8-2016

Best matching processes in distributed systems

Mohsen Moghaddam
Purdue University

Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Mohsen Moghaddam

Entitled

Best Matching Processes in Distributed Systems

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Prof. Shimon Y. Nof
Chair

Prof. Lefteri Tsoukalas

Prof. Steven J. Landry

Prof. Andrew Liu

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Prof. Shimon Y. Nof

Approved by: Prof. Abhi Deshmukh 4/28/2016

Head of the Departmental Graduate Program Date

BEST MATCHING PROCESSES IN DISTRIBUTED SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Mohsen Moghaddam

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2016

Purdue University

West Lafayette, Indiana

For my parents, Eshrat and Ahmad

ACKNOWLEDGEMENTS

Through my incredible “Advance to a Higher Degree” journey at Purdue, I had the privilege to interact with and learn from exceptionally talented and knowledgeable friends, colleagues, and mentors. Their impacts on my professional career and my personal life have been significant and will be everlasting.

I would like to extend my foremost and deepest gratitude to my advisor, mentor, coauthor, and friend, Prof. Shimon Y. Nof, who taught me how to believe in my capabilities, be professional, and always push the boundaries and “think out of the box”. He provided me with opportunities that were out of my wildest dreams, and showed me that through hard work, dedication, and focus, everything is possible. And the lesson that I will always remember from him when I go through tough times is that, no matter what, always have a sense of humor and be grateful.

I also would like to thank the members of my advisory committee, Prof. Lefteri Tsoukalas, Prof. Steven J. Landry, and Prof. Andrew L. Liu, for their valuable inputs and impacts on my PhD research.

My appreciation also goes to my colleagues and good friends at the PRISM Center, especially to Dr. Rodrigo E. Reyes Levalle, Dr. Hao Zhong, Dr. Hyesung Seok, Radhika Bhargava, Jiaxi Li, and Lu Zhang. I also would like to thank Dr. Juan D. Velásquez for his supports and inspiration in the beginning of my career at Purdue, as well as two former

PRISMers and coauthors of my first book ever, Prof. Jose A. Ceroni and Dr. Wootae Jeong. I have been, and will always be, proud to be a part of the PRISM community and the PRISM Global Network.

Last but not least, my gratitude goes to my family for their unconditional love and support throughout this journey—to my mother, Eshrat, my father, Ahmad, and my siblings, Maryam, Masoud, and Mahdi. To those loving people who demanded nothing but my success and happiness, without even being able to see me for almost four years. I will always love you and be grateful for your support. I know it has been tough, but that it was definitely worth it!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

		Page Where First Defined
ALB	Assembly Line Balancing	95
BI	Balanceability Index	117
BMP	Best Matching Problem/Protocol	49
BN	Bottleneck	45
BOMIP	Bi-Objective Mixed-Integer Programming	169
CAF	Collaborative Assembly Framework	49
CCT	Collaborative Control Theory	32
CE	Collaboration Efficiency	102
CLAP	Collaborative Location-Allocation Problem	51
CMAS	Collaborative Multi-Agent System	49
CNS	Collaborative Network of Suppliers	54
CNO	Collaborative Networked Organizations	51
CR	Collaboration Rate	172
CSS	Capacity-Sharing Supplier	80
DCS	Demand and Capacity Sharing	53
DI	Dynamic Inputs	33
DSS	Demand-Sharing Supplier	80
DX	Dimension X (of the PRISM taxonomy of best matching)	33
EA	Evolutionary Algorithm	50
EDD	Earliest Due Date	77
ES	Emergent Sets	33
e-Task	Electronic Task	167
FN	Function	33

FR	Fulfillment Rate	172
GA	Genetic Algorithm	50
GAMS	General Algebraic Modeling System	69
GP	Goal Programming	46
IP	Interdependent Preferences	50
LM	Layered Matching	33
MA	Matching Agent	82
MD	Minimum Deviation	45
MIP	Mixed-Integer Programming	63
MOMIP	Multi-Objective Mixed Integer Programming	96
MPC	Model Predictive Control	74
OS	Overall Satisfaction	33
PBMP	Predictive Best Matching Protocol	49
PR	Precedence Relation	33
p-Task	Physical Task	168
QAP	Quadratic Assignment Problem	41
RA	Resource Agent	76
RC	Resource Constraint	33
RQ	Research Question	10
RS	Resource Sharing	33
RTO	Real-Time Optimization	59
SX	Scenario X	87
SPT	Shortest Processing Time	77
SRAP	Shared Resource Allocation Protocol	76
STOP	Synchronization and Time-Out Protocol	86
TA	Task Agent	75
TAP	Task Administration Protocol	49
TRAP	Task Requirement Analysis Protocol	77
TS-Jr.	Tabu Search with Justice Rule	51
v-Organization	Virtual Organization	166

LIST OF SYMBOLS

CHAPTER 3[†]

Parameters		Page Where First Defined
$\alpha_{i'i}$	Influence of i' on the preferences of i	40
b_j	Capacity limit of j	36
$C_{jj'}$	Cost of resource sharing between j and j'	39
P_{ij}	Mutual preference of i and j	36
$\hat{P}_{ij}$	Interdependent preference of i and j	40
r_{ij}	Resource requirement of i from j	36
$s_{jj'}$	Amount of resources j shared with j'	38
w_X	Weight of X	44
Variables		
χ_{ij}	1, if i and j are matched; 0, otherwise	36

CHAPTER 4

Indices		
n	Product $(n = 1,...,N)$	63
i	Customer $(i = 1,...,I)$	63
j, j'	Supplier $(j, j' = 1,...,J)$	63
t	Period $(t = 1,...,T)$	63
Parameters		
α_i	Quantity of order from customer i	77

[†] Note: Without loss of generality, the notations are listed separately for each chapter, because some are repeated in different chapters with different definitions.

b_{njt}	Backordering cost for product n in supplier j in period t (\$/unit)	63
B_{nj0}	Initial backorder level for product n in supplier j at the beginning of planning horizon (units)	63
D_{nit}	Demand of customer i for product n in period t (units)	63
δ_i	Due date of order from customer i	77
Δ	Prediction interval	83
$f_{jj't}$	Fixed cost of collaboration between suppliers j and j' in period t (\$)	63
h_{njt}	Inventory holding cost for product n in supplier j in period t (\$/unit)	63
I_{nj0}	Initial inventory level for product n in supplier j at the beginning of planning horizon (units)	63
K_{njt}	Capacity upper bound of supplier j for product n in period t (units)	66
o_i	Order from customer i	77
p_{njt}	Production cost for product n in supplier j in period t (\$/unit)	63
$\Pr_{i,\Delta}$	Probability of ordering by customer i during Δ	83
π	Priority	77
τ_i	Processing time of order from customer i	77
$v_{njj't}$	Cost of transshipping product n between suppliers j and j' in period t (\$/unit)	63
M	A sufficiently large positive number	65

Variables

B_{njt}	Backorder level of product n in supplier j at the end of period t (units)	63
G_{njt}	Capacity-demand gap in supplier j in period t regarding product n	64
I_{njt}	Inventory level of product n in supplier j at the end of period t (units)	63
Q_{njt}	Quantity of product n produced by supplier j in period t (units)	63

$T_{njj't}$	Quantity of product *n* transshipped between suppliers *j* and *j'* in period *t* (units)	63
χ_{nijt}	1, if customer *i* is matched to supplier *j* in period *t* for receiving product *n*; 0, otherwise	63
$\lambda_{njj't}$	1, if capacity sharing proposal of supplier *j* is matched to demand sharing proposal of supplier *j'*, for product *n* in period *t*; 0, otherwise	63

CHAPTER 5

Indices

n, n'	Task $(n \in N)$	104
i, i'	Workstation $(i, i' \in I)$	104

Parameters

$\alpha_i(t)$	Progress rate of workstation *i* at time *t*	122
$\bar{C}$	Cycle time upper bound	101
C^B	Cycle time of balanceable line	107
D	Demand	101
δ	Deviation of cycle time from C^B	109
$e_{ii'}$	CE for tool sharing from *i* to *i'*	106
$f_{ii'}$	Fixed tool sharing cost from *i* to *i'*	104
IP_n	Set of immediate predecessors of task *n*	105
$O_k(t)$	Set of non-target workstations with higher workload than *k*'s targets at time *t*	122
ρ_n	Processing time of task *n*	101
T	Available production time	101
$\overline{W}$	Upper bound for the number of workstations	101
$\underline{W}$	Lower bound for the number of workstations	101
$w_i(t)$	The overall workload of workstation *i* at time *t*	122

Variables

β	BI (Balanceability Index)	

C	Cycle time	104
$s_{ii'}$	Tool sharing between workstations *i* and *i'*	106
W_t	Number of workstations in period *t*	104
χ_{ni}	1, if task *n* is matched to workstation *i*; 0, otherwise	104
$\psi_{ii'}$	1, if workstation *i* shares tools with workstation *i'*; 0, otherwise	105
Z_1	Objective 1: Number of workstations	104
Z_2	Objective 2: Cycle time	104
Z_3	Objective 3: Total collaboration cost	104

CHAPTER 6

Indices

i,k	Set I	139
j,l	Set J	139

Parameters

$\mathbf{c}$	Chromosome (solution set)	140
$\mathbf{c^m}$	Modified chromosome	148
P_{ij}	Mutual preference of $i \in I$ and $j \in J$	140
$\hat{P}_{ij}$	IP of $i \in I$ and $j \in J$	140

Variables

χ_{ij}	1, if $i \in I$ and $j \in J$ are matched; 0, otherwise	141

CHAPTER 7

Indices

$o \in O$	Organizations	171
R_t	Set of eligible resources for processing task $t \in T$	171
$r \in R$	Resources	171
$t \in T$	Tasks	171
T_o	Set of tasks associated with organization $o \in O$	171
TL	Tabu list	171

Parameters

ABSTRACT

Moghaddam, Mohsen. Ph.D., Purdue University, August 2016. Best Matching Processes in Distributed Systems. Major Professor: Shimon Y. Nof.

The growing complexity and dynamic behavior of modern manufacturing and service industries along with competitive and globalized markets have gradually transformed traditional centralized systems into distributed networks of e- (electronic) Systems. Emerging examples include e-Factories, virtual enterprises, smart farms, automated warehouses, and intelligent transportation systems. These (and similar) distributed systems, regardless of context and application, have a property in common: They all involve certain types of *interactions* (collaborative, competitive, or both) among their distributed individuals—from clusters of passive sensors and machines to complex networks of computers, intelligent robots, humans, and enterprises. Having this common property, such systems may encounter common challenges in terms of suboptimal interactions and thus poor performance, caused by potential *mismatch* between individuals. For example, mismatched subassembly parts, vehicles--routes, suppliers--retailers, employees--departments, and products--automated guided vehicles--storage locations may lead to low-quality products, congested roads, unstable supply networks, conflicts, and low service level, respectively. This research refers to this problem as *best matching*, and investigates it as a major design principle of CCT, the Collaborative Control Theory.

The original contribution of this research is to elaborate on the fundamentals of best matching in distributed and collaborative systems, by providing general frameworks for (1) Systematic analysis, inclusive taxonomy, analogical and structural comparison between different matching processes; (2) Specification and formulation of problems, and development of algorithms and protocols for best matching; (3) Validation of the models, algorithms, and protocols through extensive numerical experiments and case studies. The first goal is addressed by investigating matching problems in distributed production, manufacturing, supply, and service systems based on a recently developed reference model, *the PRISM Taxonomy of Best Matching*. Following the second goal, the identified problems are then formulated as mixed-integer programs. Due to the computational complexity of matching problems, various optimization algorithms are developed for solving different problem instances, including modified genetic algorithms, tabu search, and neighbourhood search heuristics. The dynamic and collaborative/competitive behaviors of matching processes in distributed settings are also formulated and examined through various collaboration, best matching, and task administration protocols. In line with the third goal, four case studies are conducted on various manufacturing, supply, and service systems to highlight the impact of best matching on their operational performance, including service level, utilization, stability, and cost-effectiveness, and validate the computational merits of the developed solution methodologies.

CHAPTER 1. INTRODUCTION TO MATCHING PROBLEMS

1.1 Research Motivation

Every distributed system, natural or artificial, involves certain types of interactions between its entities—from nerve cells, colonies of ants, and flocks of birds to complex networks of sensors, machines, robots, humans, and enterprises. These interactions can be collaborative (common goals), competitive (conflicting goals), or both. To ensure high quality of interactions, it is necessary for each individual entity to know *with whom to interact*, *how*, and *when*. Potential *mismatch* between those entities may lead to inefficient and suboptimal interactions, which in turn diminishes their competitive performance with respect to critical criteria such as time, cost, quality, flexibility, and stability. To ensure competitiveness, therefore, the interactions must be optimized through "best matching" of individual entities to each other; *e.g.*, bolts--nuts (selective assembly); suppliers--retailers--customers (enterprise network design); jobs--machines/--computers (scheduling); vehicles--routes (transportation planning); sensors--locations (sensor network design); interns--factories (recruitment); robots--teams (team formation).

Matching is a classic yet significant problem spanning almost every area of science, technology, engineering, mathematics, economics, and management. It is done with respect to the mutual interaction of individuals, their desirability or preferences for each other, and certain conditions. Matching preferences are diverse and context-dependent; *e.g.*, dimensional tolerance (bolts--nuts); lead-time/cost/quality (suppliers--customers); tardiness/makespan (jobs--machines); delivery time/cost (vehicles--routes); communication cost/energy consumption (sensors--locations); employee/employer satisfaction (interns--factories); conflict rate/resilience (robots--teams). Similarly, the conditions that influence a best matching process are context-specific, *e.g.*, limitation on the number of interns that a factory can admit; precedence relations among a set of jobs allocated to a single computer; lateral collaboration among suppliers through demand and capacity sharing; interpersonal biases, emotions, and relational messages among members of a social network.

Matching is a well-known combinatorial optimization problem that roots in various natural or artificial system. The problem, however, is not new and has been extensively studied for decades. It was firstly introduced by D.F. Votaw, Jr. and A. Orden in 1952 as the *assignment problem*, which involves matching the elements of two sets on a one-to-one basis such that the sum of their associated weights is minimized. Several models and algorithms have been developed since then, led by pioneering works such as Hungarian algorithm by Kuhn (1955) and deferred acceptance algorithm by Gale and Shapley (1962). Matching, as defined by Oxford dictionaries, refers to [the process of] "*corresponding or causing to correspond in some essential respect*", "*making or being harmonious*", or "*being equal to* (*something*) *in quality or strength*". Accordingly, *best* matching is defined as follows:

Definition 1.1. *Best matching*. It refers to the process of finding the best match between two or more sets considering certain conditions and criteria.

This definition implies a scope and impact for the matching problem even broader than the classic assignment problem, incorporating other significant problems such as scheduling, supplier selection, location-allocation, routing, clustering, team formation, partitioning, and so on. Nevertheless, there are some critical conflicts and shortcomings in representation of these diverse problems as matching problems, their comparative analysis, and potential extensions, which have motivated this research. The *analogy* between different and independent instances of matching is not fully utilized, for understanding and solving the existing problems, and for identifying and formalizing new problems that belong to this family of problems. More importantly, there is no clear "big picture" of the problem—no taxonomic framework for synthesis and comprehensive study of the problems as a whole. Furthermore, there are several areas that—despite importance—have received insufficient attention. The goal of this research is to elaborate on the fundamentals of best matching in distributed systems by providing solid frameworks for

- Systematic analysis of various best matching processes from different dimensions, and identification of new classes of the problem.
- Comprehensive taxonomy, analogical and structural comparison between different best matching problems and processes.
- Systematic identification of systems' hierarchy, distribution of decision-making and control functions, and nature of interactions among distributed individuals.
- Practical formulation of solutions based on a comprehensive set of tools—best matching algorithms and protocols.
- Validation of the developed concepts, models, algorithms, and protocols through extensive numerical experiments and case studies.

1.2 Definitions and Assumptions

Every distributed system involves interactions between its individual entities, and every interaction involves certain best matching processes. Although understanding of the entire scope of the problem is not easy (or even necessary for this research), some systems engineering related examples can be helpful in realizing the notions of interactions and matching in the context of this research, before moving forward with more elaborate definitions of the problem. Consider a manufacturing system with multiple layers and various cyber-physical components and collaborative units (Figure 1.1).

- The AS/RS (Automated Storage/Retrieval System) must identify the best storage locations for different parts/products. AS/RS is typically used in applications with

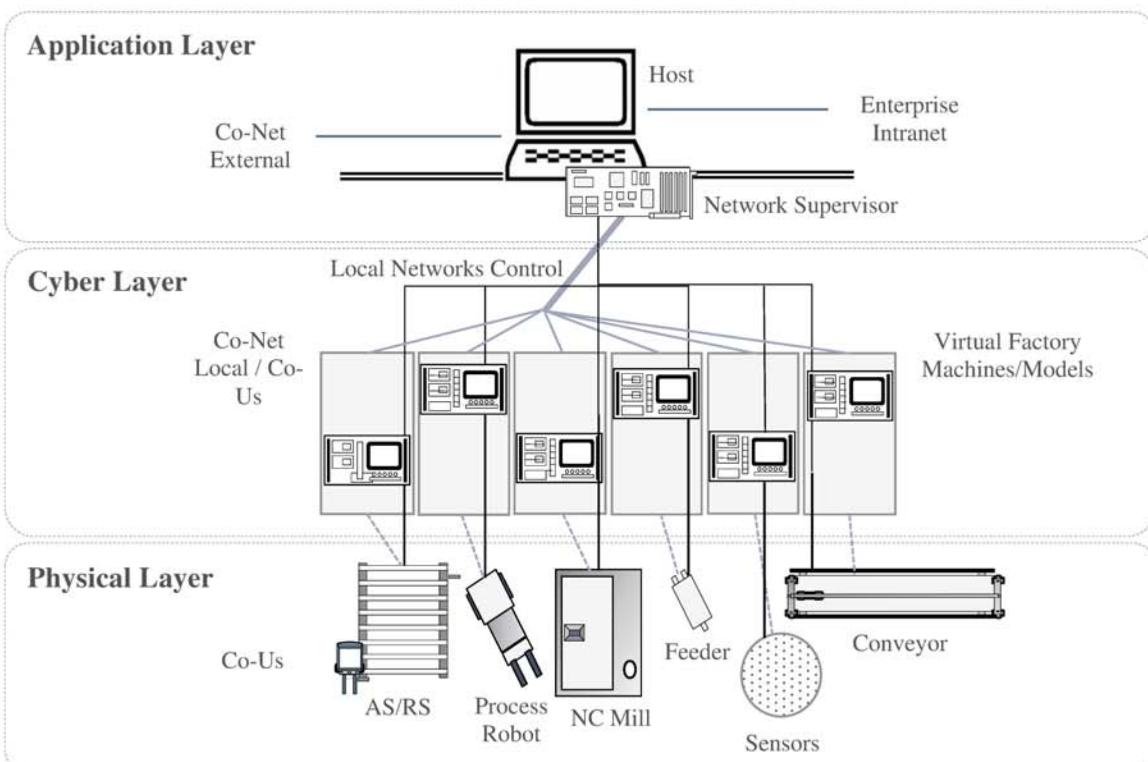


Figure 1.1.1. A cyber-physical manufacturing environment.

high volumes/rates of arrival and retrieval, strict capacity limits, and/or critical accuracy requirements. In such systems, best matching can play a significant role in enhancing efficiency and productivity.

- The process robots must be assigned the most suitable sets of tasks. Automated Guided Vehicles (AGVs), as an example of process robots, are responsible for carrying materials around the shop floor or warehouse. When multiple AGVs are working in the same facility, it is necessary to define who does what, how, and when, in order to minimize processing times and prevent conflicts (*e.g.*, collision). This can be done through dynamic best matching between materials and AGVs.
- The facility sensors must find the best peer sensors based on their network configuration/communication protocols to minimize energy consumption. Facility sensor networks are formed in different manners (*e.g.*, single-multi-hop with/without clustering), following certain communication protocols (*e.g.*, point-to-point; flooding; gossiping), in an attempt to minimize energy consumption through sending/receiving messages. One of the foremost processes in optimal configuration of sensor network as well as development of efficient communication protocols is best matching (*e.g.*, between sensors; regions; clusters; cluster heads; base stations).
- The feeder lines must rearrange/combine different component parts based on their similarity/affinity/dimensional tolerance. The classic bolt--nut best matching process is a good example to show the significant impact of best matching on selective assembly. Matching the component parts that arrive in the primary assembly line from different feeder lines (*e.g.*, based on tolerance) can remarkably

improve the quality of final products (even in presence of manufacturing deficiencies and inaccuracies), and reduce the need for rework.

- The Numerical Control (NC) milling machines must define the best tools with respect to the assigned jobs. Milling machines use various rotary cutters to remove materials from the surface of a work-piece. The rotary cutters differ in shape, size, material type, flutes/teeth, helix angle, coating, shank, *etc*., where each is suitable for different types of work-pieces. The wide range of choices and features of rotary cutters provide challenges the NC program to match the best tool to each work-piece (based on the features of both). The problem becomes even more challenging when multiple NC milling machines collaborate by sharing tools and the number/ diversity of the tools is limited.
- The virtual factory manager must identify the best virtual machine(s)/model(s) with respect to their workload/capabilities as well as the command type. The main purpose of virtual factories is to enable innovative and cost-effective production through adaptive design, virtual modeling and simulation, automated monitoring of products, processes, and factories, and knowledge integration. In this context, dynamic and optimal matching of modeling, analysis, and decision-making tasks to distributed resources (*e.g.*, computer agents; programs; machines) with various workloads and limited capacities is the key for enhancing the productivity and flexibility of the cyber and physical layers of the manufacturing system.
- The manufacturing site must categorize its suppliers based on their quality, trustworthiness, on-time deliveries, and cost. At a macro level, the manufacturing enterprise—encompassing all the aforementioned elements—must interact and

collaborate with suppliers of parts and raw materials and perhaps other similar manufacturing enterprises (*e.g.*, lateral demand and capacity sharing). In order to minimize the procurement risk, enhance quality and service level to the customers, and improve flexibility in dealing with variations in demand and spot market price, the enterprise must identify the best portfolio of suppliers for each component part based on various supplier selection criteria.

All the aforementioned (and similar) examples—in spite of their differences—have inherent matching elements that can be formalized, formulated, and optimized in a standard and general way. The interaction between each pair of individuals i and j can be formalized via a matching variable χ_{ij}, where

$$\chi_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are matched,} \\ 0, & \text{otherwise.} \end{cases}$$

The matching indeed entails certain rewards (or costs), which can be represented as mutual preferences of individuals i and j for each other, *i.e.*, P_{ij}. The following assumptions are considered throughout this dissertation.

General Assumptions

1. All input parameters, including P_{ij}, are given and known. (In some case studies, however, the uncertainties associated with the input parameters are considered and incorporated in models using different methods.)
2. The term "network" has been used several times throughout the dissertation. In the context of this research, a network refers to *a group or system of interconnected individuals* (*e.g.*, *supply network*; individuals: suppliers, retailers, customers).

3. The term “best match” is used to distinguish the best match between individuals from other potential (and no necessarily optimal) matches.

A problem with only two individuals i and j to match is trivial—the binary decision is made by merely deciding whether the respective reward (or cost) is sufficient (acceptable) or not. As the size and complexity of systems increase (and so does the possibilities of matching), however, it becomes more difficult to define what/who is the best match for what/whom. Consider a selective assembly system, where two sets of bolts I and nuts J must be matched to each other on a one-to-one basis. In this case, the quality of final products depends on the dimensional compatibility of the matched bolts and nuts, which can be formalized by P_{ij}, and is unique for each pair of bolts and nuts. In addition, the numbers/possibilities of matching are limited, as each bolt/nut must be matched to exactly one counterpart. Hence, the best match between the sets of bolts and nuts can be obtained by solving the following integer program that maximize the total quality of match with respect to the conditions for one-to-one matching:

$$\begin{aligned} \max \quad & \sum_{i \in I} \sum_{j \in J} P_{ij} \chi_{ij}, \\ \text{s.t.} \quad & \sum_{i \in I} \chi_{ij} = 1, \quad \forall\, j \in J, \\ & \sum_{j \in J} \chi_{ij} = 1, \quad \forall\, i \in I, \\ & \chi_{ij} \in \{0,1\}, \quad \forall\, i \in I,\ j \in J. \end{aligned} \tag{M1.1}$$

This is the most basic instance of matching, which can be easily solved by the Hungarian method (Kuhn, 1955) in polynomial time. Nevertheless, the complexity of the problem increases significantly by addition of more *sets*, *conditions*, and/or *criteria*, which in turn causes considerable modeling and computational challenges. As will be discussed in the next chapters, although some of these extensions have already been studied in literature, some important aspects/families of matching problems have not been properly addressed, due to lack of a systematic and holistic view of the problem. Understanding, synthesizing, and formalizing the existing and additional aspects of the problem as well as solving the problems and challenges identified along the way are the main motivations of this study, and the foundations for the research problem and questions as outlined next.

1.3 Research Problem

> *Mismatch* between individual entities of distributed systems is the challenging problem addressed in this research. Although a system with "mismatched individuals" may not necessarily collapse, it is certainly outperformed by an equivalent system with "best-matched individuals", in terms of a variety of critical criteria such as time, cost, quality, flexibility, and stability. The key challenge, therefore, is to identify those matching processes in a given distributed system, specify, structure, and formulate them in a systematic manner, and develop, test, and validate algorithms and protocols for solving them in an efficient manner.

There is a need for hierarchical representation of best matching processes in various domains by defining and quantifying various dimensions and their sub-categories, standardizing best matching processes, and enabling problem-solving capabilities as well as identification of new matching problems. Such systematic specification and taxonomy of matching processes enable identification of new problems that have been "off-the-radar" due to the lack of a holistic view. Systematic definition of matching problems with respect to standardized dimensions can enable powerful tools for scrutinizing diverse processes and systems in different manufacturing and service domains, and characterizing them as new instances in the family of best matching problems. The systematic framework for representation and identification of various and independent matching problems can trigger analogical reasoning by enabling systematic comparisons between diverse matching processes, and triggers ideas for development and validation of new algorithms and protocols based on the existing solution methodologies.

1.4 Research Questions

Structured representation, synthesis, and taxonomy of matching problems and processes enable analogical comparisons between different but analogous instances, and synergetic mechanisms for identification of new problem instances, formulations, and solution procedures, algorithms, and protocols. Hence, the first research question is outlined as follows.

RQ1. What is a good taxonomic framework for systematic syntheses, identification, and specification of matching problems in different areas? What are the most important characteristics of such framework?

Matching problems have been extensively studied in literature; however, the extensions are somewhat restricted to some certain aspects of the problem. Several instances of matching problems have not yet been addressed in literature and are not acknowledged as members of the same family of problems. This is an inevitable phenomenon due to the lack of a clear "big picture" of matching problems. The generic framework addressed by RQ1 can therefore be utilized for identification and formulation of new matching problems. Some instances of such problems have been identified and solved in this research as case studies, and some have been outlined as future research directions. Several algorithms already exist for solving relatively standard matching problems. More advanced and unstructured instances, however, require exclusive extensions of the existing algorithms or even development of new ones. With this motivation, the second research question is outlined as follows.

RQ2. What are the best approaches for structuring and formulating matching problems and processes? What algorithms and protocols can be developed to efficiently solve those best matching problems?

The developed methodologies, including technical definitions, mathematical formulations, optimization algorithms, and control protocols, must be validated to ensure their quality and impact. This must be done by performing extensive numerical experiments on various test-beds and case studies, and statistical analysis and comparison between the developed and the existing methodologies. This issue motivates the third research questions, which is outlined as follows.

RQ3. How can the developed best matching algorithms and protocols be validated? What case studies, experiments, scenarios, and statistical analysis methods must be deployed to test and highlight the relative impact of those methodologies?

1.5 Dissertation Structure

The remainder of this dissertation is organized as follows. CHAPTER 2 reviews the background and previous work on problem structure and modeling, solution methodologies, and applications of matching in different domains. CHAPTER 3 presents the PRISM taxonomy of best matching, a systematic framework for identification and formalism of various problems and processes associated with matching. CHAPTER 4, CHAPTER 5, CHAPTER 6, and CHAPTER 7 present four case studies related to different instances of matching in production and supply, manufacturing and assembly, clustering and team formation, and service enterprises. Each case study includes definitions and background, mathematical models, solution methodologies, and numerical experiments and analyses. CHAPTER 8 summarizes the dissertation, and outlines recommendations for future research.

CHAPTER 2. BACKGROUND—MATCHING CHALLENGES AND SOLUTIONS

Matching is a problem that has been widely studied and addressed in various disciplines with different, and not necessarily consistent, terminologies, assumptions, characteristics, constraints, and objectives; each with certain features and applications. Examples include, but are not limited to,

- *Manufacturing*: Production planning; Scheduling; Assembly line balancing; Group technology; Shop floor control;
- *Supply and logistics*: Supplier selection; Facility location; Warehousing; Remanufacturing; Reverse logistics;
- *Communications and networking*: Telecommunication; Power systems; Grid computing; Sensor clustering and networking; Swarm robotics;
- *Transportation and routing*: Air traffic control; Train coupling and sharing; Vehicle routing; Online travel agencies (bidding; customer sharing); Precision farming (mobile robot routing for farming and sensing);
- *Service*: Market design (students-colleges; interns-hospitals; body organs-patients); Healthcare (resident matching; doctor sharing); Project management; Social networks (teaming; partnership).

This chapter reviews different extensions of matching problems over the last 60 years, from the original assignment problem to more advanced instances with various formulations, computational complexities, and applications. Several practical applications along with solution approaches are addressed. The purpose of this review is to highlight the significance of the proposed research concerning different dimensions of best matching problems, and development of efficient algorithms and protocols for solving them[1].

2.1 Matching Problem Structures and Characteristics

The original matching problem appeared in an article by D. F. Votaw and A. Orden in 1952 on providing exact solution procedures for classification and assignment of personnel to a set of jobs. It was later formalized and presented as the assignment problem in an article by Harold W. Kuhn in 1955 on the application of the Hungarian Method for solving the assignment problem. In this dissertation, without loss of generality and consistency with literature, I refer to the assignment problem as a special class of matching problems. The term “assignment”, originating from the Medieval Latin “assignamentum”, means “the allocation of someone/something to someone/something else”, or “the attribution of someone or something as belonging” (Oxford English Dictionaries). This terminology is not all-inclusive, and restricts the problem to *unidirectional* two-sided matching, while many instances of matching problems involve bidirectional relations between the matching sets.

The original assignment problem involves one-to-one matching between two sets of individuals, where the sizes of the sets are equal, and the objective is to minimize the total cost of assignment (see Model (M1.1) in CHAPTER 1). Since the development of the

[1] Additional reviews are reported in the following chapters related specifically to the case studies.

first problem, several variations of it have been proposed and investigated in literature (Pentico, 2007). The most popular extensions are summarized below.

1. *Bottleneck matching*. This extension deals with the objective of the classic problem, minimizing the maximum cost of matching (or, maximizing the minimum satisfaction degree). Some application examples of bottleneck matching (Ravindran and Ramaswami, 1977) are (1) how to match printing jobs and press machines in order to minimize the makespan (scheduling), (2) how to transport perishable goods from warehouses to markets without spoilage, or military supplies from warehouses to command posts in case of emergency. For a maximization problem (*e.g.*, Model (M1.1) in CHAPTER 1), bottleneck matching maximizes the minimum satisfaction degree of individuals, *i.e.*,

$$\max \min_{i \in I,\ j \in J} \left\{ P_{ij} \chi_{ij} \right\}. \tag{2.1}$$

2. Σ_K *matching*. The focus of the Σ_K matching is to find a set of matches for which the sum of the K most costly matches is minimized (Grygiel, 1981).

