for AGVs. At the end of an experiment, the AGV and the client stopped moving and measuring the received signal strength, respectively; the measurement control system retrieved the measured RSSI from the client and attached a 2D location to each sample by synchronization. Such an experiment was used for two objectives: (1) characterization of radio propagation in the environment which

determines the variables of the one-loss path loss model, i.e., Equation (5.5), and which leads to an empirical OD solution, (2) monitoring of the wireless coverage when the 4 APs are deployed according to the empirical OD solution.

Table 5.4: Configurations of the environment and genetic algorithm based over-dimensioning (GAOD)

Wireless configurations	
Shadowing margin (95%)	1 dB
Fading margin (99%)	0 dB
Interference margin	0 dB
AP antenna attenuation	44 dB
WLAN standard	IEEE 802.11n
Frequency band	2.4 GHz
AP height	2 m
AP only on the wall?	Yes
Minimal AP separation (d_{APmin})	5 m
Client height	1.8 m
Required physical bitrate of a client	24 Mbps
Required minimal sensitivity of a client	-79 dBm
Genetic algorithm configurations	
Population Size	100
Elitism rate	10%
Crossover rate	90%
Mutation rate	5%
Maximum iteration	30

5.7.2 Measurement Results

To build an empirical path loss model, 3745 RF power samples were collected. Regression was applied to these data to build an empirical path loss model [50] formulated by Equation (5.5). Consequently, PL0 was 39.87 and *n* was 1.78. The R-squared value was 97.4%, indicating a high fitness level of this empirical path loss model, compared to the measured data. The GAOD then used this empirical path loss model to produce an OD solution. The obtained OD solution is illustrated in Figure 5.11. The thick black lines represent the four boundaries of the environment. The 4 APs are placed on the boundary of the environment, such that two full coverage layers are envisioned to be guaranteed. Each AP has an exact 2D location (in cm), as depicted in Figure 5.11.

Figure 5.11 also exhibits the coverage of each AP. Red colors represent the area with high RF power; blue colors stand for the area with low RF power which is whereas not lower than the required minimal sensitivity (-79 dBm, Table 5.4);

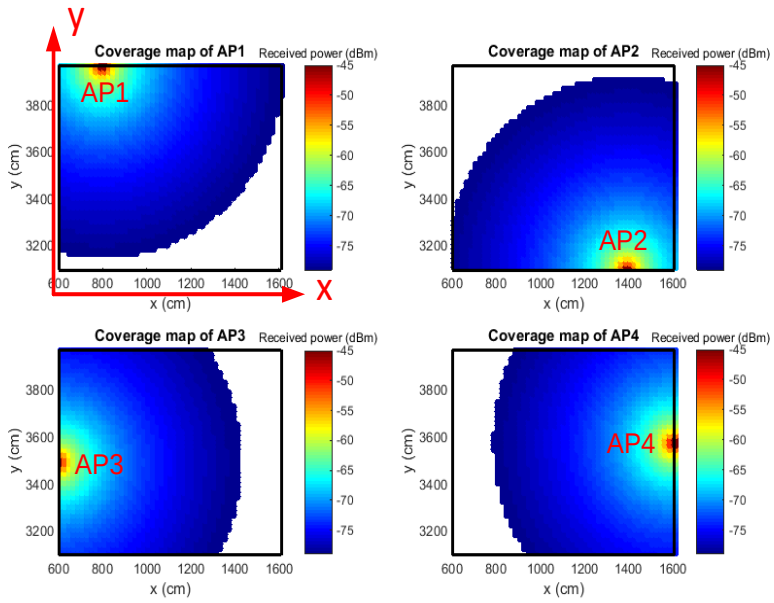


Figure 5.11: Coverage of the four over-dimensioned access points (APs), which is predicted by the path loss model. The colored area is covered by an AP, while the white area is out of coverage.

while white colors denote the area that cannot be covered by the AP. As shown, every AP cannot fully cover the environment. However, two APs can form up a complete coverage layer by combing the respective coverage, i.e., AP1 and AP2, as well as AP3 and AP4. The minimal inter-AP distance (5.13 m) is the one between AP1 and AP3. This is larger than the preset d_{APmin} (5 m). Therefore, this numerically demonstrates the effectiveness of the GAOD.

The 4 APs were then placed in the environment according to the obtained OD solution. The coverage of each AP was measured by following the experiment procedure described in Section 5.7.1. As a result, Figure 5.12 presents the received power samples that are lower than the required minimal sensitivity, i.e., the coverage hole of each AP. The coverage hole of an AP is always close to the environment boundary and on the opposite side of this AP location. The coverage power samples vary between -79 dBm and -85 dBm, which are actually below the present sensitivity (-79 dBm). Therefore, this empirically demonstrates the effectiveness of the proposed GAOD.

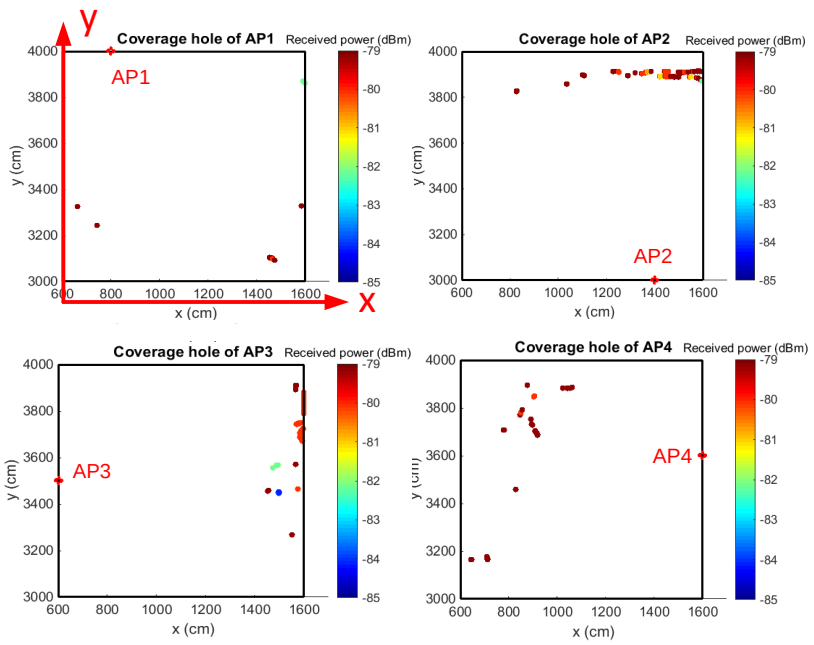


Figure 5.12: Measured coverage holes of the four over-dimensioned access points (APs). These measurements were automatically performed by an automated guided vehicle (AGV). This AGV moved around in the target environment, equipped with a wireless client to sense the RF power-based quality of service (QoS) and communicating with the central decision support system.

5.8 Numerical Experiments

The focus of numerical experiments is on the algorithmic scalability beyond the experimental validation at a small scale (Section 5.7), to adapt to the real industrial wireless deployment scale. The models and algorithms were implemented in Java. The numerical experiments were performed on a PC running 64-bit Win7 and with an Intel i5-3470 CPU (two 3.20 GHz single-thread cores) and 8 GB RAM.

5.8.1 Configurations

The two industrial indoor environments under investigation are, respectively, a factory hall of an automated guided vehicle (AGV) manufacturer and a warehouse of a car manufacturer, both located in Flanders, Belgium.

The factory hall measures $102\ m \times 24\ m$. Metal racks are placed inside for component storage. Most AGVs of varying sizes are statically placed and waiting for integration, maintenance, or shipment. Several AGVs may also be under test

Table 5.5: Numerical experiment configurations

Path loss model	
PL0	39.87 dB
n	1.78
Shadowing margin (95%)	1 dB
Fading margin (99%)	0 dB
Interference margin	0 dB
An access point (AP) as the transmitter	
Height	2 m
Gain	3 dB
WiFi standard	IEEE 802.11n
Frequency band	2.4 GHz
Maximal transmit power	7 dBm
Only on the wall?	No
Receiver of a wireless client	
Height	1.4 m
Gain	2.15 dB
Required physical bitrate	54 Mbps
Required minimal sensitivity	-68 dBpm
Environment	
Size of the factory hall (small)	$102\ m \times 24\ m$ (2600 grid points)
Size of the warehouse (large)	$415\ m \times 200\ m$ (83616 grid points)
Grid cell size (gs)	1 m
Frequency band	2.4 GHz
Antenna type	Omnidirectional
Minimal inter-AP separation	5 m
Metal rack size	$20\ m \times 3\ m \times 9\ m$
Path loss caused by one metal rack	7.37 dB
GAOD configurations	
Population size	30 (small-scale environment) 100 (large-scale environment)
Elitism rate	8%
Crossover rate	95%
Mutation rate	5%
Stop criterion	No improvement of the best fitness value during 10 consecutive cycles

by moving around. Wide WiFi coverage is needed for AGV communication and Internet access of onsite laptops.

The warehouse measures $415\ m \times 200\ m$. Metal racks are placed inside, at a height of nine meters. Wooden boxes that contain metal components are placed on the racks. Wide WiFi coverage is required to support voice picking. The pick-

ers are equipped with microphones and earphones. They communicate with the control center via WLANs to pick up and place a stuff at a specific location.

Mapping to the over-dimensioning (OD) model (Section 5.4), a metal rack in both cases is an obstacle that potentially causes evident shadowing effects to radio propagation. In the following experiments, an obstacle measures $20\ m \times 3\ m \times 9\ m$. It can be placed either horizontally (the length side is parallel to the length side of the environment) or vertically (the length side is parallel to the width side of the environment). The direction and location of an obstacle are randomly and uniformly generated in the environment. The number of racks is an input of the OD model. The grid points (GPs) that are taken up by obstacles are not considered for the path loss calculation.

The experiment parameters are shown in Table 5.5, including the path loss model, the AP transmitter, the receiver, the environment and the GAOD. All APs are powered on with maximal transmit power (TP_{max}). The grid cell size (gs) was set to one meter. It is within the distance of 10 wave lengths $(\approx 1.2\ m)$ at 2.4 GHz radio frequency band, meaning that the path loss within this distance can be considered constant without sacrificing the precision of path loss calculation [60]. The two parameters *PL0* and *n* for the one-slope path loss model were the same as these in Section 5.7. The path loss caused by a metal rack (7.37 dB) was the mean value of the measured path loss data.

Furthermore, two benchmark algorithms were used to evaluate the performance of the GAOD (genetic algorithm-based over-dimensioning): the GHOD (greedy heuristic-based over-dimensioning, Algorithm 5) and the random placement satisfying all the constraints of the OD problem (Algorithm 6). Although the latter looks simple compared to a genetic algorithm, it is a common method to deploy wireless sensors [61].

5.8.2 Results in a Small-Scale Empty Environment

The GHOD and the GAOD were first applied in the small-scale empty environment (i.e., factory hall of the AGV manufacturer), by loosening the constraint in Equation (5.9) such that no metal obstacles exist (i.e., $N_o = 0$). The performance metrics of both algorithms are shown in Table 5.6. Both algorithms satisfy the constraints of two full coverage layers and minimal AP separation (d_{APmin}) in the target factory hall. However, the GAOD outputs one AP less to solve the same OD model, and is 2.7 times faster than the GHOD.

In the solution outputted by the GHOD, the number of GPs that are covered by at least three APs is 3.3 times as the same type of number in the GAOD. This unveils an intrinsic characteristic of GAOD: it essentially minimizes the number of GPs that are covered by more than two APs, while ensuring each GP is covered by at least two APs.

Table 5.6: Algorithm performance comparison in empty environments

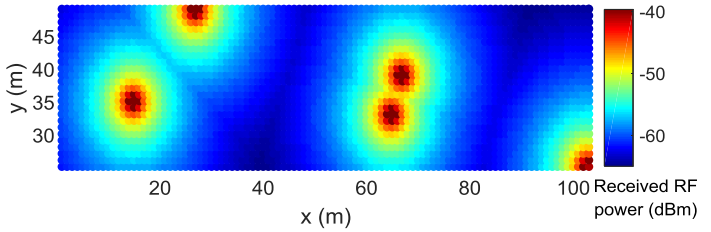
Performance metric	GHOD		GAOD		Random OD (mean/deviation)	
	S[a]	L[b]	S	L	S	L
Number of all APs	5	81	**4**	**75**	5/1	85/3
Runtime (s)	8	2633	3	19789	0/0	522/142
Percentage of GPs covered at least twice	100	100	100	100	100/0	100/0
Any AP separation within d_{APmin}?	No	No	No	No	No	No
Percentage of GPs covered more than twice	70	93	**21**	**84**	65/18	91/2

[a]Small-scale environment.
[b]Large-scale environment.

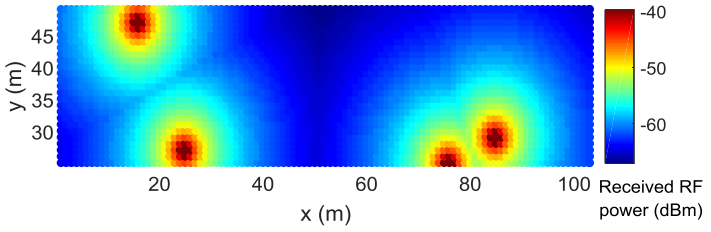
Moreover, 230 random OD solutions are generated by using Algorithm 6. As indicated in Table 5.6, on average five APs are needed with a standard deviation of one AP. This means that the median case corresponds to the GHOD, and the worst case (6 APs) outputs 50% more APs than GAOD (4 APs), which accordingly leads to about 50% more AP deployment cost than GAOD. The time to generate a random solution is negligible. This is normal since a random instance only needs to satisfy the two fundamental constraints (double full coverage and minimal AP separation) defined by Equation (5.7) and Equation (5.13), without any optimization effort. Overall, both proposed algorithms can give effective OD solutions, while GAOD is superior to GHOD in the small-scale environment regarding the number of APs that is outputted and computation time.

As a comparison, the OD solutions outputted by the GHOD, the GAOD and the random OD (Algorithm 6) are shown in Figure 5.13. The x and y axes are the length and the width of the factory hall under investigation, respectively. The highest received power of each GP is visualized. High power is represented by red, while low power is indicated by blue. As a result, the locations of the over-dimensioned APs are represented by the centers of the red dots. Figure 5.13 evidently shows that the GAOD outputs the least APs, while the random generation outputs the most APs within the same environment. In the solution of GAOD, APs tend to be evenly distributed over the environment. In the solution of random generation, APs however tend to be clustered, which also reveals why more APs are needed for satisfying the same constraints of the same OD model.

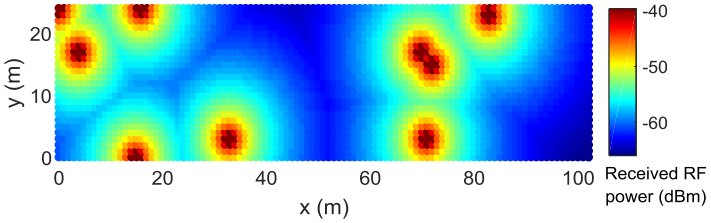
Figure 5.13 also serves as a heat map for network managers and plant managers. It vividly reveals the coverage of the whole industrial indoor environment. The minimal received power on this map is -67 dBm, which is higher than the threshold -68 dBm (Table 5.6). The minimal inter-AP separation is 9.8 m, which



(a) Solution of GHOD



(b) Solution of GAOD



(c) Random OD solution

Figure 5.13: Benchmarking in a small-scale empty environment

is larger than d_{APmin} (5 m) constrained by Equation (5.13).

5.8.3 Results in a Large-Scale Empty Environment

The GHOD and the GAOD were then performed in the large-scale empty environment (i.e., warehouse of the car manufacturer, Figure 5.14), by making loose the constraint in Equation (5.9) such that no metal obstacles exist (i.e., $N_o = 0$). The performance metrics of these two algorithms are presented in Table 5.6. Both algorithms meet the two essential constraints, i.e., double full coverage, and any AP beyond d_{APmin} of all the other APs. Most importantly, in terms of the optimization objective, the GAOD outputs six APs less. This roughly corresponds to 7.4% reduction of the network deployment cost.

The GHOD is 7.5 times faster than the GAOD. This is inverse to the phenomenon revealed in the former case. It is explained by the O(n) time complexity of GHOD, of which the fast performance shows up when the problem size grows rapidly. Nevertheless, the time taken is not a crucial factor for wireless planning,

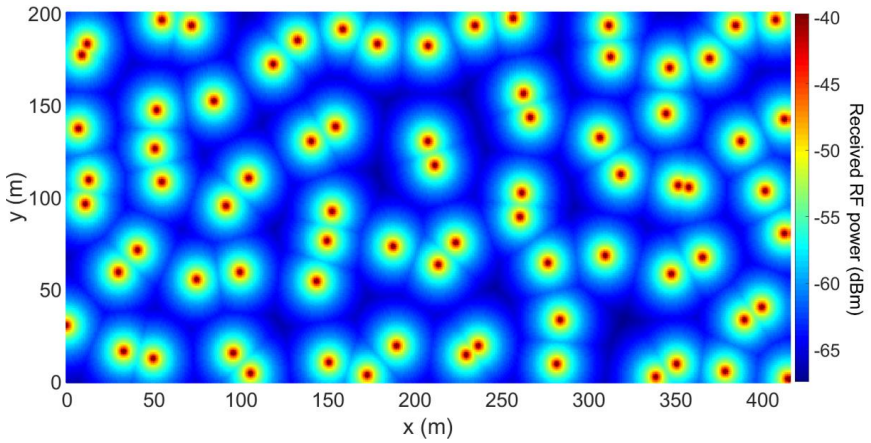


Figure 5.14: Proposed OD in a large-scale empty environment

since the planning is performed only once or at a very low frequency. Besides, the time (5.5 hours) taken by the GAOD is considered acceptable, and is significantly improved compared to 173.6 hours in the former experiment of running the d_{avg} criterion (Section 5.5).

In the OD solution given by the GHOD, 9% more GPs are covered by at least three APs. This again demonstrates the aforementioned GAOD's intrinsic characteristic of global optimization.

Furthermore, 380 random solutions are generated. As shown in Table 5.6, 85 APs on average are needed with a standard deviation of three APs. This mean AP number is 4.9% and 13.3% larger than the number of APs outputted by the GHOD and the GAOD, respectively. The best case in the random OD solution has 82 APs, which is still worse than the GHOD and the GAOD. The time for establishing a random OD solution is much shorter than the GHOD and the GAOD, which is similar as in the small scale (Section 5.8.2) due to the same reason. The percentage of GPs that are covered for more than twice is higher than the GAOD, and is at the similar level of GHOD.

The OD solution given by the GAOD is further visualized in Figure 5.14. The minimal received power on this heat map is -67.42 dBm, which is higher than the threshold. The minimal inter-AP separation among all the over-dimensioned APs is 6 m, which is higher than d_{APmin} (5 m).

5.8.4 Results in Obstructed Environments

The GHOD and GAOD algorithms were further executed in the small-scale and large-scale obstructed environments, respectively. One and ten metal racks (Table 5.5) were randomly placed in the small-scale and large-scale environments,

Table 5.7: Algorithm performance comparison in obstructed environments

Algorithm		Number of APs	Runtime (s)	Percentage of GPs covered at least by two APs
GHOD	S[a]	6	2	88
	L[b]	91	23956	91
GAOD	S	**5**	7	**66**
	L	**83**	34028	**85**
Random OD (mean/deviation)	S	7/1	0/0	75/11
	L	92/2	543/54	90/1

[a]Small-scale environment.
[b]Large-scale environment.

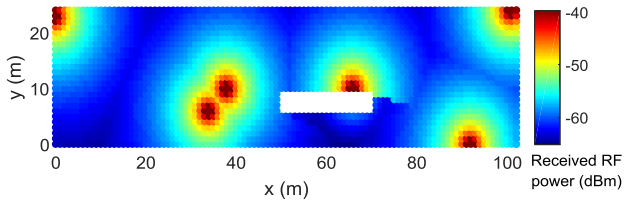
respectively. In total, 330 and 200 random OD solutions were generated in the small-scale and large-scale environment, respectively. These two large numbers of random runs aimed to guarantee the statistical performance of a random OD solution.

The coverage map of the OD solution in the small-scale obstructed environment outputted by the GHOD, the GAOD, and the random generation, is further visualized in Figure 5.15. The superiority of GAOD is clearly demonstrated, in terms of minimizing the AP number while meeting with all the constraints of the OD model.

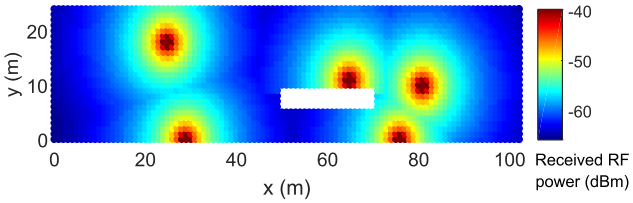
The coverage map of the OD solution in the large-scale obstructed environment outputted by the GAOD is further presented in Figure 5.16. The 10 metals racks that are randomly generated are represented by 10 white rectangles. It is observed that an increasing number of APs stay around metal racks compared to other regions without metal racks. This reveals the intrinsic property of GAOD when dealing with shadowing effects of dominant metal: additional APs are actually placed to tackle the additional path loss that is caused by these shadowing effects.

All the obtained OD solutions satisfy all the model constraints. Table 5.7 lists the other key performance metrics for comparison. All the solutions are obtained within a reasonable time span, regarding the problem size and context of wireless planning. However, the GAOD can output the lowest number of APs in both small and large environments that have metal racks. Compared to a random solution, it outputs 17%-38% and 8%-12% less APs in the small-scale and large-scale environment, respectively. The number of APs that are over-dimensioned by the GHOD is intermediate, compared with the other two algorithms.

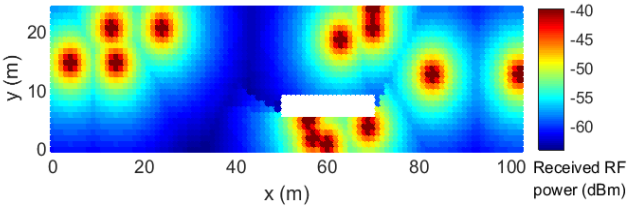
Moreover, the percentage of GPs that are covered more than twice exhibits a similar performance trend: the GAOD achieves the lowest percentage, while a random solution and the GAOD have an evidently higher percentage. In a small-scale obstructed environment, the percentage of GPs that are covered at least twice



(a) Solution of GHOD



(b) Solution of GAOD



(c) Random OD solution

Figure 5.15: Over-dimensioning solution comparison in a small-scale obstructed environment (the white rectangle represents a metal rack)

is 22% less in GAOD than that in GHOD, and is in average 9% less than that of a random solution. In a large-scale obstructed environment, the percentage of GAOD is 6% less than that of GHOD, and is in average 5% less than that of a random solution. The difference between the GAOD and the other two algorithms reduces in a large-scale environment. One important reason for this phenomenon is that more metal racks are placed in the large-scale environment, which causes more shadowing effects. Consequently, more additional APs are needed to specifically tackle these shadowing effects, while increasing the coverage layer number of the other GPs that are already covered twice.

