An efficient genetic method for multi-objective continuous production scheduling in Industrial Internet of Things

Ke Shen, Joachim David, Toon De Pessemier, Luc Martens, Wout Joseph
Department of Information Technology
Ghent University/IMEC
Ghent, Belgium
ke.shen@ugent.be, joachim.david@ugent.be, toon.depessemier@ugent.be, luc1.martens@ugent.be, wout.joseph@ugent.be

***Abstract*—Continuous manufacturing is playing an increasingly important role in modern industry, while research on production scheduling mainly focuses on traditional batch processing scenarios. This paper provides an efficient genetic method to minimize energy cost, failure cost, conversion cost and tardiness cost involved in the continuous manufacturing. With the help of Industrial Internet of Things, a multi-objective optimization model is built based on acquired production and environment data. Compared with a conventional genetic algorithm, non-random initialization and elitist selection were applied in the proposed algorithm for better convergence speed. Problem specific constraints such as due date and precedence are evaluated in each generation. This method was demonstrated in the plant of a pasta manufacturer. In experiments of 71 jobs in a one-month window, near-optimal schedules were found with significant reductions in costs in comparison to the existing original schedule.**

***Index Terms*—production scheduling; genetic algorithm; continuous manufacturing; multi-objective optimization; industrial internet of things**

I. Introduction

The general definition of continuous production was first proposed in pharmaceutical manufacturing [1], where materials and products are continuously charged into and discharged from the system throughout the duration of the process [2]. From literature review of continuous production in textile dyeing [3], construction [4], steel-making [5]–[7], and food [8], important characteristics are brought out by the comparison with traditional batch processing. Demands of energy and resources are more sensitive in continuous manufacturing, production processes are more susceptible to interference, and manufacturing systems have lower tolerance of fault. Such characteristics make continuous production scheduling difficult to tackle, remaining an obstacle in both industry and academia.

With the application of Industrial Internet-of-Things (IIoT), process control of continuous production encompasses new opportunities and risks [9]: (1) The manufacturing process is more comprehensible, while the production system is more complicated. (2) Problems and room for enhancements are easier to locate, but new uncertainties are introduced into the system with the integration of sensors. (3) Data acquisition is more flexible and undemanding, but analytics on tremendous amount of data require more effort. An ideal IoT solution should accurately and efficiently respond to customer requirements and minimize the interference in normal production processes.

In this paper, a problem of energy-and failure-aware continuous production scheduling with due data and precedence constraints (EFACPDP) is investigated. The problem comes from the workshop of a Belgium's pasta manufacturer, where an IIoT upgrade is in progress [10]. The proposed solution in our previous research [8] does not take into account due date and precedence of jobs, which is often the case in actual production. Shown in [11], these constraints make the scheduling problem difficult. Based on the acquired data form the IIoT framework, an optimization model is constructed. A genetic algorithm (GA) is proposed to search for a near-optimal solution. An introduction of the applied IIoT framework is provided as well. Our research is an early effort to investigate production scheduling in continuous manufacturing, and to integrate the method in IIoT solutions.

II. Literature Review

The studied issues in this paper involve production scheduling problems in continuous manufacturing. Existing schemes in Industrial Internet of Things (IIoT) are further inspected.

A. Continuous production scheduling

Research on continuous production scheduling was accompanied by the study of continuous manufacturing at the very beginning. Reference [1] provided a review of continuous processing in the pharmaceutical industry, including process analytical technology (PAT) for designing, analyzing and controlling manufacturing. Reference [2] presented the changed regulatory environment from batch to continuous processing in the same industry, discussing impacts of continuous manufacturing and opportunities for the future.

Reference [3] studied a scheduling problem for water-saving in the textile dyeing industry. A GA was used to reduce the freshwater consumption by optimization of scheduling based on dyeing color and depth. Such method is incomplete since single objective scheduling has great limitations for multi-purpose requirements. Reference [4] worked on an offsite production scheduling problem of precast components in construction. A hybrid simulation-GA approach was introduced to achieve the conflicting goals under uncertain workflow. The demonstration was implemented using a commercial simulation software, such solution could not be prompted to other scenarios.

Another example is the steelmaking continuous casting (SCC) problem, which is vital for iron and steel production. In literature different approaches were introduced for complicated scenarios of SCC problems: a deflected conditional subgradient algorithm for machine capacity constraints [5], a two-phase soft optimization method for uncertain processing times [6] and a bi-level heuristic algorithm for volatile energy price [7]. These approaches are highly dependent on problem-specific characteristic information, resulting in poor scalability.