3. *Balanced matching*. Similar to the bottleneck and Σ_K matching, this extension minimizes the *gap* between the maximum and minimum satisfaction degrees (Martello, 1984). Balanced matching has many practical applications (Duin and Volgenant, 1991) such as (1) cooperation between competitors on the construction of a communication network with the objective of minimizing the gap between the maximum and minimum construction and future maintenance cost, and (2) matching patients with different degrees of condition severity to different test groups such that the gap between the maximum and minimum severity conditions

is minimized. In a maximization problem, balanced matching minimizes the gap between the maximum and the minimum satisfaction degrees as follows:

$$\min \left\{ \max_{i \in I,\, j \in J} \left\{ P_{ij} \chi_{ij} \right\} - \min_{i \in I,\, j \in J} \left\{ P_{ij} \chi_{ij} \right\} \right\}. \tag{2.2}$$

4. *Minimum deviation matching*. Similar to the balanced matching but with a slight difference, the minimum deviation matching problem attempts to minimize the gap between the average and minimum satisfaction degrees (Gupta and Punnen, 1988). An application example is matching tasks and machines in a project with multiple independent phases, where the busy machines cannot be matched to the tasks of other phases until the current process is finished, and the objective is to minimize the machines idle times (Duin and Volgenant, 1991). In a maximization problem, minimum deviation matching minimizes the gap between the average and minimum satisfaction degrees as follows:

$$\min \left\{ \sum_{i \in I} \sum_{j \in J} P_{ij} \chi_{ij} - \min \left\{ P_{ij} \chi_{ij} \right\} \times \sum_{i \in I} \sum_{j \in J} \chi_{ij} \right\}. \tag{2.3}$$

5. *K-cardinality matching*. This problem instance involves two-sided matching between the sets of tasks (I) and agents (J), where (1) $|I| \neq |J|$, and (2) only K pairs of tasks and agents ($K < |I|, |J|$) are to be matched (Dell'Amico and Martello, 1997). Suggested applications by the authors include matching workers and machines, where only a subset of the workers and machines need to be matched, and assigning time slots on a communications satellite being used to transmit information from $|I|$ earth stations to $|J|$ different earth stations.
6. *Agent qualification*. This is a particular case of the problem with side constraints (*i.e.*, resource-constrained matching) where not every agent is qualified to be

matched to every task (Caron *et al*., 1999), and the objective is to maximize the overall satisfaction degree of all elements. The agent classification assumption, however, can be incorporated in the original problem by assigning *zero* preferences to the unqualified agents. Another alternative approach for modeling this problem instance is fractional programming (*a.k.a.*, fractional matching problem), where the conditions for agent qualification are incorporated in the objective function through a fractional term (Shigeno *et al*., 1995)

7. *Lexicographic bottleneck matching*. This problem instance is based on the bottleneck matching problem, but focuses on the costs of all matches other than the costliest match, *i.e.*, minimization of the second costliest, third costliest, *etc*. match, in addition to focusing on the costliest match (Burkard and Rendl, 1991; Sokkalingam and Aneja, 1998).
8. *Semi-matching*. A basic assumption of the original matching problem is that all tasks and agents are unique. Semi-matching problem deals with instances where some elements of one set (either tasks or agents) are identical while the elements of the other set are unique (Kennington and Wang, 1992; Volgenant, 1995). Examples include manpower planning, scheduling, capital budgeting and planning, and project planning (Kennington and Wang, 1992). This extension, however, can be modeled by the original problem, where the preference values of identical elements of one set over the elements of the other set are equal. The only purpose of this extension is to reduce the computational complexity of the problem through restructuring the problem and the solution procedure regarding identical elements (Volgenant, 1995).

9. *Categorized matching*. This refers to instances where the elements of one set (*e.g.*, tasks) are categorized in different groups and can be assigned to the elements of the other set (*e.g.*, agents) based on a set of inter-group and/or intra-group sequences (Punnen and Aneja, 1993). This problem is indeed analogous to task assignment with precedence relations (*e.g.*, assembly line balancing). The problem objective can take any of the aforementioned (or similar) instances. This class of problems will be further elaborated in the case study on tool sharing in collaborative assembly.
10. *Multicriteria matching*. In many problem instances, there are multiple criteria that must be considered in finding the optimal solution. Multicriteria matching problems are typically classified into the following two approaches:
    - *Combining criteria into one*, where all criteria are to be considered simultaneously (Yuan *et al.*, 1992; Geetha and Nair, 1993; Scarelli and Narula, 2002). An example of this case is the National Resident Matching Program, where the graduates of medical schools must be matched to hospitals for internships, based on several evaluation criteria (Yuan *et al.*, 1992). Parametric methods such weighted functions of all criteria are typically used in this case for defining the preferences (Scarelli and Narula, 2002). However, the decisions based on parametric approaches are not robust and depend highly on the expertise of the decision-maker.
    - *Considering criteria sequentially*, which implies instances where based on their importance, criteria are considered in sequences. For example, Lee and Schniederjans (1983) considered the problem of re-matching remedial education teachers from the schools at which they taught in the morning to the

schools at which they were to teach during the afternoon. The matching criteria were the costs of travel between the schools, the mutual preferences of teachers and schools, and the recommendations of the teachers' supervisors.

11. *Capacitated matching*. This problem deals with instances where the elements of one set (*e.g.*, agents) have limited resources, and the elements of the other set (*e.g.*, tasks) take specific amount of resources if matched to each agent of the first set. Examples include production/service capacity, budgetary limitations, degree of technical training of personnel, and time restrictions.
12. *Quadratic assignment*. This problem is formally defined as follows. There are two sets of facilities and locations, with the same size. A *distance* for each pair of locations and a *flow* for each pair of facilities are specified. The problem is how to match facilities and locations such that the sum of the distances multiplied by the flows is minimized. Similar to the majority of the extensions discussed above, this problem is also involved with the objective of assignment. An example is the assignment of doors on the opposite sides of a dock facility to the incoming and outgoing trucks, and the items in the incoming trucks are to be transported directly to the outgoing trucks, where the objective is to minimize the total travel distance for the forklifts performing the transportation (Tsui and Chang, 1992).
13. *Robust matching*. This problem instance deals with the matching decisions under uncertainty, where the robust approach aims to provide solutions that are close to optimal given any input scenario (Kouvelis and Yu, 1997). The idea is to identify appropriate *robustness scenario* out of the following options:

- *Absolute robust*, which maximizes the minimum overall satisfaction degree, over all possible scenarios.
- *Robust deviation*, which finds the best worst-case deviation from optimality considering all possible parameter value scenarios.
- *Relative robust*, which finds a solution that gives the best worst-case percentage deviation from optimality given all possible parameter value scenarios.

14. *Generalized matching*. This extension involves problems in which the elements of one set (*e.g.*, agents) can be matched to more than one element from the other set (*e.g.*, tasks), *i.e.*, one-to-many matching. Applications of generalized matching problem include (Cattrysse and Van Wassenhove, 1992) vehicle routing, fixed-charge location problems, grouping and loading in flexible manufacturing systems, scheduling projects, allocating storage, designing communication network, assigning jobs to computers, scheduling variable length TV commercials, and assigning ships to overhaul facilities. The extensions of this problem include multiple resources for each agent (Campbell and Langevin, 1995; Lee and Kim, 1998; Nowakovski *et al.*, 1999) along with bottleneck generalized matching problem (Martello and Toth, 1995; Chang and Ho, 1998), and quadratic generalized matching problem (Bokhari, 1987), both as extensions of the equivalent original problem.
15. *Multi-dimensional matching*. This extension involves matching the elements of three or more sets such as jobs-workers-machines or students-teachers-classes (*i.e.*, timetabling problem). Some of the aforementioned extensions of the classic 2D problem have also been generalized to multi-dimensional problems such as

bottleneck multi-dimensional matching (Malhotra *et al.*, 1985; Vartak and Geetha, 1990; Geetha and Vartak, 1995). The most common instance of this problem is 3D, and an interesting version is to add time, as the third factor to the classic 2D problem, *i.e.*, assignment of agents to changing tasks over time. Examples are assignment of bus drivers to routes (Carraresi and Gallo, 1984) or medical residents to rotations (Franz and Miller, 1993) over time. A limitation and open question, however, is that *integer* timeslots must be considered, which may influence the optimality of results.

Table 2.1. Matching extensions and their classification with respect to the three dimensions of matching.

Extension	Dimension		
	Sets	Conditions	Criteria
Bottleneck matching			●
Σ_K matching			●
Balanced matching			●
Minimum deviation matching			●
K-cardinality matching	●		
Agent qualification			●
Lexicographic bottleneck matching			●
Semi-matching	●		
Categorized matching		●	
Multicriteria matching			●
Resource-constrained matching		●	
Quadratic assignment			●
Robust matching			●
Generalized matching	●		
Multi-dimensional matching	●		

Table 2.1 summarizes the aforementioned extensions of the matching problem and their classification with respect to three main dimensions of matching problems; *sets*, *conditions*, and *criteria*. As observed, the majority of extensions are concerned with the matching

criteria and sets characteristics. The extensions on matching conditions are mostly involved with resource constraints and precedence relations (*i.e.*, categorized matching). These findings are important in characterizing the generic taxonomic framework for matching, as will be discussed in the next chapter.

2.2 Methodologies

The generalized matching problems are NP-hard[1]. Thus, development of computationally-efficient algorithms is critical, especially in applications with highly dynamic and unpredictable domains (where the optimal solution must be calculated quickly and frequently). Hundreds of algorithms have been developed and examined for decades for solving different instances of the matching problem. Matching algorithms can be categorized into four major classes of exact, heuristic, approximation, and relaxation. Some of the most popular algorithms are briefly described below.

1. *Hungarian method.* The Hungarian method is a combinatorial optimization algorithm that solves the original two-sided one-to-one matching problem in polynomial time. The algorithm is based on the *duality theorem* of linear programming as well as combinatorial tools in graph theory. Kuhn named the algorithm in honor of two Hungarian mathematicians, Dénes König and Jenö Egerváry, whose earlier works provided the basis for the Hungarian method (Kuhn, 1955). The Hungarian method yields the optimal one-to-one matching between the

[1] Sahni and Gonzalez (1976) proved that the assignment of tasks to agents on a one-to-many basis is an NP-hard problem. Accordingly, every matching problem that can be *reduced* to this problem is NP-hard as well. This spans all the matching problems discussed in this dissertation, except for the original two-sided one-to-one matching problem.

elements of sets I and J, based on their respective preference matrix P, where P_{ij} denotes the mutual preference of $i \in I$ and $j \in J$. The method is based on the following theorem: If a number is added to or subtracted from all of the entries of any row or column of a cost matrix, an optimal match for the resulting cost matrix is also an optimal match for the original cost matrix (Kuhn, 1955). Since the objective of the original algorithm is to minimize the cost of matching I and J, we also assume that the entries of the preference matrix are undesirable (*e.g.*, cost) and must be minimized.

2. *Deferred acceptance algorithm.* In their seminal work "College Admissions and the Stability of Marriage", Gale and Shapley (1962) developed an algorithm that does not necessarily yield the optimal matching with respect to the mutual preferences of individuals, but guarantees *stable matching*. According to the Gale-Shapley definition of stability, *a matching between sets I and J is stable if (1) no individual from either set is paired with an unacceptable match from the other set, and (2) there is no unmatched pair of individuals who both prefer each other to their current matches*. The original work of Gale and Shapley was continued by several researchers for centralized market design, with applications such as matching interns to hospitals, students to colleges, and human organs to recipients (Roth, 2008). The deferred acceptance algorithm is suitable for finding the stable one-to-one or one-to-many match between the elements of sets I and J in polynomial time. The algorithm is based on an iterative centralized negotiation mechanism, where the elements of one set offer matching proposals in each

iteration, and the elements of the other set respond by either accepting, holding, or rejecting those proposals. The proposal offering and evaluation mechanisms are based on the preferences of the individuals for each other. Hence, the algorithm can be executed in two different manners considering who is proposing (I or J), which may result in completely different matching results (Roth, 2008).

3. *Lagrangian relaxation.* A remarkable number of computationally expensive optimization problems can be viewed as “easy” problems complicated by a small number of constraints (Fisher, 1981). The Lagrangian relaxation method, due to Held and Karp (1970; 1971), is a smart use of this property for solving “hard” constrained optimization problems through relaxation of certain sets of constraints—those that cause the computational complexity. The idea is to provide useful information through approximation by replacing strict inequalities with penalty costs associated with the violation of those inequalities using Lagrange multipliers. The Lagrangian relaxation method is widely studied for solving the generalized matching problem with resource constraints (Öncan, 2007). The solutions to the relaxed problem provide suitable (upper/lower) bounds to the original problem, which can then be used for solving the original problem using iterative Lagrangian relaxation method or other methods such as branch-and-bound.
4. *Branch-and-bound method.* Branch-and-bound is a powerful discrete optimization method developed by A. H. Land and A. G. Doig in 1960. The idea is based on systematic enumeration of candidate solutions through state-space search enabled by different *branching* and *bounding* strategies. A branch produces two or more

candidate solutions with minor but known differences from the current solution. A bound, on the other hand, calculates a lower or upper bound for the objective value that is used for fathoming the candidate solutions. The logical structure of the branching and bounding procedures resembles that of a tree (Dakin, 1965). The branch-and-bound method provides an efficient (but computationally exhaustive) mechanism for solving matching problems (despite having exponential worst-case performance). A basic version of this method for solving generalized matching problems with resource constraints is discussed by Ross and Soland (1975).

5. *Genetic algorithms*. This is a leading metaheuristic introduced by John H. Holland in the 1970's. The algorithm leads to the survival of the *fittest* members of a population known as "chromosomes", considering their unique properties controlled by their basic units, their "genes". This evolutionary process imitates the process of *natural selection* supported by *crossover* and *mutation* operators, as bio-inspired mechanisms for reproduction and generation of fitter "offsprings". Genetic algorithms can be effectively designed for solving almost all variants of best matching problems, including complicated and advanced instances that are not solvable by the exact and heuristic algorithms discussed thus far. The challenge, however, is to find the best way to (1) encode a solution set into a chromosome that represents all necessary properties of the problem, and (2) reproduce new populations of chromosomes that lead to the optimal solution in an efficient manner, in terms of both computational time and solution quality. This method will be discussed in detail through the case studies.

6. *Greedy Randomized Adaptive Search Procedure* (*GRASP*). GRASP is a constructive metaheuristic due to Thomas A. Feo and Mauricio G. C. Resende (1989), which progressively (1) constructs *greedy randomized* solutions, and (2) improves them through neighborhood search. The first phase is accomplished by storing elements in a restricted candidate list (RCL) using a greedy function, and adding them to the solution, one at a time, according to their respective ranks in the RCL. In the second phase, the greedy randomized solution is improved via certain neighborhood search procedures. This procedure is repeated until the algorithm converges. GRASP is an efficient metaheuristic for best matching, especially for solving complicated instances such as biquadratic (Mavridou *et al.*, 1998) and three-dimensional (Aiex *et al.*, 2005) matching problems.
7. *Ant colony optimization.* This is a constructive metaheuristic inspired by the *foraging behavior* of ants in their search for food and the shortest path back to their nest. The original algorithm was developed by Marco Dorigo and his colleagues in the 1990's, which soon after was widely applied for solving various optimization and control problems. ACO is based on primitive behaviors of individual ants, which lead to highly intelligent behavior at the scale of colony or swarm, through efficient *interaction.* The pheromone functions as an indirect interaction mechanism between the ants, enabling a *sign-based stigmergy* that signals both the food source and a suggested path. Since pheromone is accumulated faster on shorter paths, however, this behavior of ants enables a *reinforcement learning mechanism*

that eventually leads to the detection of the shortest path. Ant colony optimization is an efficient metaheuristic for solving various hard best matching problems.

8. *Tabu search*. This is a *memory-based* local search metaheuristic invented by Fred W. Glover in 1986. The neighborhood search follows certain prohibition strategies that mark previously visited solutions (either temporarily or permanently) as forbidden or *tabu*, in order to prevent cycling and improve the efficiency of search mechanism (Glover, 1986). The algorithm starts with an initial solution and continually *explores* through its neighborhood while *exploiting* the knowledge of tabu points, until a set of convergence criteria is satisfied. Similar to the other metaheuristics discussed so far, tabu search can be used for solving an extensive range of best matching problems. A detailed application of tabu search for solving three-dimensional matching problems will be discussed in the subsequent chapters on case studies.

The literature of matching problems offers several algorithms and methodologies for solving different instances of matching (Table 2.2); however, most of those standard approaches (*e.g.*, exact algorithms; relaxation methods) are limited to a limited range of problems and are very difficult to generalize for solving more advanced and complicated instances. Hence, our approaches in this research for solving new (and indeed more complicated) instances of matching are centered around *heuristic* and *metaheuristic* methods, as will be discussed in the next chapters. The reason is that these methods are proven effective for finding approximate solutions to “hard” optimization problems by trading optimality, completeness, and accuracy for computational efficiency. Heuristics are usually *ad hoc*, suitable for particular problem instances. Higher involvement of intuition and empirical “rules-of-thumb” provides heuristic developers with more flexibility to “think out of the box” and create solutions that may be very difficult to prove in theory, but are extremely efficient in practice.

Table 2.2. Solution approaches for best matching.

Context	Method	Reference
Approximation methods	Approximation scheme	Cohen *et al*. (2006)
	Simple heuristics	Wilson (1997)
	Set partitioning heuristic	Cattysse *et al*. (1994)
	Lagrangian relaxation	Jeet Kutanoglu (2007)
Heuristics/ Metaheuristics	LP relaxation based heuristic	Trick (1992)
	Tabu search	Yagiura *et al*. (2004)
	Simulated annealing	Osman (1995)
	Genetic algorithms	Lorena *et al*. (1999)
	Neural networks	Monfared and Etemadi (2004)
	Ant colony optimization	Lourenc and Serra (2002)
	GRASP	Lourenc and Serra (2002)
Relaxation methods	Linear programming relaxation	Bender and van Nunnen (1983)
	Lagrangian relaxation	Lorena and Narciso (1996)
	Lagrangian decomposition	Yagiura *et al*. (1999)
	Variable depth search heuristics	De Farias *et al*. (2000)
Exact methods	Polyhedral analysis	De Farias *et al*. (2000)
	Branch and bound algorithms	Haddadi and Ouzia (2004)
	Branch and price algorithms	Savelsbergh (1997)
	Branch and cut and price algorithm	Pigatti and Aragoa (2004)

2.3 Applications

As discussed in the beginning of this chapter, applications of best matching are diverse, covering various production, manufacturing, and service industries. Scheduling (*e.g.*, machine; computational grids; workforce planning; batching; load balancing), transportation and routing (*e.g.*, vehicle routing), telecommunication, production planning (*e.g.*, batch loading, group technology, order selection, lot scheduling), facilities layout, and supply network design and logistics (*e.g.*, demand partitioning, sourcing, market clearing) processes can be regarded as instances of best matching in practice. Some examples of two-sided processes are (Cattrysse and Van Wassenhove, 1992) fixed-charge location problems, grouping and loading in flexible manufacturing systems, scheduling projects, storage allocation, designing communication network, assigning jobs to computers, scheduling variable length TV commercials, and assigning ships to overhaul facilities. Classic examples of problems with more sets involved include time-based allocation of bus drivers to routes (Carraresi and Gallo, 1984) or medical residents to rotations (Franz and Miller, 1993), followed by more advanced examples such as design of cyber-physical distributed systems or large-scale networks of multi-national corporations. Table 2.3 summarizes some practical applications of two-sided matching in various domains.

Table 2.3. Some practical applications of matching* (Öncan, 2007).

Problem	Matching entities	Reference
Project management	Labors, Jobs	Drexl (1991)
Load balancing	Machines, Jobs	Harvey *et al.* (2006)
Aeromedical routing	Flights, Patients	Ruland (1999)
Vehicle routing	Vehicles, Cities	Baker and Sheasby (1999)
Single egress selection	Edge links, Prefixes	Bressoud *et al.* (2003)
Wireless networks	Base stations, Terminals	Barbas and Marin (2004)
Batch loading	Batches, Jobs	Dobson and Nambimadom (2001)
Capacity planning	Periods, Batches	Mazzola *et al.* (1989)
Machine assignment	Cells, Machines	Cheng *et al.* (1996)
Group formation	Machines, Parts	Shtub (1989)
Storage assignment	Locations, Items	Lee (1992)
Dynamic ordering	Periods, Orders	Lee and Kim (1998)
Database partition	Processors, Partitions	Boffey (1989)
Location-allocation	Suppliers, Customers	Ross and Solland (1977)
Land use allocation	Activities, Land parcels	Cromley and Hanink (1999)
Worker allocation	Departments, Workers	Campbell and Diaby (2002)
Power management	Voltage levels, Tasks	Yu and Prasanna (2003)
Stock management	Demands, Stocks	Privault and Herault (1998)
Telescope scheduling	Intervals, Activities	Nowakovski *et al.* (1999)
Resource scheduling	Institutions, Activities	Zimokha and Rubinstein (1988)
Cane supply decisions	Intervals, Paddocks	Higgins (1999)
Demand partitioning	Facilities, Products	Benjafaar *et al.* (2004)
Snow disposal	Disposal/removal sites	Campbell and Langevin (1995)
Single sourcing	Warehouses, Customers	Freling *et al.* (2003)
Third party routing	Depots, Customers	Jalisi and Cheddad (2000)
Market clearing	Asks, Bids	Kalagnanam (2001)
Maximal covering	Sites, Customers	Klastorin (1979)
Labor force scheduling	Manpower, Sections	Littschwager and Tcheng (1967)
Dairy farm allocation	Factories, Suppliers	Foulds and Wilson (1997)
Capacitated clustering	Seeds, Customers	Shieh and May (2001)
Production planning	Agents, Tasks	LeBlanc *et al.* (1999)

* Öncan surveyed assignment problems that are presented in this research as matching problems, as described earlier in this chapter.

2.4 Concluding Remarks

The conclusions drawn from what we learned from our literature review are threefold. First, best matching—as mentioned in the beginning of the discussion—is a broad problem, spanning several disciplines including engineering, mathematics, economics, and management. This further highlights the significance of the problem and the impact of our contribution on these various disciplines. It is indeed a great opportunity for developing new ideas, problems, and methodologies through systematic taxonomy and analogical comparisons between different problems that may be completely different in nature—as outlined in RQ1. In addition, there is already an extensive pool of algorithms and solution approaches, which can inspire us for developing efficient algorithms and solution procedures for solving new problem instances that we identify and formulate (see RQ2 and RQ3). The last but rather the most important remark is that—as shown in Table 2.1—the research on matching problems and processes is still in progress. Its shortcoming becomes even more clear after defining and structuring the taxonomic matching framework (CHAPTER 3), as the "big picture" for comparative analysis and understanding of different instances of matching. This further highlights the lack of sufficient attention to some process characteristics that may significantly influence the outcomes of matching in practice (see RQ2). Motivated by this review, a novel taxonomic framework, the PRISM Taxonomy of Best Matching, will be presented in the next chapter as a theoretical foundation for addressing emerging matching problems in various manufacturing and service domains.

CHAPTER 3. METHODOLOGY—BEST MATCHING THEORY AND MODELS

This chapter presents a taxonomic framework for all the aforementioned (and similar) problems in manufacturing and service domains that can be recast as best matching problems. The framework, called "The PRISM Taxonomy of Best Matching", developed at the PRISM (Production, Robotics, and Integration Software for Manufacturing and Management) Center of Purdue University, formalizes best matching problems with respect to 3+1 dimensions: D1, sets; D2, conditions; D3, criteria; D+, time, progression (Figure 3.1). The PRISM taxonomy provides a systematic framework for synthesis of matching processes in distributed and collaborative/competitive systems. The framework addresses collaboration as one of the major conditions of matching processes, and models and formulates two principles of CCT (Collaborative Control Theory), *association-dissociation* and *dynamic lines of collaboration*, by incorporating the additional dimension, D+, in the decisions. The remainder of this chapter elaborates on the 3+1 dimensions of the PRISM taxonomy of best matching along with several examples and illustrations.

3.1 The PRISM Taxonomy of Best Matching

A taxonomic framework is developed to characterize and formalize matching processes with respect to 3+1 dimensions (Figure 3.1), and provide a holistic view of matching problems with respect to the research questions outlined in CHAPTER 1.

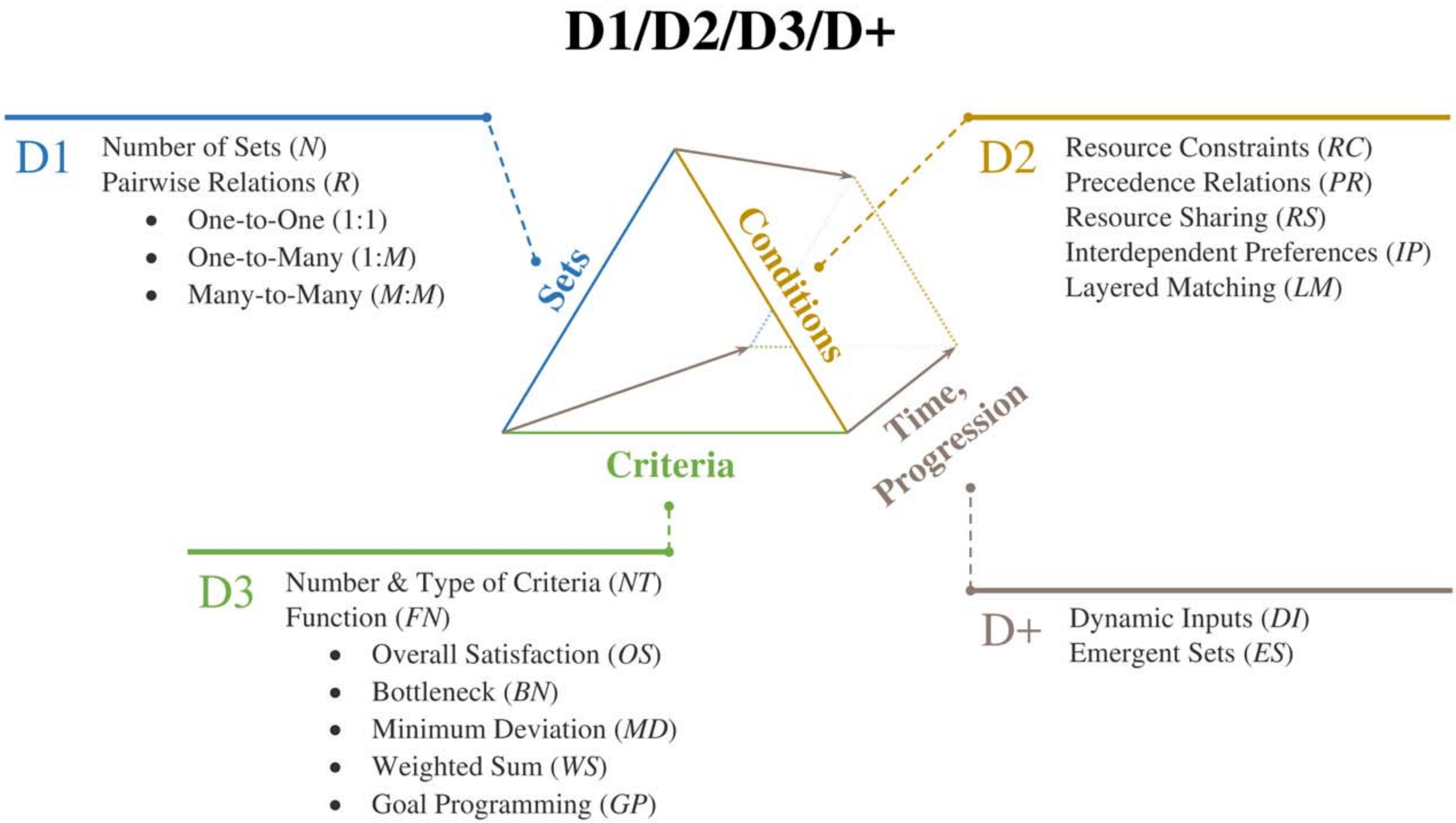


Figure 3.1. PRISM taxonomy of best matching (after Moghaddam and Nof, 2015[c]).

3.1.1 D1: Sets

This dimension formalizes the individuals to be matched and their pairwise relations, and classifies them into two or more sets. More specifically, D1 defines:

- *The number of the sets* (*N*). Individuals may be classified into two (*e.g.*, tasks-processors; students-schools) or more (*e.g.*, jobs-machines-operators) sets.
- *Pairwise relations* (*R*). The number of individuals in each set may be equal (*e.g.*, bolt-nut) or different (*e.g.*, interns-hospitals). Each individual may be matched to one or more individuals from the other set(s). Specifically, the pairwise relations between two sets of individuals may be one-to-one (*e.g.*, organs-patients), many-to-one (*e.g.*, tasks-computing resources), or many-to-many (*e.g.*, suppliers-customers).

This is one of the aspects of matching problems that have been extensively studied. The simplest case is when $N = 2$ (*i.e.*, two sets) and $R := 1:1$ (*i.e.*, one-to-one relation and thus equal size). This is indeed the original assignment problem introduced by Votaw and Orden and solved by Kuhn via the Hungarian method. Other permutations of N and R lead to other (more complicated) instances including multi-dimensional and generalized matching (see CHAPTER 2). Nevertheless, the diversity of matching problems is not limited to these limited instances, and is indeed as broad as the variety of several other *conditions* that some of which are described next.

3.1.2 D2: Conditions

Depending on the context and nature of the problem, best matching processes are sometimes *conditioned* by certain characteristics, requirements, and/or constraints. These conditions, if disregarded, may lead to misleading, inappropriate, or even infeasible matching results. In spite of their significant impact, however, best matching conditions (*i.e.*, D2) have not been properly addressed in literature compared to the other two main dimensions of best matching (*i.e.*, D1: sets; D3: criteria). The second dimension of the PRISM taxonomy of best matching therefore formalizes various conditions that may be involved in best matching processes. Accordingly, the D2 corresponding to a certain best matching process may involve one, some, or all of the conditions below (or other conditions not mentioned here).

3.1.2.1 Resource Constraints

Resource-constrained matching refers to instances where the elements of set *J*, for example, have limited resources, and each element of set *I* takes specific amount of resources if matched to an element of set *J* (*e.g.*, jobs assigned to machines). This is a classic problem with many practical applications (*e.g.*, scheduling, batching, supplier selection) and has been addressed in assignment problems through side constraints (Mazzola and Neebe, 1986). In this class of matching, there may be no limit on the number of matches for the elements of set *J*, but the number of matches will indeed depend on the availability of their resources. Let b_j denote the level of available resources of $j \in J$, and r_{ij} be the amount of resources demanded by $i \in I$ if matched to $j \in J$. This condition is incorporated in the matching by process by adding a resource constraint to the model. For example, the many-to-one matching between sets I and J with resource constrains and the objective of maximizing the overall satisfaction (OS) can be formulated as follows.

$$M{:}1 \,/\, RC \,/\, +, OS$$

$$\begin{aligned} \max \quad & \sum_{i \in I} \sum_{j \in J} P_{ij} \chi_{ij}, \\ \text{s.t.} \quad & \sum_{i \in I} r_{ij} \chi_{ij} \le b_j, \quad \forall\, j \in J, \\ & \sum_{j \in J} \chi_{ij} = 1, \quad \forall\, i \in I, \\ & \chi_{ij} \in \{0,1\}, \quad \forall\, i \in I,\ j \in J. \end{aligned} \tag{M3.1}$$

Several instances of resource-constrained matching will be discussed through the case studies.

3.1.2.2 Precedence Relations

This condition occurs where there are precedence relations between the elements of set I regarding their matching to the elements of set J that are heterogeneous and can be matched to more than one element of set I. The precedence relation constraint between two elements $i, i' \in I$ regarding the elements of set J is formulated as follows (Note: It is assumed that J's are numbered from 1 to $|J|$):

$$\sum_{j \in J} j \chi_{i'j} \le \sum_{j \in J} j \chi_{ij}, \qquad \forall\, i \in I,\ i' \in PR_i, \tag{3.1}$$

where PR_i denotes the set of immediate predecessors of element $i \in I$. That is, if $i' \in I$ has been matched to an element $j \in J$, $i \in I$ can be matched to either the same element ($j \in J$) or the next elements ($j' \in J$, $j' > j$). Matching with precedence relations may reduce the options of elements with predecessors, if the number of matches for the elements of the target set is limited, or the resources of the target set are limited (resource-constrained matching). For example, a many-to-one resource-constrained matching between sets I and J, with precedence relations between I's, and overall satisfaction degree as the matching criterion is formulated as follows:

$$M{:}1 \,/\, RC, PR \,/\, +, OS$$

$$\begin{aligned}
\max \quad & \sum_{i \in I} \sum_{j \in J} P_{ij} \chi_{ij}, \\
\text{s.t.} \quad & \sum_{i \in I} r_{ij} \chi_{ij} \le b_j, \quad \forall\, j \in J, \\
& \sum_{j \in J} j \chi_{i'j} \le \sum_{j \in J} j \chi_{ij}, \quad \forall\, i \in I,\, i' \in PR_i, \\
& \sum_{j \in J} \chi_{ij} = 1, \quad \forall\, i \in I, \\
& \chi_{ij} \in \{0,1\}, \quad \forall\, i \in I,\ j \in J.
\end{aligned} \tag{M3.2}$$

Resource-constrained matching is a special case of this problem instance—matching with precedence relations can be reduced to resource-constrained matching by setting all $PR_i = \varnothing$ for all $i \in I$. This matching condition will be further investigated through the second case study.