A performance comparison is made between the OD solutions which are outputted by the same algorithm in the same environment without and with metal, respectively. In the small-scale environment, more APs are outputted by the three algorithms if dominant metal is present. The AP number increasing rate of a random solution is the highest, at 40% in average. This is explained by a lack of effective optimization measures in random solution generation. In reverse, this demonstrates the effectiveness of GHOD and GAOD, in terms of minimizing the

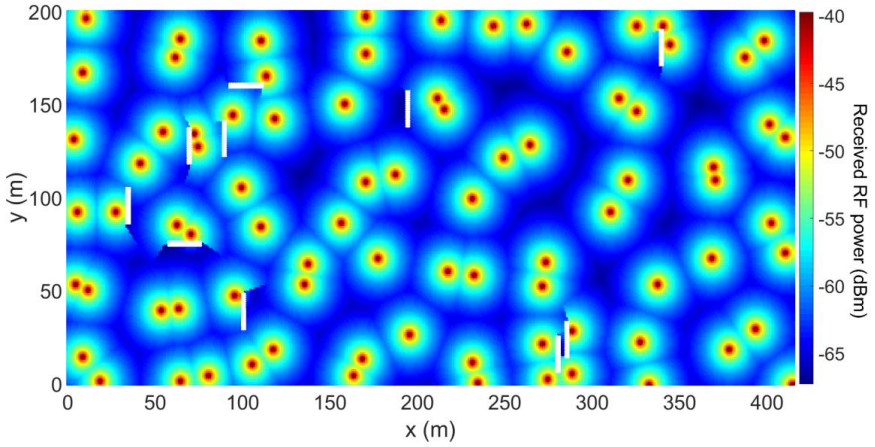


Figure 5.16: Over-dimensioning solution given by the GAOD in a large-scale obstructed environment (the 10 white rectangles represent 10 metal racks).

AP number. In the large-scale environment, similarly, more APs are needed under the presence of dominant metal. The AP number increasing rate of the three algorithms is on the same level around 10%, while the rate of GHOD is the highest, at 12%. This shows that the optimization performance of GHOD is worse than that of GAOD.

5.8.5 Deployment Cost Analysis

To analyze the deployment cost performance of the proposed GAOD, each of the following 6 algorithms went through 10 independent runs in small and large obstructed environments: GHOD-1, GAOD-1, random OD-1, GHOD, GAOD, and random OD. The former 3 algorithms are modified versions of the latter 3 algorithms, respectively, such that only one full coverage layer is ensured in a target environment, i.e., Equation (5.7) is replaced by Equation (5.16) while the objective and other constraints remain in the proposed OD model (Section 5.4).

$$\sum_{j=1}^{N_{AP}} \alpha_{ij} \geq 1, \forall i \in I \tag{5.16}$$

The number of APs is taken as the deployment cost indicator, as it is closely linked to the cost of deploying a homogeneous IWLAN. The cost gaps between deployment solutions outputted by different algorithms is thereby characterized by the gap between the numbers of APs.

As presented in Table 5.8, among the 6 algorithms, the most economical de-

Table 5.8: Deployment cost (number of APs $\times$ AP unit price) comparison of 6 different algorithms in obstructed environments (metal rack placement remains identical to that in Figure 5.15 and Figure 5.16)

Algorithm	Number of APs (mean$\pm$standard deviation)		Cost gap compared to the proposed GAOD (positive: more expensive; negative: cheaper)	
	Small-scale environment	Large-scale environment	Small-scale environment	Large-scale environment
GHOD-1[a]	3 ± 0	46 ± 0	-25%	-39%
GAOD-1[b]	3 ± 0	44 ± 1	-25%	-41%
Random OD-1[c]	4 ± 1	49 ± 2	0	-35%
GHOD	5 ± 0	81 ± 0	25%	8%
GAOD	4 ± 0	75 ± 1	0	0
Random OD	5 ± 1	85 ± 3	25%	13%

[a]Modified GHOD version that ensures only one full coverage layer.
[b]Modified GAOD version that ensures only one full coverage layer.
[c]Modified random OD version that ensures only one full coverage layer.

ployment solution is provided by the GAOD-1 (25% and 41% cheaper than the GAOD in small and large obstructed environments, respectively). As GAOD-1 only guarantees 1 full coverage layer, the proposed GAOD is cost-effective in the sense that it at most requires an additional 41% cost to deploy an IWLAN that double cover a target environment. Among the 3 algorithms that establish two full coverage layers (GHOD, GAOD, and random OD), the GAOD also remains the most cost-effective, as is needs an 8% - 25% cheaper deployment cost while ensuring two full coverage layers. Therefore, the proposed GAOD is able to provide factories cost-effective IWLAN deployment solutions.

Furthermore, the behavioral peculiarities of GHOD, GAOD and random OD are clearly exhibited in Table 5.8. The GHOD is static with a constant number of APs (zero standard deviation) in different runs, while the GAOD and random OD are stochastic. An exception is observed in the GAOD, which also outputs a constant number of APs in the small obstructed environment. This could be explained by the small problem size, where the GAOD can rapidly converge toward the global optimal or the nearly-optimal region in the same search space despite different runs.

5.8.6 Discussions

Several implications are further revealed based on the former experiments. (1) As revealed by the comparison of Table 5.6 and Table 5.7, whether obstacles are considered will evidently influence the deployment cost of an IWLAN. It is also natural to reason that an increasing number of obstacles in the OD model will lead to more APs and thus a high deployment cost. Therefore, there should be a trade-off relationship between the robustness and deployment cost of an IWLAN.

Although the layout of an industrial environment may alter causing a varying density of obstacles, it would be practical to consider a moderate density of obstacles when planning an IWLAN. Consequently, the robustness and cost performance of the deployed IWLAN can both achieve an acceptable level. (2) Despite the optimization efforts of the proposed GAOD in the deployment phase, the percentage of regions that are covered over twice is still significant, especially in a large industrial indoor environment (84%, Table 5.6). This implies a high potential to further reduce the transmit power of deployed APs at the operational phase. In this way, the regions that are covered more than twice and the produced interference can be further reduced.

In addition to the demonstrated effectiveness and efficiency in deploying an economical robust IWLAN, the proposed GAOD is compared with the recent wireless network planning studies (Table 5.1). It has two major advantages. (1) The underlying OD model fills the gap in deploying a robust IWLAN. In contrast to the commonly used Boolean disk model, it explicitly investigates the 3D shadowing effects of dominant obstacles that are prevalent in an industrial indoor environment. These shadowing effects are integrated in an empirical one-slope path loss model, which enables simple yet accurate coverage prediction. Besides double full coverage layers, an AP spatial separation is ensured to avoid that APs covering the same region are simultaneously shadowed by an obstacle. (2) Compared to most recent studies that are limited to a small or medium problem size, the proposed GAOD can additionally perform large-scale optimizations, where a large environment size, a higher spatial resolution, and a highly constrained deployment model are jointly involved. This complexity complies with the real requirement for deploying an IWLAN.

On the other hand, the proposed GAOD has two limitations (Table 5.1). (1) It focuses on homogeneous wireless network planning, while some recent studies investigate heterogeneous planning, e.g., planning of WLAN and LTE [24]. (2) The interference issue is not investigated. It would be of practical importance to limit or reduce interference in a dense wireless network. Furthermore, it needs dedicated measurement setups to empirically characterize a target environment, in order to deploy such a dense IWLAN by using the GAOD. Prior to empirical measurements, calibration should also be performed to ensure that the received power of a client is accurately sampled [6].

5.9 Conclusions and Future Work

Although wireless technologies are penetrating into the manufacturing industry, the existing research on wireless local area network (WLAN) planning is still mainly limited to small office environments. Consequently, the one coverage layer

provided by these WLAN planning approaches is vulnerable to the shadowing effects of prevalent metal obstacles in harsh industrial indoor environments. To fill this gap, this chapter investigates an over-dimensioning (OD) problem where two full coverage layers can be created at a large industrial scale for robust industrial wireless coverage. Although the second coverage layer serves as redundancy against shadowing, the deployment cost can be reduced by minimizing the number of access points (APs), while respecting the practical constraint of a minimal inter-AP spatial separation.

A genetic algorithm based OD (GAOD) algorithm is proposed to solve this problem. To enable large-scale industrial WLAN (IWLAN) planning, solution encoding, initial population generation, crossover and mutation are designed, such that the required computation time and memory are minimized. A greedy heuristic, named GHOD, is also proposed for benchmarking the performance of GAOD. Furthermore, a systematic method is proposed to promote robust wireless coverage by making use of commercial off-the-shelf wireless devices. It includes three sequential components: mobile measurement, OD, and reconfiguration. While this chapter investigates OD, Chapter 6 will study how to perform optimized reconfiguration of dense industrial wireless networks to reactively handle coverage holes while reducing interference.

A factory hall $(102\ m \times 24\ m)$ of an automated guided vehicle (AGV) manufacturer and a warehouse $(415\ m \times 200\ m)$ of a car manufacturer in Belgium are investigated as two case studies, i.e., small-scale and large-scale industrial indoor environment, respectively. Empirically, the feasibility and effectiveness of the OD model and GAOD is validated by measurements in a $10\ m \times 10\ m$ empty environment in the factory hall of the AGV manufacturer. Numerically, the effectiveness of GAOD and GHOD is extensively demonstrated in the two investigated environments, without and with the presence of metal racks, in comparison to the random OD solution generation. Compared to GAOD and GHOD, the random OD solution generation outputs up to 60% and 33% more APs, respectively. The superiority of GAOD, compared to GHOD, is demonstrated by the fact that GAOD outputs up to 20% less APs for the same OD problem in a reasonable time span.

The outcome GAOD algorithm can help network managers and plant managers to automatically plan an IWLAN which has high availability under the presence of dominant metal in the environment. Moreover, it can easily be extended to plan other robust wireless networks such as wireless sensor networks.

This presented work can have fourfold future extensions. (1) A dedicated indicator may be introduced to quantify the robustness of deployed wireless networks in harsh industrial environments, regardless of the wireless network type. For instance, such a robustness indicator may be created based on the actual monitored coverage which is correlated with the time and the upper-layer industrial application. (2) The proposed OD model and GAOD algorithm can be extended from

double to triple or even more coverage layers. This is especially important for localization and wireless sensor networks which may alternatively switch on/off each coverage layer to remain robust while conserving energy. (3) The interference of a dense network can be explicitly considered in the OD model as either a constraint or an objective for optimization. (4) Heterogeneous wireless networks can be simultaneously planned by considering other wireless technologies besides a WLAN.

References

[1] R. Drath and A. Horch. *Industrie 4.0: Hit or Hype? [Industry Forum]*. IEEE Industrial Electronics Magazine, 8(2):56–58, June 2014.

[2] W. Ikram and N. F. Thornhill. *Wireless communication in process automation: A survey of opportunities, requirements, concerns and challenges*. In UKACC International Conference on Control 2010, pages 1–6, Sept 2010.

[3] Cisco and Rockwell Automation. *Wireless Design Considerations for Industrial Applications: Design and Deployment Guide*. Technical report, Cisco and Rockwell Automation, 2014.

[4] Xiaomin Li, Di Li, Jiafu Wan, Athanasios V. Vasilakos, Chin-Feng Lai, and Shiyong Wang. *A review of industrial wireless networks in the context of Industry 4.0*. Wireless Networks, 23(1):23–41, Jan 2017.

[5] D. Plets, E. Tanghe, A. Paepens, L. Martens, and W. Joseph. *WiFi network planning and intra-network interference issues in large industrial warehouses*. In 2016 10th European Conference on Antennas and Propagation (EuCAP), pages 1–5, April 2016.

[6] X. Gong, J. Trogh, Q. Braet, E. Tanghe, P. Singh, D. Plets, J. Hoebeke, D. Deschrijver, T. Dhaene, L. Martens, and W. Joseph. *Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories*. IET Science, Measurement Technology, 10(4):375–382, 2016.

[7] E. Tanghe, D. P. Gaillot, M. Liénard, L. Martens, and W. Joseph. *Experimental Analysis of Dense Multipath Components in an Industrial Environment*. IEEE Transactions on Antennas and Propagation, 62(7):3797–3805, July 2014.

[8] Johan Åkerberg, Mikael Gidlund, Tomas Lennvall, Krister Landerns, and Mats Björkman. *Design Challenges and Objectives in Industrial Wireless Sensor Networks*. In Industrial Wireless Sensor Networks - Applications, Protocols, and Standards, Industrial Electronics. CRC Press, April 2013.

[9] Wen-Hwa Liao, Yucheng Kao, and Ying-Shan Li. *A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks*. Expert Systems with Applications, 38(10):12180 – 12188, 2011.

[10] Wen-Hwa Liao, Yucheng Kao, and Ru-Ting Wu. *Ant colony optimization based sensor deployment protocol for wireless sensor networks*. Expert Systems with Applications, 38(6):6599 – 6605, 2011.

[11] Ning Liu, David Plets, Kris Vanhecke, Luc Martens, and Wout Joseph. *Wireless indoor network planning for advanced exposure and installation cost minimization*. EURASIP Journal on Wireless Communications and Networking, 2015(1):199, Aug 2015.

[12] K. Jaffres-Runser, J. m. Gorce, and S. Ubeda. *Multiobjective QoS-Oriented Planning for Indoor Wireless LANs*. In IEEE Vehicular Technology Conference, pages 1–5, Sept 2006.

[13] Ons Abdelkhalek, Saoussen Krichen, and Adel Guitouni. *A genetic algorithm based decision support system for the multi-objective node placement problem in next wireless generation network*. Applied Soft Computing, 33(Supplement C):278 – 291, 2015.

[14] X. Liu. *A deployment strategy for multiple types of requirements in wireless sensor networks*. IEEE Transactions on Cybernetics, 45(10):2364–2376, Oct 2015.

[15] K. Mukherjee, S. Gupta, A. Ray, and T. A. Wettergren. *Statistical-Mechanics-Inspired Optimization of Sensor Field Configuration for Detection of Mobile Targets*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(3):783–791, June 2011.

[16] Suneet Kumar Gupta, Pratyay Kuila, and Prasanta K. Jana. *Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks*. Computers & Electrical Engineering, 56(Supplement C):544 – 556, 2016.

[17] Maher Rebai, Matthieu Le berre, Hichem Snoussi, Faicel Hnaien, and Lyes Khoukhi. *Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks*. Computers & Operations Research, 59(Supplement C):11 – 21, 2015.

[18] C. P. Chen, S. C. Mukhopadhyay, C. L. Chuang, T. S. Lin, M. S. Liao, Y. C. Wang, and J. A. Jiang. *A Hybrid Memetic Framework for Coverage Optimization in Wireless Sensor Networks*. IEEE Transactions on Cybernetics, 45(10):2309–2322, Oct 2015.

[19] J. H. Park. *All-Terminal Reliability Analysis of Wireless Networks of Redundant Radio Modules*. IEEE Internet of Things Journal, 3(2):219–230, April 2016.

[20] Y. D. Lin, S. L. Tsao, S. L. Chang, S. Y. Cheng, and C. Y. Ku. *Design issues and experimental studies of wireless LAN Mesh*. IEEE Wireless Communications, 17(2):32–40, April 2010.

[21] M. Rizzi, S. Rinaldi, P. Ferrari, A. Flammini, D. Fontanelli, and D. Macii. *Enhancing Accuracy and Robustness of Frequency Transfer Using Synchronous Ethernet and Multiple Network Paths*. IEEE Transactions on Instrumentation and Measurement, 65(8):1926–1936, Aug 2016.

[22] Mohammed Abo-Zahhad, Sabah M. Ahmed, Nabil Sabor, and Shigenobu Sasaki. *Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm*. Computers & Electrical Engineering, 43(Supplement C):76 – 89, 2015.

[23] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki. *Utilisation of multi-objective immune deployment algorithm for coverage area maximisation with limit mobility in wireless sensors networks*. IET Wireless Sensor Systems, 5(5):250–261, 2015.

[24] Sotirios K. Goudos, David Plets, Ning Liu, Luc Martens, and Wout Joseph. *A multi-objective approach to indoor wireless heterogeneous networks planning based on biogeography-based optimization*. Computer Networks, 91(Supplement C):564 – 576, 2015.

[25] Ning Liu, David Plets, Sotirios K. Goudos, Luc Martens, and Wout Joseph. *Multi-objective network planning optimization algorithm: human exposure, power consumption, cost, and capacity*. Wireless Networks, 21(3):841–857, Apr 2015.

[26] A. Jedda and H. T. Mouftah. *Decentralized RFID Coverage Algorithms With Applications for the Reader Collisions Avoidance Problem*. IEEE Transactions on Emerging Topics in Computing, 4(4):502–515, Oct 2016.

[27] Lin Tang, Hui Cao, Li Zheng, and Ningjian Huang. *RFID network planning for wireless manufacturing considering the detection uncertainty*. IFAC-PapersOnLine, 48(3):406 – 411, 2015. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[28] Tao Zhang and Jing Liu. *An efficient and fast kinematics-based algorithm for RFID network planning*. Computer Networks, 121:13 – 24, 2017.

[29] Massimo Vecchio and Roberto López-Valcarce. *Improving area coverage of wireless sensor networks via controllable mobile nodes: A greedy approach*. Journal of Network and Computer Applications, 48(Supplement C):1 – 13, 2015.

[30] Jie Tian, Xiaoyuan Liang, and Guiling Wang. *Deployment and reallocation in mobile survivability-heterogeneous wireless sensor networks for barrier coverage*. Ad Hoc Networks, 36(Part 1):321 – 331, 2016.

[31] Subir Halder and Sipra Das Bit. *Design of an Archimedes' spiral based node deployment scheme targeting enhancement of network lifetime in wireless sensor networks*. Journal of Network and Computer Applications, 47(Supplement C):147 – 167, 2015.

[32] Fatih Senel, Kemal Akkaya, Melike Erol-Kantarci, and Turgay Yilmaz. *Self-deployment of mobile underwater acoustic sensor networks for maximized coverage and guaranteed connectivity*. Ad Hoc Networks, 34(Supplement C):170 – 183, 2015. ADVANCES IN UNDERWATER COMMUNICATIONS AND NETWORKS.

[33] Xuemei Sun, Yiming Zhang, Xu Ren, and Ke Chen. *Optimization deployment of wireless sensor networks based on culture–ant colony algorithm*. Applied Mathematics and Computation, 250(Supplement C):58 – 70, 2015.

[34] F. Barac, S. Caiola, M. Gidlund, E. Sisinni, and T. Zhang. *Channel Diagnostics for Wireless Sensor Networks in Harsh Industrial Environments*. IEEE Sensors Journal, 14(11):3983–3995, Nov 2014.

[35] F. Barac, M. Gidlund, and T. Zhang. *LPED: Channel diagnostics in WSN through channel coding and symbol error statistics*. In 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pages 1–6, April 2014.

[36] F. Barac, M. Gidlund, and T. Zhang. *Scrutinizing Bit- and Symbol-Errors of IEEE 802.15.4 Communication in Industrial Environments*. IEEE Transactions on Instrumentation and Measurement, 63(7):1783–1794, July 2014.

[37] S. Savazzi, V. Rampa, and U. Spagnolini. *Wireless Cloud Networks for the Factory of Things: Connectivity Modeling and Layout Design*. IEEE Internet of Things Journal, 1(2):180–195, April 2014.

[38] G. Cena, L. Seno, A. Valenzano, and C. Zunino. *On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Industrial Applications*. IEEE Transactions on Industrial Informatics, 6(3):425–437, Aug 2010.

[39] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino. *An enhanced MAC to increase reliability in redundant Wi-Fi networks*. In 2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014), pages 1–10, May 2014.

[40] S. Vitturi, L. Seno, F. Tramarin, and M. Bertocco. *On the Rate Adaptation Techniques of IEEE 802.11 Networks for Industrial Applications*. IEEE Transactions on Industrial Informatics, 9(1):198–208, Feb 2013.

[41] H. Zhang, P. Soldati, and M. Johansson. *Performance Bounds and Latency-Optimal Scheduling for Convergecast in WirelessHART Networks*. IEEE Transactions on Wireless Communications, 12(6):2688–2696, June 2013.

[42] X. Jin, J. Wang, and P. Zeng. *End-to-end delay analysis for mixed-criticality WirelessHART networks*. IEEE/CAA Journal of Automatica Sinica, 2(3):282–289, July 2015.

[43] D. Herrero-Perez and H. Martinez-Barbera. *Modeling Distributed Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems*. IEEE Transactions on Industrial Informatics, 6(2):166–180, May 2010.

[44] Juan M. Novas and Gabriela P. Henning. *Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming*. Expert Systems with Applications, 41(5):2286 – 2299, 2014.

[45] Pieter Becue, Bart Jooris, Vincent Sercu, Stefan Bouckaert, Ingrid Moerman, and Piet Demeester. *Remote control of robots for setting up mobility scenarios during wireless experiments in the IBBT w-iLab.t*, pages 425–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[46] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene, and Karel Crombecq. *A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design*. J. Mach. Learn. Res., 11:2051–2055, August 2010.

[47] D. Deschrijver, F. Vanhee, D. Pissoort, and T. Dhaene. *Automated near-field scanning algorithm for the EMC analysis of electronic devices*. IEEE Transactions on Electromagnetic Compatibility, 54(3):502–510, June 2012.

[48] Sam Aerts, Dirk Deschrijver, Wout Joseph, Leen Verloock, Francis Goeminne, Luc Martens, and Tom Dhaene. *Exposure assessment of mobile phone base station radiation in an outdoor environment using sequential surrogate modeling*. Bioelectromagnetics, 34(4):300–311, 2013.

[49] David Plets, Wout Joseph, Kris Vanhecke, and Luc Martens. *Exposure optimization in indoor wireless networks by heuristic network planning*. Progress In Electromagnetics Research, 139:445–478, 2013.

[50] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme. *The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz*. IEEE Transactions on Wireless Communications, 7(7):2740–2751, July 2008.

[51] Iana Siomina, Peter Värbrand, and Di Yuan. *Pilot power optimization and coverage control in WCDMA mobile networks*. Omega, 35(6):683 – 696, 2007. Special Issue on Telecommunications Applications.

[52] Wei-Chieh Ke, Bing-Hong Liu, and Ming-Jer Tsai. *The critical-square-grid coverage problem in wireless sensor networks is NP-Complete*. Computer Networks, 55(9):2209 – 2220, 2011.

[53] W. C. Ke, B. H. Liu, and M. J. Tsai. *Constructing a Wireless Sensor Network to Fully Cover Critical Grids by Deploying Minimum Sensors on Grid Points Is NP-Complete*. IEEE Transactions on Computers, 56(5):710–715, May 2007.

[54] Mohammad Nabi Omidvar, Xiaodong Li, and Ke Tang. *Designing benchmark problems for large-scale continuous optimization*. Information Sciences, 316(Supplement C):419 – 436, 2015. Nature-Inspired Algorithms for Large Scale Global Optimization.

[55] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[56] Y. Yoon and Y. H. Kim. *An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks*. IEEE Transactions on Cybernetics, 43(5):1473–1483, Oct 2013.

[57] Yu-Shan Cheng, Man-Tsai Chuang, Yi-Hua Liu, Shun-Chung Wang, and Zong-Zhen Yang. *A particle swarm optimization based power dispatch algorithm with roulette wheel re-distribution mechanism for equality constraint*. Renewable Energy, 88(Supplement C):58 – 72, 2016.

[58] Yong Ming Wang, Hong Li Yin, and Jiang Wang. *Genetic algorithm with new encoding scheme for job shop scheduling*. The International Journal of Advanced Manufacturing Technology, 44(9):977–984, Oct 2009.