Sustainable manufacturing is getting more and more attention because of environmental issues [12], where continuous production also plays an important role in corresponding fields including oil refining, natural gas processing and chemical industry. Reference [13] proposed a power data driven scheduling method minimizing energy cost at the unit process level, demonstrated in a plastic bottle manufacturing plant. Reference [14] integrated labor cost into the model and provided a method with improved performance. However, these methods did not consider uncertainties in the production environment.

From previous literature review it is concluded that an efficient and effective schedule for continuous manufacturing is extremely difficult because of its combinatorial nature and practical complex constraints.

B. IIoT frameworks and architectures

The rapid development of IIoT provides powerful tools for handling production scheduling in continuous manufacturing [15]. A prototype of smart manufacturing using smart interconnection was introduced in [16], supporting energy consumption analysis, production statistics and forecast, physical manufacturing resource (PMR) failure prediction, and processing quality analysis. Key technologies for implementing the prototype were presented but there were no detailed explanations. Descriptions of related technologies were available in other articles: a real time production performance analysis and exception diagnosis model for smart manufacturing was proposed in [17]; a publish/subscribe-based middleware architecture was presented in [18]; an IoT-based cloud manufacturing system and its architecture were proposed in [19].

Another prototype of real-time production logistics synchronization system using smart cloud manufacturing was introduced in [20], where internal production logistics is a specific continuous manufacturing process. A four-layer framework was presented, including smart object layer, smart gateway layer, service layer and application layer.

To sum up, current research of continuous production scheduling does not take into account the benefit of IIoT. Traditional methods require enormous scenario specific information and frequently encounter problems when dealing with uncertainties. IIoT technologies make it possible for adapting these methods in the new environment, which can work as in the service/decision layer.

III. METHOD

The genetic method for the energy-and failure-aware continuous production scheduling with due data and precedence constraints problem (EFACPDP) is proposed in this section. First, the multi-objective optimization model is constructed. Afterwards, details of the algorithm to solve the model is presented.

A. Optimization model

The optimization model is constructed from the measured historical data in the workshop, including production information, machine configuration and power consumption. A combinatorial optimization formulation is applied to build the model, whose notation is presented in Table I.

The objective function is described in equation (1). The weighted sum method is used to scalarize different objectives into one single objective by multiplying user-supplied weight [21]. Therefore the optimization objective of EFACPDP is to minimize the weighted sum of energy cost, failure cost, conversion cost and tardiness cost. The methods for calculating the four cost parts are described in the following equations.

In equation (2), energy cost of each job is calculated using volatile energy price and power of machine in the corresponding time slot. In equation (3), loss of material is considered as failure cost of a job in the face of a machine failure. If a failure happened when processing a job, all materials proceeding on the machine will be lost. In equation (4), conversion cost is charged if two subsequent jobs have different product types, which is also considered the set up cost for such production changeover [22]. In equation (5), tardiness cost is positively linear-correlated with the duration of time exceeding the deadline of a job.

$$min\{\sum_{i=1}^{N}(\omega_E E_i + \omega_F F_i + \omega_C C_i + \omega_T T_i)\} \tag{1}$$

Subject to:

$$E_i = \sum_{t=T_{st_i}}^{T_{ed_i}} D_t P_t \tag{2}$$

$$F_i = h_i \cdot q_i \cdot u_{p_i} \tag{3}$$

$$C_i = \begin{cases} 0 & i = N \text{ or } p_i = p_{i+1} \\ O_{p_i p_{i+1}} & p_i \neq p_{i+1} \end{cases} \tag{4}$$

$$T_i = k \cdot L_i \tag{5}$$

TABLE I: METHOD NOTATIONS

Parameter	Description
J	set of waiting orders
U	set of product type
N	number of waiting jobs
j_i	job with index i
p_i	product type of j_i
q_i	production quantity of j_i
u_i	unit material cost of product type i
T_{st_i}	start timestamp of j_i
T_{ed_i}	end timestamp of j_i
E_i	energy cost of processing j_i
ω_E	customized weight of energy cost
D_t	energy price during time slot t
P_t	power of machine during time slot t
R_t	hazard rate of time slot t
F_i	failure cost of processing j_i
ω_F	customized weight of failure cost
h_i	failure rate of processing j_i
C_i	conversion cost of processing j_i
ω_C	customized weight of conversion cost
O_{ij}	conversion cost from product type i to j
T_i	tardiness cost of processing j_i
ω_T	customized weight of tardiness cost
k	coefficient for calculating T_i
L_i	lateness of j_i
c_i	completion time of j_i
d_i	due date of j_i