3.1.2.3 Resource Sharing

This condition addresses an extension of resource-constrained matching where the elements of the set with limited resources are allowed to laterally share resources with each other in case some of them have extra resources while the others have shortage of resources. The resource constraints in Models (M3.1) and (M3.2) limit the number of I's that can be matched to each element of set J. Hence, some elements of set I may remain unmatched due to resource shortage. On the other hand, some elements of set J are likely to end up with extra resources in case their demand is less than their capacity. Let I_1 and I_2 denote the sets of matched and unmatched elements of set I, respectively. Then, the amount of extra resources of $j \in J$ can be calculated as follows:

$$l_j = b_j - \sum_{i \in I_1} r_{ij} \chi_{ij}, \quad \forall j \in J, \tag{3.2}$$

where

$$l_j < r_{ij}, \quad \forall i \in I_2,\ j \in J. \tag{3.3}$$

The level of idle resources is actually equal to the slack variables of the resource constraints. Resource constraints may lead to "unnecessary" underutilization of resources—the elements $i \in I_2$ are left unmatched due to local resource shortage, while some elements of $j \in J$ have extra resources l_j. This limitation can be resolved by enabling the elements of set J (the suppliers) to laterally share and integrate their resources in order to serve (be matched to) more elements of set I as a whole. It will be shown through the case studies that matching with resource-sharing leads to higher resource utilization and demand fulfillment rate by matching more elements from the resource demanding set (*i.e.*, I) to the elements of the resource-sharing set (*i.e.*, set J). In mathematical terms, resource-sharing is represented as follows:

$$\sum_{j' \in j \backslash \{j\}} \left(s_{jj'} - s_{j'j} \right) \le b_j - \sum_{i \in I_1} r_{ij} \chi_{ij}, \quad \forall j \in J, \tag{3.4}$$

where $s_{jj'}$ denotes the amount of resources j shares with j'. In the next chapters, it is proven mathematically that matching with resource-sharing leads to higher resource utilization, demand fulfillment, and stability, compared to matching without resource-sharing.

Matching with resource sharing will be investigated in detail in the next chapters (case studies). It will be shown that as the costs of resource-sharing may differ from one element of set J to another, it should be optimized in addition to the main objective of matching. Given these considerations, a resource-constrained matching between sets I and J with resource sharing and the objectives of maximizing the overall satisfaction (including the total cost of resource sharing) is formulated as follows:

$$M{:}1\ /\ RC, RS\ /\ +, OS$$

$$\begin{aligned}
\max \quad & \sum_{i\in I}\sum_{j\in J} P_{ij}\chi_{ij} - \sum_{j\in J}\sum_{j'\in J} C_{jj'}s_{jj'}, \\
\text{s.t.} \quad & \sum_{i\in I} r_{ij}\chi_{ij} + \sum_{j'\in j\backslash\{j\}} \left(s_{jj'} - s_{j'j}\right) \le b_j, \quad \forall\, j \in J, \\
& \sum_{j\in J} \chi_{ij} = 1, \quad \forall\, i \in I, \\
& s_{jj'} \ge 0,\ \chi_{ij} \in \{0,1\}, \quad \forall\, i \in I,\ j, j' \in J.
\end{aligned} \tag{M3.3}$$

Matching with resource-sharing is extensively investigated through the case studies in supply networks, assembly systems, and collaborative networked organizations. It is proven that best matching between suppliers and customers considerably improves the quality of resource-sharing decisions in collaborative networks of enterprises, in terms of total collaboration cost and service level. In other words, matching is applied for optimization of resource-sharing decisions and protocols. Moreover, it will be proven that the combination of best matching and resource-sharing improves the *stability* of supply networks. It is also shown that resource-sharing has substantial impact on assembly systems in terms of resource utilization and line *balanceability*.

3.1.2.4 Interdependent Preferences

The notion of interdependencies among preferences has been investigated in utility theory (*e.g.*, Cabrales and Calvó-Armengol, 2008), where the preferences of each entity depends on the *consumption* or *well-being* of the other entities in their neighborhood. In this setting, entities may be *altruistic* or *envious* by the utility of the other entities. This idea is the initiative for the development and analysis of interdependencies among preferences in matching. Our definition of *interdependent preferences* is somewhat different from what has been discussed in the utility theory. Best matching with interdependent preferences refers to instances where the preference of $i \in I$ for $j \in J$ is influenced by and represented as a function of matching of $i' \in I$, $i' \neq i$, to $j \in J$. For instance, in team formation, the preferences of an entity over different teams may be influenced by the members of different teams. Such influences (if any), may increase or decrease the preference of an element over another element. The interdependent preference of $i \in I$ and $j \in J$ can be formulated as follows:

$$\hat{P}_{ij} = P_{ij}\left(1 + \sum_{i' \in I \setminus \{i\}} \alpha_{i'i}\chi_{i'j}\right), \qquad \forall\, i \in I,\ j \in J. \tag{3.5}$$

where $\alpha_{i'i}$ denotes the one-sided relation between i and i'—it takes positive/ negative/zero value, if i is altruistic/envious/neutral about i'; *i.e.*, $i \in I$ is

- *Altruistic* about $i' \in I$, $i' \neq i$, if $\alpha_{i'i} > 0$;
- *Envious* about $i' \in I$, $i' \neq i$, if $\alpha_{i'i} < 0$;
- *Neutral* about $i' \in I$, $i' \neq i$, if $\alpha_{i'i} = 0$.

In this context, altruism and envy respectively represent situations where the preference of an element of set I for an element of set J is increased or decreased, if another element of set I is matched to that element of set J. The mathematical formulation of matching with interdependent preferences (*i.e.*, $\hat{P}_{ij}$) is the same as matching without any interdependencies between preferences (*i.e.*, P_{ij}), except that the preferences are not fixed but functions of the decision variables $\chi_{i'j}$. For example, many-to-one matching process with interdependent preferences is formulated as the following Quadratic Assignment Problem (QAP):

$$M{:}1\ /\ RC, IP\ /\ +, OS$$

$$\begin{aligned}
\max \quad & \sum_{i\in I}\sum_{j\in J} P_{ij}\chi_{ij} + \sum_{i\in I}\sum_{j\in J}\sum_{i'\in I} P_{ij}\alpha_{i'i}\chi_{i'j}\chi_{ij}, \\
\text{s.t.} \quad & \sum_{i\in I} r_{ij}\chi_{ij} \le b_j, \quad \forall\, j \in J, \\
& \sum_{j\in J} \chi_{ij} = 1, \quad \forall\, i \in I, \\
& \chi_{ij} \in \{0,1\}, \quad \forall\, i \in I,\ j \in J.
\end{aligned} \tag{M3.4}$$

Model (M3.4) is indeed a special QAP formulation. More elaborate definition, formulation, and analysis of this model along with its applications in clustering and team formation will be presented in one of the case studies presented in the next chapters. Interdependencies between preferences may take forms other than what were discussed in this section. The preference function may not necessarily be a linear function of mutual influences of individuals on each other's preferences. The common characteristic of all variants of matching with interdependent preferences, however, is in the nature of such interdependencies—that the mutual preference of two individuals may be changed, if another matching takes place. The layered matching introduced next is another example of interdependent matches, where the quality of match in the current layer is a function of how individuals were matched in the previous layer(s).

3.1.2.5 Other Extensions

The conditions and process characteristics of matching indeed require further investigation. In line with RQ1, our aim is to identify and formulate new instances of matching with unique process characteristics, requirements, and objectives. An example of such conditions is layered matching, where the elements of two (or more) sets resulting from parallel and independent matching at one level are two be matched at a subsequent level. A practical example of this matching instance could be series-parallel assembly of component parts, where the objective is to find the best *geometric* or *shape matching* such that the overall quality of final products is improved.

3.1.3 D3: Criteria

Various criteria and objectives have been used in different studies for identifying the best match and distinguishing it from other potential matches. In this section, the most common matching criteria are reviewed and formalized. The formulations are standardized through translation of matching criteria into preference values normalized between zero and one. Depending on the application, however, the preference values may be replaced with real parameters, *e.g.*, cost, time, distance. The "best match" is essentially distinguished from other potential matches based on a set of criteria (one or more). The goal is then to enhance the performance of the system with respect to the given criterion/criteria by optimally matching the distributed individuals. Best matching criteria are diverse and depend on the context and application domain—from traditional cost, time, efficiency, and productivity factors to emerging e-Criteria (Nof, 2007) such as:

1. *Integrability*. Ability to integrate data from a number of distributed entities and increase its usefulness.
2. *Connectivity*. Type, level, and quality of internet-supported connections between distributed operating systems, and application and network layers.
3. *Agility*. Ability of a system, at individuals or network level, to respond and adapt to changes in real-time.
4. *Scalability*. Ability of a process, system, or network to handle increasing amount of tasks and adapt to growth.
5. *Reachability*. Effectiveness of interconnections and interactions between individuals in a distributed network.
6. *Viability*. Ratio of the cost of operating/sustaining distributed individuals to the rewards gained from their service.
7. *Autonomy*. Level of delegation of authority, task assignments, and decentralization in distributed networks.
8. *Dependability*. Probability of a task to be successfully executed—system *availability*; *reliability*; *sustainability*; *integrity*; *maintainability*.
9. *Resilience*. Ability to survive the unforeseen circumstances, risks, disruptions, and high impact events (*a.k.a.*, *transformability* and *adjustability*).

Although the definitions of some criteria may be completely different, their implications are the same—*desirable* criteria (*e.g.*, agility) must be increased while *undesirable* criteria (*e.g.*, cost) must be decreased. Hence, without loss of generality, all matching criteria are formalized as *preferences* of individuals for each other. Specifically, the quality of matching individuals $i \in I$ and $j \in J$ is represented by their mutual preference as

$$P_{ij} = w_i \cdot p_{i \to j} + w_j \cdot p_{j \to i}, \qquad \forall i \in I,\ j \in J, \tag{3.6}$$

where $p_{i \to j}$ and $p_{j \to i}$ denote the *normalized* preferences of $i \in I$ for $j \in J$, and $j \in J$ for $i \in I$, respectively. Coefficients w_i and w_j, $w_i + w_j = 1$, denote the relative *weights* of individuals $i \in I$ for $j \in J$ in defining the mutual preferences. The third dimension of the PRISM taxonomy of best matching formalizes the matching criteria based on the mutual preference scores. Specifically, D3 provides information on the number and type (desirable: "+"; undesirable: "–") of criteria (*NT*) along with the formulation of the respective objective function (*FN*) (see CHAPTER 2 for some of the formulations):

- Single criterion
    1. *Overall Satisfaction* (*OS*). This classic function maximizes the overall satisfaction of preference scores, *e.g.*, minimizing the total cost of production ($\text{D3} := -, OS$); maximizing the overall service level ($\text{D3} := +, OS$); minimizing the total transportation time ($\text{D3} := -, OS$).

2. *Bottleneck* (*BN*). This function maximizes the minimum satisfaction degree of individuals. Some application examples of bottleneck assignment are (Ravindran and Ramaswami, 1977):
    a. Matching printing jobs and press machines in order to minimize the makespan ($\mathrm{D3} := -, BN$).
    b. Transportation of perishable goods from warehouses to markets with minimum spoilage ($\mathrm{D3} := -, BN$).
    c. Shipment of military supplies from warehouses to command posts in case of emergency ($\mathrm{D3} := -, BN$).
3. *Minimum Deviation* (*MD*). This function minimizes the *gap* between the maximum and minimum satisfaction degrees (Martello *et al*., 1984) or the average and minimum satisfaction degrees (Gupta and Punnen, 1988). Application examples include (Duin and Volgenant, 1991):
    a. Cooperation between competitors on the construction of a communication network with the objective of minimizing the gap between the maximum and minimum construction and future maintenance cost ($\mathrm{D3} := -, MD$).
    b. Allocation of patients with different degrees of condition severity to different test groups such that the gap between the maximum and minimum severity conditions is minimized ($\mathrm{D3} := -, MD$).
    c. Assignment of tasks to machines in a project with multiple independent phases, where busy machines cannot process the tasks of other phases until

the current process is finished, and the objective is to minimize the machines idle times ($\mathrm{D3} := -, MD$).

- Multiple criteria

4. *Weighted Sum* (*WS*). This function combines the normalized values of all criteria into one—with respect to their type (*i.e.*, desirable/undesirable)—and optimizes all criteria simultaneously via the unified function. An application example is the assignment of referees to football matches in an Italian championship (Scarelli and Narula, 2002) based on multiple criteria such as reliability, evaluation of fitness, international prestige, and refereed matches/number of years worked ($\mathrm{D3} := ++++, WS$).

5. *Goal programming* (*GP*). This function prioritizes the criteria and considers them in sequence, based on certain target values and bounds. An application example is reallocation of remedial education teachers from the schools at which they taught in the morning to the schools at which they are to teach during the afternoon (Lee and Schniederjans, 1983). The matching criteria for this matching instance were (a) the cost of travel between the schools, (b) the mutual preferences of teachers and schools, and (c) the recommendations of the teachers' supervisors, where (a) $>>$ (b) $>>$ (c) ($\mathrm{D3} := ++-, LG$).

3.1.4 D+: Time, Progression

The additional dimension of matching (*i.e.*, D+)—according to the PRISM taxonomy—is related to situations where the characteristics of one (or more) of the three main dimensions (*i.e.*, D1, D2, D3) undergo changes over time (*e.g.*, the numbers/ characteristics of individuals are not fixed). A practical example is matching tasks and computational resources in Grid computing, where both the set of tasks to be processed and the set of available computational resources vary dynamically. Our proposal for solving this class of best matching problem is *predictive* and *proactive* solution mechanisms inspired by the notion of *model predictive control*. The idea of predictive best matching is to optimally match elements of two or more sets in the current timeslot, while taking into account their possible characteristics in the (near) future timeslots. These ideas and concepts will be further elaborated throughout the case studies. In mathematical terms, the time dependency can be incorporated in the model by adding a time factor t to the model. For example, the dynamic resource-constrained many-to-one matching between sets I and J can be formulated as follows:

$$M{:}1 \,/\, RC \,/\, +, OS \,/\, DI, ES$$

$$\begin{aligned}
\max \quad & \sum_{i \in I(t)} \sum_{j \in (t)} P_{ij}(t)\,\chi_{ij}, \\
\text{s.t.} \quad & \sum_{i \in I(t)} r_{ij}(t)\,\chi_{ij} \le b_j(t), \quad \forall\, j \in J(t), \\
& \sum_{j \in J(t)} \chi_{ij} = 1, \quad \forall\, i \in I(t), \\
& \chi_{ij} \in \{0,1\}, \quad \forall\, i \in I(t),\ j \in J(t).
\end{aligned} \tag{M3.5}$$

3.2 Case Studies—A Synopsis

Four case studies have been conducted on four important areas related to manufacturing and service. In line with systematic specification of matching through the PRISM taxonomy, the ultimate goal of these case studies is to shed light on different aspects and dimensions of matching by presenting and formalizing new problems and developing, testing, and validating robust procedures for solving them (see RQ1-3). The case studies are briefly introduced below (see Table 3.1).

1. *Case 1: Collaborative Supply Networks*. Demand and capacity sharing among entities within a supply network are common practice, and have become attractive strategies for competing and non-competing supply enterprises. Examples include airlines, test and assembly factories, and outsourced maintenance and logistics providers. The purpose is to maximize profit and resource utilization, enable timely delivery to customers in spite of uncertain market demands and unexpected capacity shortages, and maximize the overall stability. Demand-capacity sharing protocols are defined for the suppliers with capacity shortage to utilize excess capacities of other suppliers, thus fulfilling their current customers' demand more effectively, while eliminating excess inventory of capacity sharing suppliers. These sharing roles vary over time. High frequency of collaboration may impose additional costs to the supply network in terms of transactions, negotiations, and lateral transshipment of stocks. Best matching is thus the key to minimize the collaboration costs through dynamic matching of suppliers and customers with respect to the customers' demand and suppliers' available capacity to share. Best matching protocol is also applied for finding the best matches between the sharing proposals during collaboration negotiations among suppliers. A set of novel mixed-integer programming formulations is developed for modeling and analyzing the combined matching-sharing decisions. The models are then solved for both static (w/o D+) and dynamic (w/ D+) cases, and validated using queuing theory, Task

Administration Protocols (TAP), and Predictive Best Matching Protocols (PBMP). It is shown mathematically and through numerical experiments that the proposed collaborative frameworks outperform the previous non-collaborative models in terms of resource utilization and stability, and provide dominating strategies in terms of optimizing the total profit and service level of the supply network.

2. *Case 2: Collaborative Assembly Lines*. A Collaborative Assembly Framework (CAF), inspired by the design principles of CCT is developed in this case to enhance the *balanceability* of assembly lines. The notion of the CAF lies in dynamic utilization of idle resources to eliminate bottlenecks. The CAF is composed of two modules: (1) tool sharing protocol, which makes dynamic tool sharing decisions among fully-loaded (*i.e.*, bottleneck) and partially-loaded workstations, and (2) best matching protocol, which dynamically matches tasks and workstations, and partially- and fully-loaded workstations for tool sharing. A multi-objective mixed-integer programming model is developed for mathematical representation and a fuzzy goal programming approach is applied for optimization purposes. The objectives are to minimize the number of workstations, (2) cycle time, and (3) the total collaboration cost. The developed CAF is proven to guarantee *relative balanceability* of assembly lines, depending on pairwise tool compatibility and tool sharing performance. Moreover, a Collaborative Multi-Agent System (CMAS) enhanced with a sharing-matching protocol is developed to execute the plan, control the process, and modify the tool sharing decisions, considering dynamic changes in the system's operations (*i.e.*, D+). The numerical experiments on a set of small-sized case studies repeated and expanded from previous research

show superiority of the CAF over the existing non-collaborative approaches in terms of line efficiency, utilization, and balanceability.

3. *Case 3: Clustering with Interdependent Preferences*. Generalized matching has a variety of applications in areas such as team and network design, scheduling, transportation, routing, production planning, facility location, allocation, and logistics. The problem is indeed analogous to the *capacitated clustering problem*, where a set of individuals are partitioned into disjoint clusters with certain capacities. This case study defines, formulates, and analyzes an important behavior associated with the generalized matching: The mutual influence of the elements of the same set on each other's preferences, if matched to the same element of the other set. Such preferences are referred to as *interdependent preferences* (IP). A binary program is developed to formulate the problem and provide the basis for analyzing the impact of IP on generalized matching decisions from two perspectives: Optimal cluster formation (fixed sets) and evolution (emergent sets). A Genetic Algorithm (GA) and an Evolutionary Algorithm (EA) are then developed to handle the complexity of the cluster formation problem, and enable the network of clusters to autonomously adapt to random changes, recover, and evolve. Results from several experiments indicate (a) significant impact of IP on the optimality of cluster formation and evolution decisions, and (b) efficiency of the developed evolutionary algorithms in handling the problem's complexity, and the emergent behavior of matching. The experiments also indicate the impact of IP on the accuracy and optimality of capacitated clustering decisions.

4. *Case 4: Collaborative Service Enterprises*. The evolution of the Internet, clouds, information and communication technologies, and collaboration sciences has transformed traditional organizations of entities (*e.g.*, humans; machines; enterprises) to highly distributed, internetworked, and collaborative virtual (v-)Organizations. The emerging extensions of the notion of cloud computing to areas such as manufacturing, business, education, banking, and healthcare have enabled more systematic integration, harmonization, and sharing of distributed resources for processing of dynamic and diverse pool of tasks. This case study contributes to the design of Collaborative Networked Organizations (CNO), in terms of *location of resources* and *allocation of tasks* in the network, by incorporating both the physical and virtual dimensions of CNO in the decisions. The problem is then to dynamically find, throughout the CNO, the best locations for individual resources (*e.g.*, program; computer; sensor; robot) and the optimal allocation of each individual task to proper resources. Collaboration is enabled through cloud and cyber-supported communication technologies for sharing resources and electronic (e-)Tasks among remote organizations, such that the overall service level, network stability, and resource utilization are optimized. The problem is referred to as Collaborative Location-Allocation problem (CLAP). A bi-objective mixed-integer programming formulation is developed for modeling the CLAP. Due to its computational complexity of the CLAP, a tabu search algorithm is developed with a novel best matching heuristic inspired by the natural *justice rule* (TS-Jr.). Several numerical experiments illustrate, analyze, and highlight the unique features of the CLAP and for optimal and efficient (re)configuration of CNO.

Table 3.1. Case studies—summary of developments and contributions*.

Case	PRISM Mapping (D1/D2/D3/D+)				Validated concepts / tools
	D1	D2	D3	D+	
1. Collaborative supply networks	*M*:1	*RC*, *RS*	4, –+++, *OS*	*DI*	Network stability / TAP, PBMP
2. Collaborative assembly lines	*M*:1	*RC*, *PR*, *RS*	3, – – –, *GP*	*DI*	Line balanceability / CAF, CMAS
3. Clustering with IP	*M*:1	*RC*, *IP*	1, +, *OS*	*ES*	IP / GA, EA
4. Collaborative service enterprises	*M*:*M*:1	*RC*, *RS*	2, +–, *WS*	*DI*	CLAP / TS-Jr.

* See Figs. 4.2, 5.2, 6.2, and 7.2 in Chapters 4 to 7.
CAF: Collaborative Assembly Framework
CLAP: Collaborative Location-Allocation Problem
CMAS: Collaborative Multi-Agent System
DI: Dynamic Inputs
EA: Evolutionary Algorithm
ES: Emergent Sets
GA: Genetic Algorithm
GP: Goal Programming
IP: Interdependent preferences
M:1: Many to one
OS: Overall Satisfaction
PBMP: Predictive Best Matching Protocol
PR: Precedence Relations
RC: Resource Constraints
RS: Resource Sharing
TAP: Task Administration Protocol
TS-Jr.: Tabu Search with natural Justice rule
WS: Weighted Sum

CHAPTER 4. CASE 1—COLLABORATIVE SUPPLY NETWORKS☆

Design and coordination of supply networks is a practical example of resource-constrained matching, where—even in the simplest two-tier case—the participants must be matched according to a set of criteria. This matching problem is typically on a many-to-one basis, implying that each supplier can serve more than one customer but each customer must be connected to only one (primary) supplier at any point of time (period). The motivation of this case study is to indicate the impact of best matching on the quality of collaborative Demand-Capacity Sharing (DCS) activities throughout supply networks. DCS between the elements of supply networks is common in practice, and has turned into an attractive strategy for competing and non-competing suppliers. DCS decisions help the suppliers with capacity shortage, referred to as *demand sharing suppliers*, utilize the extra capacities of other suppliers, referred to as *capacity sharing suppliers*, in fulfilling their current demand more effectively while reducing extra inventories of capacity sharing suppliers. These DCS roles vary over time. In spite of their unique advantages, however, high frequency of DCS decisions may impose additional costs to the Collaborative Network of Suppliers (CNS) in terms of transactions, negotiations, and lateral transshipment of stocks. Best matching is

☆ The preliminary version of this case study was presented at *the 22nd International Conference on Production Research*, Brazil, 2013. The materials presented in this case study are adapted from two works of the author published in the *International Journal of Production Economics* (DOI:10.1016/j.ijpe.2013.11.015 and DOI:10.1016/j.ijpe.2015.07.038).

applied in this case as an effective tool for minimizing DCS costs through (1) dynamic matching of suppliers and customers with respect to the customers' demand as well as the suppliers' available capacity to share, and (2) finding the best matches between DCS proposals during collaboration negotiations among suppliers. It will be proven mathematically and through numerical experiments that: (1) resource-sharing (*i.e.*, DCS) outperforms traditional non-collaborative models in terms of resource utilization and stability, and (2) best matching provides a promising strategy, compared collaborative and non-collaborative without best matching, in terms of total CNS profit and service level (Moghaddam and Nof, 2013[a], 2014, and 2016[a]).

4.1 Motivation

Dynamic nature of market behavior and unforeseen changes in customers' demand are inevitable features of modern supply networks. In distributed networks of suppliers, lateral collaboration between suppliers is known as an effective strategy in reacting to dynamic market behavior and abrupt variations in demand (Jagdev and Thoben, 2001). Collaborative Network of Suppliers (CNS) refers to a set of independent suppliers collaborating laterally under specific coordination and collaboration protocols (Nof, 2003), such that *mutual* benefits are achieved. In this context, sharing *resources*, *information*, and *responsibilities*, as the three pillars of collaboration (Nof, 2007), is the key enabler in reducing the total costs and improving the global efficiency of the entire CNS (Gavirneni, 2002). In a typical CNS, each supplier has finite capacities for producing its own customer orders. Through effective collaboration, however, the suppliers can improve their stability under DCS decisions and protocols, especially in case the local orders cannot be satisfied with the available local capacity (Yoon and Nof, 2010). Accordingly, the overall inventory level of demand sharing suppliers and backorder/stockout level of capacity sharing suppliers are reduced resulting in significant mutual benefits to all collaborating suppliers.

Various DCS strategies have been developed in literature in order for individual suppliers (Kutanoglu and Mahajan, 2009; Tiacci and Saetta, 2011; Torabi and Moghaddam, 2012) or collaborative networks of suppliers (Lee *et al.*, 2007; Yoon and Nof, 2010 and 2011; Seok and Nof, 2013) to deal with uncertain and dynamic demand patterns (Table 4.1). Lateral collaboration is a promising strategy in mitigating the demand and/or supply disruption, and always outperforms the *non-collaborative* strategies in terms of cost, utilization of resources, and service level (Burton and Banerjee, 2005; Yoon and Nof, 2010; Tiacci and Saetta, 2011). A critical problem, however, is the additional costs associated with collaboration, mostly related to lateral transshipment. Several approaches have been developed and examined in literature, *e.g.*, substitution of systematic inventory level equalization policies with ad-hoc and purpose-oriented transshipment (Burton and Banerjee, 2005; Olsson, 2009), and enhancement of reactive transshipment policies with proactive redistribution of stocks (Paterson *et al.*, 2011 and 2012). The existing approaches deal with minimizing the *indirect* costs of lateral collaboration, *e.g.*, improving the service level, minimizing the stockouts. Nevertheless, significant amounts of fixed and variable costs associate with the lateral transshipment/physical distribution of stocks are still present in all existing policies. This is the major limitation of the collaborative strategies, despite their remarkable benefits to CNSs. Thus, the *frequency* and *quantity* of DCS must be considered so as to minimize the costs of negotiations, information and resource sharing, physical distribution of stocks, and other business operation.

Table 4.1. Some recent studies on enterprise collaboration.

Study	Criteria	D+
Burton and Banerjee (2005)	Lateral transshipment cost	
Lee *et al.* (2007)	Response time to demand variations; Penalty costs	
Kutanoglu and Mahajan (2009)	Overall service level; Stock costs	
Olsson (2009)	Service level; System cost	
Yoon and Nof (2010)	Global benefit of CNE; Demand fulfillment rate	●
Yoon and Nof (2011)	Total profit; Demand fulfillment rate; Impacts of low-performance parties	●
Tiacci and Saetta (2011)	Mean supply delay	
Paterson *et al.* (2011 and 2012)	Stockout probability; Safety stock; Service level	
Torabi and Moghaddam (2012)	Total profit; Lead-time; Inventories and backorders/ stockouts	
Axsäter *et al.* (2013)	System cost; Service level	
Seok and Nof (2014)	Lost sales; Capacity utilization; Long-term balance of benefits to all parties	●
Moghaddam and Nof (2013[a], 2014, 2016[a])	Total cost; Demand fulfillment; Resource utilization; Stability	●

4.2 Outline

Dynamic changes in the demand are not essentially in line with the variations in the available capacity of suppliers. Accordingly, the suppliers can be classified into either of the following categories: (1) Suppliers with capacity shortage; (2) Suppliers with extra capacity. Using the DCS protocols, the suppliers with capacity shortage are enabled to share their *unfulfilled demand* with the suppliers with extra capacity through negotiation (Yoon and Nof, 2010 and 2011). In view of that, a portion of the demand related to the suppliers with capacity shortage is *indirectly* satisfied by the suppliers with extra capacity. However, if the suppliers are not properly *matched* to the customers with respect to the relative correspondence between the demand and capacity, the efficiency of DCS decisions will diminish due to excessive rate of sharing.

Given that all suppliers are parts of a CNS seeking mutual benefits, our solution to the aforementioned problem is to enable dynamic matching of suppliers and customers in order to minimize the *gap* between the capacity of the suppliers and their allocated demands. It will be shown that this gap has a direct correlation with the lateral collaboration costs, and therefore, finding the best matching solution with the minimum gap will minimize the total cost of collaboration as well. Our objective is to improve the efficiency of collaboration decisions through (Figure 4.1)

1. *Resource-sharing decisions*. Maximizing the overall demand fulfillment rate, utilization of resources, and stability of CNS, and minimizing the inventory holding and backordering costs;
2. *Best matching decisions*. Matching suppliers and customers based on the capacity-demand gaps, and matching DCS proposals based on the fixed and variable costs of collaboration.

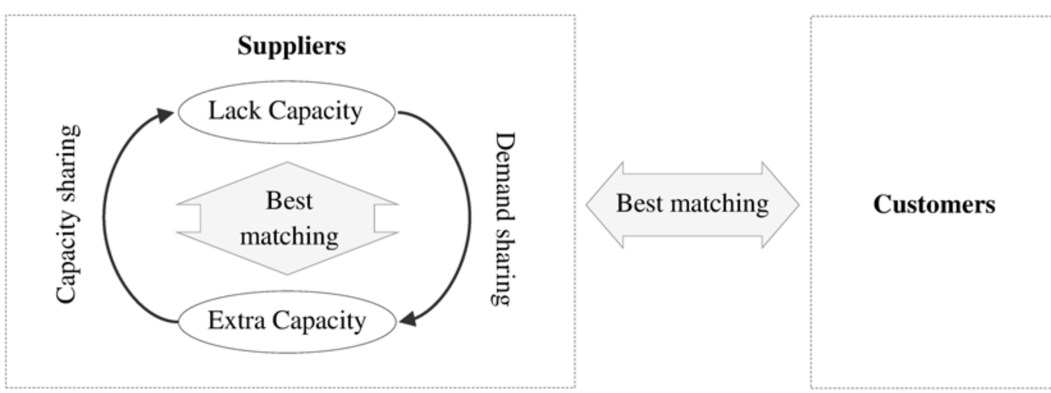


Figure 4.1. Integration of demand and capacity sharing with best matching decisions.

Consider a CNS where the dynamic demand of $|I|$ customers for N different product types must be satisfied by a set of $|J|$ collaborative suppliers over a horizon of T. Each supplier is responsible for satisfying the demand of a specific set of customers using its limited capacity. The main objective is to protect the overall competitiveness and market share of the collaborative network of suppliers. Accordingly, our fundamental assumption is that depending on the dynamic variations of demands and capacities, customers can be served by different suppliers in different periods. In each period, upon supplier-customer matching, demand-sharing (if required) is performed by the suppliers with capacity shortage through negotiations with suppliers with extra capacity, and capacity sharing is accomplished through lateral transshipment of products between suppliers. The general characteristics and assumptions of the problem are then defined as follows:

1. Each supplier has finite capacity and is able to produce all product types, which are under sharing considerations.
2. Shared fulfillment is not allowed per product type; *i.e.*, in each period, each customer must be matched to exactly one supplier regarding each product type.
3. All parameters related to production, inventory holding, backordering, and DCS cost, and capacities of the supplier in each period are known.
4. DCS cost refers to the fixed cost of DCS per proposal, *i.e.*, updating, sharing and analyzing the information and available resources, preparing and evaluating the proposals, and lateral transshipment of products per unit between suppliers in each period.