[59] Wen mei Hwu. *What is ahead for parallel computing*. Journal of Parallel and Distributed Computing, 74(7):2574 – 2581, 2014. Special Issue on Perspectives on Parallel and Distributed Processing.

[60] Simon R. Saunders and Alejandro Aragan-Zavala. *Antennas and Propagation for Wireless Communication Systems*, chapter 5. Wiley, 2005.

[61] Aarti Jain and B.V. Ramana Reddy. *Eigenvector centrality based cluster size control in randomly deployed wireless sensor networks*. Expert Systems with Applications, 42(5):2657 – 2669, 2015.

6

Transmit Power Control of Dense Industrial Wireless Networks

The industrial wireless local area network (IWLAN) is increasingly dense, due to not only the penetration of Internet of Things (IoT) to shop floors and warehouses, but also the rising need of redundancy for robust wireless coverage. Instead of simply powering on all access points (APs), there is an unavoidable need to dynamically control the transmit power of APs on a large scale, in order to minimize the interference and adapt the coverage to the latest shadowing effects of dominant obstacles in an industrial indoor environment. To fulfill this need, this chapter formulates a transmit power control (TPC) model that enables both powering on/off APs and transmit power control of each AP that is powered on. This TPC model uses an empirical one-slope path loss model considering three-dimensional obstacle shadowing effects, to enable accurate yet simple coverage prediction. An efficient genetic algorithm (GA), named GATPC, is designed to solve this TPC model even on a large scale. To this end, it leverages repair mechanism-based population initialization, crossover and mutation, parallelism as well as dedicated speedup measures. The GATPC was experimentally validated in a small-scale IWLAN that is deployed a real industrial indoor environment. It was further numerically

demonstrated and benchmarked in both small- and large-scale environments, regarding the effectiveness and the scalability of TPC. Moreover, a sensitivity analysis was performed to reveal the produced interference and the qualification rate of GATPC (in satisfying the constraint of the required coverage rate in the TPC model) in function of the varying target coverage percentage as well as the number and the placement direction of dominant obstacles. This chapter also serves as a detailed presentation of the "reconfiguration" component of the systematic method proposed in Chapter 5 for robust wireless coverage in harsh industrial indoor environments.

6.1 Introduction

As introduced in Chapter 5, the IoT or wireless communication technologies are penetrating into the manufacturing sector. An IWLAN is thus emerging as a basic infrastructure for manufacturing operations. For instance, production cell controllers can connect to other intelligent devices such as robot arms via an IWLAN on the shop floor [1], in order to realize agile production. The other industrial operations that increasingly rely on IWLANs are in the domain of intra-factory transportation by automated guided vehicles (AGVs), video monitoring, process monitoring, etc. Compared to other wireless technologies that are options for industrial applications (e.g., Bluetooth and ZigBee), an IWLAN has the advantages of low cost, high data rate and considerable coverage distance (Section 6.6.8).

However, as pointed by Chapter 5, a typical industrial indoor environment is harsh in terms of radio propagation. Besides the reason explained in Section 5.1, an industrial indoor layout may occasionally be altered with the prevalence of flexible manufacturing [2]. These dynamic shadowing effects make it increasingly difficult to maintain the expected wireless coverage in a target industrial environment. Furthermore, an IWLAN is denser compared to a public WLAN. This is not only due to the large size of an industrial indoor environment, but also driven by the increasing industrial need for redundant coverage to ensure high network availability [3]. Therefore, it is of strategic importance to conceive a TPC method to dynamically change the configuration of a dense IWLAN considering these shadowing effects, in order to guarantee robust wireless connection of personnel, machines, materials and products on a large scale.

6.1.1 Cell Breathing

Cell breathing by TPC is a well-known concept in cellular networks [4, 5]. For instance, authors in [5] investigated a problem of minimizing total WCDMA pilot power subject to a coverage constraint. A WCDMA cell shrinks or expands the coverage following the trade-off between the power consumption and the coverage

rate. Comparatively, TPC of WLANs can only be found in a limited number of studies, although dense WLANs are showing up their application significance [6, 7]. A concept of resource on demand was proposed in [8] and demonstrated in [9], where redundant APs are powered off when they are detected to remain idle according to the volume and location of user's capacity demand. However, the idea of TPC beyond simple powering-on/off was only highlighted and not investigated in these studies [8, 9].

6.1.2 Propagation Model

Empirical radio propagation or path loss modeling is essential for the coverage calculation in TPC. Power management algorithms were proposed in [10] to control the coverage of APs. However, without using any path loss model, the authors assumed that the received power of a client is proportional to the transmission power of the connected AP. Analogously, a lack of proper path loss modeling is observed in [11]. While a TPC scheme was proposed, only a linear approximation was assumed between the AP transmission power and received signal strength of a client.

The classical Boolean disk model is widely used to calculate coverage in WSN coverage related optimization problems [12, 13]. It is simple, only considering a circular area within which all grid points are covered. But its application to the IWLAN coverage related optimization problems could oversimplify the problem and degrade the the quality of the optimized solution, since it ignores the obstacle shadowing effects and cannot calculate the exact received RF (radio frequency) power of a grid point (GP) in the target environment. This RF power is further linked to interference estimation, which is an indispensable concern for dense WLANs [13]. On the other hand, it is costly and time consuming to undertake a complete site survey, in order to capture the actual coverage and interference. As highlighted in [13], a combination of site survey and planning algorithm design is a good method to reduce the required measurements without compromising much the coverage prediction.

6.1.3 Large-Scale Optimization

While large-scale optimization is increasingly desired [14, 15], most research on coverage optimization problems neglects the scalability of an optimization algorithm [12, 16–19]. Large-scale problems are characterized in at least one of the following dimensions [20]. Firstly, the search space exponentially grows with the increasing number of decision variables. Secondly, the properties of the search space may change as the number of dimensions rises. Thirdly, the fitness evaluation is expensive. Fourthly, strong interaction exists between variables. While a large

problem size is involved in the dense and robust industrial wireless network planning problem (Chapter 5), the factors that may make a TPC problem large-scale are illustrated as (1) the environment size, (2) the number of APs, (3) the number of coverage layers (or k-coverage), (4) the complexity of coverage calculation which is fundamentally based on a path loss model. To solve large-scale optimization problems, metaheuristics are extensively recognized as effective approaches [21], among which a genetic algorithm (GA) is an important method [21, 22].

6.1.4 Contributions

This chapter investigates a large-scale TPC problem for dense IWLANs. The contributions of this chapter are threefold. (1) The proposed TPC model encompasses both transmit power control and powering-on/off mechanisms. An empirical one-slope path loss model is introduced for precise yet simple coverage calculation, including the three-dimensional (3D) obstacle loss which is prevalent in harsh industrial indoor environments. (2) An efficient GA, named GATPC, is proposed to solve this TPC model on a large scale. It leverages repair mechanism-based GA operators (including population initialization, crossover, and mutation), parallelism as well as dedicated speedup measures to achieve large-scale optimization. (3) The GATPC is both empirically validated in a small-scale real industrial indoor environment and demonstrated in extensive numerical experiments regarding its effectiveness and scalability, sensitivity analysis, and benchmarking.

The rest of this chapter is organized as follows. Section 6.2 briefly presents the entire method for robust industrial indoor wireless coverage. Section 6.3 mathematically formulates this TPC problem. Section 6.4 proposes the GATPC algorithm to solve this TPC model. Section 6.5 validates the GATPC in a small empty industrial environment. Section 6.6 performs numerical experiments, benchmarking, and sensitivity analysis of GATPC. Section 6.7 draws conclusions.

6.2 Method Overview

While Chapter 5 presented the systematic method to deploy and maintain robust wireless coverage in harsh industrial indoor environments, this chapter introduces this method from the perspective of cyber-physical systems (CPSs) (Figure 6.1). Therefore, the implementation of this entire method (Chapter 5 and Chapter 6) will become more concrete and the work of this chapter will be more clearly positioned in the entire method.

As depicted in Figure 6.1, the systematic method presented in Chapter 5 can be implemented in three sub-systems: 1) measurement control (blue part), 2) monitoring (orange part), and 3) smart switch (green part). The work presented by Chapter 5 and Chapter 6 is presented by the “planning & optimization” block in

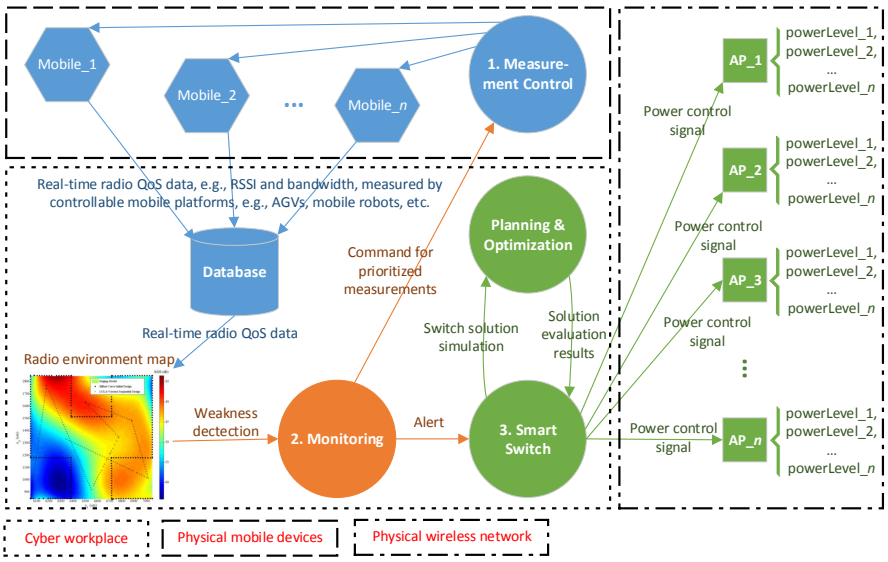


Figure 6.1: Overview of the cyber-physical system for maintaining robust wireless coverage in industrial indoor environments: 1) measurement control (blue part), 2) monitoring (orange part), and 3) smart switch (green part). This provides a complete implementation scheme for the method introduced in Chapter 5. This also demonstrates the interaction between the cyber and physcial space.

Figure 6.1. This block serves as an automated decision making engine not only for robust wireless network planning prior to the network deployment, but also for optimized and dynamic reconfiguration during the usage phase of the deployed network.

The "measurement control" block (Figure 6.1) consists of two major functions: 1) mobility control, and 2) wireless measurement control. The former function designs and provides mobility paths to various mobile devices that are available in manufacturing industry, e.g., mobile robots. Surrogate modeling [23] can be used to design the mobility path that minimizes the measurement distance/time without affecting the quality of measured data. In the latter function (wireless measurement control), various wireless quality-of-service (QoS) metrics, e.g., RSSI (received signal strength indicator) and bandwidth, can be sensed by the measurement setups that are attached to the controllable mobile devices (mobile robots) and also uncontrollable mobile devices due to their planned operations, e.g., AGVs, forklifts, and cranes. Each sample data has three major dimensions: sampling time, location, and radio frequency. At the end of a measurement campaign, the data collected by different mobile devices are aggregated and synchronized in a centralized database.

The "monitoring" block (Figure 6.1) fetches the measured wireless data from the central database and builds a holistic heat map for each of the target wireless QoS metrics. Once a weak region is detected from a hot map, the monitoring center will send a prioritized command to the "measurement control" block, such that additional measurements will be performed on that region. If the weakness is confirmed by the extra measurements, the monitoring center will create a weakness alarm not only to human network manager, but also to the "smart switch" block in the system.

The "smart switch" block (Figure 6.1) comprises software and hardware parts. The software part encompasses planning and optimization, which is the focus of Chapter 5 and Chapter 6. The hardware part controls the power on/off state and transmit power setting of each AP that is deployed. For instance, a Siemens W788-2 M12 AP provides both cabled and wireless control options. Although the cabled control is more reliable, it is more expensive due to the additional Ethernet cables and limited in the coverage distance in a large industrial indoor environment. In comparison, the wireless control is more agile and cheap, while it is less stable due to interference with the existing wireless networks.

Furthermore, Figure 6.1 demonstrates the interaction between the cyber space and the physical space in this proposed CPS. The mobile devices are controlled to sense the QoS metrics of the deployed IWLAN in the physical space. The perceived QoS data is then wirelessly transmitted to the central database and is synchronized. The monitoring of the real-time QoS metric degradation may further trigger the decision-making on the optimal TPC of the IWLAN. These types of work are performed in the cyber space. The obtained TPC solution is then sent to the corresponding AP via the wireless control channel to execute such a control action in the physical space. Therefore, this closed loop serves a typical example of the generic architecture of a CPS in Figure 1.2. While the entire closed loop is involved to make the proposed CPS complete and enable the proposed method to be validated in a real industrial environment, the focus of Chapter 5 and Chapter 6 lies in the optimization-based decision-making in the cyber space.

6.3 Problem Formulation

The problem under investigation is to optimize TPC of a dense WLAN in a metal-dominating industrial indoor environment. This IWLAN is over-dimensioned such that redundant APs are planned to create double full coverage for staying robust against shadowing effects of dominant obstacles. As a result, it is unnecessary for all APs to always work at the maximal Tx power level, which produces heavy interference. Therefore, a potential remains to minimize each AP's Tx power, including powering off.

A solution to this problem is denoted by $\vec{p}$. It is a vector of the Tx power levels of all APs (denoted as A) in the over-dimensioned IWLAN, including the decision of powering off certain APs. Table 6.1 lists all the symbols for this problem description. Sections 6.3.1-6.3.5 will present the model of this problem in five aspects: environment modeling, transmit power setting of APs, path loss calculation, interference calculation, and objective function. Section 6.3.6 gives a brief example for this model.

6.3.1 Environment

The plan of a target rectangular environment is two-dimensional (2D), i.e., horizontal and vertical. It is represented by its two extreme 2D points: $(xMin, yMin)$ and $(xMax, yMax)$. It is discretized into $gs \times gs$ small grid cells, where gs is the grid cell size that is preset as an input of the model. A grid point (GP) is represented by the upper-left vertex of a grid cell, and denoted as gp_i, where *i* is a unique index for each GP. A lexicographical order is applied to all the GPs:

$$(x0, y0) < (x1, y1) \Longleftrightarrow x0 < x1 \vee (x0 = x1 \wedge y0 < y1) \tag{6.1}$$

where $(x0, y0)$ and $(x1, y1)$ are illustrated coordinates for two arbitrary different GPs.

Consequently, a target environment is described by a set of ordered GPs denoted as Ω. The GP index *i* within Ω starts from one, corresponding to the extreme point $(xMin, yMin)$ of this environment. It increases one by one until reaching $|\Omega|$ following the lexicographical order. Then the set of GPs is denoted by their index $I = \{1, , 2, ..., |\Omega|\}$. The following formula determines the size of Ω:

$$|\Omega| = ceil\left((xMax - xMin)/gs\right) \times ceil\left((yMax - yMin)/gs\right) \tag{6.2}$$

A receiver (Rx) can be placed on each GP except the ones where APs are placed. The received power in the downlink is considered to enable the calculation of an AP's coverage. For an Rx, different physical bitrate requirements result in distinct requirements on the lowest received power, named threshold (*THLD*). The quantified relation can be found in [24].

The *i*-th GP is considered covered by the *j*-th AP, if an Rx on this GP connects to this AP and receives power values that are higher than or equal to the threshold during at least 99% of the time. This is formulated as follows:

$$\alpha_{ij} = \begin{cases} 1, \text{if } P_{ij} \geq THLD \\ 0, \text{otherwise} \end{cases} , \forall i \in I, \forall j \in J \tag{6.3}$$

where α_{ij} is the logical coverage variable for the *i*-th GP and *j*-th AP, and P_{ij} is

Table 6.1: Nomenclature of the proposed transmit power control (TPC) model

Symbol	Meaning
A	Set of over-dimensioned access points
d_{ij}	Distance between the *i*-th grid point and the *j*-th access point
d_{jmax}	Maximal radius distance the *j*-th access point can cover with its current tranmit power P_j
G	Total gain of a pair of transmitter and receiver
gs	Basic grid cell size in a discretized environment
gp_i	The *i*-th grid point in an environment
I	Set of indices of grid points
I_{ij}	Interference (dBm) of the *i*-th grid point that connects to the *j*-th access point
I_{ijmax}	Maximal interference (dBm) of the *i*-th grid point that connects to the *j*-th access point
J	Set of indices of access points
J_{on}	Access points that are powered on
J_{off}	Access points that are powered off
M	Margin (dB) considering shadowing, fading, and interference
n	Path loss exponent
N_o	Total number of dominant obstacles in an environment
N_p	Total number of transmit power levels (excluding powering off)
OL_{ij}	Obstacle loss (dB) between the *i*-th grid point and the *j*-th access point
OL_k	Obstacle loss (dB) of the *k*-th dominant obstacle
P	Set of transmit power (dBm) of an access point
P_{max}	Maximal transmit power
P_{min}	Minimal transmit power
δp	Transmit power control step (dB) of an access point
$\vec{p}$	Vector of Tx power levels of an access point
$\widehat{p}$	Set of transmit power levels
$\widehat{p}_j$	Transmit power level of the *j*-th access point
P_j	Transmit power (dBm) of the j-th access point
P_{ij}	Stable power received by the *i*-th grid point from the *j*-th access point at 99% of the time
$PL0$	Path loss (dB) at the location 1 m from a target access point
$PL(d_{ij})$	Path loss (dB) between the *i*-th grid point and the *j*-th access point
Rx	Wireless signal receiver
$THLD$	Threshold received power (dBm) of a client receiver
Tx	Wireless signal transmitter
$xMax$	Maximal horizontal coordinate of an environment
$xMin$	Minimal horizontal coordinate of an environment
$yMax$	Maximal vertical coordinate of an environment
$yMin$	Minimal vertical coordinate of an environment
Ω	Set of all grid points in an environment
ξ	Deviation between measurement and a path loss model
μ	Percentage of grid points that must be covered by at least one access point
α_{ij}	Logical coverage variable for the *i*-th grid point and the *j*-th access point
β_{ij}^k	Logical signal blockage variable for the *i*-th grid point, the *j*-th access pint, and the *k*-th dominant obstacle

Table 6.2: Mapping between physical power, power state, and the digital transmit power level of an access point

Transmit power set P (dBm)	Power state	Transmit power level set $\widehat{P}$
-	Off	0
P_{min}	On	1
$P_{min} + \delta p$	On	2
$P_{min} + 2\delta p$	On	3
...	On	...
P_{max}	On	N_p

the stable power (dBm) that an Rx on the *i*-th GP receives from the *j*-th AP at least 99% of the time. The coverage of an AP is represented by the GPs that are covered by this AP.

6.3.2 Over-Dimensioned Wireless Local Area Network

In the over-dimensioned IWLAN, $|J|$ APs are deployed (see the beginning of Section 6.3) with a minimal separation distance in the environment, where J is the set of AP index which varies from one to the total number of APs ($|A|$ or $|\vec{p}|$), i.e., $J = \{1, 2, ..., |A|\}$.

In a TPC solution $\vec{p}$, the APs are regrouped into a set of APs that are powered off (J_{off}) and a set of APs that are powered on with a certain power value (J_{on}):

$$J = J_{on} \cup J_{off} \tag{6.4}$$

where the AP indices in J_{on} and J_{off} remain the AP indices in J to identify each AP.

All APs have the same TPC range *P* (in dBm) and step δp (in dB), i.e., $P = \{P_{min}, P_{min} + \delta p, P_{min} + 2\delta p, ..., P_{max}\}$. In total, there are N_p different transmit power values in *P*, besides the possibility of powering off. Therefore, $P_j \in P$ if $j \in J_{on}$, where P_j is the transmit power of the *j*-th AP. P_j is not considered if the *j*-th AP is powered off (Table 6.2).

As indicated in Table 6.2, with the possibility of powering off, the TPC range *P* is discretized into $\widehat{P}$, which is a dimensionless set of all possible transmit power levels, i.e., $\{0, 1, 2..., N_p\}$. The discretized transmit power level of the *j*-th AP is denoted as $\widehat{P}_j$ $(\in \widehat{P})$. Specifically, the level zero stands for powering off. The level one represents the minimal transmit power level of an AP, and so on, until the maximal transmit power level N_p of an AP. Consequently, the digital variable $\widehat{P_j}$ $(\forall j \in J)$ can be used to represent all the possible power states of an AP: powering off/on and if powering on, at which transmit power this AP operates.

6.3.3 Path Loss

In [5], the path loss model was simplified as a multiplier of transmit power. In [12, 13], the Boolean disk-based path loss model only determined whether an Rx was within the circular coverage of a Tx. In the proposed TPC model, a one-slope path loss model considering metal obstacle shadowing loss along the propagation path is considered for accurate path loss calculation, which is the basis for calculating the received power of an Rx and interference. In total, there are $N_o(\geq 0)$ dominant obstacles in the investigated environment. This path loss model is formulated as:

$$PL(d_{ij}) = PL0 + 10 \cdot n \cdot \log_{10}(d_{ij}) + OL_{ij} + \xi \tag{6.5}$$

where $PL0$ (in dB) is the path loss at the distance of one meter, *n* is the path loss exponent which is a dimensionless parameter indicating the increase of path loss with the distance, d_{ij} is the distance (in m) between the Rx placed on the *i*-th GP and the *j*-th AP, OL_{ij} is the total obstacle loss (in dB) caused by the metal obstacles that block the line between the Rx placed on the *i*-th GP and the *j*-th AP, and ξ (in dB) is the deviation between the measurement and the model.

For an investigated environment, it assumes that the obstacle locations are fixed. The deviation ξ in Equation (6.5) follows a Gaussian distribution, with a mean of zero and a standard deviation σ. The gain and margin are considered in the link budget calculation to be more realistic, which was not taken into account in [5]. The total gain *G* (in dB) is the sum of the AP transmitter's gain and the Rx's gain (of omnidirectional antenna). The margin *M* (in dB) is the sum of shadowing, fading and interference margin.

$$OL_{ij} = \sum_{k=1}^{N_o} \beta_{ij}^k \cdot OL_k, \forall i \in I, \forall j \in J_{on}, \forall k \in \{1, 2, ..., N_o\} \tag{6.6}$$

$$\beta_{ij} = \begin{cases} 1, \text{if } k\text{-th obstacle blocks the line-of-sight between } i\text{-th GP and } j\text{-th AP} \\ 0, \text{otherwise} \end{cases},$$

$$\forall i \in I, \forall j \in J_{on}, \forall k \in 0, 1, ..., N_o \tag{6.7}$$

The total obstacle loss between the Rx on the *i*-th GP and the *j*-th AP is calculated in the following two equations. Equation (6.6) iterates all the dominant obstacles in the environment and accumulates the additional path loss caused by the obstacles that blocks the line-of-sight radio propagation from the *j*-th AP to the Rx on the *i*-th GP. Equation (6.7) defines the logical signal blockage variable β_{ij}^k. If the *k*-th dominant obstacle has the shadowing effect on the line-of-sight radio propagation from the *j*-th AP to the Rx on the *i*-th GP, it equals one. Otherwise, it equals zero. The attenuation (dB) of obstacle k (OL_k) is experimentally charac-

terized and is the input of this TPC model. The calculations, which are defined by Equation (6.6) and Equation (6.7), are only limited to APs that are powered on.