$$h_i = 1 - \prod_{t=T_{st_i}}^{T_{ed_i}} (1 - R_t) \tag{6}$$

$$L_i = max(c_i - d_i, 0) \tag{7}$$

$$p_i, p_{i+1} \in U \tag{8}$$

$$T_{st_i} < T_{ed_i}, \quad i \in [1, 2, ..., N] \tag{9}$$

$$T_{ed_i} \leq T_{st_{i+1}}, \quad i \in [1, 2, ..., N-1] \tag{10}$$

Cumulative multiplication is applied in equation (6) on each time slot's hazard rate to calculate the failure rate of a job. Equation (7) shows the definition of lateness. Equation (8) indicates the condition of equation (4). The solution of the optimization model is a permutation (reordering) of jobs with determined T_{st_i} and T_{ed_i} of each job. Time constraints are defined in equation (9) and equation (10).

B. Genetic algorithm

Given a candidate solution π, although its overall cost can be calculated in polynomial time using the equations proposed in this paper, there is no inference that such candidate solution is optimal or not. Therefore the problem has the NP-hardness property, to which genetic algorithm (GA) is a suitable method for a near-optimal solution [23].

The flowchart of our proposed genetic algorithm is presented in Fig 1. The pseudo-code is also given in Algorithm 1. In the initialization stage, the earliest due date (EDD) rule is applied to ensure a good start point [24]. An individual is a vector of size N, whose elements are ordered jobs following the EDD rule. The population size (number of individuals in each generation) n_g is a tuning parameter, larger number of n_g indicates a faster global convergence speed but requires more computational resources [25].

In the evaluation stage, individuals are examined whether they meet hard constraints, including precedence (e.g. j_a must be proceeded before j_b) or deadline (e.g. j_a must finish before

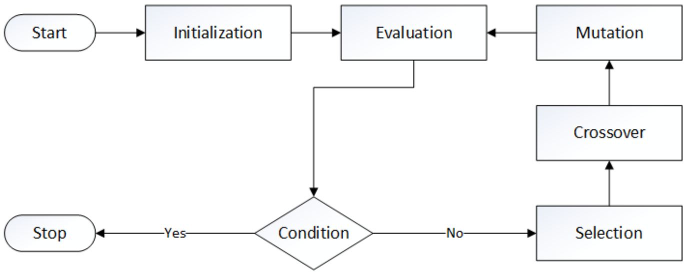


Fig. 1: Flowchart of the provided GA.

Algorithm 1 Improved genetic algorithm (IGA) for EFACPDP

Input: waiting job set J of size N, population size α, crossover rate p_1, mutation rate p_2
Output: near-optimal schedule C with cost R

```
// Initialization
individual ← Apply EDD rule on J
pop ← α × individual
while not Stop Condition do
    // Evaluation
    for i in pop do
        if i not meet constraints then
            i ← best individual in pop
        end if
    end for
    // Selection
    sub_pop ← two best individuals from pop
    winner, loser ← sort sub_pop by fitness value
    // Crossover
    if rand(0,1) < p_1 then
        keep_positions ← choose positions of loser
        keep_jobs ← loser[keep_positions]
        new_jobs ← choose non-repeat jobs from winner
        loser ← combine keep_jobs and new_jobs
    end if
    // Mutation
    if rand(0,1) < p_2 then
        point, swap_point ← choose two points of loser
        swap values of loser[point] and loser[swap_point]
    end if
end while
C ← best individual in pop
R ← cost of C
return C, R
```

timestamp T_a). If some individuals are not valid, they will be replaced by the overwhelming elite (the best individual) of this generation, which is selected according to the rank of fitness (objective) function values. Customized stop conditions comprise limits of an absolute execution time, a certain objective value or a determined iteration number.

Elitist selection [26] is applied in the following stages of genetic operations. Two best individuals are chosen from the current generation and sorted by the fitness value to determine the $winner$ and the $loser$. The $winner$ is kept in the next generation. The $loser$ is replaced by the outcome of crossover and mutation explained in our previous work [8]: mask-based crossover of $winner$ and $loser$ produces a new child, which is afterwards mutated according to random swap points.