5. DCS are performed instantaneously, *i.e.*, the negotiation and sharing times are negligible.
6. Backorders are allowed in all suppliers.
7. Demand is uncertain but the parameters are known. Uncertain demand parameters are treated as Triangular Fuzzy Numbers characterized by possibility distributions.

The problem is studied from two main perspectives. First, it is formulated as a Mixed Integer Programming (MIP) model and solved using CPLEX, in order to investigate the notion of collaborative best matching in supply networks from a long-term planning (static) perspective. Then, a Real-Time Optimization (RTO) mechanism enhanced with Task Administration Protocol (TAP) is developed for priority-based allocation/re-allocation of resources and real-time monitoring of DCS processes. The TAP is enhanced with a Predictive Best Matching Protocol (PBMP), which optimizes allocation decisions in real-

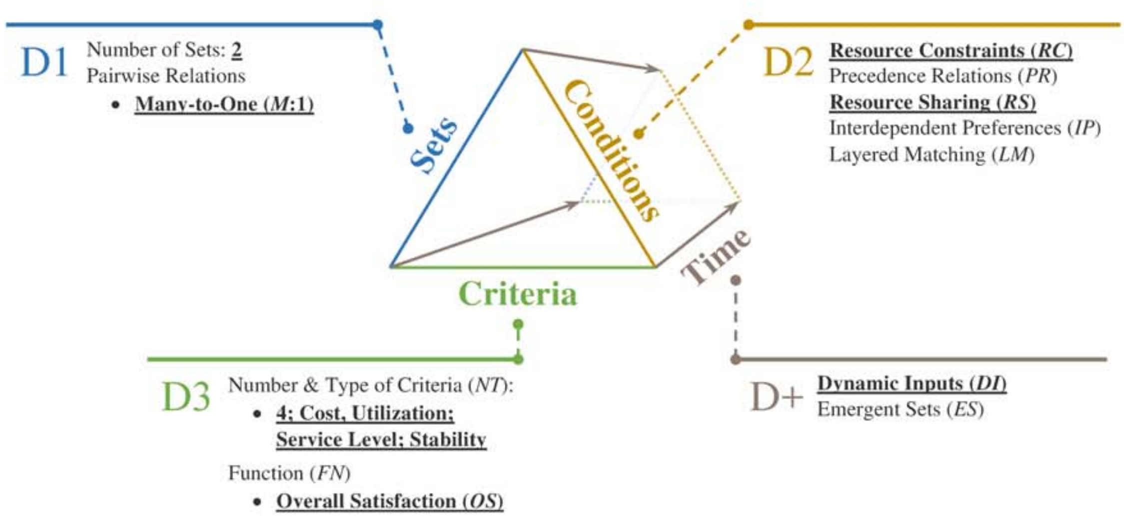


Figure 4.2. Classification of the collaborative best matching problem in supply networks according to the PRISM taxonomy of best matching.

time while taking into account potential future events. The RTO mechanism is proven in experiments to be an advantageous solution for mitigating the undesirable impacts of uncertainty and dynamicity on the CNS performance with respect to the four decision criteria. Figure 4.2 shows the status of the problem under study according to the PRISM taxonomy of best matching.

4.3 Optimization: MIP and CPLEX

A framework is developed for implementation of the resource-sharing and best matching decisions. The DCS and best matching protocols are dynamically activated in each period, taking into account the available capacities of suppliers, the demand forecasts related to each customer, and the inventory and backorder levels. The framework yields the optimal matches between the suppliers and customers, the optimal matches between the DCS proposals, and the optimal decisions recommending production and DCS plan, through the following steps (Figure 4.3):

1. The customers' demand, available capacity of suppliers, production, inventory holding, and backordering costs, fixed cost of collaboration (*i.e.*, negotiations, preparing and sharing DCS proposals), and variable cost of collaboration (*i.e.*, transshipment between suppliers). The model with fuzzy input parameters (*i.e.*, demand) is converted to its equivalent crisp model using a *possibilistic programming* method (see Moghaddam and Nof, 2014).

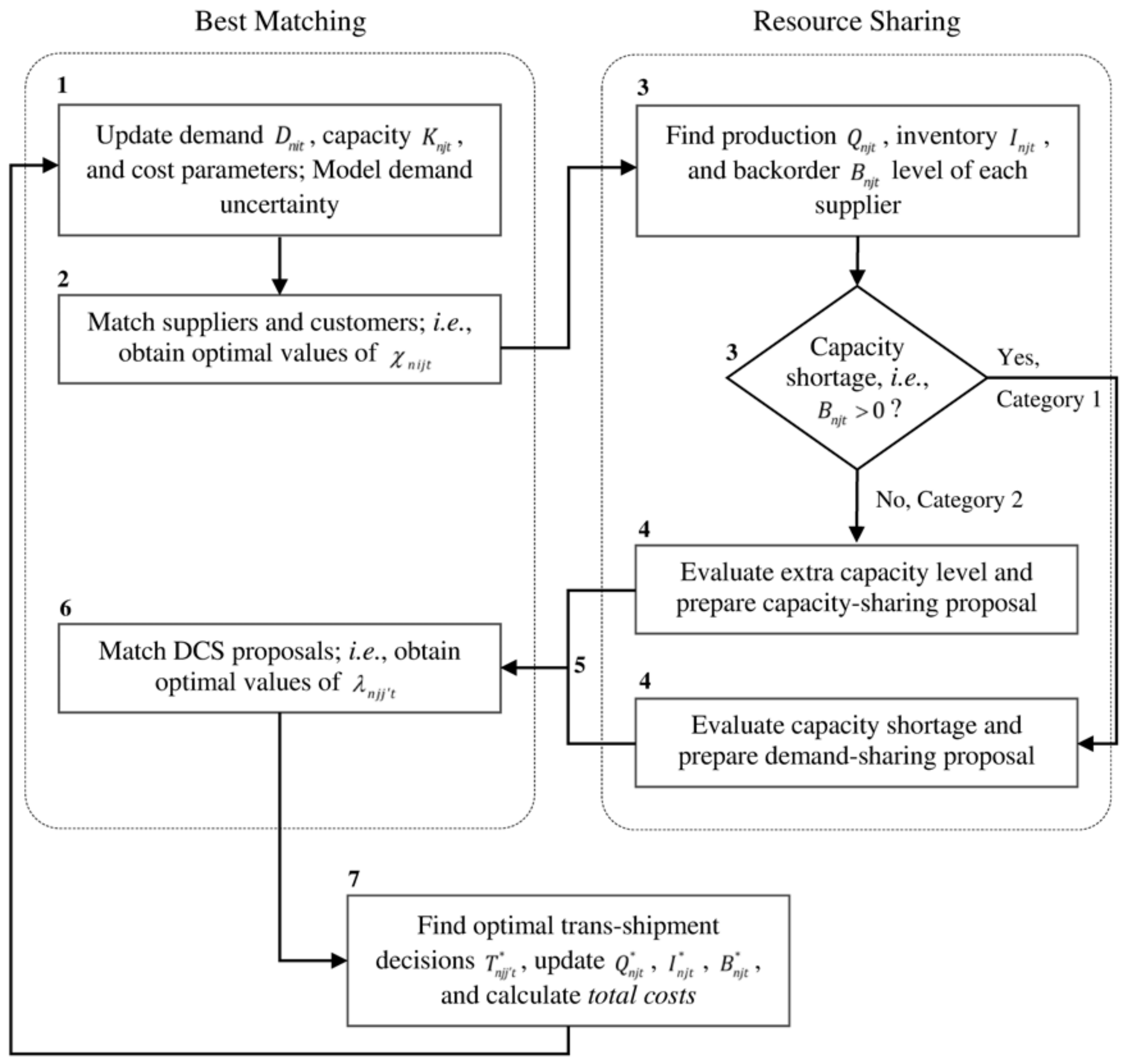


Figure 4.3. General framework of matching with resource-sharing.

2. Best matching between the sets of suppliers and customers is defined. The matching criterion is the level of correspondence between the customers' demand and the available capacity of suppliers.
3. After deciding the production, inventory, and backorder levels of each supplier, the suppliers are classified into two categories based on their available capacity.

4. Capacity shortage (*i.e.*, backorder) and extra capacity (*i.e.*, inventory) levels are evaluated at each supplier with capacity shortage and with extra capacity, respectively. Since the inventory and the backorder variables cannot take positive values at the same time, each supplier must be classified into either Category 1 or Category 2.
5. Each supplier with capacity shortage prepares and submits demand-sharing proposals. Each supplier with extra capacity analyzes the proposals, prepares and submits capacity-sharing proposals.
6. Best matching between the DCS proposals is obtained. The matching criteria are the level of correspondence between the capacity shortage of and the extra capacity of the suppliers, fixed cost of collaboration, and unit lateral transshipment cost between the suppliers. This is a *many-to-many* best matching implying that the demand shared by each supplier with capacity shortage can be fulfilled by one or more suppliers with extra capacity, and the capacity shared by each supplier with extra capacity can be utilized by one or more suppliers with capacity shortage.
7. Based on the optimal matchings between DCS proposals, the optimal decisions are then made on the production, inventory, and backorder levels, and the frequency and amount of DCS among all parties of the CNS. Note that this framework does not necessarily lead to zero inventory and backorder levels at the end of each period. There may be some inconsistency between the overall demand and capacity levels of the entire CNS. Besides, some suppliers may prefer backordering or inventory holding because of the lower costs compared to collaboration (or other reasons).

4.3.1 Mathematical Formulation

Due to the uncertainty associated with the demand, a possibilistic MIP (Mixed-Integer Programming) formulation is developed for modeling the problem of matching with resource-sharing in collaborative supply networks. In order to solve the possibilistic model, a possibilistic programming method is applied (see Moghaddam and Nof, 2014, for details). The objective function and constraints of the model is as follows[1]:

1. *Objective function.* The objective is to minimize the total cost of production, inventory holding, backordering, and lateral transshipment between the suppliers, along with fixed cost of collaboration, for all product types over the decision horizon.

$$\min \quad \sum_{n\in N}\sum_{t\in T}\sum_{j\in J}\left(p_{njt}Q_{njt} + h_{njt}I_{njt} + b_{njt}B_{njt} + \sum_{j'\in J}\left(v_{njj't}T_{njj't} + f_{jj't}\lambda_{njj't} \right) \right). \qquad (4.1)$$

2. *Inventory balance constraints*. This set of constrains guarantees inventory balance in each supplier for each product type in each period. They imply that in each period, the overall inputs at each supplier regarding each product type must be balanced with the overall outputs (see Figure 4.4). The possible strategies for holding this balance equality are: (1) inventory holding or backordering, (2) lateral collaboration through DCS with other suppliers, and (3) matching the suppliers (based on production capacity) and the customers (based on demand).

$$\sum_{i\in I}\chi_{nijt}D_{nit} = I_{nj,t-1} - B_{nj,t-1} + Q_{njt} - I_{njt} + B_{njt} + \sum_{\substack{j'\in J\\ j'\neq j}}\left(T_{nj'jt} - T_{njj't} \right), \qquad (4.2)$$
$$\forall n\in N, j\in J, t\in T.$$

---

[1] This model is an extension of Model (M3.3), *M*:1 / *RC*, *RS* / +, *OS*, presented in Chapter 3.

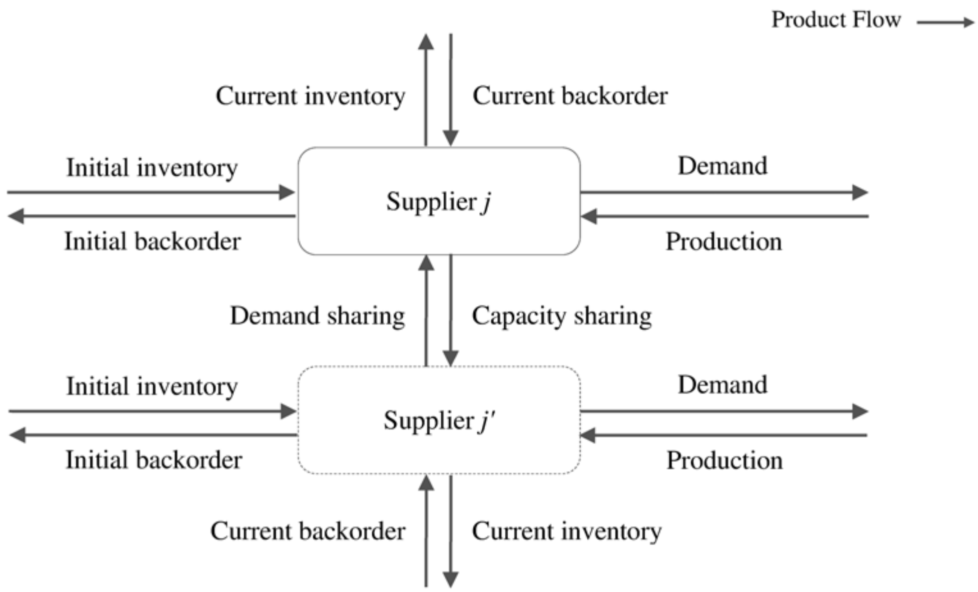


Figure 4.4. Product inflow/outflow and inventory balance in two collaborative suppliers j and j' (from the perspective of supplier j).

It can be proven that in the "collaboration with no best matching" strategy, the gap between the capacity of suppliers and the demand of the matched customers is directly correlated with the rate of lateral collaboration.

Proposition 4.1. *In the "collaboration with no matching" strategy, the gap between the capacity of suppliers and the demand of the allocated customers is directly correlated with the rate of lateral collaboration, given that collaboration is a dominant strategy.*

Proof. Let G_{njt}, $\forall n \in N, j \in J, t \in T$, denote the capacity-demand gap (see Eq. (4.2)):

$$G_{njt} = \underbrace{\sum_{i\in I} \hat{\chi}_{nijt} \tilde{D}_{nijt}}_{\text{Demand}} - \underbrace{\left(Q_{njt} + I_{nj,t-1} - B_{nj,t-1}\right)}_{\text{Capacity}} = -I_{njt} + B_{njt} - \sum_{\substack{j'\in J \\ j'\neq j}} T_{njj't} + \sum_{\substack{j'\in J \\ j'\neq j}} T_{nj'jt},$$
$$\forall n \in N, j \in J, t \in T,$$

where $\hat{\chi}_{nijt}$ is fixed and known, $\forall n \in N, j \in J, t \in T$ (*i.e.*, fixed pre-matching between the suppliers and the customers). If

$$G_{njt} = -I_{njt} + B_{njt} - \sum_{\substack{j'\in J \\ j'\neq j}} T_{njj't} + \sum_{\substack{j'\in J \\ j'\neq j}} T_{nj'jt} = 0, \qquad \forall n \in N, j \in J, t \in T,$$

given that all variables must be non-negative and minimized, we have

$$I^*_{njt} = B^*_{njt} = \sum_{\substack{j'\in J \\ j'\neq j}} T^*_{njj't} = \sum_{\substack{j'\in J \\ j'\neq j}} T^*_{nj'jt} = 0, \qquad \forall n \in N, j \in J, t \in T.$$

Now suppose for some $\hat{n} \in N, \hat{j} \in J, \hat{t} \in T$ we have (*i.e.*, capacity shortage)

$$G_{\hat{n}\hat{j}\hat{t}} = -I_{\hat{n}\hat{j}\hat{t}} + B_{\hat{n}\hat{j}\hat{t}} - \sum_{\substack{j'\in J \\ j'\neq \hat{j}}} T_{\hat{n}\hat{j}j'\hat{t}} + \sum_{\substack{j'\in J \\ j'\neq \hat{j}}} T_{\hat{n}j'\hat{j}\hat{t}} > 0.$$

It can be shown through *complementary slackness theorem* that the inventory $I_{\hat{n}\hat{j}\hat{t}}$ and backorder $B_{\hat{n}\hat{j}\hat{t}}$ variables cannot take positive values at the same time. Similarly, this statement is true for the demand and capacity sharing terms. Hence, since $G_{\hat{n}\hat{j}\hat{t}} > 0$, we have $I^*_{\hat{n}\hat{j}\hat{t}} = \sum_{\substack{j'\in J \\ j'\neq \hat{j}}} T^*_{\hat{n}\hat{j}j'\hat{t}} = 0$, and thus

$$G_{\hat{n}\hat{j}\hat{t}} = B^*_{\hat{n}\hat{j}\hat{t}} + \sum_{\substack{j'\in J \\ j'\neq \hat{j}}} T^*_{\hat{n}j'\hat{j}\hat{t}} > 0.$$

In other words, suppliers with capacity shortage can follow two strategies of either backordering or sharing the demand with the other suppliers. If collaboration is a dominant strategy, it implies that $v_{nj'jt} < b_{njt}$, $\forall n \in N, j \in J, t \in T$. Then, $B^*_{\hat{n}\hat{j}\hat{t}} = 0$ and

$G_{\hat{n}\hat{j}\hat{t}} = \sum_{\substack{j' \in J \\ j' \neq \hat{j}}} T^{*}_{\hat{n}j'\hat{j}\hat{t}}$. However, since the capacities of the suppliers are limited, a portion of this gap may be left as backorder, *i.e.*, $B^{*}_{\hat{n}\hat{j}\hat{t}} > 0$. In general,

$$G_{njt} > 0 \qquad \Leftrightarrow \qquad G_{njt} \propto \sum_{\substack{j' \in J \\ j' \neq j}} T_{nj'jt}, \quad \forall\, n \in N, j \in J, t \in T,$$

and similarly,

$$G_{njt} < 0 \qquad \Leftrightarrow \qquad \left| G_{njt} \right| \propto \sum_{\substack{j' \in J \\ j' \neq j}} T_{njj't}, \quad \forall\, n \in N, j \in J, t \in T. \ \blacksquare$$

3. *Best matching constraints.*

   - Between suppliers and customers: Constraints (4.3) imply that each customer must be matched to exactly one supplier for receiving a specific product type in each period, while there is no limitation on the number of customers matched to each supplier (except the available capacity).

$$\sum_{j \in J} \chi_{nijt} = 1, \qquad \forall\, n \in N, i \in I, t \in T. \tag{4.3}$$

   - Between DCS proposals: Constraints (4.4) ensure that suppliers can share their demand and capacity just in case their DCS proposals are matched in each period, regarding each product type, considering the fixed cost of collaboration (*M* is a very large positive number).

$$T_{njj't} \leq M\, \lambda_{njj't}, \qquad \forall\, n \in N, i \in I, j, j' \in J, t \in T. \tag{4.4}$$

4. *Capacity constraints*. This set of constraints guarantees that in each period, the production level of the suppliers for each product type does not exceed the capacity limit.

$$Q_{njt} \le K_{njt}, \qquad \forall n \in N, j \in J, t \in T. \tag{4.5}$$

5. *Feasibility of decision variables*. This set of constraints ensures that all decision variables are non-negative and the auxiliary best matching variables are binary.

$$\begin{gathered} Q_{njt}, I_{njt}, B_{njt}, T_{njj't} \ge 0, \quad \chi_{nijt}, \lambda_{njj't} \in \{0,1\}, \\ \forall n \in N, i \in I, j, j' \in J, \, t \in T. \end{gathered} \tag{4.6}$$

Note that the problem is composed of two complementary matching processes, which are in fact multi-dimensional. In both problems, besides the mentioned sets (*i.e.*, suppliers-customer; DCS proposals), there are two other dimensions—product types, and time. Since the best matching decisions for different product types and at each period are *independent*, however, the problem has been divided into several independent 2D matchings. Since the demand parameters in Constraints (4.2) are fuzzy (denoted by the superscript ~), the model is treated as a possibilistic MIP, which must be converted into an equivalent crisp model. The uncertainty associated with the demand forecasts is modeled using possibility distributions, based on the historical demand data (objective data). The applied possibilistic programming method applied for defuzzification of the model is briefly described in Moghaddam and Nof (2014). The combined DCS-best matching concept, and its impact on the utilization, service level, and stability of supply networks has been mathematically validated using queuing theory.

Proposition 4.2. *Considering the entire CNS as a single "server", if the overall arrival rate (i.e., demand) is lower than the overall service rate (i.e., capacity), all suppliers are guaranteed to have stable processes via collaboration.*

See Moghaddam and Nof (2014) for the proof of Proposition 4.2 using queuing theory.

4.3.2 Numerical Experiments

A set of experiments are conducted in this section to investigate the impact of DCS with best matching decisions on the performance of CNSs, based on the following three scenarios:

1. *No collaboration* (C_0). Each supplier has to fulfill its own demand, and no demand and capacity sharing is allowed among suppliers. This scenario is defined to highlight the relative value of lateral collaboration in CNSs.
2. *Collaboration with pre-matching* (C_1). Suppliers collaborate through dynamic sharing of their demand and capacity in each period to cope with the overall variations in their allocated demand. Through DCS, demand-sharing proposals, from the suppliers with capacity shortage, and capacity sharing proposals, from the suppliers with extra capacity, are received, analyzed, and matched. Under this scenario, the rate of lateral collaboration has a direct relation with the *gap* between the capacity of the suppliers and the demand of their fixed and *pre-matched* customers. Thus, this scenario is defined to underline the deficiency of DCS *without* dynamic best matching (with fixed pre-matching) in terms of total collaboration cost.
3. *Collaboration with best matching* (C_2). Prior to making the DCS decisions in each period, the suppliers and customers are matched according to the correspondence between their demand and capacity. It will be shown through Scenario C_2 that best matching minimizes the capacity-demand gap of each supplier, leading to lower

total cost compared to the collaboration with fixed pre-matching scenario (*i.e.*, C_1). Thus, Scenario C_2 is expected to outperform Scenario C_1; the reason is that despite maintaining the unique properties of C_1, C_2 also minimizes the fixed and variable costs of lateral collaboration.

The input parameters for the design of the CNS are available in Moghaddam and Nof (2014). Seven distributed suppliers are considered, which produce and deliver one product type to 21 customers over three consecutive periods. The data is generated randomly taking into account the meaningful relations between the values of the parameters. The General Algebraic Modeling System (GAMS) Software and the CPLEX solver are applied for solving the auxiliary crisp MIP model. Detailed information on the optimal values of decision variables under each scenario are also available in Moghaddam and Nof (2014). Under C_1, there are fixed pre-matchings between the suppliers and the customers in all periods—each supplier has to fulfill the demand of its own predefined set of customers during the decision horizon, while the suppliers and the customers are dynamically matched under C_2.

Figure 4.5 illustrates the initial supplier-customer pre-matchings related to C_0 and C_1 and the dynamic matching obtained via C_2 based on the capacity-demand gap. Figure 4.6 shows the total number of customers matched to each supplier under C_2 in different periods. Under C_2, different customers are matched to each supplier over periods one to three. That is, the best matching model optimally matches suppliers and customers depending on the relative capacity-demand gap. In this example, the matching decisions vary from one period to another, and are considerably different from the fixed pre-

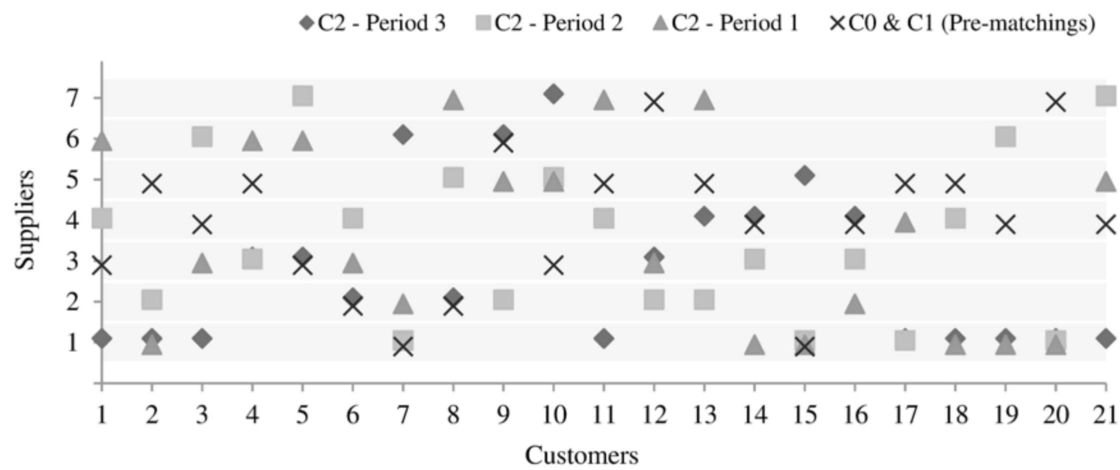


Figure 4.6. Supplier-customer matching under different scenarios.

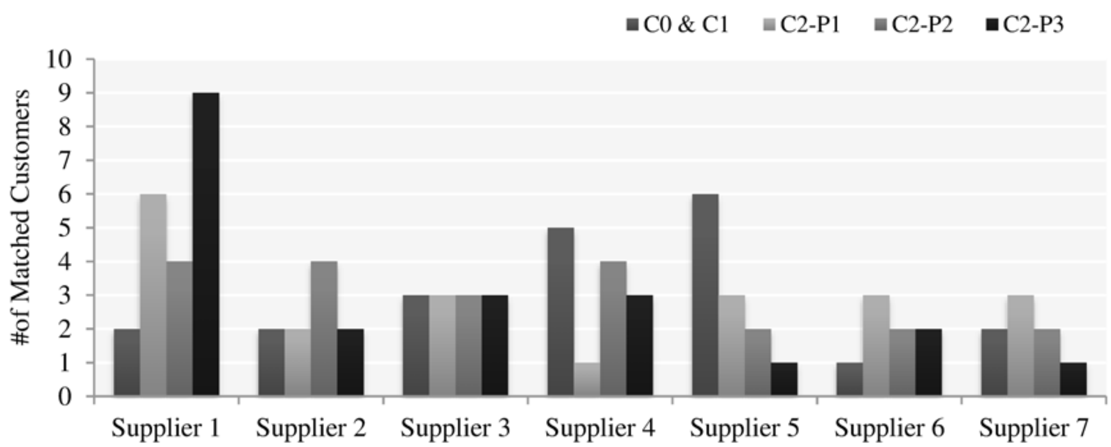


Figure 4.5. Total number of customers matched to each supplier in each period, under different scenarios.

matchings. Figure 4.6 shows that the total number of customers matched to each supplier is also different in each period, which indicates the variations in the suppliers' capacities, the customers' demands, or both. The scenarios have been examined and compared with respect to the following criteria (Figure 4.7):

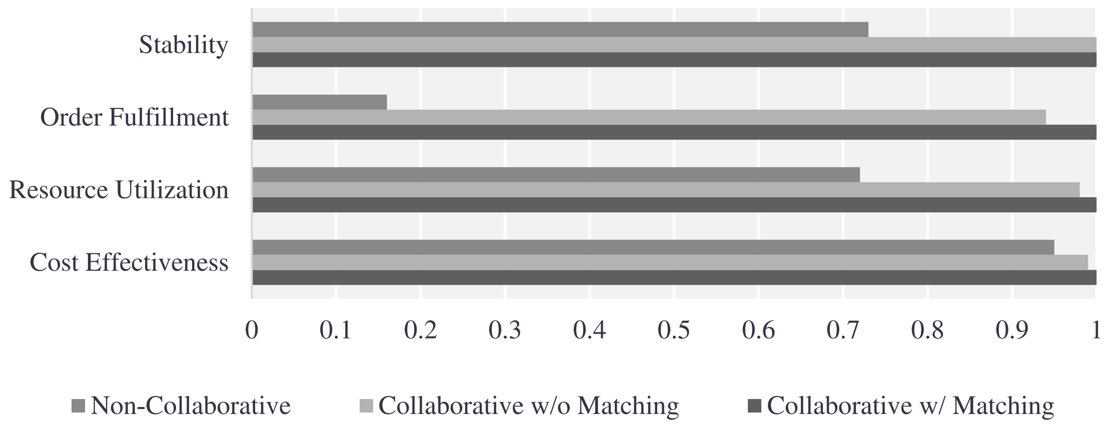


Figure 4.7. Comparative analysis of scenarios C_0, C_1, and C_2. (The average results are relatively normalized in [0, 1].)

1. *Order fulfillment*. Under C_0, the capacity-demand gap results in either excessive inventory or backorder at the end of each period. Under the collaborative scenarios (*i.e.*, C_1 and C_2), however, DCS is an attractive strategy to minimize the capacity-demand gap through lateral collaboration. Nevertheless, since the capacities of the suppliers are limited, even the collaborative scenarios may result in certain amounts of inventories/backorders. The observations indicate lower level of unfulfilled orders under C_2, compared to C_1 and C_0. The collaborative scenarios (*i.e.*, C_2 and C_1) outperform the non-collaborative scenario (*i.e.*, C_0) in terms of demand fulfillment thanks to lateral collaboration. However, C_2 is also preferred to C_1, because it enables higher service level—some suppliers may be reluctant to sharing due to high costs of collaboration relative to inventory holding/ backordering; thus, C_2 is recommended to minimize the capacity-demand gap through matching prior to making the DCS decisions, in order to improve the service level and eliminate "unnecessary collaborations".

2. *Resource utilization*. For each supplier, resource utilization is defined as the percentage of resources actually consumed relative to the amount of resources planned to be consumed. This evaluation criterion can be calculated through the following formula:

$$U_{njt} = \frac{Q^*_{njt}}{K_{njt}} \times 100, \qquad \forall n \in N, j \in J, t \in T, \tag{4.7}$$

where the optimal production level and capacity upper bound are respectively considered as the actual and the planned levels of utilization of resources. With this definition, resource utilization is 100% in all demand sharing suppliers. Through collaboration resource utilization can also be improved in the capacity sharing suppliers compared. Lateral collaboration indeed enhances the utilization of the *existing* resources rather than investing in *extra* resources for dealing with the dynamic behavior of customers' demand. That is why C_2 and C_1 outperform C_0 in terms of resource utilization.

3. *Cost effectiveness*. In addition to considerable increases in the demand fulfillment rate under the collaborative scenarios C_2 and C_1 compared to C_0, the total cost is also minimized through lateral collaboration. The substantial decrease in the total cost is due to the elimination of unnecessary inventories and backorders through DCS. Moreover, the total cost in C_2 is also lower than the total cost in C_1, which indicates the impact of best matching on the reduction of capacity-demand gaps of the suppliers, resulting in lower rates of lateral collaboration throughout the entire CNS.

4. *Stability*. Through analogical comparison of the supply processes with the *birth and death* process, the processes at the CNS and each individual supplier have been modeled as Continuous-Time Markov Chains, in order to analyze the stability of the process under each scenario. It has been proven through queuing models that if the overall demand rate is lower than the overall production rate, all suppliers are guaranteed to be stable, under collaboration (see Proposition 2). In practice, however, due to disruptions in demand and/or capacity, CNSs are not necessarily guaranteed to undergo an overall stable process. Collaboration through DCS, however, increases the stability of each supplier as much as possible. In the numerical example, the overall process of arrival and delivery of orders in the entire CNS is *unstable* or *oversaturated*. Nevertheless, our results show that even in case of overall instability, collaborative scenarios (*i.e.*, C_1 and C_2) can still improve this unstable process compared to the traditional no-collaboration scenario (*i.e.*, C_0).

4.4 Control: TAP and PBMP

A novel control mechanism is developed to rationalize, coordinate, and harmonize distributed operations, and optimize collaboration decisions in real-time. The first objective is addressed by developing a TAP (Task Administration Protocol) for effective control of DCS operations. The TAP is composed of three sub-protocols for priority-based task initialization, resource-aware task allocation, and task monitoring re-allocation (Ko and Nof, 2012). In this context, a *task* refers to an order, and the *resources* used for processing the tasks refer to the suppliers. A PBMP (Predictive Best Matching Protocol) is also developed to tackle the second objective, *i.e.*, RTO (Real-Time Optimization) of DCS decisions. The PBMP is applied for dynamic matching of orders to resources in real-time. Inspired by the notion of Model Predictive Control (MPC), the PBMP matches entities (*e.g.*, proposals, suppliers) in the current timeslot, while taking into account system characteristics in the *near future* timeslots. The RTO is then concerned with the total costs of collaboration (*e.g.*, fixed costs of negotiation, information sharing) along with variable costs of transshipment, and service level, *i.e.*, demand fulfillment rate.