Analogous to Chapter 5, compared to most coverage-related optimization problems that only rely on a 2D environment [12, 13, 16, 17, 24, 25], an obstacle is modeled as a 3D geometrical model in the decision making of line-of-sight propagation between a GP-AP pair (i.e., the logical signal blockage variable β_{ij}^{k}). Both the *j*-th AP and the Rx placed on the *i*-th GP have their own heights. An obstacle has a 3D dimension of length $\times$ width $\times$ height. An obstacle blocks the line-of-sight propagation as long as part of it crosses the straight line between the top of the *j*-th AP and the top of the Rx on the *i*-th GP. A detailed discussion on the 3D obstacle loss calculation can be found in [26].

If the *j*-th AP is powered on with the transmit power P_j, the maximal radius distance this AP can cover (d_{jmax}) can then be calculated, without considering the additional shadowing effects that may be caused by dominant obstacles. For an AP that is powered off, d_{jmax} is zero, indicating that it cannot cover any GP. This is formulated in Equation (6.8) and will be used in the speedup measures for the solution method (Section 6.4.6).

$$d_{jmax} = \begin{cases} 10^{\left(\frac{P_j+G-M-THD-PL0}{10\cdot n}\right)}, \forall i \in I, \forall j \in J_{on}, P_j \in P \\ 0, \forall j \in J_{off} \end{cases} \tag{6.8}$$

6.3.4 Interference

An inevitable goal of transmit power management for a dense WLAN is the interference among APs. While dedicated frequency planning is out of scope in this chapter, it is assumed that non-overlapping channels are effectively allocated to the dense APs. If an Rx on the *i*-th GP connects to the *j*-th AP $(j \in J_{on})$, the interference (I_{ij}, in dBm) to this Rx is then all the power this Rx can sense from the other APs that are powered on $(\forall j' \in J_{on}, j' \neq j)$ [25, 27]. The interference calculated this way is also interpreted as noise [25]. If an AP is powered off, it is not considered by this calculation. This is formulated in the following two equations:

$$I_{ij} = 10 \cdot \log_{10} \sum_{j' \in J_{on}} 10^{P_{ij'}/10}, \forall i \in I, \forall j, j' \in J_{on}, j' \neq j \tag{6.9}$$

$$P_{ij'} = P'_j + G - M - PL(d_{ij'}), \forall i \in I, \forall j' \in J_{on}, P_{j'} \in P \tag{6.10}$$

The worst case is that all APs are powered on with the maximal transmit power ($P_j = P_{max}$, $\forall j \in J_{on}$, $J_{on} = J$). Then the maximal interference (I_{ijmax}, in dBm) to an Rx can be calculated as follows:

$$I_{ijmax} = 10 \cdot \log_{10} \sum_{j' \in J_{on}} 10^{P_{ij'}/10}, \forall i \in I, \forall j, j' \in J_{on} = J, j' \neq j, P_j = P_{max} \tag{6.11}$$

6.3.5 Transmit Power Control

The TPC model is minimization of normalized total interference (Section 6.3.4) under the constraint of wireless coverage (Section 6.3.3) of a metal-dominating industrial environment (Section 6.3.1) which is deployed with over-dimensioned IWLAN (Section 6.3.2). It is described in the following three formulas.

$$\min_{\widehat{P}_j} \left(\frac{\sum_{i=1}^{N_{GP}} \sum_{j=1}^{|A|} 10^{I_{ij} \cdot \gamma_{ij}/10}}{\sum_{i=1}^{N_{GP}} \sum_{j=1}^{|A|} 10^{I_{ijmax} \cdot \gamma_{ij}/10}} \cdot 100\% \right), \forall \widehat{P}_j \in \widehat{P}, \forall i \in I, \forall j \in J \tag{6.12}$$

Subject to:

$$\sum_{j=1}^{|A|} \alpha_{ij} \geq 1, \forall i \in \mu \cdot I \tag{6.13}$$

$$\gamma_{ij} = \begin{cases} 1, \text{if Rx on the } i\text{-th GP connects to the } j\text{-th AP} \\ 0, \text{otherwise} \end{cases} \quad \forall i \in I, \forall j \in J \tag{6.14}$$

Equation (6.12) sets the goal of TPC as minimizing the normalized interference (in mW) in the whole over-dimensioned network. The essential variable that is tunable for this optimization is the transmit power level of each AP ($\widehat{P}_j, \forall j \in J$) deployed in the environment.

Equation (6.13) sets the constraint that a percentage μ of all the GPs must be covered by at least one AP, i.e., a coverage rate μ ($\mu \in (0, 1]$) must be ensured in the target environment. As the TPC model targets at the operating phase of an over-dimensioned wireless network, the redundant coverage layer planned by the OD method (Chapter 5) is powered off. This is why only one coverage layer is ensured in this constraint.

A logical variable of AP connection β_{ij} is introduced in Equation (6.14). If an Rx can sense multiple APs that are powered on, it connects to the one that achieves the highest received power at this Rx. If there are multiple APs that have the same highest received power at this Rx, the Rx randomly connects to one of these APs. An Rx can connect to at most one AP, while an AP can have multiple Rx that connect to it. While received power of a client plays a vital role in handover and AP association, further discussion on client-AP association mechanism is out of scope in this chapter.

Overall, the TPC model is formulated from Equation (6.1) to Equation (6.14),

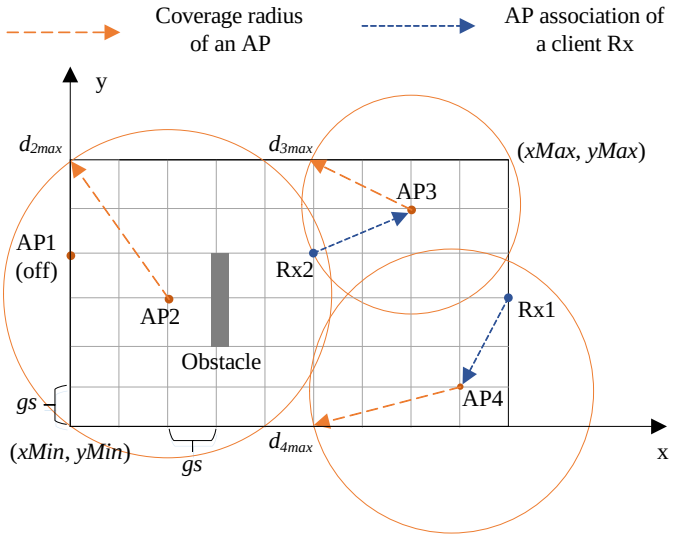


Figure 6.2: Illustration of the transmit power control model: discretized environment with the presence of a dominant obstacle, transmit power minimization of over-dimensioned access points (APs) for only one full coverage layer, AP association of two client receivers (Rx), and interference (produced by all APs that are powered on and not connected to).

and is named the interference minimization based TPC model (IM-TPC). According to the definition of large-scale problems [20], the scale of a TPC model is influenced by (1) the size of a target industrial indoor environment (which is linked to the number of APs), (2) the grid cell size (gs), (3) the number of dominant obstacles, and (4) the coverage rate (μ). The first factor determines the dimensionality of a search space and impacts the complexity of fitness evaluation. The second factor is associated to the density of a search space. The third factor defines the expense of path loss calculation and thereby fitness evaluation. The last factor controls the hardness of coverage constraint and also the interaction between transmit power levels of APs due to the planned redundancy. Therefore, a TPC model is considered as large-scale if the industrial indoor environment is large with a small grid cell size, the presence of multiple dominant obstacles, and a high coverage rate.

6.3.6 Illustrative Example

The TPC model is illustrated in a simple example (Figure 6.2). A rectangular environment, defined by $(xMin, yMin)$ and $(xMax, yMax)$, is discretized into 9

$\times$ 6 = 54 grid cells, each of which is represented by its upper-left vertex (GP). A $gs \times gs$ grid cell is considered covered by an AP, if its GP is within the coverage radius of this AP which works at its current transmit power. Although four APs are deployed as redundancy for robustness, AP1 is powered off and the other APs operate below their maximal transmit power, in order to reduce interference while guaranteeing one full coverage layer ($\mu = 1$). Therefore, the three APs that are powered on have their respective coverage radius: d_{2max}, d_{3max}, and d_{4max} (without considering any obstacles). There are two clients: Rx1 and Rx2. While Rx1 connects to AP4 as it is only covered by AP4, Rx2 has to decide between AP2 and AP3. Due to the obstacle shadowing between AP2 and Rx2, Rx2 receives lower signal strength from AP2 than from AP3. Rx2 thus connects to AP3. As a result of these AP associations, AP2 and AP3 produce interference to Rx1, while AP2 and AP4 cause interference to Rx2. As there are multiple transmit power control solutions for one full coverage layer, the optimized solution has to be determined in terms of minimal interference. An optimization algorithm is needed for automatic decision making of the transmit power of each AP, when the TPC problem size increases and power control is frequently performed due to dynamic shadowing effects in harsh industrial indoor environments.

6.4 Solution Algorithm

As an optimum rectangular grid coverage problem is NP-complete [28], metaheuristics are an effective technique to solve this type of problem. As a well-known metaheuristic, a GA gives a near-optimum solution for an NP-complete combinatorial problem within a reasonable time [29]. This characteristic complies with the objective of the investigated TPC problem to fast obtain a high-quality solution without necessarily requiring the real optimum. Therefore, a GA is used for this TPC problem.

The design of this GA follows the objective of simultaneously minimizing memory usage and CPU time for solving a large-scale TPC model. This GA based TPC algorithm is named GATPC. Given that the GATPC checks the coverage of a GP by iterating all APs that can potentially cover this GP, the following definitions are made to facilitate the presentation of GATPC in the subsections.

Definition 1: *covered GPs* refer to a set of GPs that are covered by at least one AP.

Definition 2: *uncovered GPs* represent a set of GPs that are not yet covered by any AP with the current TPC solution $\vec{p}$.

Definition 3: *new covered GPs* of a given AP stand for a subset of *uncovered GPs* that can be covered by this AP at its current transmit power level.

Definition 4: a *GP-AP link* shows that the investigated GP can be covered by

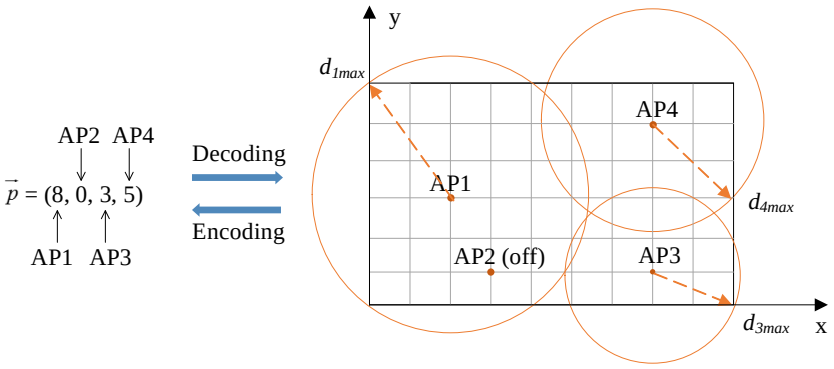


Figure 6.3: Example of transmit power control solution encoding and decoding. A solution is denoted by a vector $\vec{p}$ that indicates an access point (AP) and its transmit power level by its element index and element value, respectively. As a result of this $\vec{p}$, the relationship among coverage radiuses of all APs are: $d_{1max} > d_{4max} > d_{3max}$, while AP2 is powered off.

the investigated AP at its maximal transmit power level. It thus shows an AP's potential to cover a GP.

Definition 5: the *nearest potential AP* of an uncovered GP is the AP that is the nearest to this GP among all the APs that have *GP-AP links* with this GP.

6.4.1 Solution Encoding and Fitness Evaluation

As introduced in Section 6.3, a TPC solution is $\vec{p}$, a vector containing $|\vec{p}|$ discretized AP transmit power levels, including powering off (Table 6.2). The index of a value in $\vec{p}$ corresponds to the index of the AP that is over-dimensioned in the environment. The list of APs in the over-dimensioned IWLAN is sorted by applying the lexicographical order (Equation (6.1)) to the GPs on which these APs are placed. Therefore, the j-th value in $\vec{p}$ corresponds to the transmit power level of the j-th AP. Figure 6.3 illustrates an example of encoding and decoding $\vec{p}$ in this manner.

This encoding scheme has two advantage points. Firstly, it facilitates the definition of crossover and mutation, which will be presented in Section 6.4.3 and Section 6.4.4, respectively. Secondly, it enables efficient memory utilization. As each scalar in $\vec{p}$ can be represented only by a 32-bit integer, little encoding memory is needed even for evolutionary optimization of a large-scale TPC problem. For example, 1,000 TPC solutions for 100 APs only occupy a memory of 0.38 MB.

The fitness of a TPC solution is the normalized interference, which is defined by Equation (6.12). The entire TPC model (Section 6.3) is required for this calculation. Therefore, the population evolution in a GA will autonomously mitigate

Algorithm 9 Generation of a random transmit power control (RTPC) solution

Input: none
Output: a qualified random TPC solution $\vec{p}$ satisfying the coverage constraint defined by Equation (6.13)

1: $\vec{p} \leftarrow |A|$ random numbers $\in \widehat{P}$
2: $\vec{p} \leftarrow\ repair(\vec{p})$ using Algorithm 10

Algorithm 10 Repair of an unqualified transmit power control solution

Input: a TPC solution $\vec{p}$ without knowing its qualification
Output: a qualified TPC solution $\vec{p}$ satisfying the coverage constraint defined by Equation (6.13)

1: covered GPs $\leftarrow$ *new covered GPs* of APs $\in A$ with $\vec{p}$
2: **if** $|covered\ GPs < \mu \cdot \Omega|$ **then**
3: *uncovered GPs* $\leftarrow \Omega$
4: remove *covered GPs* from *uncovered GPs*
5: **for** $j \leftarrow 1 : |A|$ **do**
6: **if** $\vec{p}(j) < N_p$ **then**
7: set up *GP-AP links* between *uncovered GPs* and $AP(j) \in A$
8: **end if**
9: **end for**
10: **while** *uncovered GPs* $\neq \oslash$ & $|covered\ GPs| < \mu \cdot \Omega$ **do**
11: find *nearest potential AP* of a random GP $\in$ *uncovered GPs*
12: **if** *nearest potential AP* $= \oslash$ **then**
13: remove this random GP from *uncovered GPs*
14: **else**
15: assign *nearest potential AP* with the minimal transmit power level that can cover this random GP
16: remove *new covered GPs* of *nearest potential AP* from *uncovered GPs*
17: *covered GPs* $\leftarrow$ *new covered GPs* $\cup$ *covered GPs*
18: **if** transmit power level of *nearest potential AP* $= N_p$ **then**
19: remove *GP-AP links* between *nearest potential AP* and all the related GPs in *uncovered GPs*
20: **end if**
21: **end if**
22: **end while**
23: **end if**

the interference, which is the objective of TPC.

6.4.2 Population Initialization

It is not obliged to generate only qualified initial individuals, since unqualified individuals will be either eliminated by the population evolution or improved by the crossover and mutation operations. However, any generation of unqualified individuals will produce computation redundancy to the GA search and thus reduce the

optimization efficiency. Especially for evolutionary optimization of a large-scale TPC problem, the computation time to get an acceptable solution is quite sensitive to computation redundancy. Hence, the proposed initial population generation algorithm aims to produce 100% qualified initial individuals.

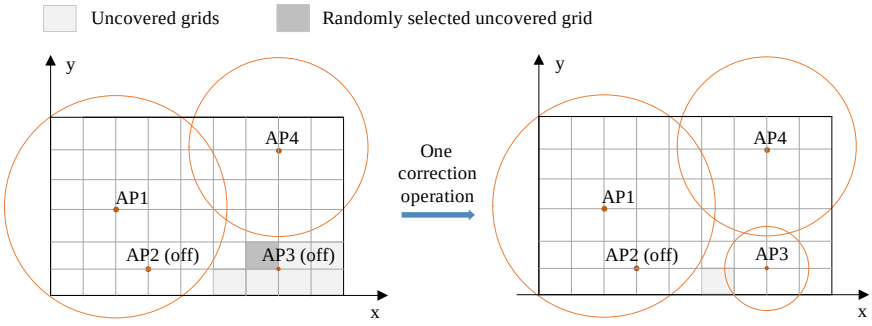
Algorithm 9 describes a two-step sequential method to randomly produce a qualified individual. Step 1 (line 1) generates a random TPC solution without considering the coverage constraint defined by Equation (6.13). Step 2 (line 2) checks and ensures the random solution's satisfaction of coverage constraint by potentially raising the transmit power of selected APs to the minimal degree. This procedure for qualification check and potential repair is proposed in Algorithm 10.

At the start of Algorithm 10, the GPs' coverage information is updated by setting APs with these input transmit power levels (line 1). If the required coverage rate is not yet achieved (line 2), this random TPC solution will be corrected (lines 3-22). The idea of repair is to iteratively cover the *uncovered GPs* using the potential APs (whose current transmit power is below the maximal level) while maintaining the increase in transmit power as slight as possible (to comply with interference minimization). At the first step of correction (lines 3-9), *uncovered GPs* are all found and their *GP-AP links* are all established. The second step of correction (lines 10-22) is an iteration. In each iteration, a GP is randomly selected from uncovered GPs and its *nearest potential AP* is found (line 11). The rationale behind the random selection is to increase the diversity of population and prevent premature convergence. In a minor case where a randomly selected GP is shadowed by an obstacle such that it cannot be covered by any AP, this GP is removed from *uncovered GPs* (lines 12-13). Otherwise, the transmit power of *nearest potential AP* is set to the minimal level that can cover this random *uncovered GP* (line 15), followed by updating GP coverage information and *GP-AP links* (lines 16-20). Figure 6.4 further presents an example to iteratively correct an unqualified solution (lines 3-22, Algorithm 10).

To produce the entire initial population, Algorithm 9 is iterated for a number of times equal to the size of population. As it is extremely hard to produce even a qualified solution for this TPC problem, Algorithm 9 also serves as a random TPC solution generation algorithm (named RTPC) for benchmarking.

6.4.3 Selection and Crossover

Parent solutions are selected using roulette wheel selection for a crossover operation. A crossover operation enables two parent solutions to breed two new child solutions by swapping the parents' genes. A one-point crossover is realized in GATPC using two steps. Step 1 (lines 1-4, Algorithm 11) exchanges partial chromosomes of two parents around a randomly selected loci. Figure 6.5 illustrates this one-point crossover operation. It is simple and intuitive due to the encoding



(a) Before this correction operation (b) After this correction operation

Figure 6.4: Example of iterative correction of an unqualified transmit power control (TPC) solution. (a) Seven grid cells are uncovered by any APs. One grid is randomly selected (dark grey) and then AP3 is selected from the four APs as it is the closest to this random uncovered grid. (b) One grid remains uncovered after AP3 is set to the minimal transmit power that can cover this random uncovered grid (a grid is represented by its upper-left vertex). Another iterative correction is thereby needed to fully cover the entire environment.

Algorithm 11 Crossover for the genetic algorithm based transmit power control

Input: two selected parent TPC solutions *parent1* and *parent2*
Output: two offspring TPC solutions *offspring1* and *offspring2*

1: $loci \leftarrow$ a randomly selected index of *parent1*
2: divide *parent1* and *parent2* into two parts along this random *loci*, respectively
3: *offspring1* $\leftarrow$ 1st part of *parent1* + 2nd part of *parent2*
4: *offspring2* $\leftarrow$ 1st part of *parent2* + 2nd part of *parent1*
5: **for** *indiv* $\in \{$*offspring1*, *offspring2*$\}$ **do**
6: $\quad indiv \leftarrow repair(indiv)$ using Algorithm 10
7: **end for**

scheme (Section 6.4.1) and lexicographical ordering of all APs in Equation (6.1). Step 3 (lines 5-7 in Algorithm 11) first checks whether each child solution achieves full coverage. If the coverage rate, required by Equation (6.13), is not yet achieved, the *uncovered GPs* will be addressed one by one with their *nearest potential APs*. This potential correction of an unqualified solution follows the repair procedure described in Algorithm 10.

6.4.4 Mutation

A GA is known as a global optimization algorithm. A mutation operation plays a vital role to this end. The mutation of GATPC is defined by Algorithm 12, comprising two steps. Step 1 (lines 1-5) powers off one AP that already reaches

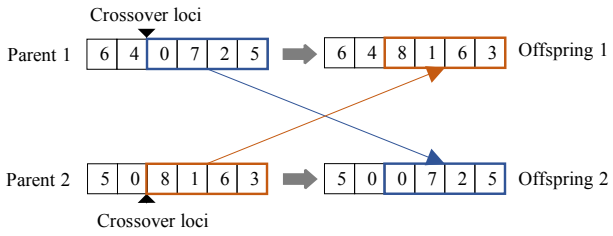


Figure 6.5: Example of one-point crossover of two parent transmit power control (TPC) solutions, which exchange a segment of transmit power levels in vector $\vec{p}$. The offspring would need to be corrected.

Algorithm 12 Mutation for the genetic algorithm based transmit power control

Input: an offspring TPC solution outputted by crossover and selected for mutation
Output: a new TPC solution

1: $selectedAPs \leftarrow$ AP(s) in the input individual that reach(es) the highest transmit power level (N_p)
2: **if** $|selectedAPs| > 1$ **then**
3: *selectedAP* $\leftarrow$ a random AP in *selectedAPs*
4: **end if**
5: new $\vec{p} \leftarrow$ power off *selectedAP*
6: $\vec{p} \leftarrow repair(\vec{p})$ using Algorithm 10

the highest transmit power level, i.e., without potential to increase transmit power any more. This aims to increase the diversity in the solution space and prevents the GA search from being trapped in a local optimum. A new individual is then created at the end of step 1. Step 2 (line 6 in Algorithm 12) also employs the repair mechanism (Algorithm 10). It corrects the new individual produced by the former step 1 with the "best effort", if the environment cannot be covered at the required coverage rate at the end of step 1.

6.4.5 Parallel Genetic Algorithm

The efficiency of a GA search is sensitive to a large-scale optimization. A conventional GA structure can be found in [29]. The fundamental GA operations include: (1) population initialization, where a fixed size of individual solutions are generated in a random manner; (2) crossover, which swaps part of genes of two chromosomes (i.e., individual solutions); (3) mutation, which swaps genes (i.e., part of a solution) of a chromosome; and (4) elitism, which retains a fixed size of the best individuals in a parent generation as members in the child generation.

All these GA operations and fitness calculation exhibit a common characteristic for applying "map-and-reduce" [30] or "divide-and-conquer" [13, 20, 31] parallel computation strategy: each GA operation contains multiple independent

sub-operations of the same type and with different individuals. Therefore, the sub-operations can be conducted in parallel, such as by multithreads of a processor [32]. The results of sub-operations are then collected one by one at the end of each sub-operation. As a result, the GA search gains speedup as multithreads physically work in parallel in different cores of a processor. A flow chart of the proposed parallel GA can be found in Figure 5.9 in Chapter 5.

6.4.6 Additional Speedup Measures

As aforementioned, the design of GATPC in the former subsections follows the idea of decreasing computation time and memory, to enable large-scale optimization of TPC. More specifically, the potential for computation time reduction lies in the unnecessary computation. It is illustrated as (1) the extensive calculation of a complete set of variables, whereas only a small subset is actually required by the algorithm instance, (2) the repetitive calculation of the same set of variables, while these variables do not have to be updated, (3) the repetitive implicit reading from and writing to the I/O (e.g., files), while only one reading and one writing are actually sufficient. The following measures are taken to speed up the GATPC by reducing the computation redundancy.