Complexity analysis of the algorithm is performed using Bachmann-Landau notations [27]. The initialization stage takes quasilinear time $O(N \log N)$ for sorting using EDD rule. In the evaluation stage, calculation of each type of cost is linear time $O(N)$. Sorting individuals is quasilinear time $O(\alpha \log \alpha)$, afterwards finding the best individual and possible replacement require constant time. Each selection, crossover, and mutation stage also takes constant time. Finding the best individual and the corresponding cost is quasilinear time $O(\alpha \log \alpha)$. To sum up, time complexity of the provided algorithm is $O(N \log N + \beta * (N + \alpha \log \alpha) + \alpha \log \alpha)$, where β is the number of iterations to reach. Normally, β is much larger than N and N is larger than α, therefore the time complexity is reduced to $O(\beta * N)$.

IV. CASE STUDY

A case study was performed in a pasta manufacturer in Belgium, where a one-month production planning problem was investigated for demonstrating the proposed method. An introduction of the applied IoT solution in the customer company is additionally provided by the end.

A. Application of the provided method

The implemented scheduler has two working modes: an expected mode designed for normal use and a demonstration mode compatible with existing data records. The major difference lies in the availability of time-related attributes, including job production durations and machine failures. For a demonstration, determined values can be derived from historical data. For future planning, expectations or predictions for those attributes would be used. This paper illustrates examples from actual industrial scenario using the demonstration mode.

A visualization of existing production records from 2016-11-03 to 2016-11-28 is presented in Fig. 2a. During this time window, 71 jobs from 7 different types (HORENTJE, MACARONI, PENNE, SPIRELLI, VERMICELLI, ZITTI and Other) of pasta were processed on the investigated production line. Small breaks are represented by special jobs with type None. Theoretically continuous manufacturing does not have disruptions, but in reality such uncertainties are inevitable. According to job shop operators, short machine idle periods could be used for real-time maintenance [28], while long idle periods should be prevented. From this existing schedule, room for improvement is detected: (1) Frequent conversions of different types of jobs were performed on the production line, causing high conversion cost. (2) Long breaks happened at a certain time, but during a two-week period there was no real-time maintenance.

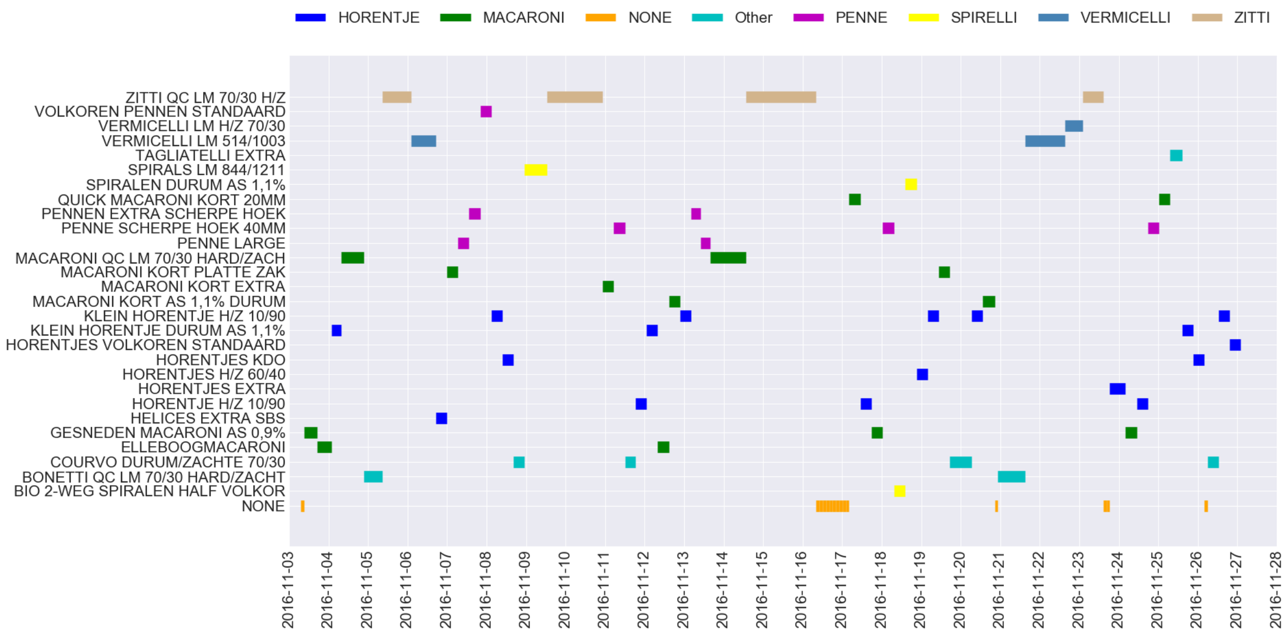
With the same parameter settings ($\omega_E = 1$, $\omega_F = 1$, $\omega_C = 100$, $\omega_T = 1$, where the conversion cost weight is set much larger than other cost weights) and duplicate inputs derived from historical data, the proposed scheduler using Algorithm 1 provides a near-optimal schedule in Fig. 2b. The aforementioned drawbacks of the original schedule are solved: jobs of same types are scheduled subsequently in small groups, and long breaks are divided into small pieces distributed over the entire time horizon. The conversion cost (of the original schedule) is reduced from €4800.0 to €3400.0 (of the near-optimal schedule), saving 29.2%. The total cost is reduced from €5207.7 to €4368.9, saving 16.1%.