The proposed mechanism requires cyber-supported collaboration infrastructures for effective information sharing and enhanced connectivity among distributed participants. Advances in collaborative e-Work over the last two decades have provided effective computer-supported and communication-enabled solutions for design, engineering, and control of CNS (Nof *et al.*, 2015). Development of the TAP and PBMP then relies highly on agent-based technologies in order to proactively identify resources, provide real-time value-added information, and reduce potential conflicts and errors (Klusch, 2001), as well as workflow technologies to enable scalability, availability, and reliability of processes. It is shown that deployment of agents, coordination protocols, and workflows—as the first theoretical foundation of collaborative e-Work (Nof, 2007)—coupled with planning models provide a powerful *design-control loop* that enhances the quality of collaboration decisions. The PBMP is indeed an agent-based optimization technique based on *mediator architecture* where a mediator agent monitors, synchronizes, and optimizes the activities of other distributed agents (Barbati *et al.*, 2012).

4.4.1 General Logic

The TAP and PBMP are developed for real-time execution of the generated plan in each period. The protocols frequently identify the current state of each individual supplier through distributed agents and following certain workflows, and take necessary actions to handle dynamic and unforeseen situations. Task administration—the process of receiving, processing, and delivering orders—ranges from priority-based initialization of tasks (*e.g.*, based on due dates or order size) to allocation of resources and monitoring time-out conditions. By adding a "time dimension" (*i.e.*, D+ of the PRISM taxonomy) to the generated plan, the protocols are triggered for real-time control of the decisions. The TAP is composed of three sub-protocols that are the core of the proposed collaborative control mechanism for coherent and integrated administration and synchronization of distributed processes. The PBMP is triggered as the second stage of task administration, as a complimentary protocol for improving the performance of the TAP, in this case, in terms of total cost, resource utilization, service level, and stability of the CNS. This protocol is composed of three sub-protocols as follows (Figure 4.8):

1. *TRAP*. Each task (*i.e.*, order) has unique characteristics such as type, quantity, and due date, which define its priority compared to other tasks waiting in the queue to be processed by the same server (*i.e.*, supplier). These characteristics are identified by Task Agents (TAs) responsible for the ongoing tasks in the queue of a given supplier (*i.e.*, there are $|J|$ TAs in the system). The role of the TRAP is timely identification of task requirements, and prioritization of them for being served by a busy server.

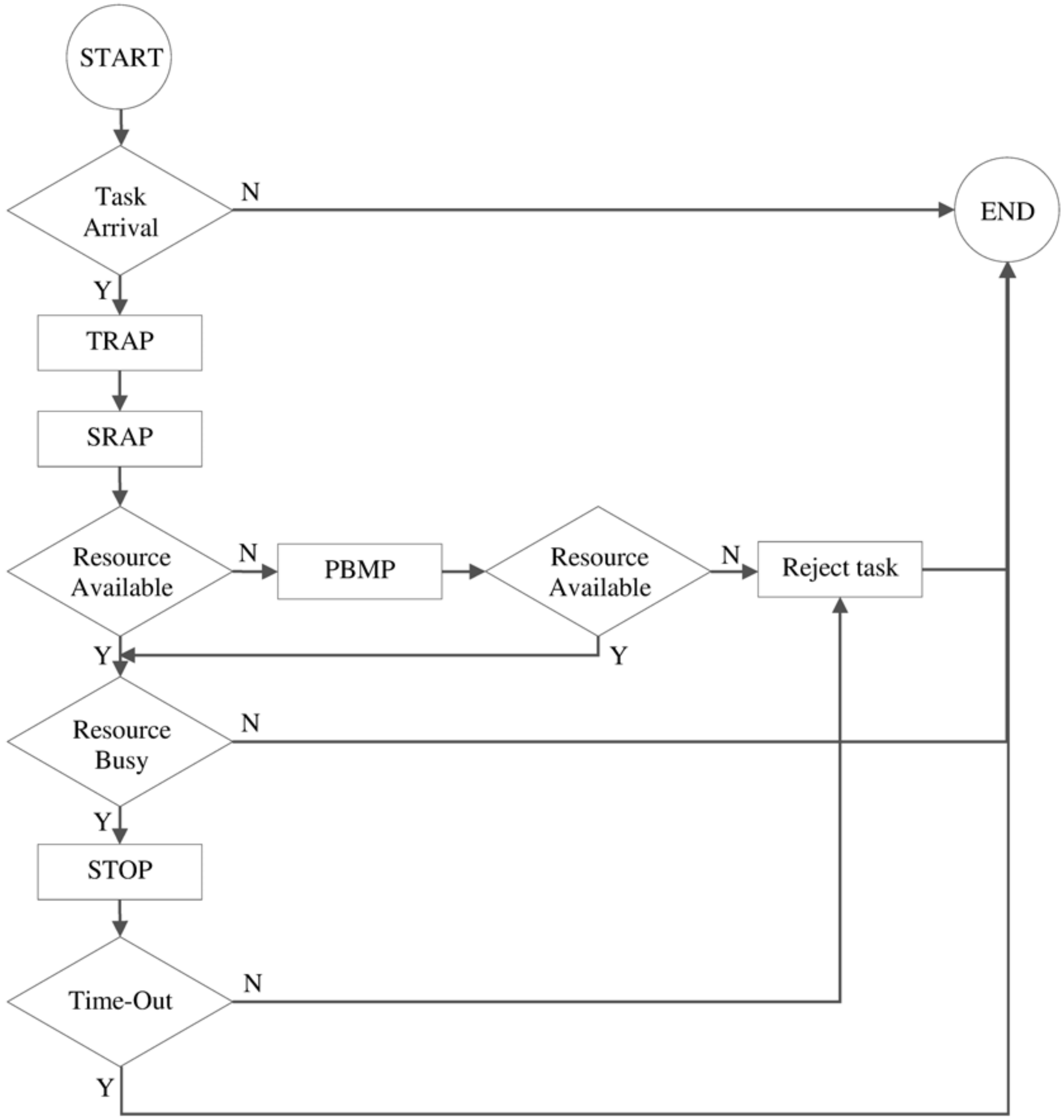


Figure 4.8. Workflow of TAP for distributed and dynamic matching.

2. *SRAP*. Distributed Resource Agents (RAs) define the best resources for processing each task, considering the updated priorities and available resources. Each RA corresponds to a specific supplier (*i.e.*, there are |*J*| RAs in the system). The primary resource allocation decisions are defined by the collaborative plan at the beginning of the period (*i.e.*, production level, supplier-customer matching, and DCS

decisions). However, in unforeseen situations, those decisions are updated by the RAs using PBMP, taking into account events that may occur over a predefined interval starting from the current timeslot.

3. *STOP*. In some situations, the task in process must be timed-out if certain conditions hold. Accordingly, the process is suspended by the RA, and the task is released or its load is relaxed. The STOP is triggered if at least one of the following conditions holds: Excessive occupation of resources by the task in process; Preemption by urgent tasks.

4.4.2 TRAP—Task Requirement Analysis

Each customer order is directed to the corresponding supplier according to the plan generated at the beginning of the period: Order o_i arrives at supplier *j*, if $\chi_{ij}^{*}=1$, where superscript * denotes optimality. The order is specified as $o_i=\{\tau_i, \alpha_i, \delta_i\}$, where τ_i, α_i, and δ_i denote its processing time, quantity, and due date, respectively. (The values of these parameters are assumed to be known.) Upon arrival of a new order, the TA updates the priority of all orders in the queue. If all the priority values are the same, or the priority of the order is less than the other orders, it is simply added to the end of the queue, following the FIFO (First-In, First-Out) discipline. Otherwise, the queue is reordered according to the updated relative priorities. Various disciplines exist for prioritizing a set of tasks in a queue. Without loss of generality, however, the priorities are calculated following the Earliest Due Date (EDD) and Shortest Processing Time (SPT) disciplines to protect service level, *i.e.*, lower the rate of backorders/stockouts and increase the throughput, respectively [1]. Following the EDD and SPT disciplines, the priority of order o_i is obtained through the following formula:

$$\pi\left(o_i,t\right)=w_1\underbrace{\left(1-\frac{\delta_i-t}{\sum_{o_{i'}(t)}\left(\delta_{i'}-t\right)}\right)}_{\text{EDD}}+w_2\underbrace{\left(1-\frac{\tau_i}{\sum_{o_{i'}(t)}\tau_{i'}}\right)}_{\text{SPT}}, \tag{4.8}$$

where t denotes the current timeslot, $o_{i'}\left(t\right)$ denotes the orders already assigned to the RA at timeslot t, including the new order o_i. w_1 and w_2 denote the weights of the EDD and SPT disciplines, respectively. After defining the priorities, each TA reports the tasks to the corresponding RA that is responsible for assigning resources to the tasks based on their relative priorities.

4.4.3 SRAP—Shared Resource Allocation

The RAs follow the production and DCS decisions defined by the plan. Hence, the production level at each supplier j over the entire period is equal to Q_j^*. In addition, if $\lambda_{jj'}^*=1$, it implies that the RAs corresponding to suppliers j and j' must share capacity/demand with the quantities of $T_{j'j}^*$ or $T_{jj'}^*$, where $T_{j'j}^*T_{jj'}^*=0$ (which can be proven through the *complementary slackness theorem*). Nevertheless, the plan may not be feasible in real-time due to the complexity and dynamic behavior of the system, which, in turn, may cause certain conflicts during the plan execution. The most substantial negative impact will be on the service level, *i.e.*, late/partial fulfillment or even rejection of orders due to capacity shortage, conflicts in DCS (*e.g.*, willingness to collaborate), and disruptions in

[1] See Ko and Nof (2012) for a more comprehensive priority evaluation function.

demand/capacity. Accordingly, supplier j is a demand sharing supplier at timeslot t, if at least one of the following conditions holds:

$$K_j(t) - \sum_{o_{i'}(t)} \alpha_{i'} < 0, \tag{4.9}$$

$$\delta_i - t - \sum_{o_{i'}(t)} \tau_{i'} < 0, \quad \{i' \mid \pi(o_{i'}, t) > \pi(o_i, t)\}. \tag{4.10}$$

In Eq. (4.9), $K_j(t)$ denotes the available capacity of supplier j at timeslot t, which implies that supplier j does not have enough capacity for fulfilling its orders. Eq. (4.10) implies that supplier j is under time limitation, *i.e.*, there is (at least) an order o_i that will be late if processed after other orders with higher priorities, and its priority cannot be increased. Two collaborative approaches are proposed for handling such situations:

1. *Decentralized DCS*. In the decentralized approach, the RAs are responsible for generating DCS proposals, and negotiating with their most preferred counterparts (Yoon and Nof, 2010). Accordingly, the RAs make DCS decisions according to their own local benefits and objectives—they compete in case there are conflicts of interest, which does not necessarily lead to global optimal solutions for the entire CNS. Nevertheless, in competitive environments, which may be cooperative but not collaborative, decentralized approaches provide promising solutions (see Seok and Nof, 2014). Note that this is a discrete-event-based procedure, *i.e.*, it starts automatically by any changes identified in the state of the system (*e.g.*, arrival or departure of tasks). The decentralized DCS procedure is composed of the following steps (Figure 4.9):

a. Each RA classifies itself as "demand sharing" or "capacity sharing". If there is no demand sharing/capacity sharing RA, the procedure stops. Otherwise, it proceeds to the next step.

b. The demand sharing RAs prepare demand-sharing proposals. The proposals include requests for partial or complete (or both) fulfillment of unsatisfied orders. Accordingly, the demand sharing RAs request *capacity sharing* for orders shared partially, and *direct delivery* from the capacity sharing suppliers in the case of complete sharing of an order (to minimize the shipment cost).

c. The demand sharing RA j assigns a priority value to each target capacity sharing supplier j' using the following formula:

$$\pi_{j'} = 1 - \frac{v_{jj'}}{\sum_{l \in CSS_j} v_{jl}}, \tag{4.11}$$

where CSS_j denotes the set of all target capacity sharing suppliers for demand sharing supplier j. The target suppliers are sorted according to their unit transshipment costs (*i.e.*, variable cost of collaboration). The goal is to minimize the collaboration costs.

d. The demand sharing RAs submit their demand sharing proposals to the capacity sharing RAs following the priority values obtained from Eq. (4.11). If no capacity sharing supplier is willing to collaborate, the demand sharing RA rejects the excessive orders or keeps them as backorders.

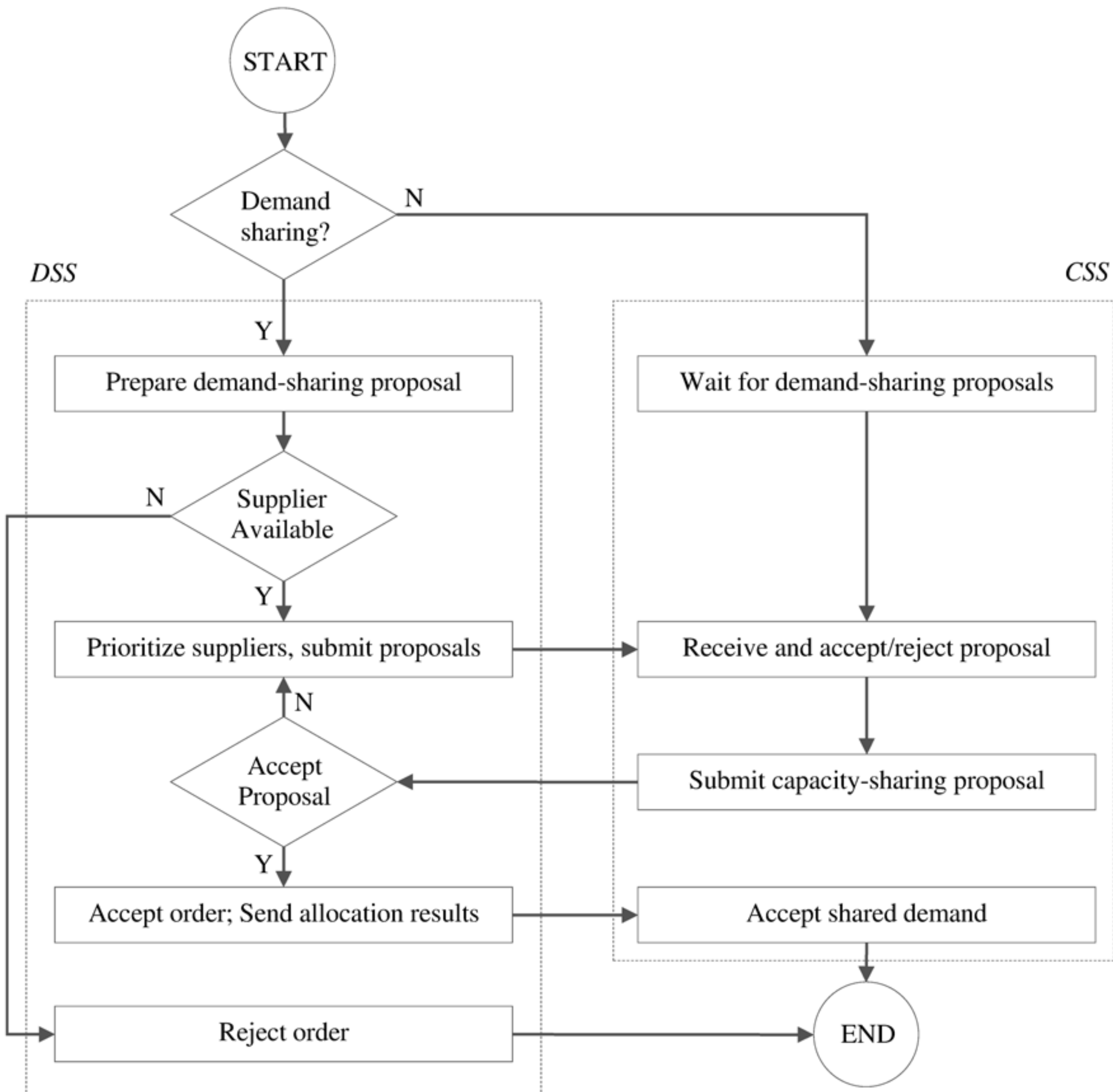


Figure 4.9. Workflow of the decentralized DCS procedure. (CSS: Capacity Sharing Supplier; DSS: Demand Sharing Supplier)

e. The capacity sharing RA j analyzes the received proposals (if any), and accepts order $o_{j'}$ received from DSS j' if

$$\alpha_{j'} < K_j(t) - \sum_{o_i(t)} \alpha_i, \tag{4.12}$$

where $o_i(t)$ denotes the tasks already in the queue of capacity sharing supplier j. If Eq. (4.12) does not hold, order $o_{j'}$ is *partially* accepted due to capacity shortage, and the quantity of the capacity-sharing proposal will be equal to the whole capacity available at the capacity sharing supplier *j*. The demand sharing proposals are processed by the capacity sharing RAs following FIFO discipline.

f. The capacity sharing RAs prepare capacity-sharing proposals and return to the corresponding demand sharing RAs.

g. If there is no pending demand-sharing proposal, the procedure stops. Otherwise, it returns to Step (d).

2. *Centralized DCS—PBMP*. In the decentralized procedure, the demand sharing RAs prepare proposals considering their own local benefits. On the other hand, the capacity sharing RAs evaluate the incoming proposals following FIFO discipline and accept/reject based on their own local benefit as well as their available capacities. This procedure does not guarantee the global optimality of DCS decisions for the entire CNS in terms of total cost and demand fulfillment rate. In addition, the existing approaches are not predictive and ignore possible changes in the state (*i.e.*, capacity) of the system in the near future. For instance, a capacity sharing supplier accepts a demand-sharing proposal at time t, and then encounters capacity shortage due to unanticipated increase in its own customer order at time $t+\Delta$. This may diminish the efficiency of collaboration decisions and cause conflicts during DCS. Hence, our proposal is a centralized predictive DCS mechanism organized by a Matching Agent (MA). The MA, integrated with all the distributed TAs and RAs, is a *mediator agent* (Barbati *et al.*, 2012) that controls the

DCS decisions based on the current states of all collaborating suppliers, taking into account potential future events. The proposed PBMP comprises the following steps (Figure 4.10):

a. Upon an updates in the system (*i.e.*, arrival or departure of a task), the MA is triggered and requests updates from the distributed RAs. (The current timeslot is set to zero, $t = 0$).

b. Demand sharing RA j (if any) evaluates and submits an order (*i.e.*, demand-sharing proposal) to the MA as $o_j = \{\tau_j, \alpha_j, \delta_j\}$.

c. The MA receives the orders, calculates the *prediction interval* using the following formula, and submits the result to the capacity sharing suppliers:

$$\Delta = \max_{j \in DSS} \{\tau_j\}. \tag{4.13}$$

The prediction interval determines the length of the future horizon considered by the MA for making current DCS decisions through PBMP, and encompasses the entire time interval that all the existing demand-sharing proposals will remain in the system.

d. Capacity sharing RA j calculates its expected capacity level considering future events during Δ. The future events involve planned orders from the customers matched to j. Accordingly, capacity sharing RA j submits the following expected capacity level:

$$\hat{K}_{j,\Delta} = K_j - \sum_{\{i | M_{ij}^{1^*} = 1\}} \mathrm{Pr}_{i,\Delta}\, \bar{\alpha}_i, \tag{4.14}$$

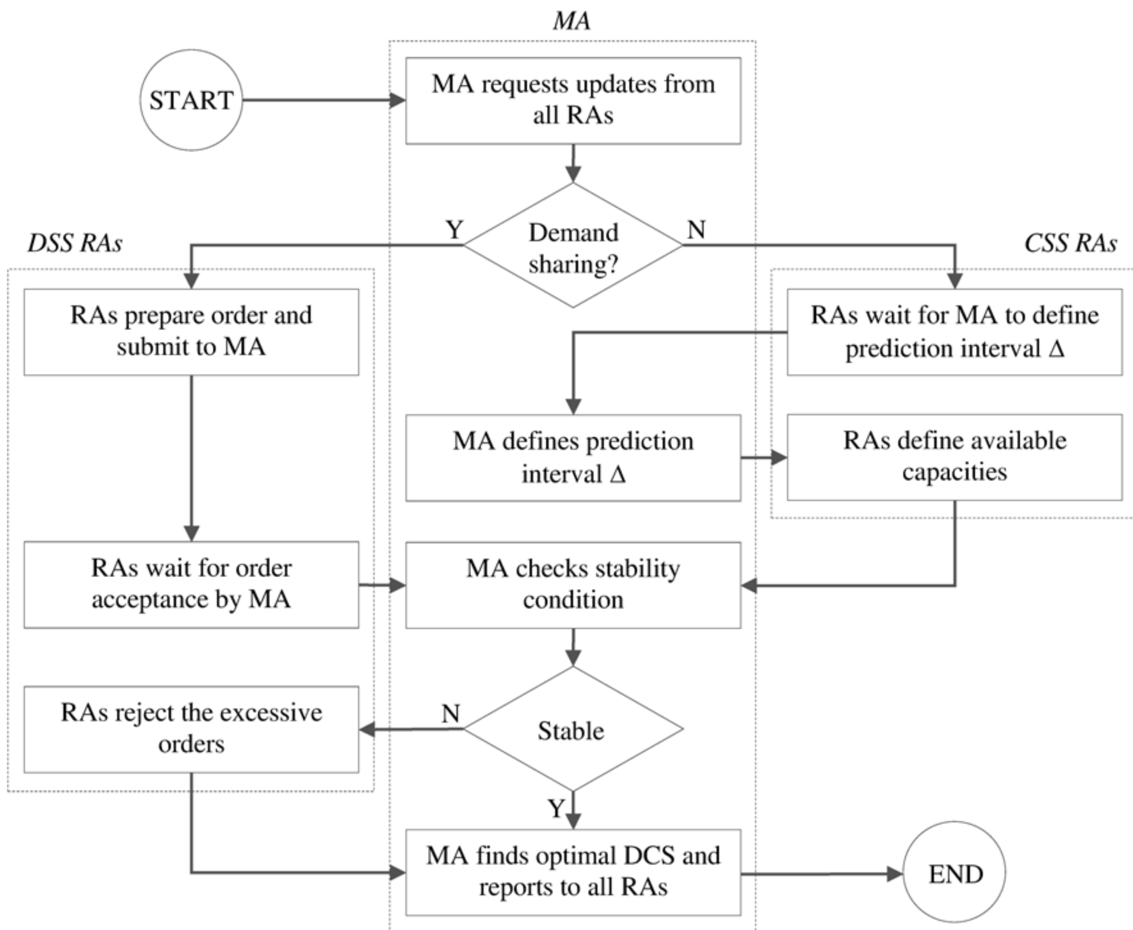


Figure 4.10. Workflow of the centralized DCS procedure—PBMP. (CSS: Capacity Sharing Supplier; DSS: Demand Sharing Supplier)

where $\mathrm{Pr}_{i,\Delta}$ denotes the probability of receiving an order from customer i during Δ, and $\bar{\alpha}_i$ denotes the expected quantity of orders received from customer i. Note that order probabilities and quantities must be estimated by the corresponding RAs, according to the history of orders received from each customer. It is assumed that the required estimates are known and available.

e. The MA evaluates the *stability* of the DCS process by checking the following inequality:

$$\sum_{j \in DSS} \alpha_j \leq \sum_{j' \in CSS} \hat{K}_{j',\Delta}. \tag{4.15}$$

If the stability Eq. (14) holds, the PBMP proceeds to the next step. Otherwise, the MA rejects one or more demand-sharing proposals through the following procedure:

i. Define the capacity-demand gap G

$$G = \sum_{j \in DSS} \alpha_j - \sum_{j' \in CSS} \hat{K}_{j',\Delta}. \tag{4.16}$$

ii. Find the order (i.e., demand-sharing proposal) o_j that minimizes $\left|\alpha_j - G\right|$.

iii. Eliminate o_j from the demand-sharing proposals and update G in (15). If $G < 0$, go to Step (f). Otherwise, return to Step (ii).

This procedure improves the *service level* by minimizing the number of demand-sharing proposals rejected due to capacity shortage, and the *resource utilization* through minimizing the gap between the available capacity and updated demand-sharing proposals.

f. The MA defines the optimal match between orders placed by demand sharing suppliers and capacities shared by capacity sharing suppliers through solving the following MIP:

$$
\begin{aligned}
\min \quad & \sum_{j \in DSS} \sum_{j' \in CSS} \left(f_{jj'} \lambda_{jj'} + v_{jj'} T_{jj'} \right), \\
\text{s.t.} \quad & \sum_{j \in DSS} T_{j'j} \le \hat{K}_{j',\Delta}, \quad \forall\, j' \in CSS, \\
& \sum_{j' \in CSS} T_{j'j} = \alpha_j, \quad \forall\, j \in DSS, \\
& T_{j'j} \le \gamma\, \lambda_{j'j}, \quad \forall\, j \in DSS, j' \in CSS, \\
& T_{j'j} \ge 0,\ \lambda_{j'j} \in \{0,1\}, \quad \forall\, j \in DSS,\ j' \in CSS.
\end{aligned} \tag{M4.1}
$$

The objective function of Model (M4.1) minimizes the fixed and variable costs of collaboration. The first set of constraints prevents capacity limit violations in each capacity sharing suppliers. The second set of constraints guarantees fulfillment of each demand-sharing proposal. The third set of constraints implies that suppliers collaborate only if their DCS proposals are matched. The last set of constraints ensures the feasibility of decision variables.

4.4.4 STOP—Synchronization and Time-Out

STOP is a background protocol activated during the process of each task by the RAs to monitor the process in real-time. A task is timed out by STOP if at least one of the following conditions holds:

1. *Excessive resource occupation.* Refers to situations where the *actual* processing time $\hat{\tau}_i$ of order o_i is much higher than their expected values, which, in turn, may delay the rest of the tasks waiting in the queue. A predefined threshold is defined for checking time-out conditions as follows:

$$\hat{\tau}_i - \tau_i > \varepsilon, \tag{4.17}$$

where ε denotes the threshold. However, there are two instances where the task is not timed out, even if Eq. (4.17) holds: (1) There is no other task waiting in the queue; (2) The in-process task will be delayed if timed-out.

2. *Preemption by urgent tasks*. Refers to situations where a task may be late if it is not processed before the current process is completed. Although in the TRAP prioritizes the tasks based on the EDD policy, this situation is likely to occur due to dynamic changes in due dates and/or processing times. The preemption procedure is performed by the RAs as follows:
   a. The RA checks the following condition for all orders in the queue:

$$\delta_i - t - \tau_i < 0. \tag{4.18}$$

   b. If Condition (4.18) holds for order o_i in the queue and for order $o_{i'}$ in process (*i.e.*, $\delta_i > \tau_{i'}^r(t)$, where $\tau_{i'}^r(t)$ denotes the remaining processing time of $o_{i'}$ at timeslot *t*), $o_{i'}$ is preempted by o_i and will be resumed after completion of o_i.

4.4.5 Numerical Experiments

Four scenarios are considered in order to investigate different aspects of the developed RTO mechanism compared to similar existing approaches:

1. *No collaboration with fixed pre-matching* (S_1). Each supplier is responsible for the demand of its own fixed set of customers where no DCS takes place among

suppliers. The purpose of S_1 is to underline the relative impact of collaborative DCS along on the performance of individual suppliers and the entire CNS.

2. *Collaboration with fixed pre-matching* (S_2). Suppliers collaborate through DCS, but (a) the set of customers being served by each supplier, and (b) the demand/capacity sharing suppliers are fixed. The purpose of S_2 is to highlight the role of TAP in RTO of the DCS decisions.
3. *TAP without PBMP* (S_3). The real-time control mechanism is activated at the beginning of each period after plan generation, and follows all sub-protocols of TAP to optimize the process in real-time. The SRAP, however, follows the decentralized DCS procedure. The purpose of S_3 is to investigate the role of MA and PBMP in optimization of DCS decisions, especially in terms of total cost.
4. *TAP with PBMP* (S_4). The SRAP is performed through the centralized process performed by the MA. This scenario focuses on minimizing total collaboration cost in real-time, along with resource utilization, demand fulfillment rate, and stability, compared to S_1, S_2, and S_3.

All scenarios are simulated based on the generated plans presented in Section 4.3.2, and the results corresponding to each period are analyzed and compared. The statistical significance of all the resulting observations is then analyzed. The evaluation criteria and the findings are as follows:

1. *Order fulfillment*. The variations in capacity-demand gaps of each supplier may result in shortage in some suppliers, and thus late deliveries or rejected orders. Lateral collaboration is a promising solution for minimizing these gaps, through

enabling demand sharing suppliers to make use of excess capacities available at capacity sharing suppliers. Our experiments indicate that collaborative scenarios (*i.e.*, S_2 to S_4) outperform the non-collaborative scenario S_1. Real-time execution of the collaboration plans, however, may encounter some deficiencies due to unforeseen variations in capacities and/or demands at different points of time. The developed RTO aims to minimize such gaps through dynamic requirement planning, allocation, and monitoring of plan execution, based on the TAP. The results shown in Figure 4.11 indicate improvements in fulfillment of customer demands by the TAP-enabled scenarios (*i.e.*, S_3 and S_4) compared to non-TAP scenarios. Moreover, the results show superiority of S_4 to S_3 in terms of demand fulfillment rate, which is due to the predictive control mechanism of the PBMP during DCS process.

2. *Resource utilization*. The non-collaborative scenarios are inferior compared to the collaborative scenarios in terms of resource utilization, which is due to their inflexibility in reducing the capacity-demand gaps. Collaboration enables the capacity sharing suppliers to improve utilization of their resources through sharing them with the demand sharing suppliers. The collaboration plans may need modifications in real-time. Results of simulation show that the TAP-enabled scenarios outperform the other scenarios in terms of resource utilization (Figure 4.11).

3. *Stability*. This criterion is evaluated for each individual supplier as the ratio of late/rejected orders to the entire orders received over each period. Collaboration significantly improves the stability of demand sharing suppliers by reducing their

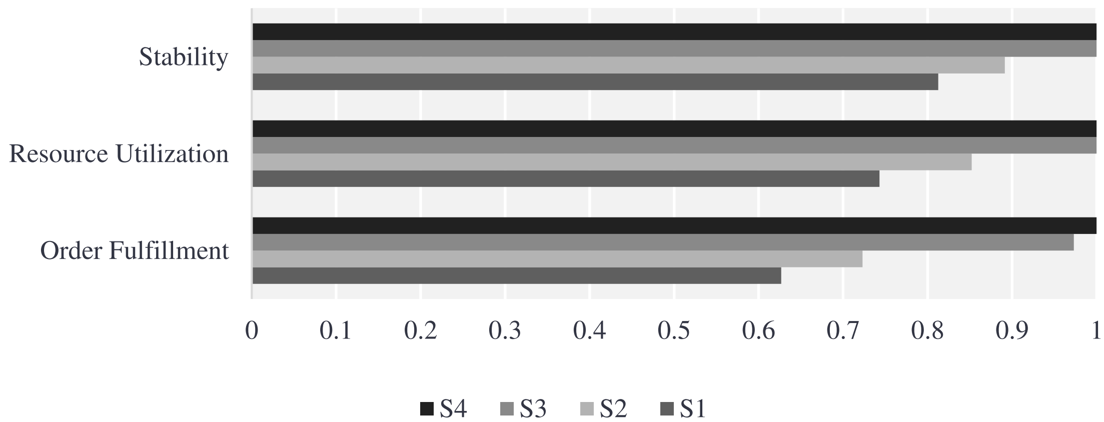


Figure 4.11. Comparative analysis of scenarios based on order fulfillment, resource utilization, and stability. (Results: Mean values of 30 independent simulations; Relatively normalized in [0, 1])

capacity-demand gap (see Proposition 2). The proposed real-time control mechanism improves stability of suppliers through timely detection of shortages and optimal allocation of available resources. The simulation results shown in Figure 4.11 indicate superiority of TAP-enabled scenarios to the non-TAP scenarios in terms of stability.