An AP's maximal coverage distance $(d_{jmax}, \forall j \in J)$ is extensively calculated by Algorithms 9-12. To speedup, d_{jmax} is calculated by Equation (6.8) before the actual start of a GA search. In total, N_p different d_{jmax} values are pre-calculated according to N_p different AP transmit power values of an AP, and stored as a constant vector. All the d_{jmax}-related calculation during the GA search process will then simply look up to this vector, instead of repeating the path loss calculation millions of times.

GP-AP links of all APs are very frequently established or removed in the correction procedure of Algorithms 9-12, which requires an extensive iteration of all possible GP-AP pairs ($|\Omega| \cdot |A|$ in the worst case). This certainly becomes a tedious and time-consuming operation for a large environment that has more than 10,000 GPs as well as at least dozens or hundreds of APs. The corresponding speedup measure consists of the following four sequential steps. (1) For the *j*-th AP, search in the aforementioned d_{jmax} vector for its maximal coverage distance corresponding to the current transmit power level. (2) Set up a $d_{jmax} \times d_{jmax}$ rectangular region that is centered at the *j*-th AP. (3) Iterate the GPs within this rectangular area and set up *GP-AP links* of the *j*-th AP. (4) Iterate all APs and conduct steps (1-3) in each iteration. The obtained speedup is especially significant for a large environment, since the area to set up *GP-AP links* is substantially reduced from the entire environment to the $d_{jmax} \times d_{jmax}$ small square.

Besides, Algorithms 9-12 frequently judge whether an obstacle shadows the signal between the *j*-th AP and an Rx on the *i*-th GP, and then calculate the accu-

mulated obstacle loss, i.e., Equation (6.6) and Equation (6.7). The speedup measure is inspired from the fact that, for a certain GA search, all the obstacles and APs are static in terms of quantity and location. Consequently, the signal blockage between the *i*-th GP and the *j*-th AP can be judged before the GA search, and the corresponding obstacle loss (including zero loss) can be pre-stored in a table. The GA search will then only need to inquire the pre-stored table of obstacle loss by inputting the indexes of GP and AP, instead of on-the-fly judgment.

Last but not least, Algorithms 9-12 extremely frequently judge whether a GP is covered by an AP at its current transmit power level, i.e., Equation (6.3). Thereby, the path loss calculation considering the shadowing effects of dominant obstacles should extensively be performed. As a speedup measure, Equation (6.3) is implemented in the following sequential steps for the *i*-th GP and the *j*-th AP. (1) Look up to the aforementioned d_{jmax} vector for the corresponding d_{jmax} of the *j*-th AP. (2) Set up a $d_{jmax} \times d_{jmax}$ square that is centered at the *j*-th AP. (3) Calculate the path loss between the *j*-th AP and the *i*-th GP, without considering the shadowing effects. (4) Look up to the aforementioned obstacle loss table, and add the obstacle loss to the path loss that is obtained in step (3), and get the final path loss value. (5) Obtain P_{ij} with the final path loss value and judge whether it is above the preset sensitivity threshold.

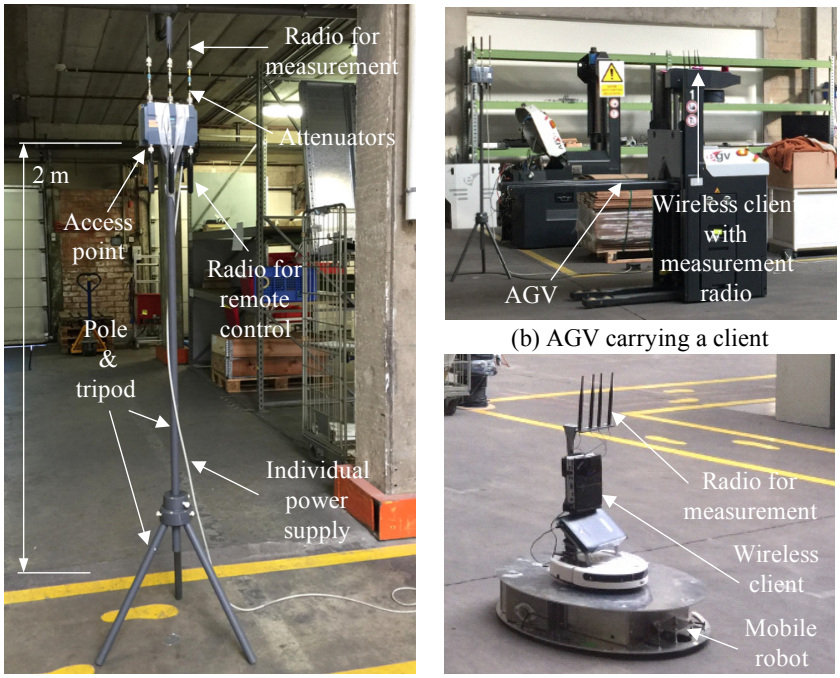
6.5 Experimental Validation

The TPC model and the GATPC algorithm were validated in a small open industrial environment ($10\ m \times 10\ m$) in the factory hall of a manufacturer of automated guided vehicles (AGVs), in Flanders, Belgium.

6.5.1 Configurations

A measurement control system [24] accommodating the GATPC and four Siemens industrial APs (Scalance W788-2 M12) with individual power supply (Figure 6.6a) were used.

More specifically, an AP has two radio ports (Figure 6.6a). One was configured for measurements at 2.4 GHz and the other was configured for remote control at 5 GHz, such that the interference between measurements and control is mitigated. For an AP, 44 dB attenuation was added to each of the three ports of the measurement radio, to mimic a larger environment needing four APs for double full coverage. Individual power supply plus an extension power cable was applied to every AP, to enable deployment without the distance limitation. The remote AP control was realized by SSH (secure shell). The central PC thus sent wireless control commands to an AP, such as setting the transmit power and powering on/off a radio.



(a) Deployment of access point

(b) AGV carrying a client

(c) Mobile robot carrying a client

Figure 6.6: Experimental facilities, including a measurement control computer system, four commercial off-the-shelf industrial access points, an automated guided vehicle (AGV) with a wireless client, and a mobile robot with a wireless client.

The four APs were over-dimensioned on the boundary of the environment, such that each side was placed with one AP and double full coverage was planned [24]. The AP locations are indicated in Table 6.3, of which the coordinates are these used by the localization system of an automated guided vehicle (AGV).

The coverage measurement facilities that were used have been introduced in [24] in detail. They mainly include a measurement control software system, two Zotac mini-PCs as two individual wireless clients, four poles with tripods to support the APs at the height of 2 m (Figure 6.6a), an AGV as a controllable mobile vehicle which carries one client on the top (Figure 6.6b), a w-iLab.t mobile robot [33] which carries the other client on the top (Figure 6.6c).

Instead of manual measurements, the two clients automatically kept on moving around in the environment and measuring the coverage of the AP that they connected to, and fed the collected samples back to the central PC for monitoring. These samples were stored in database of the measurement control system. Samples from the same AP and within the same spatial grid cell were further aggre-

Table 6.3: Configurations of the measurement campaign

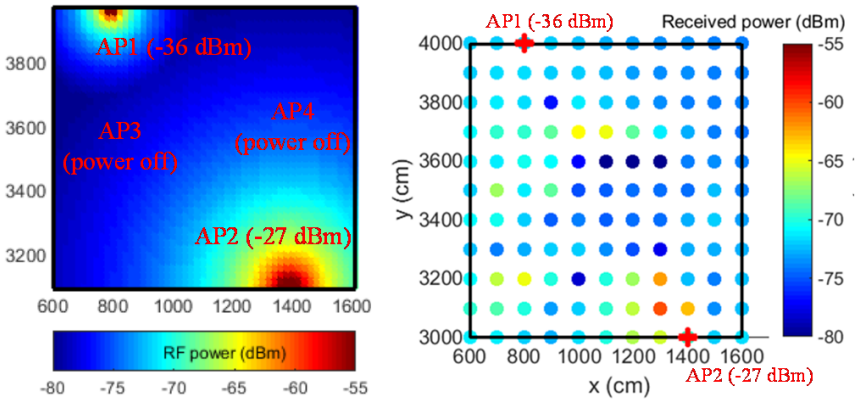
Variable	Setting
AP transmit power range with attenuation	-39:1:-27 dBm
WLAN standard	IEEE 802.11n
AP working frequency band	2.4 GHz
AP remote control frequency band	5 GHz
AP height	2 m
AP1 location	(8, 40) m
AP2 location	(14, 31) m
AP3 location	(6, 35) m
AP4 location	(16, 36) m
Required physical bitrate of a wireless client	24 Mbps
Required receiving sensitivity	-79 dBm
Mobility speed of the automated guided vehicle and mobile robot	20 cm/s
Grid cell size for coverage monitoring	1 m
Shadowing margin (95%)	1 dB
Fading margin (99%)	0 dB
Interference margin	0 dB
GATPC stop criterion	30 iterations

gated to one value (dBm) to enable stable coverage monitoring. For the minority of grid cells that might contain no sample, interpolation [23, 24] was applied based on the surrounding samples. Table 6.3 lists the key measurement configurations.

In total, 3745 RF power samples were collected. Regression [34] was applied to these data to build an empirical path loss model formulated by Equation (6.5), where *PL0* was 39.87, *n* was 1.78, and the obstacle loss OL_{ij} $(\forall i \in I,\ \forall j \in J)$ was zero dB due to the empty environment. The R-squared value was 97.38%, indicating that the path loss model was highly fitted to the samples.

6.5.2 Validation Results

The TPC solution given by the GATPC algorithm is illustrated by Figure 6.7a. AP1 and AP2 are powered on at -36 dBm and -27 dBm, respectively. AP3 and AP4 are both powered off. The colored GPs (grid points) represent the highest received RF power from the existing APs. All the received RF power values are above the required lowest sensitivity (-79 dBm, Table 6.3), indicating one full coverage layer in the environment. In a conventional full power-on scheme, all the four APs are simply powered on with the maximal transmit power (-27 dBm). In comparison, in the obtained TPC solution, AP1 decreases the transmit power to -36 dBm, and AP3 and AP4 are powered off.



(a) Solution suggested by GATPC (b) Coverage monitored by two robots

Figure 6.7: Transmit power control solution given by the proposed genetic algorithm based transmit power control (GATPC): (a) the predicted coverage using this solution, and (b) the actual coverage monitored by an automated guided vehicle (AGV) and a mobile robot carrying wireless clients. The predicted and measured coverage maps are highly matched.

The power states and transmit power levels of the four deployed APs were then set according to this optimized TPC solution. The coverage was monitored. As shown in Figure 6.7b, the received RF power values vary between -60.4 dBm and -78.8 dBm. They are above the threshold sensitivity (-79 dBm, Table 6.3), demonstrating that the environment is fully covered. Therefore, the solution given by GATPC is effective to satisfy the major constraint (i.e., coverage defined by Equation (6.13)) of the TPC model.

6.5.3 Demonstration of Mobile Measurement

The integration of measurement setups on an AGV and a mobile robot was carried out, respectively, to demonstrate the feasibility of mobile measurements. The experiments by this AGV (Figure 6.6b) were conducted in the former environment. This area was clear of obstacles such that the radio propagation was always line-of-sight (LoS) between Tx and Rx. Besides, this area is surrounded by metal racks. A Siemens Scalance-W700 industrial AP was deployed and equipped with a directional antenna of 18 dBi that has a beam width of 17°. The AP operated at a frequency of 5240 MHz and had a transmit power of 5 dBm.

Figure 6.8 shows the RSSI measurement results that are captured by the setup on the AGV and collected by the central controller. It clearly presents that this AGV automatically follows a closed trajectory which is formed up with parallel straight lines and big 180° turns between the ends of straight lines. The mea-

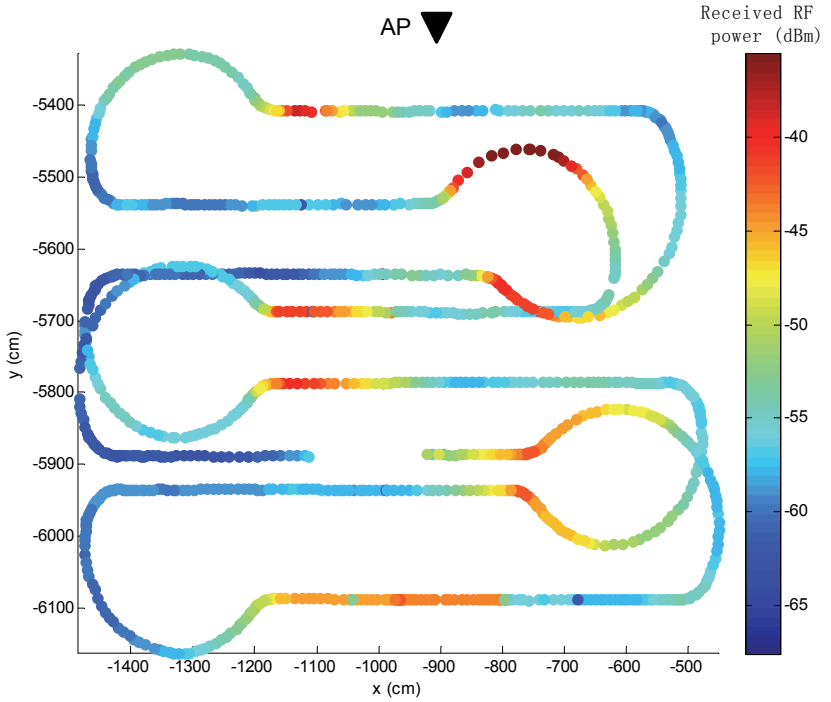


Figure 6.8: Radio signal strength (RSS) measurements by integration of a measurement setup on an automated guided vehicle (AGV). The access point (5240 MHz, 5 dBm, equipped with a directional antenna of 18 dBi) is placed about 1 m outside the top-center region of the REM, and is oriented towards the neighborhood of the point (-900, -6100). The whole figure, shown on the central controller, is also a representation of the spatial AGV trace for conducting the measurement. The x and y coordinates are negative, since the axis origin is located in another hall of the AGV factory.

sured received power values (RSSI mapped to the received power) vary between -35.54 dBm and -67.54 dBm, 62% of which stay between -50 dBm and -60 dBm. This reveals a little variation, which can be explained that only line-of-sight radio propagation is involved and this environment is not sufficiently large to have a significant large scale fading.

The measurements by a mobile robot (Figure 6.6b) were conducted in another hall in the AGV factory. The environment is the same as the one in the AGV measurements above, except that the AP has a transmit power of 20 dBm. The robot followed a path of the 2nd order Hilbert curve as the initial design, and some additional paths of straight lines as the sequential design [24]. To map the measured RSSI to a location on the plan, the robot uses a dead reckoning technique

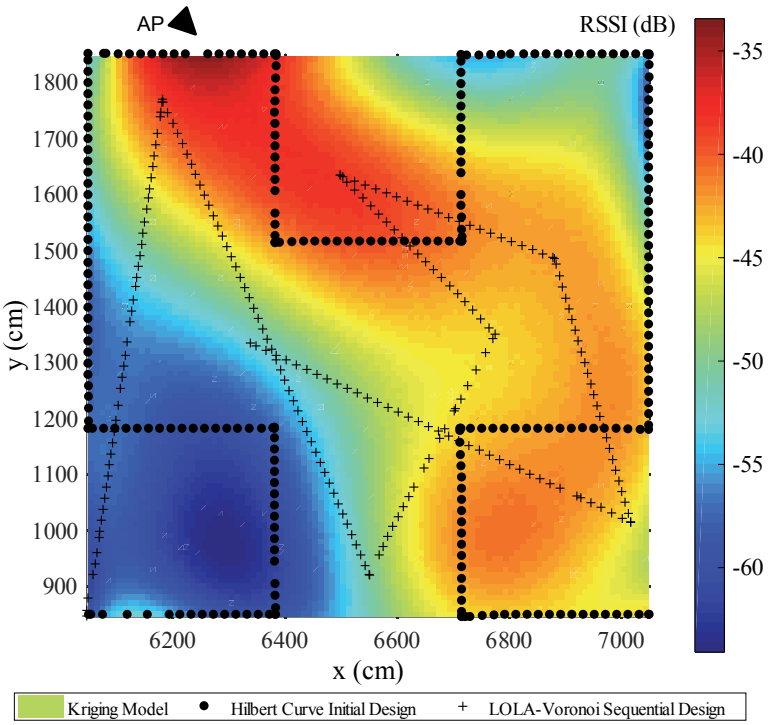
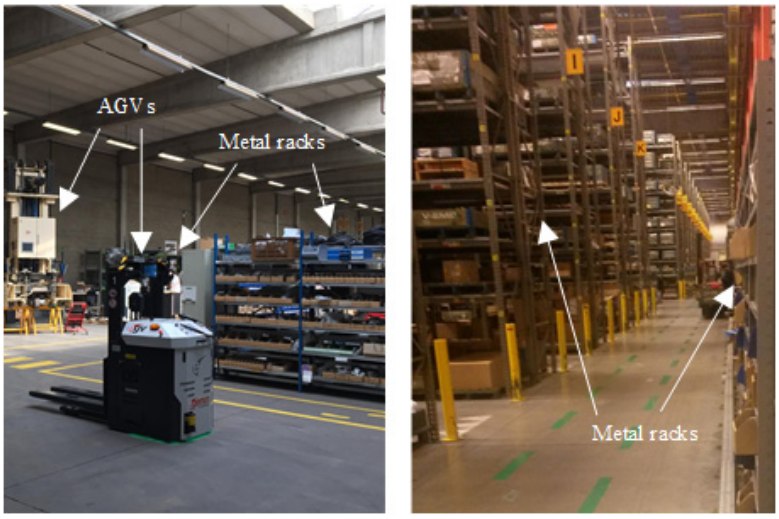


Figure 6.9: The radio environment map (REM), shown on the central controller, specifically indicating RSSI spatial distribution. The access point (5240 MHz, 20 dBm, equipped with a directional antenna of 18 dBi) is placed about 3 m outside the top-left of the target region, and is oriented towards the neighborhood of the point (7000, 1400). The mobile robot automatically performs measurements, by following the 2nd order Hilbert curve (dots) first, and then the LOLA-Voronoi sequential design (crosses). The Hilbert curve measurement path design tries to cover the test field as evenly as possible, whereas the LOLA-Voronoi algorithm focuses on complementary measurements in locations where the RSSI values are highly dynamic.

for estimating its location and thus communicates its location to the central controller in real time. For example, given a start position, the speed and angle from the internal logic are used to determine the new position of the robot. The Rx antenna was placed on the top of the robot. However, its height is not a crucial parameter in this investigation, since the inventory height has little effect on the large scale fading in line-of-sight circumstances [34]. Based on the 719 RSSI samples by using the initial design and sequential design [24], a Kriging surrogate model-based coverage heat map was built and illustrated in Figure 6.9. Such a heat map thus enables an identification of coverage holes in a target environment.



(a) Solution suggested by GATPC (b) Coverage monitored by two robots

Figure 6.10: The two industrial indoor environments for numerical experiments: (a) a factory hall of an automated guided vehicle (AGV) and (b) a warehouse of a car manufacturer. Both environments are full of metal racks, which creates a challenge for radio propagation or robust wireless connection.

6.6 Numerical Experiments

Numerical experiments were further conducted on the proposed GATPC algorithm. A 64-bit Win7 PC was used, with an Intel i5-3470 CPU and 8 GB RAM.

6.6.1 Configurations

The two investigated industrial indoor environments are a factory hall of an automated guided vehicle (AGV) manufacturer and a warehouse of a car manufacturer, both located in Flanders, Belgium.

The AGV factory hall (Figure 6.10a) measures $102\ m \times 24\ m$. It represents a small-scale industrial indoor environment. It is full of metal racks for component storage. Vehicles of varying sizes are usually placed without moving and waiting for integration, maintenance, or shipment. Wide WiFi coverage is needed for vehicle communication and Internet access of the workers' laptops.

The warehouse (Figure 6.10b) measures $415\ m \times\ 200\ m$. It represents a large-scale industrial indoor environment. It is full of metal racks at a height of nine meters. These racks are filled with wooden boxes that contain metal components. Wide WiFi coverage is required to support the voice picking. Human pickers are

Table 6.4: Configurations of the transmit power control (TPC) numerical experiments

Path Loss Model		
PL0		39.87 dB
n		1.78
Shadowing margin (95%)		7 dB
Fading margin (99%)		5 dB
Interference margin		0 dB
Access point (AP)		
Height		2m
Gain		3 dB
WiFi standard		IEEE 802.11n
Transmit power range		{-5:1:7} dBm
Locations		Outputted by over-dimensioning
Number of APs		4 (small-scale environment) 75 (large-scale environment)
Wireless Client		
Height		1.4 m
Gain		2.15 dB
Required physical bitrate		54 Mbps
Required minimal sensitivity		-68 dBm
Environment		
Factory hall	Size (small scale)	2448 m^2 $(102\ m \times 24\ m)$
	Grid point number	2600
Warehouse	Size (large scale)	83,000 m^2 $(415\ m \times 200\ m)$
	Grid point number	83,616
Grid cell size (gs)		1 m
Radio frequency		2.4 GHz
Antenna type		Omnidirectional
Metal rack size		$20\ m \times 3\ m \times 9\ m$
Path loss caused by one metal rack		7.37 dB
GATPC algorithm		
Population size		60 (small-scale environment) 100 (large-scale environment)
Elitism rate		4%
Crossover rate		70%
Mutation rate		40%
Stop criterion		50 iterations

equipped with microphones and earphones. They communicate with the control center via WLANs, to pick up a stuff from and place it to a specific location.

For the TPC model, a metal rack is in both cases an obstacle that potentially causes evident shadowing effects to radio propagation. In the following numerical experiments, an obstacle measures $20\ m \times 3\ m \times 9\ m$. It can be placed on the plan of a target environment either horizontally (i.e., the length side is parallel to

the length side of the environment) or vertically (i.e., the length side is parallel to the width side of the environment). The direction and location of an obstacle were randomly generated following a uniform distribution, with a rack entirely enclosed in the environment. The number of racks is an input of the TPC model. The GPs occupied by obstacles are not considered in the path loss calculation.

The network parameters are summarized in Table 6.4, including the path loss model, AP Tx, Rx, and environment. APs deployed by using the over-dimensioning algorithm such that two full coverage layers are created in the target environment [24, 35]. Each AP has 14 different transmit power levels, including powering off.

As pointed out in [13], the grid cell size (gs) influences the computational accuracy of coverage, and gs should be as small as possible without significantly compromising the computational complexity. Consequently, gs is set to one meter, which is within 10 wave lengths at 2.4 GHz (1.2 m). This means that the path loss within this distance can be considered as constant without sacrificing the precision of path loss calculation [36]. The two parameters $PL0$ and n of the one-slope path loss model are same as these in Section 6.5. The additional path loss caused by a metal rack (7.37 dB) is the mean of measured path loss samples. The GA parameters are tabulated in Table 6.4. The stop criterion is 50 iterations, during which the GA was found to usually stagnate.

In the large-scale WLAN design in [25], the sizes of two environments are $68m \times 59\ m$ and $12\ m \times 67\ m$, and 30 APs are involved. Comparatively, our investigated two environments have sizes of $102\,m \times 24\,m$ and $415\,m \times 200\,m$, and up to 75 APs are involved, which is a hyper-large problem size for optimization.