The scheduler could also work for a single objective in response to customized requirements. Fig. 3 shows an example of energy-oriented scheduling based on the previous original schedule, with cost weight settings $\omega_E = 1$, $\omega_F = 0$, $\omega_C = 0$, $\omega_T = 0$. Time-of-use pricing (ToUP) policy [29] is applied in the company, where on-peak and off-peak hours are visualized in Fig 3a. Energy consumption of the original schedule and the near-optimal schedule is presented in Fig. 3b and Fig. 3c, where energy consumption of each time slot (and each job) is shown. By rearranging the jobs and real-time maintenance windows, the energy cost is reduced from €402.1 to €386.9, saving 3.78%. The original schedules in Fig 3b is same as that in Fig 2a, while the candidate schedule in Fig 3c is different from that in Fig 2b.

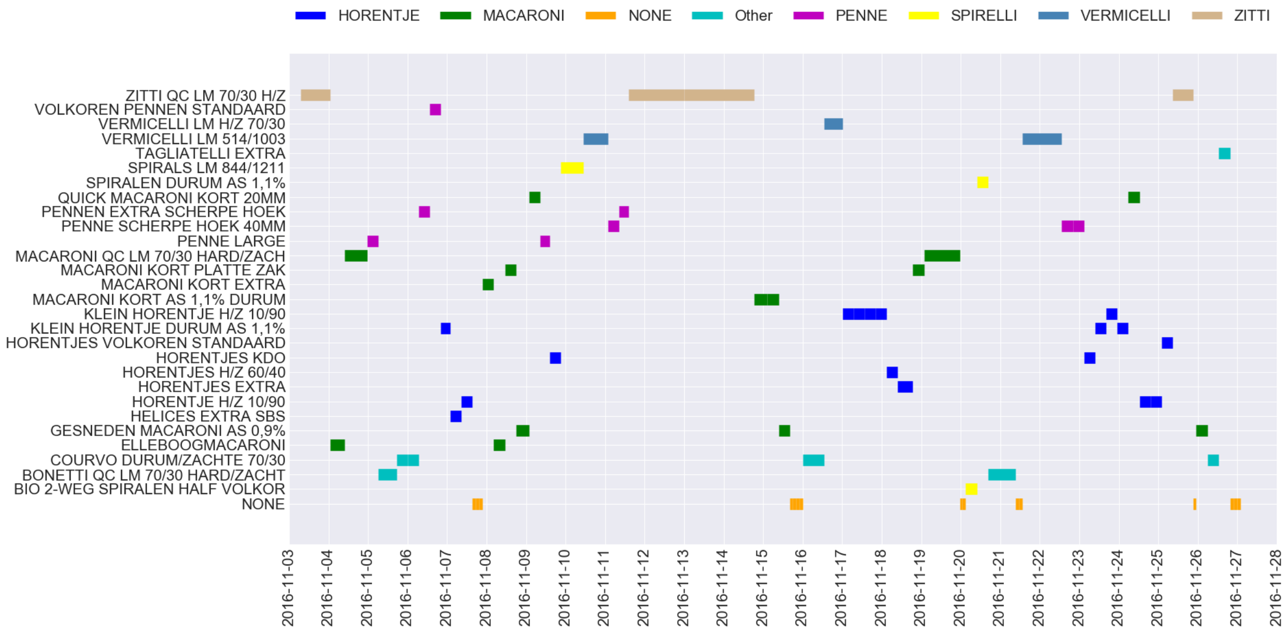
The convergence of the provided GA is guaranteed by the Holland's schema theorem [30]. Search trends of the algorithm in two aforementioned examples are presented in Fig. 4. For each generation, the best, worst, and mean fitness (objective) values of all individuals are calculated and depicted in subfigures. In both examples, the algorithm results in declining mean fitness values when number of iterations grows. The lowest cost value along with the corresponding best individual of the last iteration are returned as a candidate solution. For the multi-objective schedule, the algorithm takes 121.14s on a normal PC (i7-7700 CPU 16G RAM) for 5000 iterations. For the energy-oriented (single-objective) schedule, the algorithm takes 98.55s, respectively. The required duration for execution slightly changes in different trials, therefore the time mentioned above are typical values with ±5s.