4. *Cost effectiveness*. The effect of collaboration on the total cost (Figure 4.12)—according to the experiments—is twofold. Although lateral collaboration minimizes the *undesirable* costs of inventory holding and backordering, it imposing extra capacity-demand sharing costs. In some situations, the unit cost of collaboration may be much higher than the holding and backordering costs, and therefore, suppliers may not be willing to collaborate or even may leave the CNS in case this is a long-term situation (see CHAPTER 6). Nevertheless, even in such cases, the indirect cost of inventory holding or backordering, and their impact the long-term reputation and profit of suppliers should not be underestimated. The

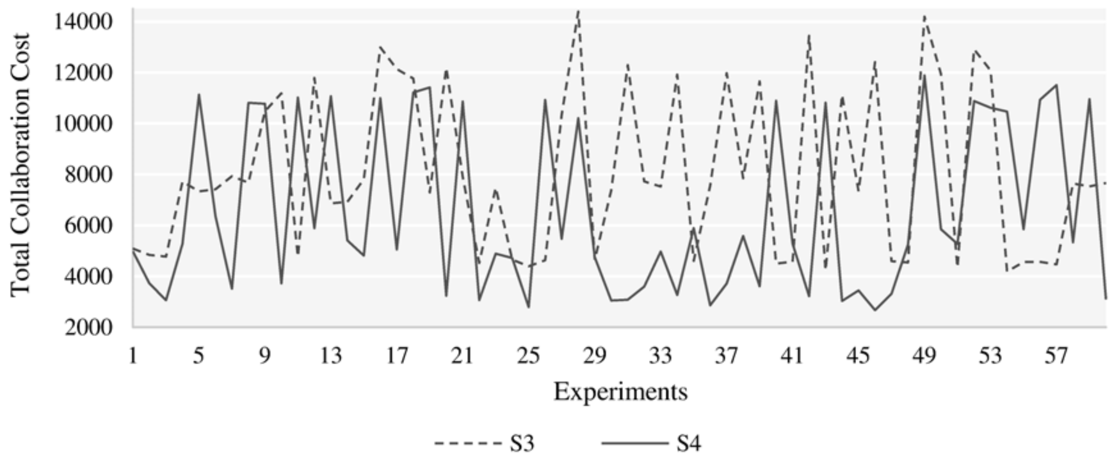


Figure 4.12. Total collaboration cost under the decentralized DCS procedure (S_3) and PBMP (S_4) in 60 independent experiments.

developed PBMP (*i.e.*, S_4) provides more effective mechanism for optimization of collaboration costs in real-time compared to the decentralized DCS approach (*i.e.*, S_3). Statistical analyses also indicate a significant difference (confidence level of 95%) between the simulation results shown in Figure 4.12, which indicates superiority of S_4 over S_3 in terms of cost effectiveness (see Moghaddam and Nof, 2016[a]).

4.5 Concluding Remarks

Collaboration is proven as a powerful strategy for mitigating the undesirable *mismatches* between demand and capacity levels and their unforeseen variations over time. Compared to traditional non-collaborative approaches, the supply process can be performed more efficiently through dynamic DCS between the suppliers. DCS significantly improves demand fulfillment rate, as a representative of service level (Kutanoglu and Mahajan, 2009; Olsson, 2009; Yoon and Nof, 2010 and 2011; Paterson, 2011 and 2012; Axsäter *et al.*, 2013), and resource utilization, as a criterion highly correlated with the stock (Kutanoglu and Mahajan, 2009; Torabi and Moghaddam, 2012) and service (Olsson, 2009; Axsäter *et al.*, 2013) costs, safety stock level and stockout probability/costs (Paterson, 2011 and 2012). It is also shown that DCS guarantees stability of each individual supplier, given that the entire CNS is stable. Even in case of overall instability in the network (*i.e.*, overall demand higher than the total available capacity), DCS alleviates the instability effect on different suppliers to the minimum possible extent. Accordingly, in line with the previous studies (*e.g.*, Burton and Banerjee, 2005; Yoon and Nof, 2010; Tiacci and Saetta, 2011), our observations indicate substantial improvements made by lateral collaboration as opposed to the conventional non-collaborative strategies.

Lateral collaboration strategies, in general, impose extra costs associated with the fixed costs of negotiation and transactions along with variable costs of transshipment of stocks (Burton and Banjeree, 2005). The rate of lateral collaboration (and the corresponding costs) without best matching is directly correlated with the gap between the capacity of each single supplier and the aggregate customer demand allocated to it. Several policies and strategies have been proposed and discussed in literature, aiming at minimizing the lateral collaboration costs. Some studies attempted to deal with the lateral collaboration cost through considering direct shipments from the capacity sharing suppliers to the customers of the demand sharing suppliers. Nevertheless, in all those cases, the fixed cost of collaboration is not negligible.

A *dynamic best matching* framework is developed and examined as an approach towards minimizing the cost of lateral collaboration through making supplier-customer best matching decisions prior to lateral collaboration. It is shown through this case study that dynamic best matching between the suppliers and the customers is an appealing strategy in reducing these extra costs, and even improving the service level through minimizing the rate of unfulfilled demand compared to the collaborative scenarios with fixed pre-matchings.

Some experts and practitioners, however, may prefer fixed matching between suppliers and customers as an advantage because of less complexity and easier implementation. Nevertheless, from the customers' viewpoint, timely efficient delivery and higher demand fulfillment rate seem to be more important factors compared to the source of supply. Moreover, from the viewpoint of the CNS, lower total costs, while maintaining the inherent benefits of lateral collaboration is superior as well. Besides collaborating suppliers, dynamic sharing of customers could be a beneficial strategy for competing suppliers, depending on the level of competition, transshipment cost, and differentiations between the suppliers (Zhao and Atkins, 2009). However, in case the capacity level of the suppliers and the demand level of the customer are less prone to abrupt fluctuations in long term, the supplier-customer matchings are also expected to have less variation. Thus, running the model in a long run could result in specific *clusters* of potential suppliers for each customer. Accordingly, collaboration within the clusters can combine the benefits of flexibility in matching under varying conditions, while maintaining some level of mutual loyalty between suppliers and customers in each cluster (see CHAPTER 6).

This case study also pinpoints the impact of real-time collaborative control on the quality and efficiency of lateral collaboration processes. The presented experimental results and analyses indicate the impact of RTO mechanisms on further improvement of DCS decisions under dynamic and unforeseen changes in the behavior of the system. The idea is to reduce the length of planning period, monitor and revise the plans in real-time with the aid of agent-based systems, effective TAP, and the respective workflows. The new collaborative control mechanism provides the basics for further automation of order fulfillment processes in CNS. Dealing with different various interrelated aspects of task administration, the TAP consolidates the initialization of ordering processes, allocation of resources, and process monitoring through logical workflows of their respective sub-protocol (*i.e.*, TRAP; SRAP; STOP). The TAP is known as an effective mechanism for improving enterprise collaboration decisions. The PBMP—the core of the new RTO mechanism—enhances the TAP using an agent-based system capable of optimizing the supplier-customer and supplier- supplier matching decisions in real-time. The primary focus of the centralized approach, enabled by the PBMP, is to minimize the collaboration costs by transforming blind/random sharing procedures to more intelligent and optimized matching mechanisms supported by multi-agent frameworks and collaborative control protocols.

CHAPTER 5. CASE 2—COLLABORATIVE ASSEMBLY LINES☆

This case investigates an instance of best matching with precedence relations and resource-sharing. A common example of this extension of matching problems is Assembly Line Balancing (ALB), where a set of tasks with certain precedence relations must be matched to a set of workstations. This example is selected as the second case, because the insights provided by this case are complementary to those provided by the previous case. In Case 1, matching (of suppliers-customers) was applied as a mechanism to improve the already existing sharing (of demand-capacity) decisions. In Case 2, in contrast, the classic matching (of tasks-workstations) decisions are improved by incorporating a sharing (of "tools") mechanism in the decisions. In this case, thus, a Collaborative Assembly Framework (CAF), inspired from the design principles of CCT, the Collaborative Control Theory, is developed to enhance the *balanceability* of assembly lines (Figure 5.1). The notion of the CAF lies in dynamic utilization of idle resources to eliminate bottlenecks. The CAF is composed of (1) tool sharing protocol for making dynamic *tool-sharing* decisions among fully loaded (*i.e.*, bottleneck) and partially loaded workstations, and (2) best matching

---

☆ The preliminary version of this case study was presented at the *11th IFAC Workshop on Intelligent Manufacturing Systems*, Brazil, 2013. The materials presented in this case study are adapted from two works of the author published in the *IIE Transactions* (DOI:10.1080/0740817X.2015.1027456) and *Mechatronics* (DOI:10.1016/j.mechatronics.2014.10.001) journals.

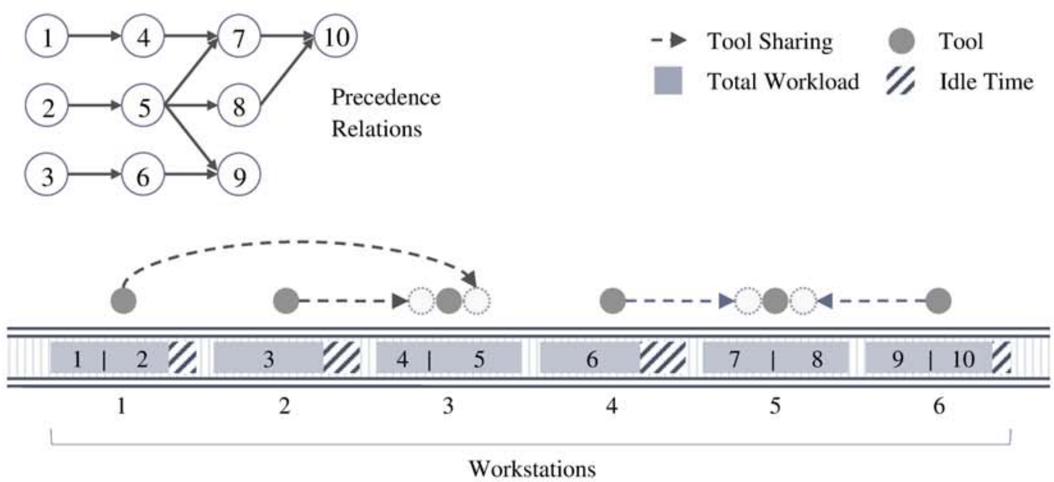


Figure 5.1. Balancing assembly lines with collaborative workstations (after Moghaddam and Nof, 2015[b]).

protocol, which dynamically matches (a) tasks and workstations, and (b) collaborating workstations. A Multi-Objective Mixed-Integer Programming (MOMIP) formulation is developed for mathematical representation and optimization of the problem.

The CAF is proven to guarantee *relative balanceability* of assembly lines, depending on the pairwise tool compatibility and tool sharing performance. A Collaborative Multi-Agent System (CMAS) enhanced with a sharing-matching protocol is also developed for real-time execution of the plans generated by the optimization model, control of processes, and modification of the tool sharing decisions, considering dynamic changes in the system's operations. Experiments show that the CAF framework significantly outperforms classic approaches in terms of cycle time, utilization of tools, and balanceability. In addition, the control mechanism is proven to augment the line flexibility against the inherent uncertainties of assembly processes compared to *static* CAF frameworks (*i.e.*, involvement of D+).

5.1 Motivation

ALB (Assembly Line Balancing) is a classic resource-constrained matching problem that involves assignment of tasks of different duration and precedence relations to a sequence of interconnected workstations such that their workload is balanced with respect to the required production throughput (Nof *et al.*, 1997). Every assembly process consists of a sequence of tasks that usually cannot be *subdivided* and must be processed at a specific workstation (Rekiek and Delchambre, 2005). ALB problems are typically classified into two types:

1. *Type*-I. Minimize the number of workstations for a given cycle time.
2. *Type*-II. Minimize the cycle time (or maximize the production throughput) for a fixed number of workstations.

Further objectives have also been investigated in literature; *e.g.*, maximizing line/operator efficiency (Song *et al.*, 2006), and minimizing the costs of workforce (Sprechter, 1999; Sarin *et al.*, 1999; Gamberini *et al.*, 2006) and task duplication (Bukchin and Rabinowitch, 2006) for design/reconfiguration of various assembly lines with specific product models, line layout/configuration, line control mechanisms, automation level, and industrial applications (see Ghosh and Gagnon, 1989; Becker and Scholl, 2006; Boysen *et al.*, 2007 and 2008). All the aforementioned objectives, however, are usually limited by the basic characteristic of ALB problems—*task indivisibility*. This characteristic causes unbalanced workload between workstations and lower flexibility in terms of production throughput. Bottleneck is an inevitable phenomenon in almost every assembly line, which restricts the production throughput and diminishes the line *balanceability*.

In practice, fully balanceable assembly lines are difficult (or, in most cases, impossible) to achieve (Anuar and Bukchin, 2006), due to the indivisibility of task along with processing time variations and precedence relation constraints. Accordingly, workstations are typically classified as (A) *fully loaded*, with workloads *equal to* the cycle time, or (B) *partially loaded*, with workloads *lower than* the cycle time. Set A is essentially non-empty, *i.e.*, there is always at least one fully loaded workstation (*i.e.*, the bottleneck) specifying the cycle time. Set B, however, may be empty but with a low likelihood and in case the line is fully balanceable (*i.e.*, no possibility for further improvement).

In traditional assembly lines, workstations are isolated and operate independently. In such settings, increasing the line throughput requires improving the performance of the workstation(s) in Set A, *i.e.*, the bottleneck(s). Nevertheless, this is not an economically justifiable strategy since it requires additional investments (*e.g.*, equipment duplication), while a part of the already existing equipment is idle in the workstation(s) in Set B. The notion of *work sharing* was suggested over the last decade to address this drawback (Askin and Chen, 2006; Anuar and Bukchin, 2006; Guo *et al.*, 2008; Bukchin and Sofer, 2011), where tasks are allowed to be processed in multiple workstations in each cycle. The idea is based on substituting moving workers with moving tasks to avoid the movements of workers between the workstations and thus, improve the line productivity (Bukchin and Sofer, 2011). The work sharing idea, however, disregards the primary assumption of ALB problems on the indivisibility of tasks (Rekiek and Delchambre, 2005). This assumption can be justified from economic, technical, and technological perspectives; *e.g.*, equipment duplication cost (Bukchin and Sofer, 2011), setup and configuration restrictions, space limitation. Moreover, some tasks may not be dividable or may require considerable setup times (*i.e.*, setup time duplication). Furthermore, dynamic changes in the system's configuration may enforce reconfiguration of the workstations or even rebalancing of the line over time. Therefore, the redundant equipment may be useless in some cycles/periods.

5.2 Outline

The CAF is based on the utilization of *available* idle resources rather than investing in additional resources for improving the line balanceability. The CAF is composed of two major components—tool sharing, best matching. Tool sharing has been partly discussed by a few studies in the ALB literature, *e.g.*, part-sharing (Chan *et al.*, 2009) and operator-sharing (Rabbani *et al.*, 2012) among specific production facilities. A practical example of tool-sharing in assembly and test utilities is sharing of computer integrated testers and inspection tool, especially when their actual use is only during a small fraction of the assembly process at each assembly station (Esfarjani and Nof, 1998). In the context of the CAF, a "tool" refers to any type of *resources* (*e.g.*, equipment; robot; operator) that are capable of processing the tasks and can be shared between the workstations.

The matching decisions involve finding the best correspondence between the sets of tasks and workstations, taking into account tool sharing alternatives, processing times, precedence relations, and cycle time upper bound. Tool sharing decisions must be made after identifying any idle workstation, and involve preparing a tool sharing proposal with distinct characteristics including available tools to be shared and the sharing period. Based on the tool sharing proposals received from Set B (*i.e.*, partially loaded) workstations, the best matching module matches the proposal to the best workstation in Set A, considering tool-task compatibility and pairwise linear distance between the corresponding workstations. An MOMIP formulation is developed for mathematical representation and optimization purposes. The objective is to improve balanceability of the line with the minimum possible cost through minimizing the number of workstations (*Type*-I ALB), minimizing the cycle time (*Type*-II ALB), and minimizing the total cost of collaboration (*i.e.*, tool sharing).

It is proven that the balanceability of assembly lines depends highly on: (1) The *compatibility* of the shared tools with the ongoing tasks, and (2) The tool sharing performance. Hence, the CAF guarantees balanceability of assembly line in case all tools are compatible with all tasks and the pairwise distances between workstations are negligible. It is also proven that the CAF, in general, leads to relative balanceability, where the deviation of the optimal cycle time from the ideal cycle time (*i.e.*, cycle time of fully balanceable line) is a function of tool compatibility and pairwise distances between workstations. Accordingly, the main prerequisite of the CAF to improve the balanceability of the line is to augment the capability of tools in processing the tasks related to other workstations; *e.g.*, cross-trained operators (Bartholdi and Eisenstein, 1996; Hopp *et al.*, 2004); multi-purpose facilities/robots. A set of numerical experiments is conducted to illustrate the advantages of the developed CAF over the classic non-collaborative approaches.

A CMAS (Collaborative Multi-Agent System) is also developed to enhance the automation of the CAF. Intelligent and autonomous agents, distributed among the assembly system, are considered to operate in accordance with a tool sharing-best matching decisions and protocol. The real-time, collaborative control mechanism is adapted and extended to provide feedback to the off-line plan generated and updated continuously using the MOMIP model. Several experiments and analyses are conducted in to illustrate the applicability of the CAF in significantly improving the performance of assembly lines in terms of flexibility (*i.e.*, cycle time reduction), utilization of tools, and balanceability *in real-time* (*i.e.*, including D+ of the PRISM taxonomy). Figure 5.2 shows the status of the problem under study according to the PRISM taxonomy of best matching.

5.3 Optimization: MOMIP and Goal Programming

Consider an assembly process with dynamic demand of D_t over a $|T|$-period horizon. The assembly process is decomposed into $|J|$ tasks with certain processing times ρ_{jt} and precedence relations. Due to technological and economic limitations, an upper bound is defined for the number of workstations. An upper bound is also defined for the cycle time with respect to the demand rate D_t and the available production time A_t as follows:

$$\bar{C}_t = \frac{A_t}{D_t}, \qquad \forall t \in T. \tag{5.1}$$

The theoretical lower bound for the number of workstations in each period is then defined according to the overall workload (*i.e.*, sum of processing times) and the cycle time upper bound as follows:

$$\underline{W}_t = \left\lceil \frac{\sum_{j \in J} \rho_{jt}}{\bar{C}_t} \right\rceil, \qquad \forall t \in T. \tag{5.2}$$

Give these definitions, the general characteristics and assumptions of the CAF are as follows: (1) The processing times are independent from the matched workstation and

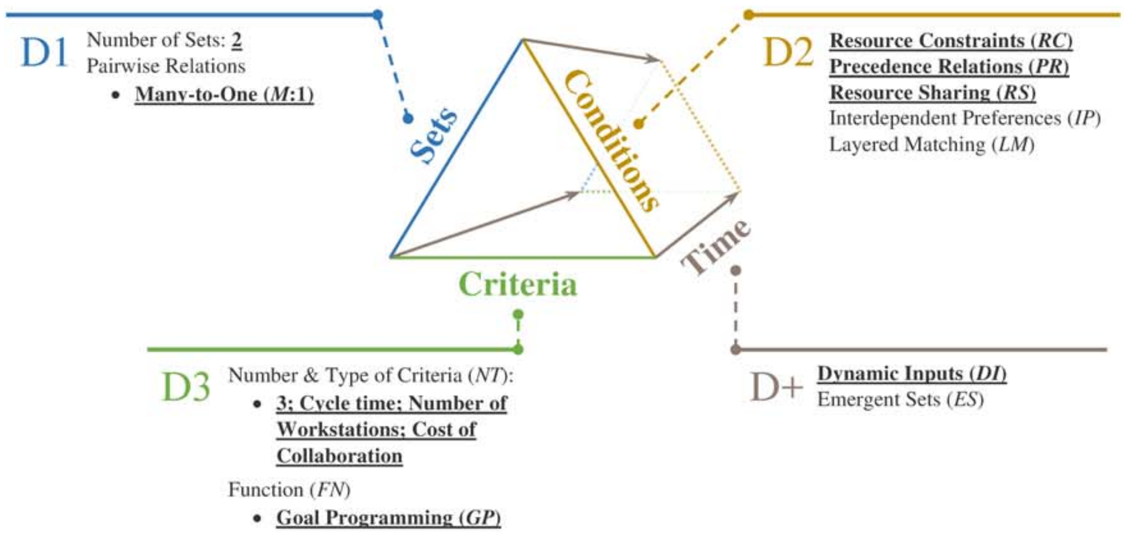


Figure 5.2. Classification of the collaborative best matching problem in assembly systems according to the PRISM taxonomy of best matching.

are fixed in all periods. (2) Each task must be processed in one workstation and cannot be subdivided between multiple workstations. (3) Idle workstations are allowed to share their tools with other workstations during their idle times. (4) No work-in-process inventory is allowed. (5) Demand rate, processing time, and precedence relations of tasks, available production time, upper bound for the number of workstations, and pairwise tool-task compatibility ratios are known. Accordingly, the CAF finds the best tool sharing-best matching decisions based on the following logic (Figure 5.3):

1. The input data is collected for demand forecasts, processing times, precedence relations, cycle time upper bound, upper bound and lower bounds for the number of workstations, available production time for all periods, and pairwise Collaboration Efficiency (CE) between the workstations. CE $e_{ii'} \in [0,1]$ represents the capability of the tools of workstation i in processing the tasks of workstation i'

through tool sharing, as well as the pairwise linear distance between the workstations.

2. Best matching protocol optimally matches tasks and workstations. The matching criteria are processing times and precedence relations, cycle time upper bound, and the lower bound on the number of workstations. This is a *many-to-one* matching problem, implying that each task must be matched to only one workstation, but each workstation may process more than one task.
3. Tool sharing protocol classifies the workstations into Sets A and B, identifies the tool sharing offers, and calculates the potential tool sharing benefits according to the pairwise CEs.
4. Best matching protocol matches the tool sharing proposals prepared by the workstations in Sets A and B. The matching criterion is the pairwise CEs. This is a *many-to-many* matching problem, which implies that each workstation of Set A can be assisted by more than one workstation of Set B, and each workstation of Set B is allowed to share its tools with more than one workstation of Set A.

5.3.1 Mathematical Formulation

The primary objective of the CAF is to improve the balanceability of assembly lines. This can be realized through minimizing (1) the number of workstations (the first objective, Z_1), (2) the cycle time (the second objective, Z_2), and (3) the total cost of collaboration (the third objective, Z_3). The functions associated with these objectives are formulated as follows[1]:

1. *Number of workstations* (*Objective 1*). Minimizes Z_1 considering the costs of opening and facilitating workstations;

$$\min\ Z_1 = \sum_{i \in I} \sum_{t \in T} \sum_{j \in J} i\chi_{jit}. \tag{5.3}$$

According to Eq. (5.3), the optimal number of workstations in each period can be obtained using the following formula:

$$W_t^* = \max_{i \in I,\ j \in J} \left\{ i\chi_{jit}^* \right\}, \qquad \forall\ t \in T, \tag{5.4}$$

where * denotes optimality. The optimal number of workstations must lie between the theoretical lower bound and upper bound.

2. *Cycle time* (*Objective 2*). Minimizes Z_2 via tool sharing, *i.e.*, smoothing the overall workload between partially loaded and bottleneck workstations;

$$\min\ Z_2 = \frac{\sum_{t \in T} C_t}{T}. \tag{5.5}$$

3. *Collaboration costs* (*Objective 3*). Minimizes Z_3 by optimal matching of tasks and workstations, such that the rate of tool sharing between the workstations is minimized. Considering $f_{ii'}$ as the fixed cost of collaboration between workstations i and i', the total collaboration cost over the $|T|$-period horizon is calculated as follows:

[1] This model is an extension of Model (M3.2), *M*:1 / *RC*, *PR* / +, *OS*, and Model (M3.3), *M*:1 / *RC*, *RS* / +, *OS*, presented in Chapter 3.

$$\min\ Z_3 = \sum_{i\in I}\sum_{i'\in I}\sum_{t\in T} f_{ii'}\psi_{ii't}. \tag{5.6}$$

4. *Resource constraints*. This set of constrains ensures that the workload (*i.e.*, the overall processing times of the tasks matched to each workstation) does not exceed the cycle time upper bound.

$$\sum_{j\in J}\rho_{jt}\chi_{jit} \le \bar{C}_t, \qquad \forall\ i\in I,\, t\in T. \tag{5.7}$$

5. *Precedence relations constraints*. This set of constraints guarantees that the pairwise precedence relations of the tasks are not violated, *i.e.*, each task is allowed to be matched to the same workstation as its immediate predecessor(s) is (are) matched, or to the succeeding workstations.

$$\sum_{i\in I} i\chi_{j'it} \le \sum_{i\in I} i\chi_{jit}, \qquad \forall\ j\in J,\ j'\in IP_j,\, t\in T. \tag{5.8}$$

6. *Best matching constraints.*
   - *Tasks-workstations*. This set of constraints ensures that each task is matched to exactly one workstation in each period. The only limitation on the number of the tasks matched to each workstation is the cycle time upper bound, according to (5.7).

$$\sum_{i\in I}\chi_{jit} = 1, \qquad \forall\ j\in J,\, t\in T. \tag{5.9}$$

   - *Collaborating workstations*. This set of constraints ensures that tool sharing is performed from workstation i (Set B) to workstation i' (Set A), if the corresponding tool sharing proposal has been approved with respect to the pairwise collaboration costs and pairwise CEs.

$$S_{ii't} \leq M\,\psi_{ii't}, \qquad \forall\, i, i' \in I,\ t \in T. \tag{5.10}$$

7. *Tool sharing constraints*. In these constraints, tool sharing variables are defined in order for the workstations in (a) Set A to make use of the tools shared by other workstations, and (b) Set B to share their tools during their idle times with other workstations, such that the cycle time is minimized. Tool sharing variables represent "virtual extra times" shared by the workstations in Set B with the workstations in Set A, which in turn augment their performances by reducing their workload. The efficiency of tool sharing depends on the pairwise compatibility and distance between the workstations, *i.e.*, CE, where $e_{ii'} = 1$ implies 100% compatibility and negligible distance, while $e_{ii'} = 0$ implies no compatibility and/or extremely long pairwise distance between workstations i' and i.

$$\sum_{j \in J} \rho_{jt} \chi_{jit} + \sum_{i' \in I} \left(S_{ii't} - e_{i'i} S_{i'it} \right) \leq C_t, \qquad \forall\ i \in I,\ t \in T. \tag{5.11}$$

Since the conflicting objectives of minimizing the number of workstations and cycle time are considered simultaneously, some *redundant* workstations may be opened in some periods, without any matched tasks, just for the sake of tool sharing. The following constraints are then defined to eliminate such conflicts through allowing only *active* workstations to share their tools.

$$\sum_{i' \in I} S_{ii't} \leq \xi \sum_{j \in J} \chi_{jit}, \qquad \forall\ i \in I,\ t \in T, \tag{5.12.a}$$

$$\sum_{i' \in I} S_{i'it} \leq \xi \sum_{j \in J} \chi_{jit}, \qquad \forall\ i \in I,\ t \in T. \tag{5.12.b}$$

8. *Feasibility of decision variables*. These constraints ensure that all decision variables are non-negative and the best matching variables are binary.

$$S_{ii't}, C_t \geq 0, \quad \chi_{jit}, \psi_{ii't} \in \{0,1\}, \qquad \forall\, j \in J, i,\, i' \in I, t \in T. \tag{5.13}$$

A fuzzy goal programming method with imprecise goal hierarchy (Aköz and Petrovic, 2007) is applied for solving the above multiobjective model. The method enables the decision-makers to linguistically assign *soft* pairwise importance relations between the three conflicting objectives. The method utilizes an additive achievement function, as a linear combination of the achievement degrees of the goals, and the satisfaction degrees defined according to the hierarchical importance relations. Details of the fuzzy goal programming method with imprecise goal hierarchy are provided in Moghaddam and Nof (2015[a]).

The CAF is guarantees the balanceability of assembly lines in cases where all tools are compatible and the pairwise distances between the workstations are negligible.

Definition 5.1. *An assembly line is "balanceable" if the overall workload is equally distributed between all active workstations*.

Following the above definition, the necessary condition for a non-collaborative assembly line to be balanceable is

$$\sum_{j \in J} \rho_j \chi_{ji} = C^B, \qquad \forall\, i \in I, \tag{5.14}$$

where C^B denotes the cycle time of an equivalent balanceable line. Accordingly,

$$C^B = \frac{\sum_{j \in J} \rho_j}{|I|}. \tag{5.15}$$

Proposition 5.1. *The CAF guarantees balanceability of an assembly line if* $e_{ii'} = 1$, $\forall i, i' \in I$.

Proof. Let *A* and *B* denote the sets of fully- and partially-loaded workstations, respectively ($A \cup B = I$). Their workloads can then be calculated as follows:

$$L_i = \begin{cases} \sum_{j \in J} \rho_j \chi_{ji} - \sum_{i' \in B_i} e_{i'i} S_{i'i}, & \forall\ i \in A, \\ \sum_{n \in N} \rho_j \chi_{ji} + \sum_{i' \in A_i} S_{ii'}, & \forall\ i \in B. \end{cases} \tag{5.16}$$

According to Eq. (5.14), an assembly line is balanceable if all workstations have the same workloads equal to the cycle time of the equivalent balanceable line, *i.e.*, $L_i = C^B$, $\forall\ i \in I$, which, according to Eq. (5.15), can be written as

$$\sum_{i=1}^{|I|} L_i = |I| \cdot C^B = \sum_{j \in J} \rho_j. \tag{5.17}$$

According to Eq. (5.17), the line is balanceable if the overall workload of all workstations is equal to the sum of the tasks' processing times. From Eq. (5.16), it can be concluded that

$$\sum_{i=1}^{|I|} L_i = \sum_{j \in J} \rho_j + \left(\sum_{i' \in A} S_{ii'} - \sum_{i' \in B} e_{i'i} S_{i'i} \right) \geq \sum_{j \in J} \rho_j. \tag{5.19}$$

Thus, the line is balanceable if the second term on the right-hand side of Eq. (5.19) is equal to zero. By definition, $s_{ii'}$ represents the portion of time in which the tools of workstation i shares with workstation i' . Hence, $\sum_{i' \in A} S_{ii'} = \sum_{i' \in B} S_{i'i}$, and therefore, since $0 \leq e_{ii'} \leq 1$, $\forall i, i' \in I$, the necessary condition for balanceability of the (collaborative) assembly line is $e_{ii'} = 1$, $\forall i, i' \in I$. ■

Even if $\exists i, i' \in I$, $e_{ii'} \neq 1$, the CAF still yields a *relatively balanceable* assembly line, where the overall workload is "almost" equally distributed between the workstations. The cycle time of a relatively balanceable line is higher than the cycle time of a fully balanceable line.

Definition 5.2. *An assembly line is relatively balanceable if the overall workload is almost equally distributed between all workstations due to having some CE coefficients less than one, i.e.,* $\exists i, i' \in I$, $e_{ii'} < 1$. *The cycle time of a relatively balanceable line is greater than the cycle time of a fully balanceable line.*

Proposition 5.2. *If* $\exists i, i' \in I$, $e_{ii'} < 1$, *the CAF leads to relative balanceability of an assembly line, where the lower bound for the gap between the cycle time and* C^B *is*

$$\delta = \frac{\sum_{i \in I} \sum_{i' \in I} \left(S_{ii'} - e_{i'i} S_{i'i} \right)}{|I|}. \tag{5.20}$$

Proof. Dividing both sides of Eq. (5.20) by the number of workstations, we have

$$\frac{\sum_{i=1}^{|I|} L_i}{|I|} = C^B + \frac{\sum_{i \in I} \sum_{i' \in I} \left(S_{ii'} - e_{i'i} S_{i'i} \right)}{|I|}. \tag{5.21}$$

The left-hand side of Eq. (5.21) yields the average workload of all workstations that is greater than or equal to the cycle time of the line; since the workload is *almost* equally distributed among all workstations. Therefore, $\delta = C - C^B \geq \sum_{i \in I} \sum_{i' \in I} \left(S_{ii'} - e_{i'i} S_{i'i} \right) \Big/ |I|$. ■

5.3.2 Numerical Experiments

Three test-problems with 12, 16 and 24 tasks (denoted by P12, P16 and P24, respectively) are adapted from Kim *et al.* (2009) for numerical illustration and analysis. Figure 5.4 shows the precedence relation diagrams with acyclic networks where nodes and arcs respectively represent the task numbers and processing times, and the precedence relations between the starting and the ending node tasks. The tasks are numerically labeled with respect to their precedence relations. The periods are considered monthly and three periods are taken into account in all test-problems. Assuming 8 working hours per day, and 22 working days per month, the available production time is 176 hours in each period. The demand (units), the cycle time upper bound (hours), and the upper bound and the theoretical lower bound for the number of workstations are presented in Table 5.1. The data related to pairwise CEs and collaboration costs is available in Moghaddam and Nof (2015[a, b]).