6.6.2 Effectiveness in Empty Environments

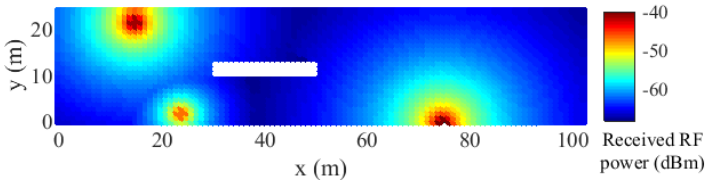
The GATPC was first performed in the small-scale and large-scale environments without any presence of metal obstacles while one full coverage layer was guaranteed ($\mu = 1$). Two other transmit power management schemes were used for benchmarking. One is the RTPC scheme (Algorithm 9). The other is the full power-on scheme, where all APs are powered on with maximal transmit power, i.e., no TPC is deployed.

6.6.2.1 Small-Scale Empty Environment

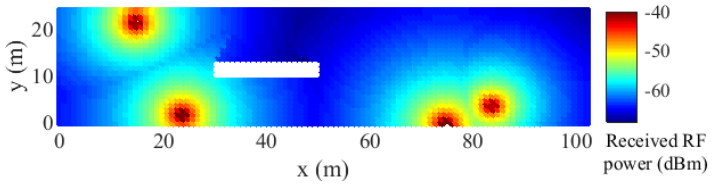
Figure 6.11 presents the coverage maps for the TPC solutions given by the GATPC, the RTPC, and the full power-on schemes, respectively, in the small-scale empty environment. Table 6.5 exhibits the interference produced by these three schemes as well as their runtime in this environment. Overall, the GATPC demonstrates to have notable superiority over the two benchmark schemes, in terms of reducing

transmit power of wireless nodes and minimizing total interference in the network.

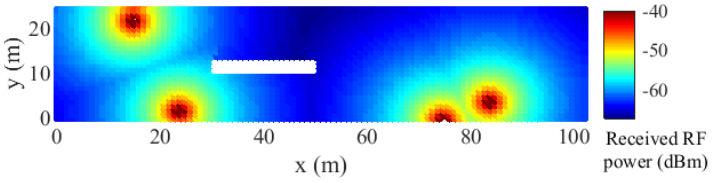
The GATPC significantly decreases the transmit power of two APs while ensuring one full coverage layer in the environment. The transmit power of the four APs in the over-dimensioned IWLAN [35] is -4 dBm, 6 dBm, -3 dBm and 7 dBm, respectively, from the left to the right of Figure 6.11a. In contrast, the RTPC exhibits very limited performance in reducing the redundant transmit power. Its coverage map (Figure 6.11b) is close to that of the full power-on scheme (Figure 6.11c). Its suggested transmit power is 6 dBm, 6 dBm, 6 dBm and 5 dBm, respectively.



(a) GATPC



(b) RTPC



(c) Full power-on

Figure 6.11: Three transmit power control schemes for an empty small-scale environment. The proposed GATPC algorithm is notably superior in reducing the transmit power or coverage of over-dimensioned wireless nodes while ensuring full coverage in the environment.

As indicated in Table 6.5, the GATPC evidently reduces the total interference (-32.04 dBm). In comparison, the RTPC shows limited capacity in mitigating interference. Its interference level (-24.95 dBm) is close to that produced by the worst case (-23.83 dBm in full power-on scheme).

Besides, the runtime of GATPC is short (64 s, Table 6.6). The RTPC has nearly zero runtime (Table 6.5), since optimization is not involved and the environment is small.

Table 6.5: Interference of different transmit power control (TPC) schemes and runtime of RTPC

Environment type		Small empty	Small obstructed	Large empty	Large obstructed
GATPC	Interference (dBm)	-32.04	-26.59	-9.71	-10.02
RTPC	Interference (dBm)	-24.95	-24.97	-9.29	-9.56
	Runtime (s)	0	0	103	131
Full power-on	Interference (dBm)	-23.83	-24.04	-7.02	-7.52

Table 6.6: Speedup performance of the GATPC algorithm using high-performance computing (HPC)

Runtime	Small empty	Small obstructed	Large empty	Large obstructed
With HPC (s)	64	73	102,841	167,504
Without HPC (s)	94	172	3,866,700	4,670,300
Reduction rate (%)	31.9	57.6	97.3	96.4
Speedup times	0.5	1.4	37.6	27.9

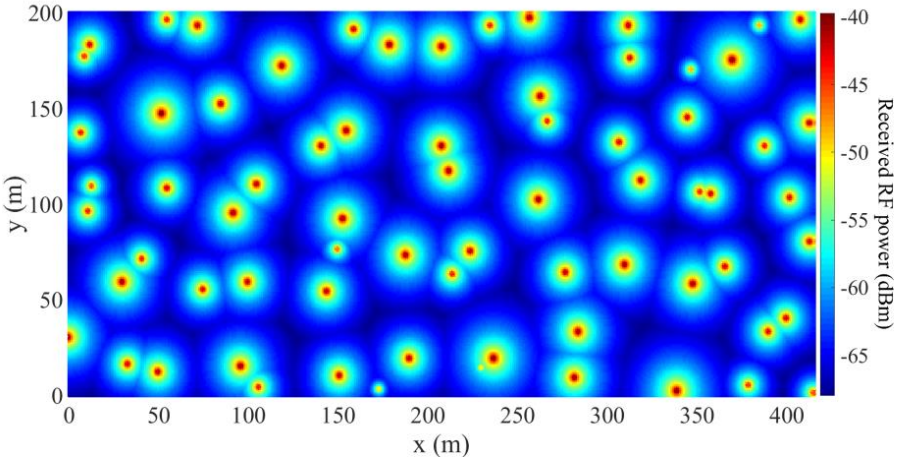


Figure 6.12: Transmit power control solution suggested by the GATPC algorithm for an empty large-scale environment. Among the 75 APs in the over-dimensioned IWLAN [35], four are powered off and most are set to transmit power lower than the maximum, while still having full coverage.

6.6.2.2 Large-Scale Empty Environment

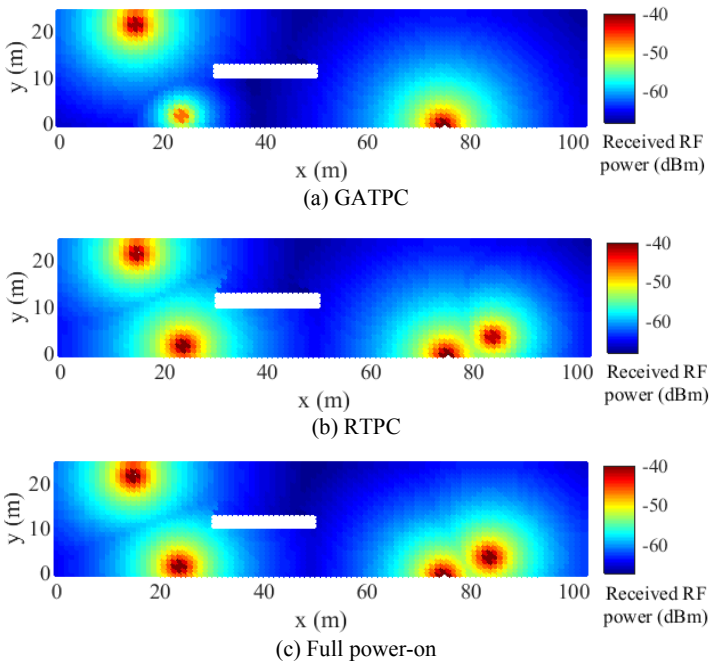
The GATPC exhibits superior interference minimization performance in the large-scale empty environment, compared to the RTPC and the full power-on schemes. It achieves an interference level of -9.71 dBm in comparison to -9.29 dBm for the RTPC scheme and -7.02 dBm for the full power-on scheme (Table 6.5).

The GATPC is effective in AP transmit power reduction (Figure 6.12). Besides the four powered-off APs, most of the powered-on APs are set to a transmit power level that is lower than the maximum (7 dBm, Table 6.4), and many are even set to a level which is very close or equal to the minimum (-5 dBm, Table 6.4).

The runtime of the GATPC significantly increases (102,841 s or about 28.5 h, Table 6.6), in comparison with that in a small-scale empty environment (64 s, Table 6.6). This is explained by the 34 times larger area and the consequently 603 times more AP-GP pairs in the large-scale environment.

6.6.3 Effectiveness in Obstructed Environments

The GATPC was then performed in these small-scale and large-scale environments which are obstructed while one full coverage layer is still guaranteed ($\mu = 1$). To mimic the shadowing effects in industrial indoor environments, one metal rack (Table 6.4) was placed in the small-scale environment and ten in the large-scale environment with a 100% qualification rate (meaning that the GATPC can always



(a) GATPC

(b) RTPC

(c) Full power-on

Figure 6.13: Three transmit power control schemes for a small-scale environment with a metal rack. The proposed GATPC algorithm is evidently superior in reducing the transmit power or coverage of over-dimensioned wireless nodes while ensuring full coverage in the environment.

produce a TPC solution that fully satisfy the required coverage rate with these obstacle settings). The two aforementioned benchmark schemes were also used to measure GATPC's performance.

6.6.3.1 Small-Scale Obstructed Environment

Figure 6.13 presents the coverage maps for the TPC solutions given by the GATPC, the RTPC, and the full power-on schemes, respectively, in the small-scale obstructed environment. Table 6.5 exhibits the interference produced by these three schemes as well as their runtime in this environment.

The GATPC obviously demonstrates superior TPC effectiveness in the small-scale obstructed environment, compared to the other two schemes. According to its output solution, it not only powers off one of the four APs, but also decreases the other two's transmit power (0 dBm and 6 dBm) while keeping the fourth one at the maximum (Figure 6.13a). In contrast, the RTPC scheme exhibits little capacity to reduce the transmit power. Its output TPC solution (Figure 6.13b) is quite close

to that of the full power-on scheme (Figure 6.13c).

The GATPC also shows up as the best in interference mitigation in the small-scale obstructed environment. It suppresses the inference down to -26.59 dBm (Table 6.5). This is lower than -24.97 dBm in the RTPC scheme and -24.04 dBm in the full power-on scheme (Table 6.5).

The runtime of the GATPC is short (73 s, Table 6.6), due to the small scale of the investigated environment. It slightly increases compared to that in the small-scale empty environment. This is because of the additional obstacle loss calculation (Equation (6.6) and Equation (6.7)), though the GPs occupied by the metal rack are excluded in the interference calculation. For the same two reasons explained in Section 6.6.2.1), the RTPC scheme has almost zero runtime.

6.6.3.2 Large-Scale Obstructed Environment

The GATPC also exhibits superiority in interference mitigation in the large-scale obstructed environment compared to the other two schemes. It achieves a total interference level of -10.02 dBm, while the RTPC and full power-on schemes produce interference of -9.56 dBm and -7.52 dBm, respectively (Table 6.5).

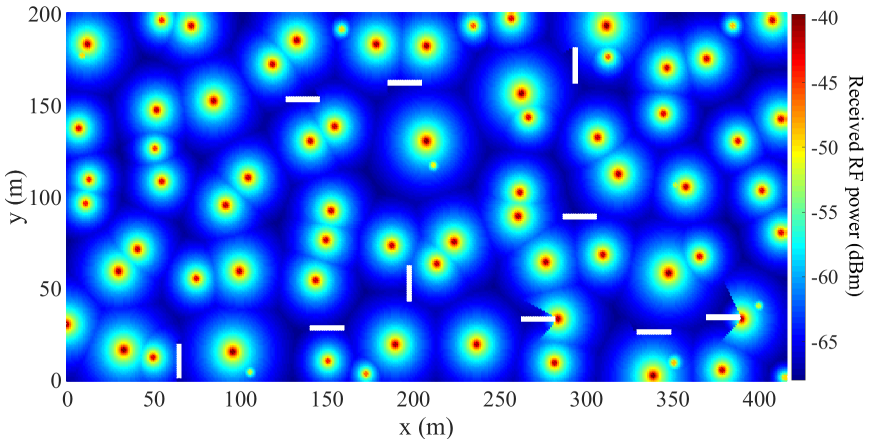


Figure 6.14: Transmit power control solution suggested by the GATPC algorithm for an obstructed large-scale environment (the 10 white rectangles represent 10 randomly placed metal racks). Besides one AP that is powered off, many of the rest APs reduce their transmit power close to the minimum, while still ensuring full coverage.

The effectiveness of GATPC in TPC is further demonstrated in Figure 6.14, which presents the corresponding coverage map. One AP is powered off and three APs are powered on with the minimal transmit power of -5 dBm. Among the APs that are powered on, many have transmit power levels that are lowered close to the

minimum.

For the same two reasons explained in Section 6.6.2.2, the runtime of the GATPC rises to 167,504 s compared to 73 s in the small-scale obstructed environment (Table 6.6). Due to the additional obstacle loss calculation, it is also larger than that in the large-scale empty environment (102,841 s, Table 6.6).

6.6.4 Effectiveness in Speedup

To further benchmark the GATPC's speedup performance, the GATPC without high-performance computing (HPC) was used as a variant version. It includes the parallel processing (Section 6.4.5) and speedup measures (Section 6.4.6). As it turned out to be very time-consuming to obtain an optimized solution in the large-scale environment (at the unit of months), the following means were taken to gauge its runtime.

First, the runtime to generate one random solution in the initial population was measured (at the scale of thousands of seconds). It was then multiplied by the population size to get the total runtime for population initialization. This GATPC variant was rerun by enabling HPC in population initialization (Section 6.4.2) and followed by population evolution (Section 6.4.3 and Section 6.4.4) without HPC. Once the population went through one evolution and its corresponding runtime was obtained (at the scale of ten thousands of seconds), this algorithm stopped and this runtime was multiplied by the number of evolutions to get the runtime for evaluating the entire population. Finally, the estimated overall runtime was the sum of the runtime for the population initialization and that for the population evolutions.

The GATPC with HPC demonstrates significant speedup performance, as presented in Table 6.6. In the small-scale environment, its speedup times stay around one. The runtime of both algorithms are acceptable. However, in the large-scale environment, the speedup times boost to around 30. This makes it feasible to run the GATPC algorithm in a dramatically-reduced time horizon (1 - 2 days), in contrast to the infeasible runtime of the version without HPC. Given that a factory's major layout cannot change too frequently, this optimized runtime is acceptable from the perspective of adapting TPC to a factory layout while minimizing the network interference.

6.6.5 Sensitivity of Quantification Rate

A qualification rate means the probability for this algorithm to intrinsically satisfy the TPC model's fundamental constraint (i.e., coverage, Equation (6.13)) when all APs are powered on with the maximal transmit power. As the "best effort" philosophy (Sections 6.4.2-6.4.4) is applied in the GATPC algorithm, the qualification

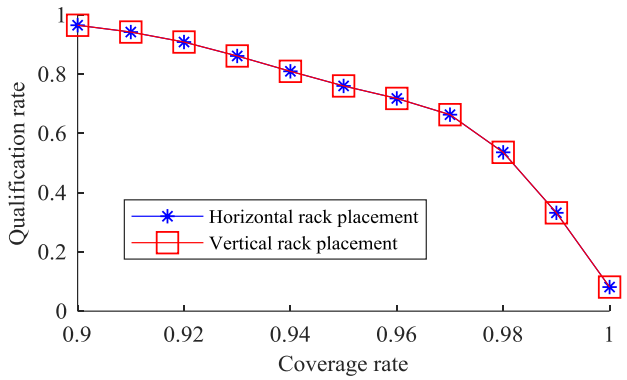


Figure 6.15: Transmit power control (TPC) qualification rate to satisfy the required coverage rate in a metal-dominating environment. For each coverage rate, the rack location iterates over all the possible grid points with horizontal and vertical placement direction. As shown, 90% coverage can be guaranteed by the GATPC algorithm in more than 95% of the shadowing cases.

rate was investigated. During this experiment, the corresponding interference was calculated each time when one metal rack was shifted to a different GP in the small-scale environment. This calculation was iterated over all the possible GPs. Consequently, the relationship between this qualification rate and the required coverage rate was captured.

As depicted in Figure 6.15, 90% coverage can be guaranteed in more than 95% shadowing cases, demonstrating the GATPC's effectiveness in a general obstructed environment. The qualification rate is insensitive to the placement direction of a metal rack. It achieves as high as 96.5% at the coverage rate of 90%. It gradually decreases with the rising coverage rate, and finally drops to 8.1% in the case of full coverage. This decrease is explained by some specific rack locations, where a rack shadows some specific GPs such that no AP can provide effective coverage for them. If a coverage level higher than 90% is desired for 95% of the shadowing cases, this improvement would rely on the over-dimensioning algorithm, instead of the TPC algorithm.

6.6.6 Sensitivity of Interference

The correlation between the interference and required coverage rate was further investigated under a varying number of metal racks placed in the small-scale environment. For each configuration, 30 independent runs were conducted and the average interference was collected, in order to get representative optimization results.

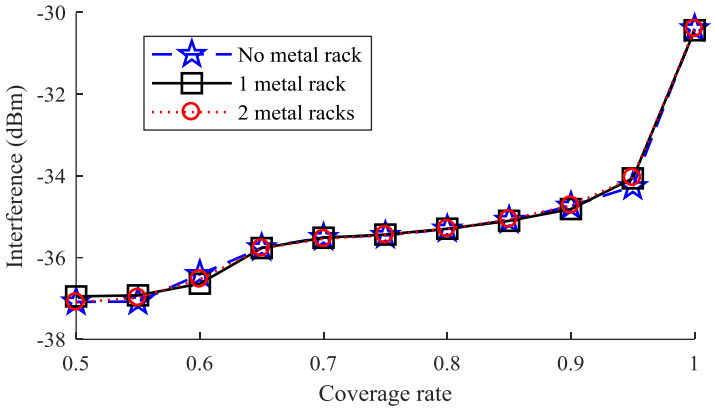


Figure 6.16: Overall network interference under varying coverage rate and number of metal racks.

As indicated in Figure 6.16, the interference declines from about -30 dBm at full coverage to -37 dBm at 50% coverage, regardless of the number of metal racks. This insensitivity to the number of metal racks implies that the limited number of GPs occupied by metal racks does not contribute much to the overall interference.

This drop is explained by the continuously decreased AP transmit power to satisfy the TPC model's coverage constraint which consistently becomes less strict. This is further demonstrated by the optimization results, where the number of APs that are powered off generally increases with the reduction of required coverage rate.

Furthermore, the 10% coverage reduction from 100% to 90% contributes to more than 60% of the overall decreased interference (Figure 6.16). This implies that lowering the required coverage rate to less than 90% cannot be highly effective, because when the coverage rate falls in the range between 90% and 65%, the interference nearly remains stable. This is further proved by the optimization results, where the number of APs that are powered off almost remains 1 when the coverage declines from 95% to 65%. In spite of the slight decrease in interference when the coverage continues to drop from 65% to 50% (Figure 6.16), the seriously-affected coverage should dramatically overweight this gentle decrease. Therefore, when using GATPC, a coverage rate between 90% and 100% not only guarantees a high coverage level for wireless clients, but also is an effective range to control the overall interference.

6.6.7 Performance Comparison with Benchmark Algorithms

Besides the demonstrated effectiveness and scalability of the proposed GA, its performance is further compared with other state-of-the-art algorithms. The bench-

mark algorithms include (1) the proposed GATPC, (2) the proposed GA with the Boolean disk model which is commonly used in literature (GABD) [12, 13], (3) the proposed GA without repair mechanism (GAW), (4) a discrete version of particle swarm optimization (PSO) algorithm [37], and (5) an ant colony optimization (ACO) algorithm [38] which were used in two similar wireless coverage problems, as well as (6) the former PSO with the repair mechanism proposed in this chapter (Algorithm 10), which is named PSOR.

Only the option of powering on/off APs is enabled in the GABD, due to the Boolean disk model. As no repair mechanism is used in GAW, PSO, and ACO, a solution is assigned the maximal fitness (which means the worst solution in this experiment) if it does not satisfy the required coverage rate, aiming to eliminate unqualified solutions. To get tailored for the TPC problem, the ACO in [38] has the following adaptations. The construction graph in Figure 3 of [38] is an $(N_p+1)\times A$ matrix. An ant goes from the leftmost to the rightmost column and selects one vertex in each column. This constructs a TPC solution. The upper bound of the solution ($\widehat{C}$) is the interference when all APs are powered on. The construction rule guides an ant to select the transmit power level of an AP. The pheromone is deposited between two APs and is calculated by Equation (10) in [38]. The heuristic information is based on the increment in the actual coverage rate. The probability to select a transmit power level is calculated by Equation (12) in [38], except that the index *k* refers to a transmit power level of an AP. An ant selects a transmit power level using Equation (13) in [38]. The pheromone is updated by Equation (15) in [38]. The local search procedure is similarly to check and power off redundant APs after a best-so-far solution is updated at the end of each iteration.

For all algorithm instances, the required coverage rate was 100%; the maximal number of iterations was 50; the other configurations remained these in the original literature. 30 runs and 1 run of each algorithm instance were performed for a small and large problem size, respectively, in order to evaluate the optimization efficiency and scalability effectiveness of these algorithms, respectively. The performance comparison of these 6 algorithms was conducted in three dimensions: the interference produced by the optimized TPC solution, the runtime of an algorithm, and the percentage of runs complying with the required coverage rate. An algorithm instance terminated in case of a runtime longer than 48 hours.

Figure 6.17 further presents the convergence and runtime trends of these six algorithms in a small obstructed environment, both of which are averaged over 30 runs to have statistical significance. As demonstrated by Figure 6.17a, GATPC achieves the fastest convergence by rapidly dropping during the first 5 generations, staying nearly stable after the 10th generation, and slightly decreasing at the 26th generation. PSOR has a slower convergence trend compared to GATPC by steadily dropping until the 34th iteration. ACO has a relatively unstable convergence trend.

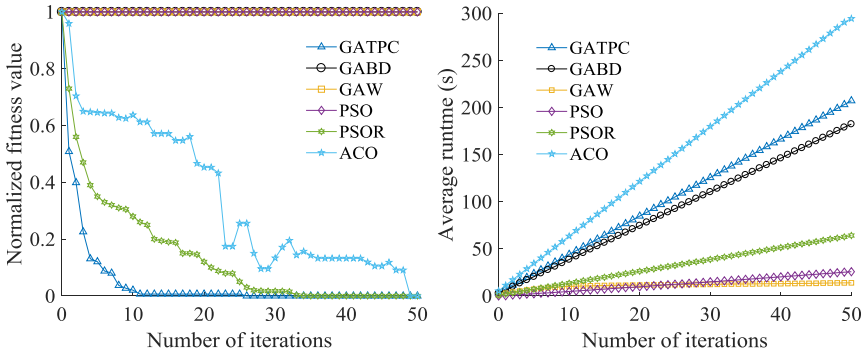


Figure 6.17: Convergence and runtime trends of six optimization algorithms in a small obstructed environment (both trends are averaged over 30 independent runs). A smaller fitness value indicates a superior performance for interference minimization.

It fast drops in the first 4 iterations, and stays on a plateau with a gentle decrease until the 22nd iteration. This process iterates until the 49th iteration. A reason for this phenomenon could be that ACO cannot fully get stable in 50 iterations, revealing its relatively weaker search competence compared to GATPC and PSOR.