An important discussion point is that each run of the program could produce a different result since GA is stochastic. In most cases, larger number of iterations provides a better result. Therefore in the previous examples the algorithm was set to run 5000 iterations, which is a relatively large number to ensure a stable result.

The investigated problem in our previous research [8] can also be solved using the current scheduler with a specific parameter setting ($\omega_E = 1$, $\omega_F = 1$, $\omega_C = 0$, $\omega_T = 0$, where

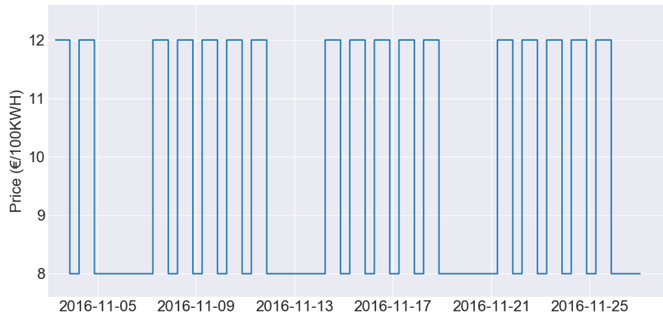


(a) The original schedule in the historical record (Conversion cost: €4800.0, Total cost: €5207.7).

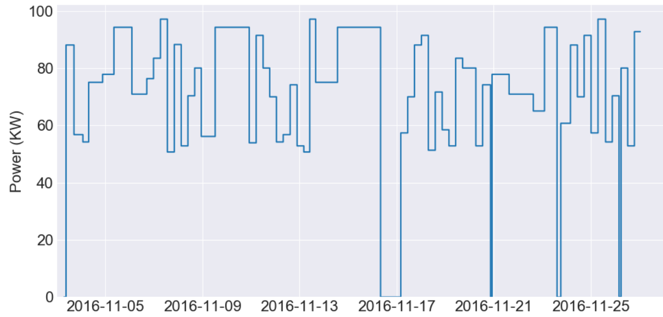


(b) The near-optimal schedule provided by the scheduler (Conversion cost: €3400.0, Total cost: €4368.9).

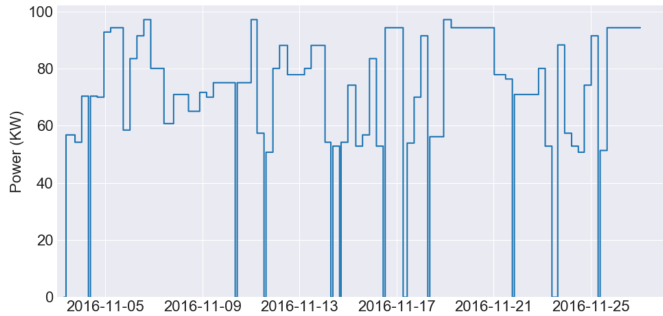
Fig. 2: Gantt charts of a multi-objective schedule ($\omega_E = 1$, $\omega_F = 1$, $\omega_C = 100$, $\omega_T = 1$) from 2016-11-03 to 2016-11-28: (a) The original schedule in historical records (b) The near-optimal schedule provided by the scheduler.



(a) Energy price from 2016-11-03 to 2016-11-28.



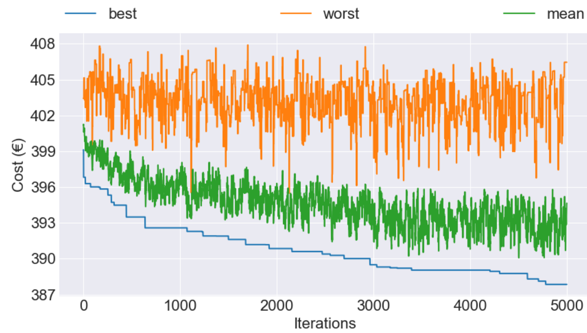
(b) Machine power of each job in the original schedule.



(c) Machine power of each job in the near-optimal schedule.

Fig. 3: Energy consumption of an energy-oriented schedule ($\omega_E = 1$, $\omega_F = 0$, $\omega_C = 0$, $\omega_T = 0$) from 2016-11-03 to 2016-11-28: (a) Time-of-use energy price (b) Machine power of each job in the original schedule from historical records, total energy cost is €402.1 (c) Machine power of each job in the near-optimal schedule provided by the scheduler, total energy cost is €386.9.