To illustrate the superiority of the CAF over classic non-collaborative approaches, two scenarios are defined. The *non-collaborative* scenario, denoted by S_1, considers cases where workstations are isolated and no collaboration through tool sharing occurs between them. The second scenario, denoted by S_2, represents the CAF where collaboration is performed through tool sharing and best matching decisions.

Table 5.1. Demand, cycle time upper bound, and lower bound (L#W) and upper bounds (U#W) for the number of workstations.

Period	Demand			Cycle time			L#W			U#W		
	P12	P16	P24	P12	P16	P24	P12	P16	P24	P12	P16	P24
1	21.13	15	11	8.33	11.73	16.00	3	6	9	5	8	10
2	28.16	18	10	6.25	9.78	17.60	4	7	8	5	8	10
3	21.13	16	10	8.33	11.00	17.60	3	6	8	5	8	10

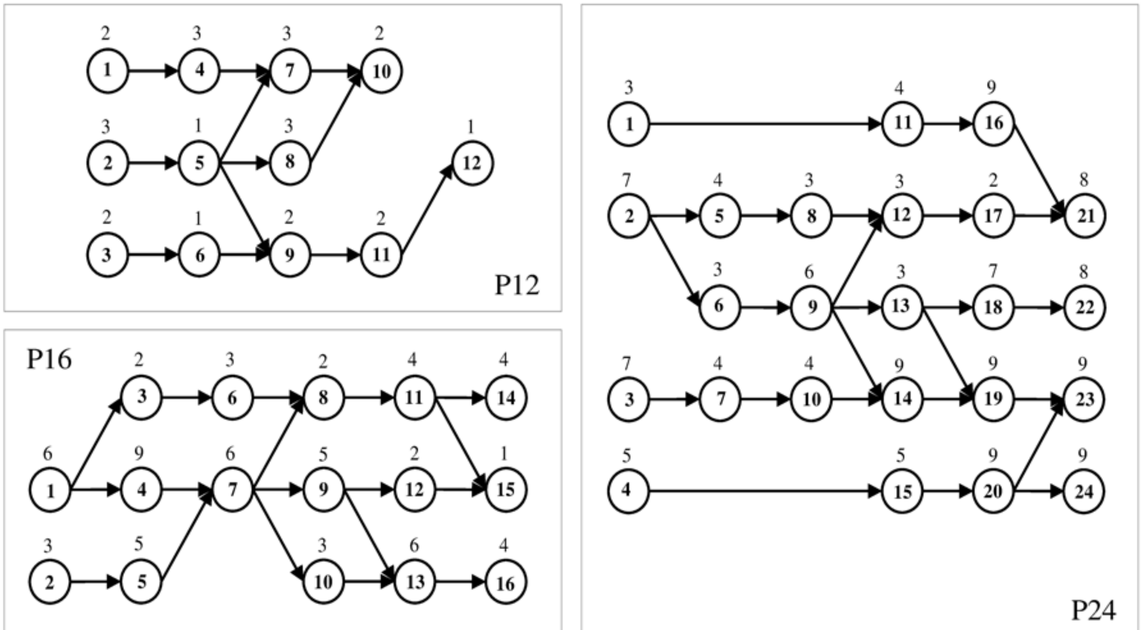


Figure 5.4. Processing times and precedence relations.

The GAMS software package and the CPLEX solver have been used for solving the MOMIP model (Tables 5.2 to 5.4). The experimental results show that demand variability and dynamic tool sharing between the workstations have resulted in different matching of the tasks to the workstations in different periods.

Table 5.2. Line balancing decisions for P12 under S_1.

Work-station	Tasks			Workload		
	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3
1	1-2-4	1-2	2	8 (A)*	5 (B)	3 (B)
2	3-5	3-4-5	3-5-6-9	3 (B)	6 (A)*	6 (B)
3	6-8	7-8	8	4 (B)	6 (A)	3 (B)
4	7-9-11	6-9-10	1-11	7 (B)	5 (B)	4 (B)
5	10-12	11-12	4-7-10-12	3 (B)	3 (B)	9 (A)*

*: Cycle time

Table 5.3. Line balancing decisions for P16 under S_1.

Work-station	Tasks			Workload		
	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3
1	1-2	1-2	1-2	9 (B)	9 (A)*	9 (B)
2	4	3-5	3-5-6	9 (B)	7 (B)	10 (A)*
3	3-5-6	4	4	10 (B)	9 (A)	9 (B)
4	7-8-10	6-7	7	11 (A)*	9 (A)	6 (B)
5	9-12	8-9-12	8-9-10	7 (B)	9 (A)	10 (A)
6	13-16	10-13	11-13	10 (B)	9 (A)	10 (A)
7	11-14-15	11-14-15	12-14-16	9 (B)	9 (A)	10 (A)
8	-	16	15	-	4 (B)	1 (B)

Table 5.4. Line balancing decisions for P24 under S_1.

Work-station	Tasks			Workload		
	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3
1	2-5-6	1-2-6-11	1-4-11	14 (B)	17 (A)*	12 (B)
2	1-8-9-12	5-8	2-5-8	15 (B)	7 (B)	14 (B)
3	11-13-16	9-12-13-17	6-9-12-17	16 (A)*	14 (B)	14 (B)
4	4-15	3-7-10	13-15	10 (B)	15 (B)	8 (B)
5	3-20	14-18	18-20	16 (A)	16 (B)	16 (B)
6	7-24	4-19	3-16	13 (B)	14 (B)	16 (B)
7	10-14	15-16	22-24	13 (B)	14 (B)	17 (A)*
8	18-19	20-22	7-10-14	16 (A)	17 (A)	17 (A)
9	17-23	24	19	11 (B)	9 (B)	9 (B)
10	21-22	21-23	21-23	16 (A)	17 (A)	17 (A)

Tables 5.2-5.4 show that the workloads of different workstations are significantly different under S_1. That is, without collaboration, Set B workstations suffer considerable idle times, while the line throughput is bounded due to the presence of fully loaded workstations (*e.g.*, P12: workstations 1, 2-3, and 5 in Periods 1 to 3; P16: workstation 4 in period 1; P24: workstations 1, 8, and 10 in Period 2). Therefore, in order to improve the balanceability of the line (*i.e.*, to simultaneously improve the overall utilization of the workstations and the line efficiency) tool sharing and best matching decisions must be applied, as suggested by

the CAF (*i.e.*, S_2). Details of the tool sharing decisions for P12, P16, and P24 are presented in Tables 5.5 to 5.7, respectively.

Table 5.5. Tool sharing and best matching decisions for P12 under S_2.

i	$S_{ii't}$			Target workstations (i')			Workload		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	-	0.06	2.08	-	2	2-5	4.98	5.06*	5.08*
2	2.00	-		1	-	-	5.00	4.99	4.97
3	1.07	-	2.05	1-4	2	5	5.07*	5.02	5.05
4	-	-	0.98	-	3	5	4.93	4.87	4.98
5	2.07	2.06		4	2-3-4	-	5.07	5.06	4.92

Table 5.6. Tool sharing and best matching decisions for P16 under S_2.

i	$S_{ii't}$			Target workstations (i')			Workload		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	0.34	-	-	3	-	-	9.34*	8.33*	7.91
2	0.34	1.27	-	3	1-3	-	9.34	8.27	8.06
3	-	-	-	-	-	-	9.17	8.00	8.36
4	-	-	2.38	-	-	2-3	9.26	8.00	8.38*
5	2.19	-	-	3-4-6	-	-	9.19	8.05	7.92
6	-	-	-	-	-	-	9.21	8.08	7.97
7	0.34	-	-	6		-	9.34	8.10	8.02
8	-	4.27	7.38	-	3-4-5-6-7	1-2-5-6-7	-	8.27	8.38

Table 5.7. Tool sharing and best matching decisions for P24 under S_2.

i	$S_{ii't}$			Target workstations (i')			Workload		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	0.08	-	2.18	2	-	7	14.08*	14.00	14.18
2	-	7.12	0.18	-	1-4-5-8	3	13.84	14.12*	14.18
3	-	0.12	0.36	-	4	7	13.74	14.12	14.18
4	4.08	-	6.19	2-3-5	-	5-6-8	14.08	13.99	14.19*
5	-	-	-	-	-	-	14.01	13.61	14.12
6	1.08	0.12	-	5	8	-	14.08	14.12	12.43
7	1.08	0.12	-	5-8	8	-	14.08	14.12	13.04
8	-	-	-	-	-	-	13.97	13.96	12.71
9	3.08	5.11	5.18	8-10	8-10	8-10	14.08	13.11	13.18
10	-	-	-	-	-	-	14.02	14.12	14.19

The results shown in Tables 5.5, 5.6, and 5.7 indicate that S_2 leads to "smoother" workloads through dynamic tool sharing between Set A and Set B workstations. Since the CEs are functions of pairwise linear distances between the workstations, the workstations in Set B prioritize their adjacent (or closer) bottleneck workstations for tool sharing purpose (see "target workstations" columns in Tables 5.5-5.7). Figure 5.5 shows that tool sharing and best matching decisions (*i.e.*, S_2) result in smoother workloads in different workstations compared with the non-collaborative scenario (*i.e.*, S_1) solutions. The numbers on each plot represent the cycle times. Figure 5.6 shows the improvements in the line efficiency (*i.e.*, cycle time improvement) and utilization of resources (*i.e.*, idle times reduction). The optimal collaboration costs per cycle are as follows. P12: \$17.67 in Period 1, \$26.99 in period 2, and \$53.71 in Period 3, P16: \$135.8 in Period 1, \$116.7 in Period 2, and \$152.2 in Period 3, and P24: \$100.0 in Period 1, \$125.2 in Period 2, and \$115.3 in Period 3. These values correspond to the costs of tool sharing among workstations (*e.g.*, additional setup; tool transfer; operator training), which is indeed lower than the costs of work sharing strategies, since under the CAF, there is no need to duplicate the tools for each single collaboration among the workstations.

Since the tools of each workstation may not be fully compatible with the ongoing tasks of the other workstations, *i.e.*, $\exists\, i, i' \in \boldsymbol{I}$, $e_{ii'} < 1$, the CAF leads to partially balanceable assembly lines. Efforts must be made to improve the compatibility of the tools (*e.g.*, cross-trained operators, multi-purpose facilities and robots) and augment tool sharing processes to achieve fully balanceable assembly lines through collaboration. To

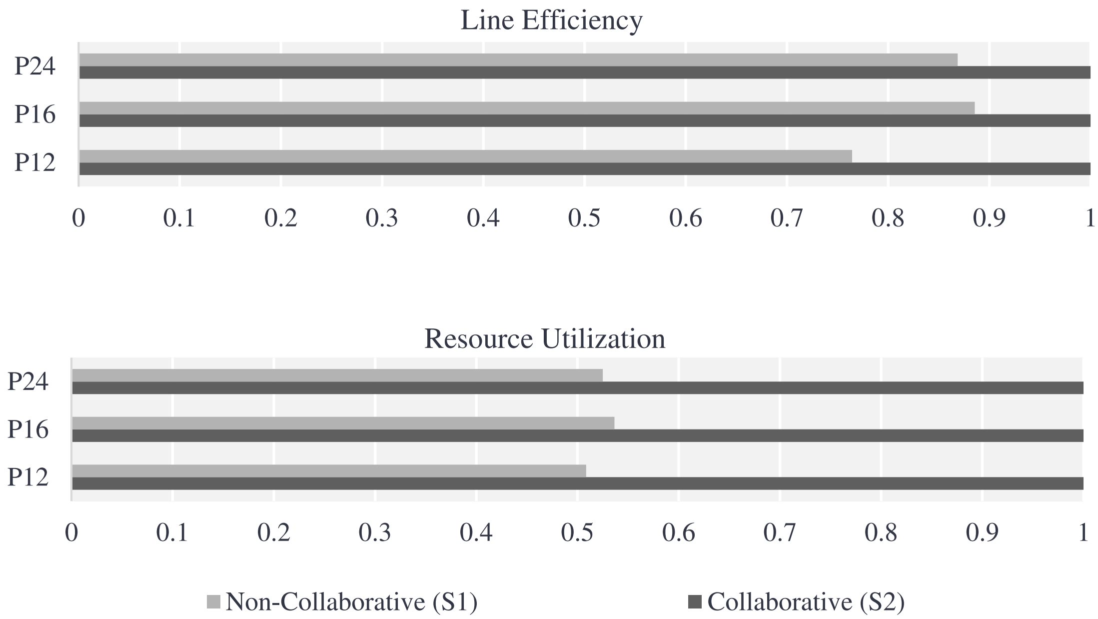


Figure 5.5. Line efficiency and resource utilization, under S_2 and S_1. (The average results of multiple periods are relatively normalized in [0, 1].)

numerically analyze the impact of CEs on the balanceability of the assembly lines, a sensitivity analysis is conducted on the P12, P16 and P24 test problems (Figure 5.7).

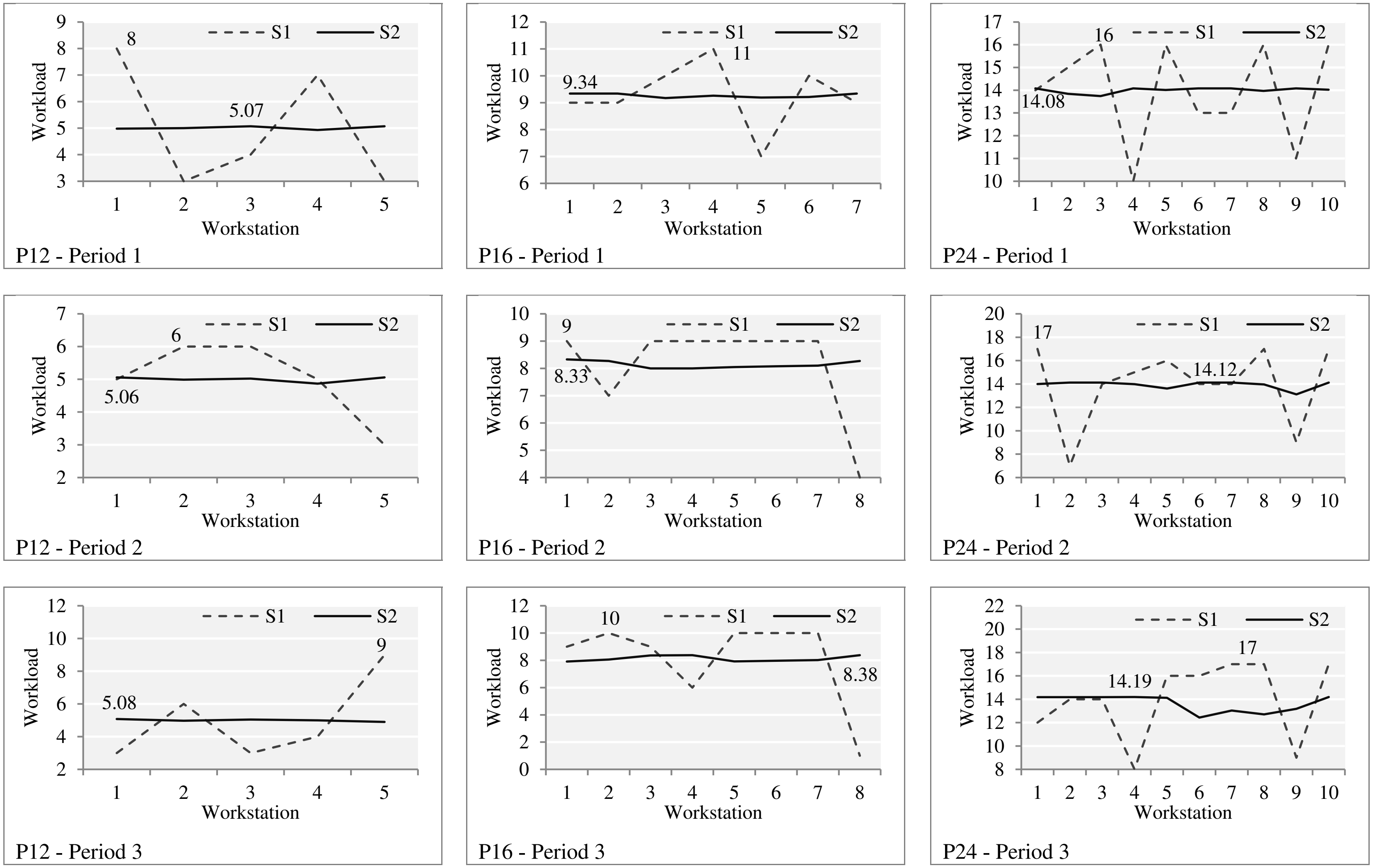


Figure 5.6. The workloads and optimal cycle time in different periods—the non-collaborative (S_1) *vs*. the CAF (S_2).

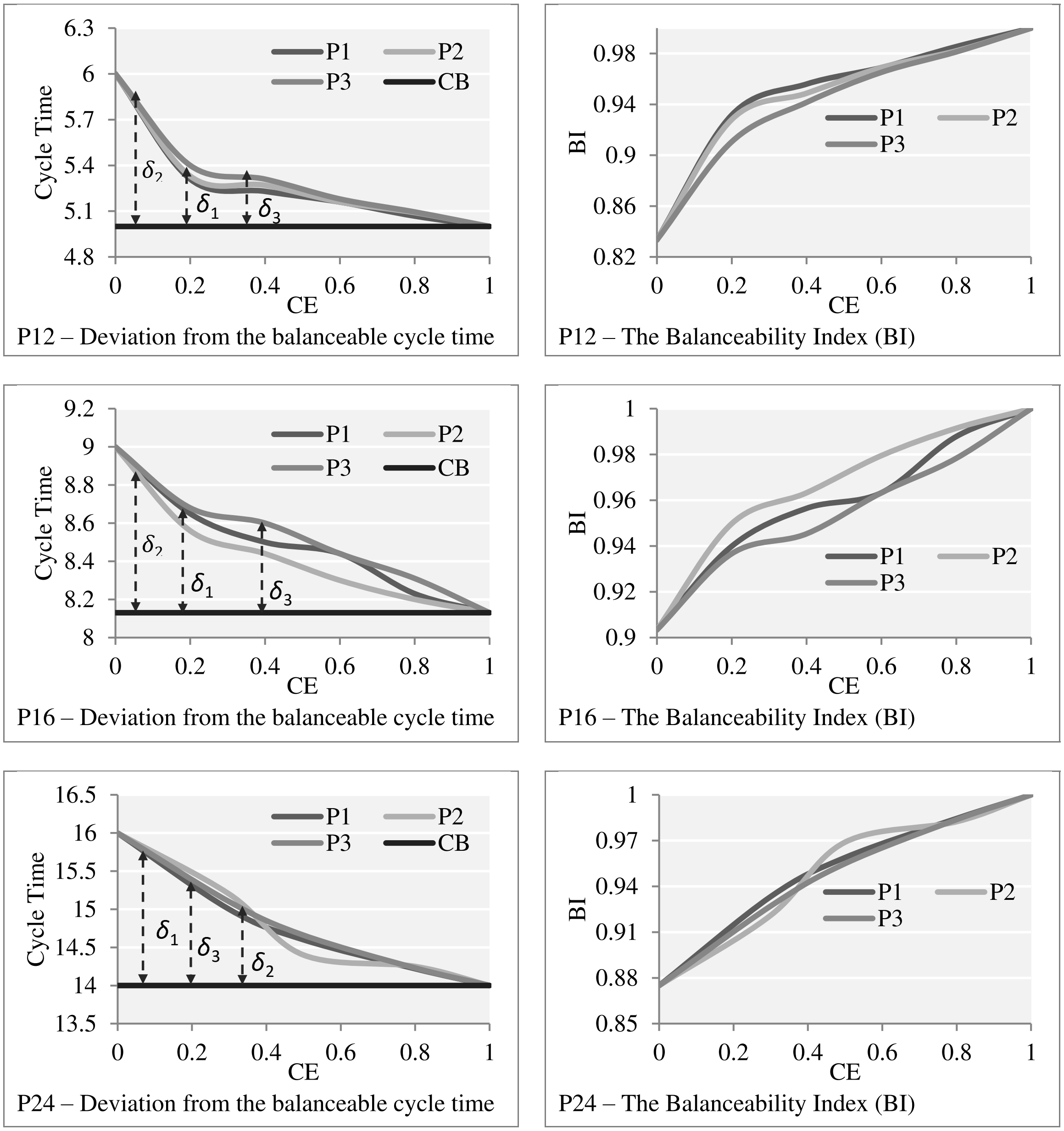


Figure 5.7. Cycle time variations/deviation from the balanceable cycle time and the BIs against different CEs, under the CAF. (PX: Period X; CB: Cycle time of balanceable line)

5.4 Control: CMAS

A CMAS (Collaborative Multi-Agent System) is developed for real-time implementation, monitoring, and modification of the generated plan, and providing feedback based on the system performance. The CMAS is composed of a loosely coupled network of heterogeneous agents that collaborate to improve the flexibility and balanceability of assembly lines. Two types of agents are incorporated in the model:

1. *Workstation agent* (*WA*). These agents are responsible for monitoring the progress of the tasks' processes, and calculating the estimated workload in each cycle. The WAs can be equipped with simple vision sensors and processors for calculation of the assembly task progress in real-time.
2. *Tool agent* (*TA*). These agents are responsible for negotiations and decision-making regarding collaboration with other workstations through the tool sharing protocol. As mentioned earlier, "tool" refers to any entity that processes the tasks, from a human operator, an assembly device (manual, semi-automated or fully automate), an automated assembly tester, to a fully automatic robot.

The developed multi-agent system is *collaborative* in nature; because each agent (either WA or MA) is defined as an intelligent software system (with some computational capability) that communicates and cooperates with other software systems to handle the dynamicity of the collaboration and matching decisions in real-time—something beyond the capability of each individual software system (Shen *et al.*, 2006). Both WAs and TAs must comprise the three principal c of an agent, according to Jennings *et al.* (1998):

- The agents must be *situated*, *i.e.*, must be able to receive sensory inputs from their environment. The WAs receive inputs through their sensors (*e.g.*, vision sensor), while the TAs receive their required inputs from other TAs and WAs. The input information, however, is limited and incomplete and thus, the agents need to collaborate under certain protocols for achieving a more flexible and balanceable assembly line.

- Both WAs and TAs must be *autonomous* in making decisions. This implies autonomous workload estimations by the WAs, and sharing/matching decisions by the TAs, without any human intervention.
- The TAs and WAs must be *flexible*; *i.e.*, be responsive to changes, proactive, and opportunistic, and interact with other peers, resolve potential conflicts, and synchronize their actions.

These characteristics can be achieved through combination of the agents' intelligence with effective coordination protocols. Various agent-based paradigms have been discussed in literature for distributed control of manufacturing and assembly systems (*e.g.*, Seliger and Krützfeldt, 1999; Bussmann and Sieverding, 2001; Shen *et al.*, 2005; Monostori *et al.*, 2006; Leitão, 2009), which deploy different types of agents to represent the ongoing tasks in the system. For example, Seliger and Krützfeldt (1999) introduce scheduler agents (*e.g.*, assembly, transportation, and supplier schedulers) along with carrier and assembly agents at different levels to construct a society of agents for assembly control. The CMAS developed here and the two types of agents, the WA and TA, can be embedded to any of such designs for a more holistic and practical representation of the system. The focus of this study, however, is on the real-time plan execution and feedback mechanisms for implementation of the CA framework.

In terms of distribution of control, agent-based paradigms are mainly categorized into three classes of fully hierarchical (Class I); semi-heterarchical (Class II); fully heterarchical (Class III). Unlike hierarchical (or vertical) control, *master-slave* relations (*i.e.*, hierarchies) cannot be clearly identified in a heterarchical (or horizontal) control system, which in turn, provides many advantages such as higher level of autonomy, along with some critical limitations such as incapability of long-term optimization (Trentesaux, 2009). Thus, the architecture of the integrated planning-control mechanism developed in this case follows Class II, *i.e.*, semi-heterarchical control, which is indeed a heterarchical control system with *supervisory control*. That is, despite the *heterarchy* between the agents (*i.e.*, WAs and TAs), their specific activities and responsibilities are periodically defined and updated by a *supervisor*; the off-line plan generated by the MOMIP model every T time units (Figure 5.8).

Effective interaction and communication between the distributed TAs and Was as well as synchronized local computations require a systematic collaboration protocol. The

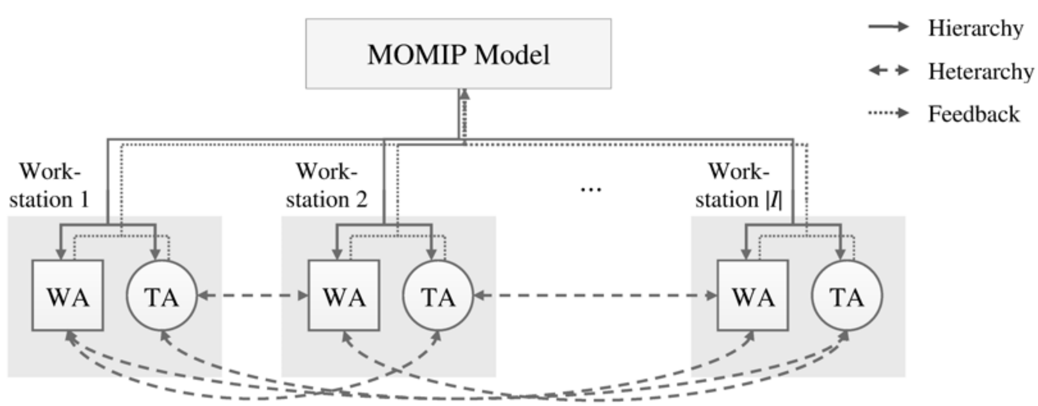


Figure 5.8. Semi-heterarchical architecture of the CMAS.

role of the collaboration protocol is to control the tool sharing activities and optimize the decisions in real-time through matching idle tools with overloaded workstations. The matching type is *one-to-one* at any given time, implying that an idle tool can serve only one workstation at a time. An idle tool, however, can be matched to more than one workstation and serve them in sequence. The assembly system operates according to the off-line plan generated by the MOMIP model, as a supervisory control mechanism. The reason is that since the plan is updated continuously, the likelihood of deviations from the optimal plan caused by the changes in the system parameters should not be high. In view of that, the need for modification of the plan is minimized and the system is expected to follow the plan that is proven optimal. The tool sharing-best matching protocol is based on continuous monitoring of the system's states and discrete events.

5.4.1 Tool Sharing-Best Matching Protocol

The plan is generated by solving the MOMIP model and obtaining $S^*_{ii'}$, χ^*_{ji}, C^*, and W^* (optimal number of workstations), and is executed for $\lceil T / C^* \rceil$ consecutive cycles, where T denotes the running time of the model, and * denotes optimality. In each cycle, the TS-BM protocol is activated as described below.

1. For $\lceil T / C^* \rceil$ consecutive cycles, repeat Steps 2 to 9. At the beginning of each cycle, set the current time t to zero.
2. After becoming idle at time t, the TA of the corresponding workstation, denoted by k, estimates its "progress rate" as follows:

$$\alpha_k(t) = \frac{\sum_{j \in J} \rho_j \chi^*_{jk}}{t}, \tag{5.22}$$

and sends a message to the WAs of other workstations requesting their progress rates.

3. The WAs that received the request calculate and report their progress rate as follows:

$$\alpha_i(t) = \frac{\varepsilon_i(t) \sum_{j \in J} \rho_j \chi^*_{ji}}{t}, \qquad \forall\, i = 1, \ldots, W^*, i \neq k, \tag{5.23}$$

where $\varepsilon_i(t)$ denotes the fraction of the workload of workstation i that is successfully processed by time t.

4. After receiving the progress rate reports, the TA of workstation *k* estimates the overall workload of all workstations (including its own workstation) as follows:

$$w_i(t) = \frac{1}{\alpha_i(t)} \sum_{j \in J} \rho_j \chi^*_{ji} + \sum_{i' \in I} \left(\frac{S^*_{ii'}}{\alpha_i(t)} - \frac{e_{i'i} S^*_{i'i}}{\alpha_{i'}(t)} \right), \qquad \forall\ i = 1, \ldots, W^*. \tag{5.24}$$

5. The TA of workstation *k* then finds the set of workstations that (a) are not in its predefined tool sharing plan, and (b) their overall workload is higher than the overall workload of all workstations that are in its predefined tool sharing plan (denoted by set X_k), *i.e.*,

$$O_k(t) = \left\{ i \mid S^*_{ki} = 0,\ w_i(t) > w_{i'}(t),\ \forall\, i' \in X_k \right\}, \tag{5.25}$$

where $X_k = \left\{ i \mid S^*_{ki} > 0 \right\}$. Then,

   a) If $O_k(t) = \varnothing$ and $\max_{i \in I} \left\{ w_i(t) \right\} \neq w_k(t)$, or $O_k(t) = \varnothing$ and $w_i(t) = w_{i'}(t)$, $\forall i, i' \in I$, the TA of workstation *k* follows its predetermined plan, and the procedure goes to Step 7. Otherwise, it proceeds to Step 5-b.

b) If $O_k(t)=\varnothing$ and $\exists i \in I$, $w_k(t) > w_i(t)$, the TA of workstation k revises its tool sharing plans and levels its workload through solving Model (M5.1):

$$\begin{aligned}
\min \quad & \hat{C}, \\
\text{s.t.} \quad & w_i(t) + e_{ki}\hat{S}_{ki} \le \hat{C}, \qquad \forall\ i \in X_k, \\
& w_k(t) - \sum_{i \in X_k} \hat{S}_{ki} \le \hat{C}, \\
& w_i(t) \le \hat{C}, \qquad \forall\ i = 1,\ldots,W^*,\ i \notin X_k,\ i \ne k, \\
& \hat{S}_{ki}, \hat{C} \ge 0, \qquad \forall\ i \in X_k.
\end{aligned} \tag{M5.1}$$

The procedure then goes to Step 7. Otherwise, it proceeds to Step 5-c. The superscript ^ denotes revised values and is used to distinguish from the optimal values of the plan.

c) If $O_k(t) \ne \varnothing$, the TA of workstation k revises the plan and levels the workload through solving Model (M5.2):

$$\begin{aligned}
\min \quad & \hat{C}, \\
\text{s.t.} \quad & w_i(t) + e_{ki}\hat{S}_{ki} \le \hat{C}, \qquad \forall\ i \in X_k, \\
& w_i(t) - e_{ki}\hat{S}_{ki} \le \hat{C}, \qquad \forall\ i \in O_k(t), \\
& w_i(t) \le \hat{C}, \qquad \forall\ i = 1,\ldots,W^*,\ i \notin X_k \cup O_k(t),\ i \ne k, \\
& \hat{S}_{ki}, \hat{C} \ge 0, \qquad \forall\ i \in X_k \cup O_k(t).
\end{aligned} \tag{M5.2}$$

The procedure then proceeds to Step 6.

6. The TA updates its set of workstations to be served through tool sharing, *i.e.*, $X_k \leftarrow X_k \cup \left\{ i \mid i \in O_k(t),\ \hat{S}_{ki} > 0 \right\}$.

7. The TA of workstation k always selects workstation $i \in X_k$ as its next destination for TS, where

$$|k-i| = \min_{i \in X_k} \{|k-i|\}. \tag{5.26}$$

8. The procedure fixes the value of $\alpha_k(t)$ for the rest of the cycle, then returns to Step 2, and is applied to the next tool that becomes idle after t and before $\hat{C}$ (if any).
9. At the end of each cycle, the real-time performances are recorded as feedback for the next generations of the off-line plan developed by the MOMIP model.