On the other hand, GABD, GAW, and PSO cannot effectively improve the finess of the best solution across iterations (Figure 6.17a). This phenomenon is explained by two reasons. For GABD, this would be caused by a lack of fine-tuned transmit power control mechanism, such that the best solution in the initial population cannot be further enhanced by only powering on/off APs. For GAW and PSO, this is due to the missing repair mechanism for unqualified solutions. Consequently, a population cannot make effective progress toward better solutions through iterations.

In terms of average runtime trends of these six algorithms (Figure 6.17b), all algorithms have a runtime which linearly increases with the number of iterations, except GAW of which the rise of runtime slows after the 16th iteration. This common linear increase in runtime is explained by the fact that the number and type of operations performed in each iteration of these optimization algorithms are more or less the same. ACO has the most rapidly-increasing runtime, which is evidently longer than that of other algorithms in each iteration. PSO has the most slowly-rising runtime before the 25th generation compared to the other algorithms, while GAW's runtime rises the most slowly after the 25th generation. GATPC has a runtime which increases slightly faster than that of GABD. This implies that the consideration of fine-tuned transmit power control does not obviously increase the computational burden of an optimization algorithm. However, the consideration of repair mechanism for unqualified solutions evidently slows down the evolutionary search, by comparing the runtime of GATPC and GAW in Figure 6.17b. It also

moderately slows down the particle swarm search, by comparing the runtime of PSOR and PSO (Figure 6.17b). This highlights the computational overhead of repairing unqualified solutions during an optimization process.

Table 6.7 tabulates the performance comparison of these six algorithms on both small and large problem sizes, regarding interference minimization, runtime, and compliance with full coverage requirement. Although GAW and PSO produce solutions with the lowest mean interference (-27.80 dBm and -27.83 dBm) in a small obstructed environment, they cannot fully satisfy the required coverage rate and lead to the highest variance in interference (2.54 dBm and 3.86 dBm, which are both at least one order of magnitude larger than that variance in interference of other algorithms). On the other hand, GATPC and PSOR, which both use the proposed repair mechanism (Algorithm 10), achieve the lowest mean interference (-26.51 dBm and -26.52 dBm) among algorithms that 100% comply with the coverage requirement as well as a low variance in the produced network interference (0.11 dBm and 0.12 dBm). This comparison underlines the undeniable importance of the proposed repair mechanism in reducing ineffectiveness of a search. Comparatively, assigning maximal fitness values to unqualified solutions is not an effective method to search for qualified solutions in the solution space.

Similarly, due to a lack of repair mechanism, GAW and PSO have the shortest mean runtime (14 s and 20 s) of on a small scale (Table 6.7), respectively, while breaking the hard constraint of required coverage percentage. GATPC achieves both moderate average (207 s) and variance (4 s) in runtime on a small scale. A remarkable observation is on PSOR. It has a mean runtime evidently shorter than that of GATPC (80 s vs. 207 s), while remaining a very similar level in interference and full coverage requirement compliance. Conversely, the mean interference and runtime of ACO are inferior on a small scale (-25.47 dBm and 290 s), though it can effectively eliminate unqualified solutions by assigning maximal fitness values to unqualified solutions (implying that these solutions are the worst for parent selection during population evolution). A reason would be that an ACO-based algorithm highly needs a tailored design for a specific problem (e.g., transmit power control in this chapter), in terms of construction graph, construction rule, pheromone management, and local search procedure. These observations of runtime in a small-sized problem in Table 6.7 comply with the runtime trend depicted in Figure 6.17b.

The contribution of the proposed path loss model or the coverage calculation method is also underlined by comparing GABD to GATPC in the small obstructed environment (Table 6.7). As only the option of powering on/off is available in GABD, the produced mean interference is the highest (-25.28 dBm). Its zero variance in interference is also explained by a lack of fine-tuned transmit power control mechanism such that the local optimum can be easily found by randomizing a number of solutions in the initial population and no better solution is found

Table 6.7: Performance comparison of six optimization algorithms for the transmit power control (TPC) problem in small and large-scale obstructed environments (best performance is in bold).

Algorithm	Small obstructed environment[a]			Large obstructed environment[b]		
	Interference (dBm)	Runtime (s)	Coverage requirement compliance	Interference (dBm)	Runtime	Coverage requirement compliance
GATPC[d]	-26.51 $\pm$ 0.11	207 $\pm$ 4	**100%**	-10.02	167,504	**100%**
GABD[e]	-25.28 $\pm$ 0	182 $\pm$ 2	**100%**	n.a.[c]	$>$ 187 h	n.a.
GAW[f]	-27.80 $\pm$ 2.54	**14** $\pm$ 5	85%	-9.72	172,563	100%
PSO[g]	**-27.83** $\pm$ 3.86	20 $\pm$ 3	87%	**-12.85**	**45,170**	72%
PSOR[h]	-26.52 $\pm$ 0.12	80 $\pm$ 5	**100%**	-10.24	105,070	**100%**
ACO[i]	-25.47 $\pm$ 0.07	290 $\pm$ 8	**100%**	n.a.	$>$ 600 h	n.a.

[a] Mean $\pm$ standard deviation are used to evaluate the performance of 30 independent runs.
[b] The optimized solution obtained by one run due to the long runtime required by a large-scale TPC problem.
[c] Unavailable performance data due to a runtime longer than 48 h.
[d] The proposed genetic algorithm-based transmit power control.
[e] The proposed genetic algorithm with the Boolean disk model.
[f] The proposed genetic algorithm without repair mechanism.
[g] The discrete version of particle swarm optimization algorithm without the proposed repair mechanism.
[h] The discrete version of particle swarm optimization algorithm with the proposed repair mechanism.
[i] The ant colony optimization algorithm.

through the evolutionary search. This reason is further justified by Figure 6.17a, where GABD cannot improve the best solution in the initial population with the rising number of generations.

These observations remain in the large obstructed environment (Table 6.7) except two points. Firstly, although the mean runtime of GABD (182 s) is comparable to that of GATPC (207 s) for a small problem size, this gap significantly rises ($>$ 187 h vs. 167,504 s, i.e., more than 4-times difference) for a large problem size. This should be due to the characteristic of the proposed repair mechanism. Once an AP is powered on, it achieves the maximal transmit power, losing the potential to collaborate with other APs for further coverage. This thus triggers lines 18-20 in Algorithm 10 much more frequently to remove the invalid GP-AP links due to a loss of capability to fine-tune the coverage. Secondly, ACO is still the slowest optimization algorithm ($>$ 600 h). But it has an even larger runtime gap with other algorithms. This indicates that ACO especially requires tailored design for large-scale optimization despite the common challenge faced by the other algorithms to work on a large problem size.

6.6.8 Comparison of Wireless Technologies for the Industry

The industrial wireless network planning and reconfiguration solution presented in Chapter 5 and Chapter 6 advances the state-of-the-art of wireless planning and management for robust factory indoor radio communications. Integrated with industrial facilities, the remotely-controllable mobile measurements effectively enable more conscious deployment, and efficient real-time monitoring and problem-solving from the perspective of clients. Here, WiFi is considered as a representative wireless technology and is integrated with this proposed solution for enhancing the robustness of communications in harsh industrial environments. Empirical and simulation studies are both included for the demonstration.

Table 6.8 presents a summary and comparison of representative wireless technologies, which have the potential for factory indoor communications. WiFi in this table refers to the conventional WiFi without the proposed solution in Chapter 5 and Chapter 6. Robust WiFi refers to the WiFi integrating the proposed solution, with the technological improvement aspects in bold in Table 6.8. The comparisons are described as follows, with an emphasis on the position of robust WiFi among the other wireless technologies.

Frequency band and spectrum availability: robust WiFi avoids extra expenditure of purchasing any license for a radio spectrum.

System bandwidth, max data rate, and latency: robust WiFi can achieve the highest communications performance. This provides the largest possibility to satisfy industrial requirements for wireless indoor communications.

Coverage: robust WiFi has a moderate coverage of around 100 m. This adapts

to the general size of a factory.

Network planning: integrated with the proposed over-dimensioning solution (Chapter 5), robust WiFi can have optimized planning and deployment of APs on a target shop floor.

QoS: indicated by the received signal strength that is measured by the proposed "mobile measurement" (Chapter 5), QoS can be maintained in robust WiFi by real-time management from the perspective of clients on a target shop floor.

Managed by operators, ease of deployment & use, and total cost of ownership: robust WiFi enables a network administrator to have a full control or management of the network at a relatively low cost (Chapter 6), without having to update the wireless standard and requiring specified hardware that is commercially unavailable.

Applications for the industry: robust WiFi offers the widest application range for industry to build the necessary information and communications technology infrastructure towards factories of the future.

Measures for robustness: the common techniques used by the other wireless technologies, such as forward error correction (FEC), work under the precondition that the wireless link is well maintained and require a change of the wireless standard. In addition to this, robust WiFi is able to deal with the situation where the link is physically broken or the coverage is shadowed by dominant obstacles in harsh industrial environments.

Among the other wireless technologies in Table 6.8, HetNet currently has few relevant applications for industry and is very costly. There may be a trend of applying HetNet in the industry [39] for some strong reasons like the possibility of using TDMA for guaranteed real-time traffic, sharing of link quality information in both uplink (UL) and downlink (DL), and a performance close to Shannon limit. However, most commercial off-the-shelf devices do not incorporate this type of wireless radio. It is then hard to retrofit HetNet into industrial environments.

Therefore, wireless personal area network (WPAN) and WLAN are more suitable for factory indoor radio communications. The applications include IoT, monitoring, M2M, and so on. Among the four WPAN and WLAN technologies, robust WiFi outperforms the other three concerning the following aspects: (1) providing the highest data rate (1.3 Gbps), lowest latency (1.5 ms), and largest coverage (100 m); (2) over-dimensioning based network planning (Chapter 5), which offers redundancy in APs and radio links for robust communications in harsh environments; (3) real-time maintenance of vendor-independent QoS from the perspective of clients ("mobile measurement" component of the proposed method); (4) having the largest set of radio applications for the industry; (5) dynamically tackling the robustness problem where radio links seriously degrade or break down, or APs encounter failure in the harshness of industrial environments (Chapter 6); (6) possibility to incorporate other cutting-edge research on improving WiFi for in-

Table 6.8: Comparison of potential wireless technologies for factory indoor communications (bold: technological aspects of WiFi that can be enhanced by the proposed method in Chapter 5 and Chapter 6).

Key aspects	WPAN		WLAN	WWAN
Representative technology	Bluetooth 4.2	ZigBee	WiFi	HetNet/small cells
Frequency band	2.4 GHz ISM	2.4 GHz ISM	2.4 GHz ISM &5 GHz U-NII	Up to 4 GHz
Spectrum availability	Unlicensed	Unlicensed	Unlicensed	Licensed & unlicensed
System bandwidth	1, 2 MHz	3 MHz	20, 22, 40, 80, 160 MHz	1.4, 3, 5, 10, 15, 20, 100 MHz
Max theoretical data rate	3 Mbps	250 kbps	1.3 Gbps	DL[a]: 1 Gbps UL[a]: 500 Mbps
Latency	2.5 ms	20-30 ms	1.5 ms	DL: 10 ms, UL: 23.4 ms
Coverage	10 m	10-30 m	100 m	10 m (femtocell) 100 m (picocell) 1000 m (microcell)
Network planning	No	Yes	No	Yes
Quality of service (QoS)	Yes	Yes	No	Yes
Managed by operators	No	No	No	Yes
Ease of deployment & use	Very easy	Easy	Easy	Very hard
Total cost of ownership	Very cheap	Cheap	Cheap	Very expensive
Application for the industry	Cable replacement at a short range, e.g., IoT[a] connection between PDA[a] and headphone	Intermittent data transmissions, e.g., environment monitoring, asset tracking, remote control	Corporate network connection, Internet services, M2M[a], e.g., inventory management, metering, video monitoring	M2M
Major measures for robustness	FEC[a] & ARQ[a] for bit error detection	Frequency agility, mesh, redundancy in sensor data	FEC, STBC[a], OFDM[a]	Channel coding, HARQ[a] SC-FDMA[a], OFDMA[a], advanced MIMO[a]

[a]M2M: ARQ: automatic repeat request, DL: downlink, machine-to-machine communication, FDMA: frequency-division multiplexing, FEC: forward error correction, IoT: Internet of things, MIMO: multiple-input multiple-output, OFDMA: orthogonal FDMA, PDA: personal digital assistant, STBC: space-time block code, SC-FDMA: single-carrier FDMA, UL: uplink.

dustrial usage, e.g., by adding a TDMA MAC to have deterministic packet timing guarantee on packet delivery [40].

More specifically for the above third enhancement aspect, by integrating low-cost measurement setups onto available industrial mobile facilities, such as AGVs, mobile robots, and forklifts, the proposed method can be carried out in a controllable manner from the perspective of users. This is in contrast to commercial off-the-shelf network management tools which reveal information from the perspective of infrastructures, e.g., Cisco Prime Infrastructure, Alcatel-Lucent OmniAccess Wireless Base Software, and Ruckus ZoneDirector. This allows the proposed solution to be directly associated to the QoS from a user's side on the shop floor, further facilitating a proper measure to enhance the industrial wireless robustness.

6.7 Conclusions and Future Work

With the ongoing trend toward factories of the future, IoT penetrates to shop floors and warehouses, which includes not only wireless sensors networks (WSNs) but also other wireless technologies such as wireless local area networks (WLANs). This chapter formulates a transmit power control (TPC) model for dense industrial WLAN (IWLANs). It addresses the drawbacks of existing coverage-related optimization models, by focusing on scalability, simple yet accurate coverage prediction considering three-dimensional (3D) shadowing effects in harsh industrial indoor environments, complete power management schemes (with both powering on/off and transmit power control mechanisms), and empirical validation. It also provides a detailed presentation of "reconfiguration" part of the systematic method proposed in Chapter 5 for robust wireless coverage in harsh industrial indoor environments. To solve this TPC model, this chapter proposes a genetic algorithm based TPC (GATPC). Repair mechanism-based population initialization, crossover and mutation are designed to reduce the GA search redundancy. Parallelism and dedicated speedup measures are further proposed to speed up a GATPC instance for both small- and large-scale optimization.

The GATPC was experimentally validated with a real IWLAN deployed in a small-scale industrial environment, and numerically demonstrated in both small and large problem sizes. The solution quality of the GATPC was proven in terms of effectively conducting adaptive coverage and minimizing interference even in the presence of metal obstacles. The speedup performance of GATPC was measured to be as high as 37 times compared to the serial GATPC without speedup measures. The effectiveness and scalability of GATPC was further demonstrated by comparing to other state-of-the-art algorithms. Through sensitivity studies, the produced interference and qualification rate of GATPC are revealed according to varying required coverage rate as well as the number and placement direction of

dominant obstacles.

The formulated TPC problem and the proposed GAPTC algorithm can also be applied to other types of wireless network besides WLANs, e.g., optimized coverage maintenance of WSNs [16] as well as RFID network planning and configuration [41]. Regarding the future work, further speedup measures or high-performance algorithm design paradigms may be explored to additionally reduce the runtime of the GAPTC. While the path loss model used in Chapter 5 and Chapter 6 considers line-of-sight propagation and obstacle shadowing, further investigations can be performed to demonstrate whether diffraction has a significant influence on the coverage calculation and whether it should also be considered in a path loss model for accurate yet simple wireless coverage prediction.

References

[1] G. Cena, L. Seno, A. Valenzano, and C. Zunino. *On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Industrial Applications*. IEEE Transactions on Industrial Informatics, 6(3):425–437, Aug 2010.

[2] Jiafu Tang, Chengkuan Zeng, and Zhendong Pan. *Auction-based cooperation mechanism to parts scheduling for flexible job shop with inter-cells*. Applied Soft Computing, 49(Supplement C):590 – 602, 2016.

[3] S. Roswitha. *A wireless future for industry*. Technical report, Siemens, 2014.

[4] Margot Deruyck, Emmeric Tanghe, Wout Joseph, and Luc Martens. *Characterization and optimization of the power consumption in wireless access networks by taking daily traffic variations into account*. EURASIP Journal on Wireless Communications and Networking, 2012(1):248, Aug 2012.

[5] Iana Siomina, Peter Värbrand, and Di Yuan. *Pilot power optimization and coverage control in WCDMA mobile networks*. Omega, 35(6):683 – 696, 2007. Special Issue on Telecommunications Applications.

[6] F. G. Debele, M. Meo, D. Renga, M. Ricca, and Y. Zhang. *Designing Resource-on-Demand Strategies for Dense WLANs*. IEEE Journal on Selected Areas in Communications, 33(12):2494–2509, Dec 2015.

[7] Y. Kim, G. Hwang, J. Um, S. Yoo, H. Jung, and S. Park. *Throughput Performance Optimization of Super Dense Wireless Networks With the Renewal Access Protocol*. IEEE Transactions on Wireless Communications, 15(5):3440–3452, May 2016.

[8] Amit P. Jardosh, Konstantina Papagiannaki, Elizabeth M. Belding, Kevin C. Almeroth, Gianluca Iannaccone, and Bapi Vinnakota. *Green WLANs: On-Demand WLAN Infrastructures*. Mobile Networks and Applications, 14(6):798, Dec 2008.

[9] Marco Ajmone Marsan, Luca Chiaraviglio, Delia Ciullo, and Michela Meo. *A Simple Analytical Model for the Energy-efficient Activation of Access Points in Dense WLANs*. In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pages 159–168, New York, NY, USA, 2010. ACM.

[10] P. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi. *Cell Breathing in Wireless LANs: Algorithms and Evaluation*. IEEE Transactions on Mobile Computing, 6(2):164–178, Feb 2007.

[11] W. Ikram, S. Petersen, P. Orten, and N. F. Thornhill. *Adaptive Multi-Channel Transmission Power Control for Industrial Wireless Instrumentation*. IEEE Transactions on Industrial Informatics, 10(2):978–990, May 2014.

[12] Y. Yoon and Y. H. Kim. *An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks*. IEEE Transactions on Cybernetics, 43(5):1473–1483, Oct 2013.

[13] X. Y. Zhang, J. Zhang, Y. J. Gong, Z. H. Zhan, W. N. Chen, and Y. Li. *Kuhn-Munkres parallel genetic algorithm for the set cover problem and Its application to large-scale wireless sensor networks*. IEEE Transactions on Evolutionary Computation, 20(5):695–710, Oct 2016.

[14] Jayanthi Manicassamy, S. Sampath Kumar, Mohana Rangan, V. Ananth, T. Vengattaraman, and P. Dhavachelvan. *Gene Suppressor: An added phase toward solving large scale optimization problems in genetic algorithm*. Applied Soft Computing, 35(Supplement C):214 – 226, 2015.

[15] Gaoji Sun, Ruiqing Zhao, and Yanfei Lan. *Joint operations algorithm for large-scale global optimization*. Applied Soft Computing, 38(Supplement C):1025 – 1039, 2016.

[16] C. P. Chen, S. C. Mukhopadhyay, C. L. Chuang, T. S. Lin, M. S. Liao, Y. C. Wang, and J. A. Jiang. *A Hybrid Memetic Framework for Coverage Optimization in Wireless Sensor Networks*. IEEE Transactions on Cybernetics, 45(10):2309–2322, Oct 2015.

[17] X. Liu. *A deployment strategy for multiple types of requirements in wireless sensor networks*. IEEE Transactions on Cybernetics, 45(10):2364–2376, Oct 2015.

[18] Ning Liu, David Plets, Kris Vanhecke, Luc Martens, and Wout Joseph. *Wireless indoor network planning for advanced exposure and installation cost minimization*. EURASIP Journal on Wireless Communications and Networking, 2015(1):199, Aug 2015.

[19] Maher Rebai, Matthieu Le berre, Hichem Snoussi, Faicel Hnaien, and Lyes Khoukhi. *Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks*. Computers & Operations Research, 59(Supplement C):11 – 21, 2015.

[20] Mohammad Nabi Omidvar, Xiaodong Li, and Ke Tang. *Designing benchmark problems for large-scale continuous optimization*. Information Sciences, 316(Supplement C):419 – 436, 2015. Nature-Inspired Algorithms for Large Scale Global Optimization.

[21] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. *Metaheuristics in large-scale global continues optimization: A survey*. Information Sciences, 295(Supplement C):407 – 428, 2015.

[22] Z. Y. Li and M. S. Zhang. *Pin assignment optimization for large-scale high-pin-count BGA packages using genetic algorithm*. IEEE Transactions on Components, Packaging and Manufacturing Technology, 5(2):232–244, Feb 2015.

[23] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene, and Karel Crombecq. *A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design*. J. Mach. Learn. Res., 11:2051–2055, August 2010.

[24] X. Gong, J. Trogh, Q. Braet, E. Tanghe, P. Singh, D. Plets, J. Hoebeke, D. Deschrijver, T. Dhaene, L. Martens, and W. Joseph. *Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories*. IET Science, Measurement Technology, 10(4):375–382, 2016.

[25] A. M. Gibney, M. Klepal, and D. Pesch. *Agent-Based Optimization for Large Scale WLAN Design*. IEEE Transactions on Evolutionary Computation, 15(4):470–486, Aug 2011.

[26] D. Plets, E. Tanghe, A. Paepens, L. Martens, and W. Joseph. *WiFi network planning and intra-network interference issues in large industrial warehouses*. In 2016 10th European Conference on Antennas and Propagation (EuCAP), pages 1–5, April 2016.

[27] V. Angelakis, S. Papadakis, V. Siris, and A. Traganitis. *Adjacent channel interference in 802.11a: Modeling and testbed validation*. In 2008 IEEE Radio and Wireless Symposium, pages 591–594, Jan 2008.

[28] L. Kong, M. Zhao, X. Y. Liu, J. Lu, Y. Liu, M. Y. Wu, and W. Shu. *Surface Coverage in Sensor Networks*. IEEE Transactions on Parallel and Distributed Systems, 25(1):234–243, Jan 2014.

[29] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[30] Wen mei Hwu. *What is ahead for parallel computing*. Journal of Parallel and Distributed Computing, 74(7):2574 – 2581, 2014. Special Issue on Perspectives on Parallel and Distributed Processing.

[31] X. Li and X. Yao. *Cooperatively Coevolving Particle Swarms for Large Scale Optimization*. IEEE Transactions on Evolutionary Computation, 16(2):210–224, April 2012.

[32] Yong Ming Wang, Hong Li Yin, and Jiang Wang. *Genetic algorithm with new encoding scheme for job shop scheduling*. The International Journal of Advanced Manufacturing Technology, 44(9):977–984, Oct 2009.

[33] Pieter Becue, Bart Jooris, Vincent Sercu, Stefan Bouckaert, Ingrid Moerman, and Piet Demeester. *Remote control of robots for setting up mobility scenarios during wireless experiments in the IBBT w-iLab.t*, pages 425–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[34] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme. *The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz*. IEEE Transactions on Wireless Communications, 7(7):2740–2751, July 2008.

[35] Xu Gong, David Plets, Emmeric Tanghe, Toon De Pessemier, Luc Martens, and Wout Joseph. *An efficient genetic algorithm for large-scale planning of dense and robust industrial wireless networks*. Expert Systems with Applications, 96:311 – 329, 2018.

[36] Simon R. Saunders and Alejandro Aragan-Zavala. *Antennas and Propagation for Wireless Communication Systems*, chapter 5. Wiley, 2005.