(a) Search trend in the multi-objective schedule ($\omega_E = 1$, $\omega_F = 1$, $\omega_C = 100$, $\omega_T = 1$).



(b) Search trend in the energy-oriented schedule ($\omega_E = 1$, $\omega_F = 0$, $\omega_C = 0$, $\omega_T = 0$).

Fig. 4: Search trends of the provided genetic algorithm in two aforementioned examples, represented by the best, worst, and mean objective values over iterations: (a) The multi-objective schedule (b) The energy-oriented schedule.

energy cost and failure cost are considered, conversion cost and tardiness cost are omitted). Two cases were studied to verify the performance of the previously proposed algorithm with time complexity of $O(n^2)$, one small case of 8 jobs in one week and one large case of 1122 jobs in two years. Those studied cases are not applicable in the real production. In this paper, important constraints including due date and precedence of jobs have been considered. The algorithm has also been optimized to $O(n)$.

B. *Introduction of the applied IoT solution*

The implemented scheduler is part of the IoT solution under construction in the customer company, mentioned in Fig. 5. The architecture of the smart manufacturing system is abstracted into three different logical layers. The connection layer is at level 0, where near-field data acquisition sensors are installed on production and packaging lines, using regional wireless network to communicate with controllers and monitors. The control layer is at level 1, service providers (from inside or outside the company) have access to production data and sensor behavior management. In the decision layer at level 2, intelligent analytics are performed based on the data report. Our provided algorithm works in the decision layer, yielding a near-optimal production plan. The data flow and the response stream is represented using solid and dotted arrows. High-level decisions are executed by active sensors in the connection layer, or by job shop operators.

V. Conclusion

The research problem in this paper arises from complicated production environments and different cost saving requirements of manufacturing enterprises. The problem is formulated using a weighted-sum multi-objective optimization model and

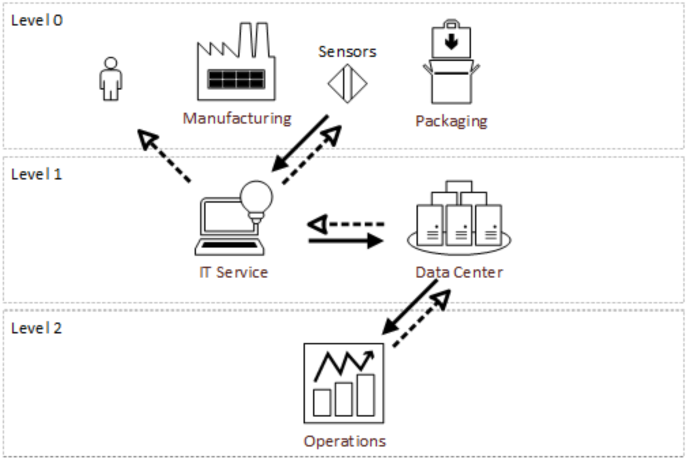


Fig. 5: Design architecture of the applied IoT solution.

solved by an improved genetic algorithm, where non-random initialization, elitist selection and constraint evaluation policies are adopted.

The proposed method is demonstrated in an upgrading IoT solution framework in a Belgian pasta manufacturer, working at the decision level for efficient production planning. In the examples of a one-month schedule, the proposed method obtains a near-optimal solution in fewer than 2 minutes and has significant improvement in results. Given N waiting jobs and β iterations limitation, the algorithm takes linear time $O(\beta * N)$.

For future work, integration of preventive maintenances and predicted machine failure durations into the optimization model for better use of the scheduler's expected mode will be investigated. The communication stream from high to low level of the IoT solution will be further explored for potential extensions of our method.

Acknowledgment

This work was executed within the imec.icon project ELITE, a research project bringing together academic researchers and industry partners. The ELITE project was co-financed by imec and received project support from Flanders Innovation & Entrepreneurship (project nr. HBC. 2017.0149).

References

[1] K. Plumb, "Continuous Processing in the Pharmaceutical Industry", *Chemical Engineering Research and Design*, vol. 83, no. 6, pp. 730-738, 2005.

[2] S. Lee et al., "Modernizing Pharmaceutical Manufacturing: from Batch to Continuous Production", *Journal of Pharmaceutical Innovation*, vol. 10, no. 3, pp. 191-199, 2015.

[3] L. Zhou, K. Xu, X. Cheng, Y. Xu and Q. Jia, "Study on optimizing production scheduling for water-saving in textile dyeing industry", *Journal of Cleaner Production*, vol. 141, pp. 721-727, 2017.