[37] M. Hooshmand, S.M.R. Soroushmehr, P. Khadivi, S. Samavi, and S. Shirani. *Visual sensor network lifetime maximization by prioritized scheduling of nodes*. Journal of Network and Computer Applications, 36(1):409 – 419, 2013.

[38] Y. Lin, J. Zhang, H. S. H. Chung, W. H. Ip, Y. Li, and Y. H. Shi. *An Ant Colony Optimization Approach for Maximizing the Lifetime of Heterogeneous Wireless Sensor Networks*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(3):408–420, May 2012.

[39] T. Pötsch, S. N. K. Khan Marwat, Y. Zaki, and C. Gorg. *Influence of future M2M communication on the LTE system*. In 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC), pages 1–4, April 2013.

[40] Margot Deruyck, Jeroen Hoebeke, Eli De Poorter, Emmeric Tanghe, Ingrid Moerman, Piet Demeester, Luc Martens, and Wout Joseph. *Intelligent TDMA heuristic scheduling by taking into account physical layer interference for an industrial IoT environment*. Telecommunication Systems, Jul 2017.

[41] Chuanxin Zhao, Changzhi Wu, Jian Chai, Xiangyu Wang, Xinmin Yang, Jae-Myung Lee, and Mi Jeong Kim. *Decomposition-based multi-objective firefly algorithm for RFID network planning with uncertainty*. Applied Soft Computing, 55(Supplement C):549 – 564, 2017.

Part III

The End

7

Conclusions and Future Research

7.1 Conclusions

In this dissertation, the energy-aware evolutionary optimization is applied to two cyber-physical systems (CPSs) in Industry 4.0: (1) the production system under demand response (DR), and (2) the wireless communication system in harsh industrial indoor environments. The conclusions will be drawn on these two CPSs and the evolutionary optimization.

7.1.1 Production System

Smart grids have recently penetrated into the manufacturing industry, as the industrial energy consumption occupies a significant part in the overall energy consumption of a society. Price-based DR is an important mechanism in smart grids, under which the electricity price is dynamic over time to foster adapted consumption behaviors of the end users. While this mechanism extensively emerges in household applications, the amount of relevant research in the manufacturing industry is very limited. On the other hand, industrial Internet-of-Things (IIoT) is under rapid development in the manufacturing industry. This facilitates the fine-grained mon-

itoring of energy consumption behaviors of machines or production lines and/or other important process variables, which are conventionally unknown to factories. Both emerging developments thus provide new opportunities to scheduling of production systems on the shop floor.

In this context, an energy- and labor-aware production scheduling method was proposed, demonstrated, and analyzed in Chapters 2-4 of this dissertation. Generally, it is an effective and efficient method that enables factories to execute economic production even under time-varying electricity and labor pricing, and even with stringent yet practical timing constraints, e.g., labor shift and due date that must be satisfied and small time slot (e.g., one second) for scheduling and control. This method could serve as a showcase for both IIoT and industrial DR. For IIoT, problem modeling, decision making, and quantitative economic analysis are performed in an enhanced manner, based on empirical data that are collected from the shop floor environment. In this way, the production cost of a factory is reduced and the operation efficiency is improved. For industrial DR, production loads are scheduled in an intelligent manner to adapt to the dynamic electricity price while considering the other important production constraints (e.g., machine interdependency, labor, and timing). Consequently, the energy cost for production is optimized while considering its relation (e.g., trade-off) with other production performance metrics (e.g., makespan and labor cost).

More specifically, Chapter 2 proposed an energy-aware production modeling, simulation, scheduling, and rescheduling method. The energy cost reduction performance of this method has been demonstrated using empirical energy data from a surface grinding machine and three real electricity price schemes, i.e., time-of-use pricing (ToUP), real-time pricing (RTP), and critical peak pricing (CPP). An empirical sensitivity analysis showed that this method reduced the energy cost for production on a single machine by 10% in average and up to 60%, compared to conventional production schedules which have no energy awareness. This energy cost effectiveness was also demonstrated to remain to the maximal extent by energy-aware rescheduling upon unforeseen disruptions during the execution of production jobs. The method proposed in Chapter 2 can be applied to automatically design production schedules for a single manufacturing/re-manufacturing machine and a production line that has a unique major energy consumption process. Electricity prices with more dynamic variations over time will increasingly require automated decision making, and thus enhance the economic impact of this method.

Chapter 3 proposed an integrated energy- and labor-aware production scheduling method under dynamic electricity prices and labor wages over time. This method encompasses production modeling, simulation, optimization, and empirical quantitative analysis. More specifically, it simultaneously schedules jobs and human workers on a single machine that has a set of energy states, taking the due

date as a hard constraint. The operations of the machine and human workers were framed in labor shifts, such that a weekend or a period prohibiting production may split a production job or a machine changeover into multiple subparts. Sensitivity analyses and Pareto front approximation quantitatively revealed the general trade-off relation between the energy cost and the labor cost. The electricity price, the weekend production, the number of jobs, and the load duration were found to be sensitive factors that impact the joint energy and labor cost of production. Compared to a schedule only with energy awareness or labor awareness, such an integrated energy- and labor-aware production schedule demonstrated stable and superior economic performance, regarding energy cost, labor cost, and a sum of these two cost parts. Therefore, both energy awareness and labor awareness are recommended to be integrated in production scheduling to unlock more production cost reduction potential. Although the portion of both cost parts in the overall production cost has case-by-case dependency, the method proposed in Chapter 3 enables better-matched human and machine resources for a single machine or a production line which has a unique bottleneck process, without ignoring the trade-off between these two cost parts.

Chapter 4 proposed a many-objective energy- and labor-aware flexible job shop scheduling method under time-varying electricity pricing and labor wages. Beyond the work in Chapter 2 and Chapter 3, this method extended the investigated production system from a single machine to a flexible job shop, which is one of the most complex shop floor configurations for production. Compared to the prevalent full flexible job shop, five additional considerations have been modeled in this method to enhance its general applicability: (1) the job recirculation (i.e., a job may return to a machine such that multiple operations of a job may be processed by one machine), (2) the partial flexible job shop (i.e., an operation cannot be processed on all machines, or each machine has its own set of capable operations), (3) the operation sequence-dependent machine setups, (4) the state-based energy consumption of each machine, (5) the attached human worker and labor shift on each machine. Compared to the existing production scheduling research which is limited to at most three optimization objectives, this method simultaneously optimizes five production metrics: makespan, total energy cost, total labor cost, maximal workload, and total workload. Through numerical experiments, some important insights on the production system are revealed as follows. (1) It is of economic importance to model the labor aspect in an energy-aware flexible job shop scheduling problem (FJSSP) under dynamic electricity pricing, due to the quantified conflict between the energy cost and the labor cost. Although the portion of both cost parts in the overall production cost varies in different industrial cases, the consideration of both energy and labor aspects in a FJSSP increases the flexibility of this proposed scheduling method. (2) There exists a trade-off relation between the makespan and the total energy cost, as well as between the total

energy cost and the total labor cost, regardless of the dynamic electricity pricing schemes. The trade-off between the former pair of metrics is stronger than that between the latter. (3) The total labor cost has a conflict relation with the maximal workload, while the maximal workload is in a harmonious relation with the total workload. These relations are stronger under RTP than under ToUP.

7.1.2 Wireless Communication System

With the development of IIoT in the manufacturing industry, diverse wireless networks are foreseen to be rapidly and densely deployed on the shop floor and in the warehouses, though the industrial indoor environment is evidently harsh compared to the office or residential environment. It thus becomes an unavoidable challenge for factories to deploy and reconfigure these wireless networks in an economic and robust manner. To tackle this challenge, Chapter 5 and Chapter 6 of this dissertation proposed a systematic method to deploy, monitor, and reconfigure a wireless network in a harsh industrial indoor environment. As an overall conclusion, this method is effective and efficient to achieve this goal.

More specifically, Chapter 5 proposed an over-dimensioning (OD) method for automated and economic planning of dense and robust wireless networks in harsh industrial indoor environments. This method can create two full coverage layers at a large industrial scale for robust wireless coverage even under various three-dimensional (3D) obstacle shadowing effects which are common on the shop floor or in the warehouse. Although the second coverage layer serves as redundancy against shadowing effects, this method reduces the deployment cost by minimizing the number of wireless nodes (e.g., wireless local area network or WLAN as an exemplary wireless network), while respecting the practical constraint of a minimal spatial separation between each pair of wireless nodes. This method was implemented on a computer-based wireless network monitoring system. This system and a small-size wireless network with four over-dimensioned access points (APs) were deployed in a real industrial indoor environment. The effectiveness of this method was empirically validated by monitoring the actual coverage of these four APs. Its effectiveness and efficiency (regarding minimizing the deployment cost while satisfying various defined deployment constraints) were further demonstrated and benchmarked in a set of numerical experiments. Compared to benchmarking planning methods, the solution provided by this method was shown to reduce the deployment cost by 20% to 60%. This method can help network managers and plant managers to automatically plan an IWLAN which has high availability under the presence of dominant obstacles (e.g., production machines, robots, and automated guided vehicles or AGVs) in an industrial indoor environment. Moreover, it can be used to plan other types of robust wireless networks in harsh industrial environments in terms of coverage, such as wireless sensor net-

works (WSNs) and radio-frequency identification (RFID) networks.

Chapter 6 proposed a transmit power control (TPC) method for interference mitigation in dense wireless networks that operate in harsh industrial indoor environments. This method addressed the drawbacks of existing wireless system coverage-related optimization methods, by the focus on method scalability, the simple yet accurate coverage prediction considering 3D shadowing effects of dominant obstacles, the complete power management schemes (with both powering on/off and transmit power calibration mechanisms), and the empirical validation. This method was implemented in a computer-based wireless network monitoring and control system. This system and a wireless network of four over-dimensioned APs were deployed in a real industrial indoor environment. The network control (e.g., configuration of transmit power of an AP and power on/off of a transmitter or Tx) was remotely performed by this central computer system. The effectiveness of this method was empirically validated through controlling this network using the reconfiguration solution provided by this method and monitoring the actual coverage in a distributed manner. The effectiveness and efficiency of this method (regarding interference mitigation while satisfying various defined network reconfiguration constraints) were further demonstrated and benchmarked in extensive numerical experiments. Sensitivity studies showed that the produced network interference by using this TPC method can have the most significant drop when the required coverage rate decreases from 100% to 90%. Besides, 90% coverage over a target area can be guaranteed by this TPC method in more than 95% obstacle shadowing cases. Overall, the method proposed in Chapter 6 can help network managers and plant managers to automatically reconfigure the coverage of a wireless network based on the monitored quality-of-service (QoS), while minimizing the desired wireless network performance metrics (e.g., interference). It can also be applied to various types of wireless networks for coverage-related optimization problems.

7.1.3 Evolutionary Optimization

Despite the large computational intelligence community, evolutionary algorithms (EAs), as a highly representative subdomain, still receive some criticism or doubt from the conventional mathematical optimization community (which usually use mathematical programming for optimization). This typically includes (1) a lack of theoretical convergence proof for the real optimum or the real Pareto optimal front [1], (2) a lack of mathematical expression for the evolutionary search behavior [2], and (3) huge efforts or need for expertise in selecting, implementing, and tuning an EA for a specific problem in hand. On the one hand, this thesis tackles the third criticism, by tailoring a set of EAs and other metaheuristics for five typical optimization problems (Chapters 2-6) on CPSs in Industry 4.0, which

are essentially in the traveling salesman problem (TSP) or vehicle routing problem (VRP), and set cover problem classes. On the other hand, it proposes an adaptive multi-objective memetic algorithm (AMOMA) for multi-objective optimization of a production scheduling problem. This thus highlights the irreplaceable advantage of EAs: a natural enabler for fast and high-quality decision making on highly complex problems or non-deterministic polynomial (NP) hard problems even in a large problem size. In contrast, mathematical programming, as an alternative optimization technique, widely suffers from poor scalability and common intractability. For instance, in [3], the number of binary variables and constraints of a scheduling problem increased with the number of time slots, such that the problem size was only limited to several time slots. The scheduling formulation in [4] was highly dependent on the model, the method, and the objective which were involved. In [5], it took 100,000 seconds to solve a scheduling problem, while some problem instances could not be solved with this time budget. The detailed discussions around EAs are described below.

Compared to designing a specific heuristic on a problem basis, the implementation efforts are insignificant when an EA is extended from one problem to another in the same problem class, thanks to the common domain knowledge and problem structure as well as the open-source frameworks (e.g., MOEA [6] and jMetal [7]). Chapter 2 implemented a genetic algorithm (GA) to solve the single-objective energy-aware single-machine scheduling problem. Permutation was utilized to encode a scheduling solution. Roulette selection, one-point crossover, and swap-based mutation were employed in population evolution. Chapter 3 reused this GA implementation to solve the single-objective energy- and labor-aware single-machine scheduling problem, except (1) the fitness function which was updated according to the new objective, and (2) the job timing method which was adapted to the labor shift. This extension effort is clearly limited compared to starting from scratch or designing a problem-dependent heuristic. The major components of this GA were still reusable when Chapter 3 extended to the nondominated sorting genetic algorithm-II (NSGA-II) for multi-objective optimization, regarding the solution encoding, selection, crossover, and mutation. Chapter 4 tailored the nondominated sorting genetic algorithm-III (NSGA-III) to solve the many-objective energy- and labor-aware flexible job shop scheduling problem. Due to the increased problem constraints and objectives, the implementation efforts relatively rose, in terms of re-designed solution encoding and decoding, crossover, and mutation. Nonetheless, the fundamental principle of tailoring an EA for a scheduling problem remained in Chapter 4: (1) the permutation fits well in solution encoding for a scheduling problem, and (2) the timing issue should be carefully handled in an EA if there are some critical timing constraints.

In addition to the scheduling or TSP/VRP problem domain, Chapter 5 and Chapter 6 shifted to another completely different problem domain: wireless cov-

erage optimization problem or set cover problem. The extension from the GA in Chapter 5 to that in Chapter 6 was also insignificant, although the GA tailoring efforts increased compared to these in Chapter 2 and Chapter 3. In the single-objective dense and robust wireless network planning problem in Chapter 5, three factors play a vital role in increasing the tailoring efforts: (1) the hard constraint of double coverage layers, (2) the hard constraint of minimal AP separation distance, (3) the large-scale optimization. The former two factors significantly increase the difficulty in producing a qualified solution (which satisfies all the constraints). Consequently, a heuristic (which depends on a specific problem) was designed for population initialization, crossover, and mutation, respectively. The last factor makes the execution of every genetic operator computationally expensive. It thus increasingly requires an efficient design and implementation requirement of each genetic operator. To this end, the parallel GA paradigm and the speedup measures were proposed in Chapter 5. The single-objective dense and robust wireless network reconfiguration problem in Chapter 6 shares the structure of the planning problem in Chapter 5, regarding the discretized environment, the dominant 3D obstacles, the Tx, the receiver (Rx), and the coverage. The domain knowledge in Chapter 5 can also be leveraged, e.g., the coverage optimization-related definitions and the propagation model. The large-scale optimization characteristic also exists in Chapter 6. As a result, the parallel GA paradigm proposed in Chapter 5 was reused in Chapter 6, while the population initialization, crossover, and mutation were redesigned and commonly supported by the proposed repair heuristic (for repairing unqualified solutions and thus reducing the redundancy in an evolutionary search).

Furthermore, Chapter 3 proposed the AMOMA, aiming to fast converge toward the Pareto trade-off front without a loss in the population diversity. It synergistically integrates in the NSGA-II a convergence-driven tabu search (CTS) and a diversity-driven tabu search (DTS), respectively. As two local search operators, CTS and DTS are reactively launched upon a cross-dominance-based convergence rate of zero. Besides a premium group for local searches, an ordinary group is used to raise the refinement frequency when no qualified local optimum is found from the former group. This prioritized grouping strategy was demonstrated to evidently stimulate the convergence. With only a 2-min time budget on an ordinary PC (with an Intel Core i5-3470 @ 3.2 GHz and 8 G RAM) and empirical shop floor data, an AMOMA instance was proven to be able to stop even earlier by comparing the monitored and maximal stagnation times. Extensive benchmarking demonstrated the superiority of AMOMA compared to the NSGA-II, the greedy randomized adaptive search procedure (GRASP), and other 4 variants of AMOMA, regarding the number of nondominated solutions, the convergence, and the diversity. In other words, although the real Pareto front is unknown when working with EAs, the approximation set given by the AMOMA is superior to these provided by other

widely-used multi-objective metaheuristics and its own variants. The hybridization philosophy highlighted by the AMOMA can be analogously applied to many other existing EAs and their corresponding appropriate local searches [8, 9]. Besides the proposed novel EA, the single-/multi-/many-objective optimization effectiveness and efficiency of all the tailored EAs in this thesis have also been demonstrated by benchmarking using other widely-used metaheuristics and/or rules-of-thumb, sensitivity analysis, and visualization. Therefore, these investigations strongly demonstrate the fit of an EA for fast and high-quality decision making.

7.2 Future Research

The future extension work can be performed on optimization of these two CPSs. For the production system under dynamic electricity prices, the extension work can follow several directions. (1) As a production system is highly complex and constrained, there is a constant need for more advanced and integrated problem modeling. For instance, while labor is integrated to the energy-aware production scheduling model in this dissertation, other important production aspects can also be modeled in an appropriate manner and integrated to the former scheduling model, e.g., intra-factory transportation, intermediate storage/buffer between machines, machine life time, and machine maintenance. The intention of this shop floor-wide modeling philosophy is not to increase the complexity of a problem itself, but to enhance the flexibility of the model, to reveal more insights on process/system behavior, to diagnose process/system issues, and to test process/system improvement ideas. (2) As fast yet high-quality decision makings are constantly required for scheduling a production system, it is of practical importance to design more novel EAs or metaheuristics, to achieve faster yet higher-quality single-, multi-, and many-objective optimization. The innovation may come from hybridization of existing optimization methods by combining the advantages of each one, design for large-scale optimization, introduction of new nature-inspired paradigms, and so on. (3) Dynamic production scheduling/dispatching deserves more investigations, as diverse disruptions may occur during the execution of production jobs on the shop floor while the relevant research is insufficient. Reactive rescheduling [10, 11] was proposed in Chapter 2 to this end. But it may cause the nervousness in the cyber-physical production systems. As future work, robustness will explicitly be defined to characterize the dynamic scheduling performance, though it is seldom done in literature. Event-driven reactive re-dispatching, cyclic re-dispatching, and dispatching of preventative operations for anticipated random events may be jointly investigated to derive this robust multi-objective dispatching algorithm. (4) It will increase the technology readiness level by applying the proposed energy- and labor-aware production scheduling method to the real pro-

duction on the shop floor and benchmarking by empirical measurements on the production lines. (5) An extension work of setting off/idle modes to a production system can be integration of machine life time model in this energy conservation strategy. The trade-off between energy conservation and machine life time can then be quantified. And machine maintenance can be scheduled before the actual machine failure. (6) Empirical energy modeling will be an important factor that links the whole energy-aware production scheduling research to the physical shop floor environment. Therefore, it deserves more investigations. The extensions include state-based energy modeling with statistical significance, machine life time estimation and machine failure prediction based on the collected power data, as well as integrated scheduling of production jobs and machine maintenance based on the power data-driven life time and failure prediction.

For the wireless system in harsh industrial indoor environments, the extension work can follow several directions. (1) A dedicated indicator may be introduced to quantify the robustness of deployed wireless networks in harsh industrial environments, regardless of the wireless network type. For instance, such a robustness indicator may be defined based on the actual monitored coverage which is correlated with the time and the upper-layer industrial application. (2) The proposed over-dimensioning (OD) model and GAOD algorithm can be extended from double to triple or even more coverage layers. This is especially important for localization and wireless sensor networks which may alternatively switch on/off each coverage layer to remain robust while conserving energy. (3) Additional decision makings can be integrated in the model and solution algorithms, regarding radio frequency planning when designing a wireless network and frequency hopping during the operating phase of a wireless network. (4) Heterogeneous wireless networks can be simultaneously planned by considering other wireless technologies besides a wireless local area network (WLAN). (5) Further speedup measures or high-performance algorithm design paradigms may be explored to additionally reduce the runtime of the GAPTC. (6) While the path loss model used in Chapter 5 and Chapter 6 considers line-of-sight propagation and obstacle shadowing, further investigations can be performed to demonstrate whether diffraction has a significant influence on the coverage calculation and whether it should also be considered in a path loss model for accurate yet simple wireless coverage prediction.

References

[1] K. Sindhya, K. Miettinen, and K. Deb. *A Hybrid Framework for Evolutionary Multi-Objective Optimization*. IEEE Transactions on Evolutionary Computation, 17(4):495–511, Aug 2013.

[2] H. Ishibuchi, N. Akedo, and Y. Nojima. *Behavior of Multiobjective Evolu-*

tionary Algorithms on Many-Objective Knapsack Problems. IEEE Transactions on Evolutionary Computation, 19(2):264–283, April 2015.

[3] Hubert Hadera, Rachid Labrik, Juha Mäntysaari, Guido Sand, Iiro Harjunkoski, and Sebastian Engell. *Integration of Energy-cost Optimization and Production Scheduling Using Multiparametric Programming*. In Zdravko Kravanja and Miloš Bogataj, editors, 26th European Symposium on Computer Aided Process Engineering, volume 38 of *Computer Aided Chemical Engineering*, pages 559 – 564. Elsevier, 2016.

[4] Andres F. Merchan, Hojae Lee, and Christos T. Maravelias. *Discrete-Time MIP Methods for Production Scheduling in Multistage Facilities*. In Zdravko Kravanja and Miloš Bogataj, editors, 26th European Symposium on Computer Aided Process Engineering, volume 38 of *Computer Aided Chemical Engineering*, pages 362 – 367. Elsevier, 2016.

[5] Nikos H. Lappas and Chrysanthos E. Gounaris. *Comparison of Continuous-Time Models for Adjustable Robust Optimization in Process Scheduling under Uncertainty*. In Zdravko Kravanja and Miloš Bogataj, editors, 26th European Symposium on Computer Aided Process Engineering, volume 38 of *Computer Aided Chemical Engineering*, pages 391 – 396. Elsevier, 2016.

[6] David Hadka. *MOEA Framework: a Free and Open Source Java Framework for Multiobjective Optimization*. http://www.moeaframework.org/, 2017.

[7] Juan J. Durillo and Antonio J. Nebro. *jMetal: A Java framework for multi-objective optimization*. Advances in Engineering Software, 42(10):760 – 771, 2011.

[8] Raúl Mencía, María R. Sierra, Carlos Mencía, and Ramiro Varela. *Memetic algorithms for the job shop scheduling problem with operators*. Applied Soft Computing, 34:94 – 105, 2015.

[9] C. C. Liao and C. K. Ting. *A Novel Integer-Coded Memetic Algorithm for the Set k-Cover Problem in Wireless Sensor Networks*. IEEE Transactions on Cybernetics, PP(99):1–14, 2017.

[10] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A generic method for energy-efficient and energy-cost-effective production at the unit process level*. Journal of Cleaner Production, 113:508 – 522, 2016.

[11] Xu Gong, Toon De Pessemier, Wout Joseph, and Luc Martens. *A stochasticity handling heuristic in energy-cost-aware scheduling for sustainable production*. Procedia CIRP, 48(Supplement C):108 – 113, 2016. The 23rd CIRP Conference on Life Cycle Engineering.