[4] Z. Wang, H. Hu and J. Gong, "Framework for modeling operational uncertainty to optimize offsite production scheduling of precast components", *Automation in Construction*, vol. 86, pp. 69-80, 2018.

[5] K. Mao, Q. Pan, T. Chai and P. Luh, "An Effective Subgradient Method for Scheduling a Steelmaking-Continuous Casting Process", *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 1140-1152, 2015.

[6] S. Jiang, M. Liu and J. Hao, "A Two-Phase Soft Optimization Method for the Uncertain Scheduling Problem in the Steelmaking Industry", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 416-431, 2017.

[7] H. Hadera, I. Harjunkoski, G. Sand, I. Grossmann and S. Engell, "Optimization of steel production scheduling with complex time-sensitive electricity cost", *Computers & Chemical Engineering*, vol. 76, pp. 117-136, 2015.

[8] K. Shen, T. Pessemier, X. Gong, L. Martens and W. Joseph, "Genetic Optimization of Energy- and Failure-Aware Continuous Production Scheduling in Pasta Manufacturing", *Sensors*, vol. 19, no. 2, p. 297, 2019.

[9] E. Sisinni, A. Saifullah, S. Han, U. Jennehag and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724-4734, 2018.

[10] "imec project - ELITE", *Imec-int.com*, 2019. [Online]. Available: https://www.imec-int.com/en/what-we-offer/research-portfolio/elite. [Accessed: 09- Apr- 2019].

[11] M. Pinedo, *Scheduling*. Berlin: Springer, 2016.

[12] M. Akbar and T. Irohara, "Scheduling for sustainable manufacturing: A review", *Journal of Cleaner Production*, vol. 205, pp. 866-883, 2018.

[13] 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", *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016.

[14] X. Gong et al., "Energy- and Labor-aware Production Scheduling for Sustainable Manufacturing: A Case Study on Plastic Bottle Manufacturing", *Procedia CIRP*, vol. 61, pp. 387-392, 2017.

[15] D. Mourtzis, E. Vlachou and N. Milas, "Industrial Big Data as a Result of IoT Adoption in Manufacturing", *Procedia CIRP*, vol. 55, pp. 290-295, 2016.

[16] F. Tao, J. Cheng and Q. Qi, "IIHub: An Industrial Internet-of-Things Hub Toward Smart Manufacturing Based on Cyber-Physical System", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2271-2280, 2018.

[17] Y. Zhang, W. Wang, N. Wu and C. Qian, "IoT-Enabled Real-Time Production Performance Analysis and Exception Diagnosis Model", *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1318-1332, 2016.

[18] W. Kang, K. Kapitanova and S. Son, "RDDS: A Real-Time Data Distribution Service for Cyber-Physical Systems", *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 393-405, 2012.

[19] F. Tao, Y. Cheng, L. Xu, L. Zhang and B. Li, "CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System", *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1435-1442, 2014.

[20] T. Qu, S. Lei, Z. Wang, D. Nie, X. Chen and G. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing", *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1-4, pp. 147-164, 2015.

[21] R. Marler and J. Arora, "The weighted sum method for multi-objective optimization: new insights", *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853-862, 2009.

[22] N. Giannelos and M. Georgiadis, "Efficient scheduling of consumer goods manufacturing processes in the continuous time domain", *Computers & Operations Research*, vol. 30, no. 9, pp. 1367-1381, 2003.

[23] J. Jiang, J. Zhang, L. Zhang, X. Ran and Y. Tang, "Passive Location Resource Scheduling Based on an Improved Genetic Algorithm", *Sensors*, vol. 18, no. 7, p. 2093, 2018.

[24] S. Jiang, M. Liu and J. Hao, "A Two-Phase Soft Optimization Method for the Uncertain Scheduling Problem in the Steelmaking Industry", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 416-431, 2017.

[25] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester: John Wiley & Sons, 2004.

[26] I. Harvey, "The microbial genetic algorithm", *European Conference on Artificial Life*, 2009.

[27] T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to algorithms*. Cambridge (Inglaterra): Mit Press, 2009.

[28] Q. Liu, M. Dong and F. Chen, "Single-machine-based joint optimization of predictive maintenance planning and production scheduling",

Robotics and Computer-Integrated Manufacturing, vol. 51, pp. 238-247, 2018.
[29] 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.
[30] W. Banzhaf and C. Reeves, *Foundations of genetic algorithms*. San Francisco, Calif.: Morgan Kaufmann Publishers, 1999.