Real-time control and modification of the plan via the tool sharing-best matching protocol are based on the "relative performance" of the tools in different workstations, represented by the progress rates $\alpha_i(t)$, where $\alpha_i(t) < 1$, $\alpha_i(t) > 1$, and $\alpha_i(t) = 1$ respectively imply under-achievement of the expected performance values, over-achievement of the expected performance values, and full correspondence between the expected and realized performance values. If the progress rates of all workstations are similar, then regardless of the values, there is no need to modify the generated plan (*i.e.*, the second condition of Step 5-a always holds). Otherwise, the plan must be modified such that workstations with higher overall workloads receive extra aid from the other workstations, and thus the cycle time is improved. The modifications are performed by the TAs once their corresponding tools become idle.

As mentioned in Step 5, the TAs of the idle tool may encounter three different situations: (a) the bottleneck is among their corresponding workstations planned for tool sharing or there is no bottleneck, (b) their own workstation is the bottleneck (considering the workload of their originally assigned tasks along with the time required for TS defined by the off-line plan), or (c) the bottleneck is neither their own workstation nor among their corresponding workstations planned for tool sharing. In the first situation, the TA follows the initial plan expecting to receive extra aid from the other TAs that will be idled during the cycle time. In the second situation, the TA needs to revise its own tool sharing plan using Model (M5.1), as its own workstation is happened to be the bottleneck (considering the sharing times). In the first two sets of constraints in Model (M5.1), the TA revokes a portion of its tool sharing commitments and saves their times for their own workstation. The last set of constraints ensures that the overall workloads of other workstations do not exceed the revised cycle time. Similarly, in the third situation, the TA uses Model (M5.2) to revise the plan through revoking a portion of its tool sharing commitments (the first set of constraints), and either saving the time for its own workstation if it is bottleneck (the second set of constraints) or sharing with other bottleneck workstations (the third set of constraints).

If the TA has more than one target workstations for tool sharing, the sequence of services is defined based on their linear distance (Eq. (5.26))—at any point of time, the closets workstation to the shared tool has the highest priority to be served first. Note that our assumption is that the assembly line is straight, and the workstations are numbered sequentially. The tool sharing-best matching protocol levels the workloads of all workstations and thus improves the flexibility and balanceability of the line during the execution in the face of uncertainties and dynamicity of the system's operations and characteristics. These objectives are achieved through dynamic matching of the idle tools to the overloaded workstations. At the end of each cycle, the realized values of processing times, progress rates, and CEs are reported to the planning stage as feedback for the next generation.

5.4.2 Numerical Experiments

These experiments are intended to show the impact of the developed real-time control mechanism (the CMAS) on the flexibility of the designed system in terms of dealing with dynamic changes in the system's performance over time. Two scenarios are therefore defined in order to investigate different aspects of the dynamic CAF:

- *Static CAF* (S_1). This scenario merely considers the design aspect of the CAF; i.e., the results presented in Section 5.3.2. That is, the plan is generated off-line and no control mechanism is applied to modify the plan and provide feedback in real-time. The purpose is to highlight the role of the CMAS and the tool sharing-best matching protocol handling the uncertainties of the assembly process.
- *Dynamic CAF* (S_2). All aspects of the dynamic CAF, from off-line planning to real-time control are incorporated in this scenario. The assembly line is simulated to investigate capability of the developed CMAS and the tool sharing-best matching protocol in maintaining the flexibility and balanceability of the line in uncertain and dynamic environments compared to S_1.

Theoretically, it can be shown that the static scenario S_1, in its best performance, yields the same results as the dynamic scenario S_2 (see Moghaddam and Nof, 2015[b]). This situation occurs where all tools have the same progress rate (no matter if it is better than planned, the same, or worse). In other words, the second condition of Step 5-a of the tool sharing-best matching protocol always holds throughout the cycle, and therefore, no changes are made in the off-line plan. Otherwise, if the progress rates are not the same, the off-line plan can be improved by solving either Model (M5.1) or Model (M5.2), resulting in a better cycle time. Accordingly, considering this inherent superiority of S_2 over S_1, the focus of these experiments is on showing how large the gap between the cycle times of S_2 and S_1 can be, in a simulated environment.

The assembly system is simulated for 10 independent experiments on each individual plan associated with each test problem. The uncertainty is incorporated in the processes through adding random noises to the processing times. Random processing time values are generated following Normal distribution, as one of the best probability distributions for representing the uncertainties in processing time, where the mean values are as shown in Figure 5.4 and the standard deviations are equal to 0.2 of the mean. For simplicity of calculations, the progress rate of each workstation is approximated as the average progress regarding each task; *i.e.*,

$$\alpha_i = \frac{\sum_{n\in N}\left(\rho_n\chi_{ni}^{*}/\hat{\rho}_n\chi_{ni}^{*}\right)}{\sum_{n\in N}\chi_{ni}^{*}}, \qquad \forall\ i=1,\ldots,W^{*}, \tag{5.27}$$

where $\hat{p}_n$ denote the actual (simulated) processing times. Calculation of α_i, however, is performed by the WAs in practice.

Figure 5.9 compares the cycle time values obtained from S_2 and S_1 for P12, P16, and P24. Results show that the developed real-time control mechanism significantly improves the performance of collaborative assembly systems through modification of the tool sharing plan with respect to the dynamic variations in the system's characteristics. Moreover, our observations show that the gap between S_1 and S_2 is proportional to the cycle time. That is, as the cycle time of a system increases the positive/negative impact of high/low performance workstation, *i.e.*, workstations with $\alpha_i > 1$ or $\alpha_i < 1$, on the gap between the cycle times of S_1 and S_2 becomes more evident. The average improvements in the cycle time by S_2, compared to S_1, are 8.8%, 22.1%, and 25.1% for P12, P16, and P24, respectively. This phenomenon is also further intensified as the number of workstations increases, which usually results in a more diverse population of high and low performance workstations.

Compared to S_1, S_2 also yields lower variations in the cycle time in different independent experiments, which implies higher flexibility of the assembly system in the face of dynamic and unforeseen changes. In other words, minimization of the cycle time through S_2 increases the capability of the system in increasing its throughput (without any additional resources) in the cases of abrupt increases in the demand. These findings indicate the superiority of the developed control mechanism over classic off-line planning approaches, and its role in upholding the performance of assembly systems in real-time.

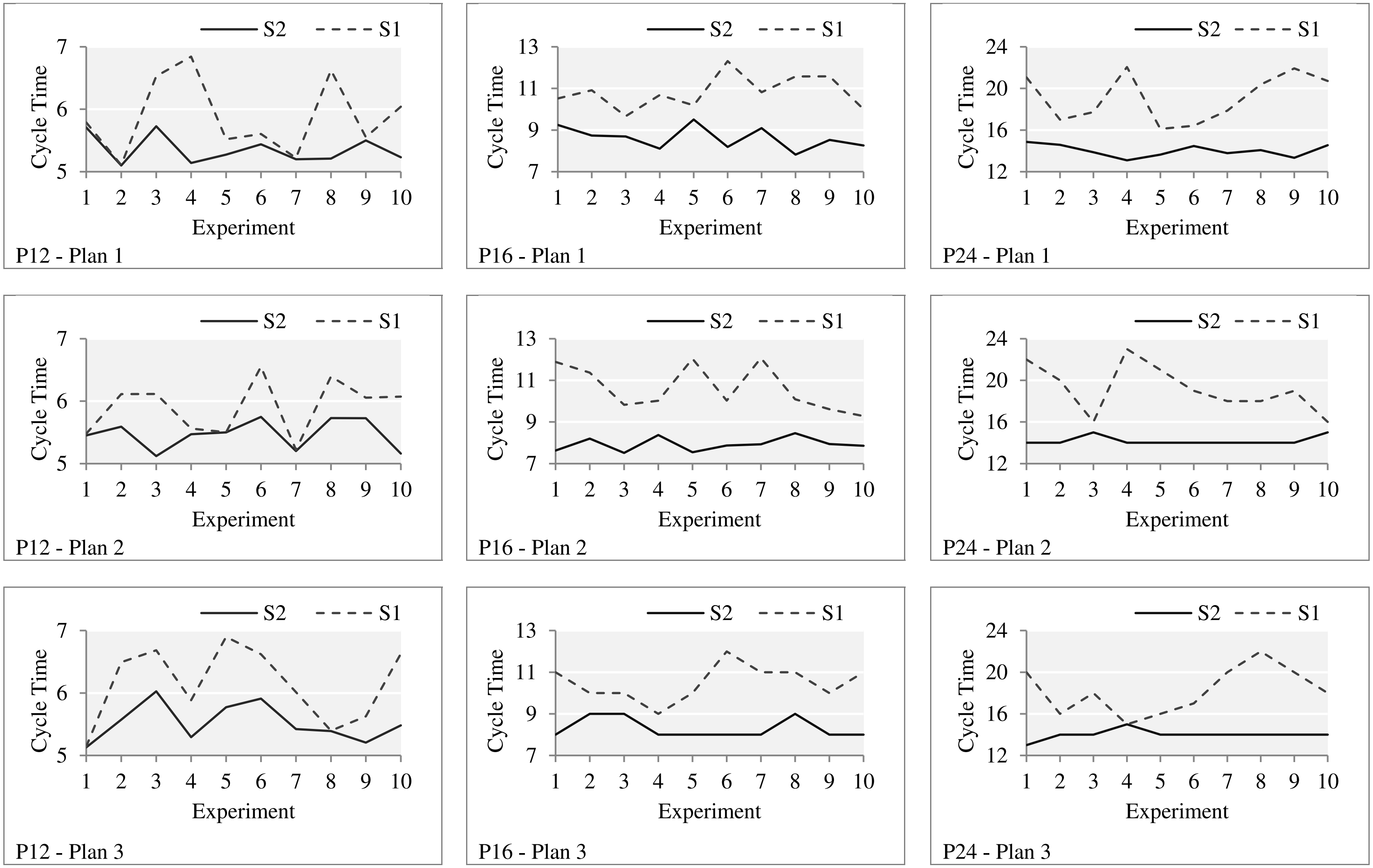


Figure 5.9. Cycle time variations in different experiments—the static CAF (S_1) *vs*. the dynamic CAF (S_2).

5.5 Concluding Remarks

It was shown, both mathematically and through several numerical experiments, that the *relative* balanceability of assembly lines is guaranteed by the CAF, where, for a fixed line configuration, the deviation of the optimal cycle time from its ideal value (*i.e.*, the cycle time of balanceable line) is a function of CE coefficients. That is, assuming an ideal assembly line with "all-purpose" tools located in different workstations along with negligible tool transfer times, the CAF results in a fully balanceable assembly line. The experimental results highlight the impact of CEs on the balanceability of assembly lines. Improved balanceability, in turn, leads to higher flexibility of the assembly line in dealing with dynamic variations of the demand.

Collaboration through tool sharing and best matching increases the maximum capacity of the assembly line in terms of throughput rate (*i.e.*, minimizes the minimum CT). Accordingly, in case of unforeseen increases in the demand, the CAF enhances the possibility timely response to such disruptions without extra investments. On the other hand, in case the required throughput rate is lower than the maximum throughput rate (*i.e.*, balanceable line), tool sharing may be temporarily avoided to reduce unnecessary collaboration costs. The unique advantage of the CAF compared to other collaborative approaches (*e.g.*, work sharing) is the emphasis on optimal identification, assignment, and utilization of the *existing* idle resources in elimination of bottleneck workstations, rather than investing in extra resources (*e.g.*, equipment duplication). The developed mathematical formulation provides a flexible and powerful multiobjective optimization tool for the CAF, which can be used for solving small-sized collaborative ALB problems.

The framework developed and discussed in this case provides necessary and general insights for understanding of the notion of collaborative processes in assembly lines and further investigations in the future. Thus, although the CAF is validated in this example with cases applied in previous research based on realistic small-sized ALB problems, a next step is to implement this framework in more complex industrial assembly lines. A multi-agent system is designed and the detailed workflow, from the design (the MOMIP model) to the real-time execution and control (the CMAS) is developed. The statistical analyses on the results of different scenarios indicate significant improvements in the cycle time by the newly developed design-control mechanism for the CAF, in comparison with the classic ALB approaches without tool sharing-best matching, and the static CAF (Moghaddam and Nof, 2015[b]).

Without loss of generality, a set of simplifying assumptions have been made, which need further investigation from the systems engineering aspect. The CE coefficients are defined as the representatives of tool sharing success between each pair of workstations, considering tool-task compatibility, setup times, movement times, and so on. Clearly, case-specific technical and technological limitations may prevent the CEs from taking their highest values, and thus the assembly line from being fully balanceable. Flexibility and capability of the shared tools in joining the process of a task at any point, and the possibility of processing a task with multiple tools at the same time are other challenges to be addressed in future work.

CHAPTER 6. CASE 3—CLUSTERING WITH INTERDEPENDENT PREFERENCES[☆]

The generalized matching problem is analogous to the *capacitated clustering problem*—the problem of partitioning a number of individuals into disjoint clusters with certain capacities. That is, capacitated clustering can be recast as a problem of finding the best two-sided match between the sets of entities (set I) and clusters (set J). For example, customers, tasks, and interns may represent the sets of entities to be respectively matched to suppliers, machines, and hospitals, each representing a specific cluster with limited capacity (*e.g.*, a supplier can serve a limited number of customers; a machine can process a limited number of tasks; a hospital can admit a limited number of interns). The difference between the many-to-one matching problem and the capacitated clustering problem, however, is in their objectives. The objective of the many-to-one matching problem is to maximize a set of matching criteria subject to certain capacity limits and requirements, while the capacitated clustering problem merely ensures that the clusters' capacities are not violated (Osman and Christofides, 1994).

Matching criteria, as thoroughly discussed in CHAPTER 3, are diverse; from cost to distance (*e.g.*, between suppliers and customers), performance (*e.g.*, machines

---

[☆] The preliminary version of this case study was presented at the *Industrial and Systems Engineering Research Conference*, Nashville, USA, 2015. The materials presented in this case study are adapted from two works of the author published in *Decision Support Systems* (DOI: 10.1016/j.dss.2015.08.005) and partially in the *International Journal of Advanced Manufacturing Technology* (DOI:10.1007/s00170-015-7806-7).

processing tasks with different speeds), quality (*e.g.*, dimensional tolerance of assembly products), and stability, depending on the application and scope. These matching criteria can be formalized as *preferences* of the elements of different sets for each other. Such preferences may or may not be fixed and independent of environmental and decisional factors. The purpose of this case is to investigate an important behavior associated with the generalized matching problems: The potential influences of matching decisions on the individuals' preferences. Specifically, the motivation is to investigate two-sided matching instances between sets I and J, where the preference of $i \in I$ for $j \in J$ may be influenced by and represented as a function of matching $i' \in I$, $i' \neq i$, to $j \in J$.

To better comprehend the notion of interdependencies among preferences in best matching, consider the following example (Figure 6.1). Suppose you have just arrived at a

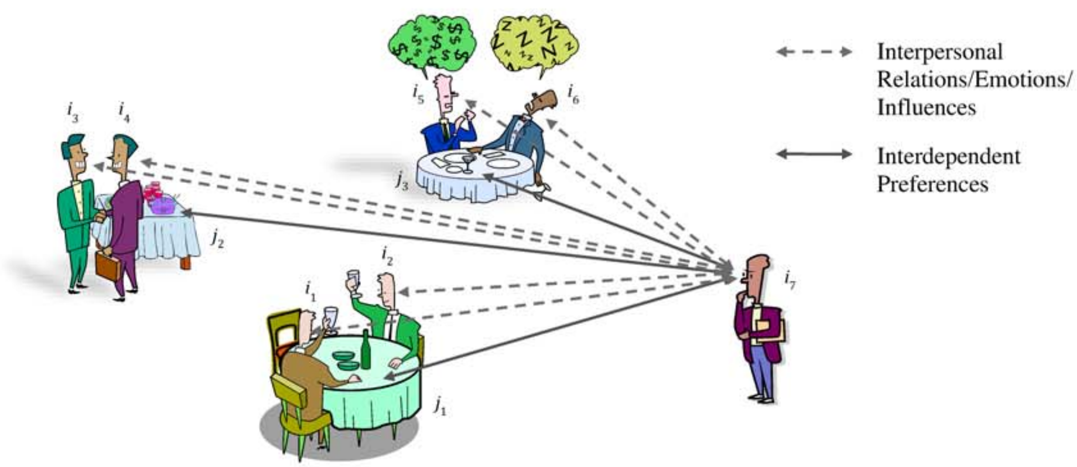


Figure 6.1. The social example—the initial preferences of the recently arrived guest (i_7) for the tables (j_1 to j_3) is influenced by his interpersonal relations/emotions with regard to the other guests already sitting at the tables (i_1 and i_2 at j_1; i_3 and i_4 at j_2; i_5 and i_6 at j_3). Attendance of i_7 to each table may drastically change the topology of the entire network. For instance, if i_7 chooses j_3, and i_5 is extremely *envious* about him, i_5 may leave j_3 right away. Similarly, if i_4 is *altruistic* about i_7, he may leave j_2 and join j_3.)

social gathering and are going to choose a table to sit at (Guests: Set *I*; Tables: Set *J*). Besides your initial preference for each individual table (*e.g.*, location; food), your choice may be influenced by the people who are already sitting at each table. It is a natural phenomenon; we have different perceptions and attitudes about different people. The preferences, as described earlier, are merely an abstraction of various matching criteria, and thus, such interdependencies may have certain implications in different application domains. Some practical examples are:

- *Enterprise collaboration*. The *profitability* of a particular coalition for an enterprise may be influenced (increased/decreased) by the members (*i.e.*, other enterprises) of that coalition.
- *Wireless sensor networks*. The choices of an individual sensor for different clusters in terms of *energy consumption* may be influenced by the type, number, and energy level of the sensors in each cluster.
- *Swarm robotics*. The *efficiency* of an individual robot may be influenced by its assignment to different teams, depending on the depending on the type, number, and functionality of the robots in each team.
- *Scheduling*. The optimal allocation of a task to a machine in terms of *makespan* or *cost* may be influenced by the processing and/or setup requirements of the tasks that are already in process or in the queue of each machine.
- *Storage assignment*. The best storage location for a particular product in terms of *total movement time* of material handling devices may be influenced by its *affinity* with the already allocated products.

6.1 Motivation

Interdependencies between preferences can dramatically influence best matching decisions and lead to non-optimal or even paradoxical decisions, if disregarded. The notion of IP (Interdependent Preferences), coined by Gaertner and Pollak in the 1970s, has been extensively investigated in utility theory, as an indication of the dependencies of the individuals' preferences on the *consumption* or *well-being* of other individuals in their neighborhood (Tomes, 1986; Postlewaite, 1998; Koçkesen, 2000; Bell, 2002; Sobel, 2005; Cabrales and Calvó-Armengol, 2008). Also referred to as *peer influence*, *neighborhood effect*, *bandwagon effect*, and *conformity* (Yang and Allenby, 2003), IP leads to either *altruistic* or *envious* behaviors—instead of considering their own absolute payoffs, individuals tend to evaluate their payoffs *relative to* those of others (Risse, 2011; Jamison, 2012).

In social sciences and psychology, the notion of IP is known as *interpersonal relations*/*behaviors*/*emotions* (*e.g.*, Lee *et al.*, 2011; Morita and Burns, 2014), and is proven to have significant impacts on social interactions and team/group activities (Manning *et al.*, 2008; Yilmaz and Peña, 2014). Examples include success/failure of collaborative marketing and sales teams (Niculescu, 2013), conflicts, job satisfaction, effectiveness, and turnover of interactive nursing units (Cox, 2001), efficiency and rate of errors/ miscommunication in surgical units and operating rooms (Lingard *et al.*, 2002; Romanowski *et al.*, 2013), performance, throughput, and cost of construction projects (Ling and Tran, 2012), all influenced by certain mutual interactions among individuals.

The major motivations of this case are therefore the extensive applications of IP, the lack of generic and formal analysis of it in the matching literature, and the impact of IP on capacitated cluster formation and evolution. Accordingly, this case study defines, formulates, and analyzes an extension of the generalized matching with IP, where the elements of the same set influence each other's preferences for the elements of the other set(s), if matched to the same element.

6.2 Outline

The many-to-one best matching problem with IP (henceforth, BMP-IP) is considered, where each element of set *I* can be matched to up to one element from set *J*, considering interdependencies among the preferences of the elements of set *I*. The BMP-IP under study is indeed a capacitated clustering problem, where each element of set *J* can be matched to a limited number of elements from set *I*. The clustering must be performed with respect to the mutual preferences of *I*'s and *J*'s, the capacity limits of *J*'s, and the influences of *I*'s on each other's preferences. According to the PRISM taxonomy of best matching, the BMP-IP can be characterized as shown in Figure 6.2.

The BMP-IP is first investigated from the cluster formation perspective: Sets *I* and *J* are fixed, and all elements are matched simultaneously, given their preferences and respective IP, capacity requirements and limits. The outcome of the capacitated many-to-one matching is a set of clusters, each corresponding to a specific element of set *J*. These clusters may not be necessarily disjoint and may interact and collaborate with each other; *e.g.*, demand-capacity sharing among suppliers (Moghaddam and Nof, 2014); tool sharing

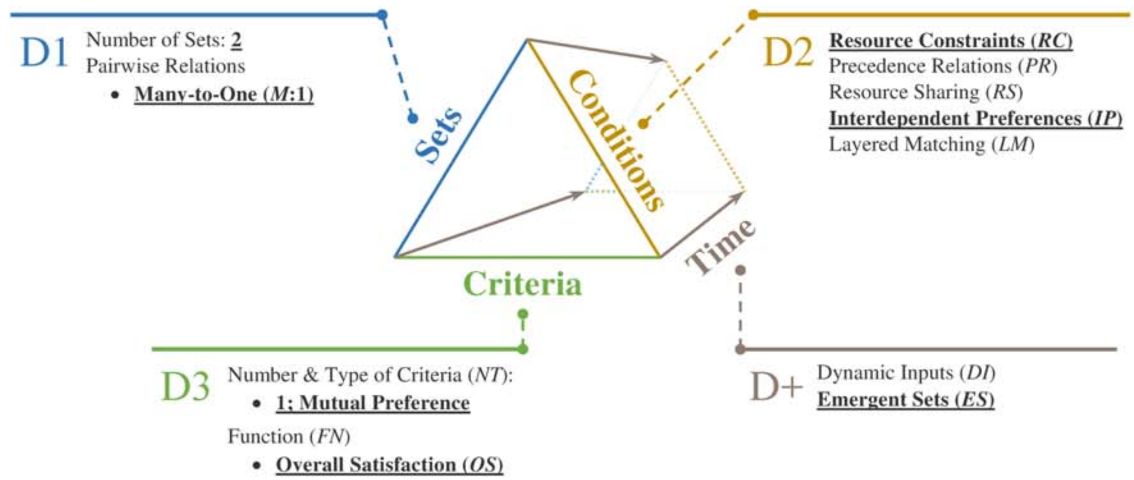


Figure 6.2. Classification of the BMP-IP according to the PRISM taxonomy of best matching.

among machines (Moghaddam and Nof, 2015[a, b]); task sharing enterprises (Moghaddam *et al.*, 2016). The problem is formulated as a binary program, with IP defined as a linear combination of binary best matching variables. Accordingly, the BMP-IP turns into a Quadratic Assignment Problem (QAP), which is known as an NP-hard problem. Therefore, a Genetic Algorithm (GA) is developed to handle the inherent computational complexity of the BMP-IP.

Following the formation, a mechanism is developed for real-time administration and control of the dynamic changes in the structure of the capacitated clusters. In this work, evolution or *emergence* (Barabási and Albert, 1999) of the networked clusters refers to association of new individuals or dissociation of existing individuals (to/from either set I or J). These behaviors are formalized by two principles of the collaborative control theory (Nof, 2007): *association/dissociation* and *emergent lines of collaboration and command* (Figure 6.2; D+). Such evolutionary behaviors are influenced by two main characteristics of the BMP-IP: The limited capacities of clusters and the interdependencies between the individuals' preferences. An Evolutionary Algorithm (EA) is therefore developed to model and analyze the impact of IP on the real-time administration, optimization, and control of the evolution of the networked clusters. Several experiments are performed on various test-problems to investigate the impact of IP on the formation and evolution of capacitated clusters, the sensitivity of best matching decisions to the intensity of interdependencies among preferences, and the computational efficiency of the developed algorithms.

6.3 Cluster Formation: QAP and GA

The mathematical formulation of the BMP-IP was presented earlier in CHAPTER 3 (Model (M3.4); *M*:1 / *RC*, *IP* / +, *OS*) as a QAP (Quadratic Assignment Program). The QAP is NP-hard (Sahni and Gonzalez, 1976), and thus is the BMP-IP. Several exact and heuristic approaches have been developed and examined in literature for solving the QAP (see, *e.g.*, Öncan, 2007; Burkard *et al.*, 2009), among which Ant Systems (Gambardella *et al.*, 1999; Maniezzo and Colorni, 1999; Talbi, *et al.*, 2001), and GA (Ahujaa et al., 2000; Drezner, 2003 and 2008) are the most common techniques. In this case, GA is applied due to its simplicity and capability of solution representation and regeneration, ability to work with multiple solution sets, and compatibility with unique specification of different problems in terms of encoding and decoding schemes (see CHAPTER 2).

The notion of GA is based on the evolution through *natural selection*, which is the foundation for the evolutionary mechanisms developed in this work. To avoid the infeasibility of the solutions (in terms of capacity limits) and increase the efficiency of the algorithm, a greedy heuristic is developed for generation of the initial population, and a reproduction scheme based on path re-linking method (Glover, 1994) is developed for reproduction through crossover and mutation, as described next.

6.3.1 Encoding

In GA, each potential solution set is encoded as a *chromosome*. The efficiency and performance of the algorithm, in general, and the reproduction functions, in particular, depend on the applied encoding scheme. For the BMP-IP, a natural encoding scheme is applied: an array of length $|I|$, where each specific *gene* of a chromosome $\mathbf{c}$ corresponds to a specific element of set I, and each *allele* of each gene takes a value in J (Figure 6.3). Note that if the overall capacity of set J is less than the size of set I, some elements of set I may remain unmatched. In the proposed encoding scheme, the alleles of the unmatched genes (if any) take zero values. Note that due to the limitations on the number and capacity of the elements of set J, a generated chromosome may be infeasible. Therefore, a greedy heuristic is developed for initialization of the algorithm, as described next.

6.3.2 Initialization

The performance of GA substantially depends on the initial population generation, in terms of diversity, fitness, and, in this case, feasibility of chromosomes. To ensure these qualifications in the initial population, a greedy heuristic is developed, which generates each chromosomes of the initial population through the following steps:

1. Generate a random permutation of the elements of I and store them in R in that order.
2. Set $\hat{P}_{ij} \leftarrow P_{ij}$, $\forall i, j$.

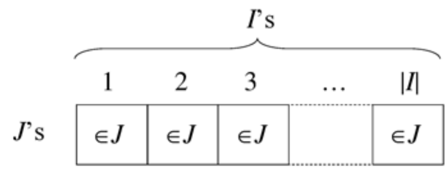


Figure 6.3. Chromosomes (**c**) representation.

3. Pick the first element of R, denote it by k, and assign it to $l \in J$ such that.

   $\hat{P}_{kl} = \max_{j \in J} \left\{ \hat{P}_{kj} \right\}$.

4. Set $M_l \leftarrow M_l - 1$, and $R \leftarrow R \setminus \{k\}$.

5. Set $\mathbf{c}(k) = l$, $\chi_{kl} = 1$, and update all $\hat{P}_{ij}$'s using Eq. (3.5) in CHAPTER 3.

6. If $M_l > 0$, go to the next step. Otherwise, set $J \leftarrow J \setminus \{l\}$, and then go to the next step.

7. If

   a) $R = \varnothing$, stop.

   b) $R \neq \varnothing$ and $J = \varnothing$, set $\mathbf{c}(i) = 0$, $\forall\, i \in R$, and stop.

   c) $R \neq \varnothing$ and $J \neq \varnothing$, return to Step 3.

The diversity of the initial solution is guaranteed by random ordering of I's while generating each chromosome (Step 1). In addition, the generated chromosomes are expected to have acceptable fitness values, because each individual is matched to its most preferred "available" match (Step 3). The generation also avoids capacity violation (*i.e.*, infeasibility) in each chromosome by eliminating full J's from the list of potential choices for the next I's (Steps 6 and 7-b). If the available capacity of J's is less than the number of I's, the alleles of some genes take zero values in the chromosome.

6.3.3 Fitness Evaluation

The fitness value of each chromosome **c**, which represents the overall satisfaction degree of mutual preferences/IP is calculated using the objective function of Model (M3.5) in CHAPTER 3, where

$$\chi_{ij} = \begin{cases} 1, & \text{if } \mathbf{c}(i) = j \\ 0, & \text{otherwise} \end{cases}$$

6.3.4 Reproduction

After generation of the first population and evaluation of their fitness values, the next step is to produce the next population from the existing population of chromosomes. In a GA, reproduction can be performed in a variety of manners and through different genetic operators. *Parent* chromosomes must be selected to produce offsprings. According to the theory of natural selection and evolution, natural reproduction must result in stronger or *fitter* chromosomes in the next generations. The reproduction mechanisms of the GA must then be designed in a way that mimics this trait of biological systems. In this case, in addition to the fitness of the generated offsprings, their feasibilities must also be taken into account—to uphold the performance of the algorithm, infeasible solutions must be avoided during the reproduction procedure. Hence, a reproduction scheme is developed based on *path crossover* and *path mutation* mechanisms.

- *Path crossover*. Crossover is a major reproduction mechanism in natural evolution, which combines the genes of two (or more) parents to produce offsprings. The crossover operator is critical for the success of GA. The operator must be able to probabilistically *explore* new solutions (to ensure diversity), and *exploit* good traits of previous generations (to ensure fitness). Therefore, a modified version of the path crossover method (Glover, 1994) with insert transformation (Ahuja *et al.*, 2000) is developed, which besides diversity and fitness, guarantees the feasibility of the produced offsprings (Figure 6.4):

1. Randomly select two parent chromosomes $\mathbf{c}_1$ and $\mathbf{c}_2$, and *fix* the genes with similar alleles (exploitation of good traits of parents). Randomly select a gene, and set $Z = 0$.
2. If the current gene is fixed, set $Z \leftarrow Z + 1$ and go to Step 3 (no offspring is generated). Otherwise, perform *insert transformation*:
    a) In parent $\mathbf{c}_1$, randomly select an *unfixed* gene (other than the current gene), which has the same allele as the one in the current gene of $\mathbf{c}_2$. If there is no such gene in $\mathbf{c}_1$, go to Step (c) (no offspring will be generated).
    b) Insert the allele of the selected gene at the current gene of $\mathbf{c}_1$, and shift the alleles of other unfixed genes to the right.
    c) Apply (a) and (b) to $\mathbf{c}_2$, then go to the next step.
    d) Place the *original* parents (*i.e.*, $\mathbf{c}_1$ and $\mathbf{c}_2$) and the generated offsprings (*i.e.*, *transformed* parents) into the chromosome pool *CP*.
3. If $Z = |I|$, go to Step 4. Otherwise, move to the next gene, from left to right in a cyclic fashion, and return to Step 2.

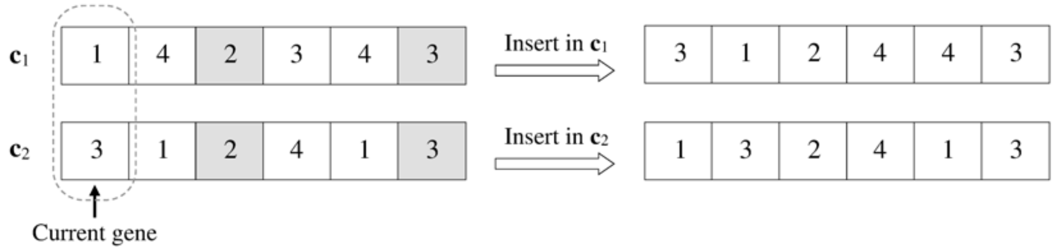


Figure 6.4. Example of path crossover operation on two parent chromosomes $\mathbf{c}_1$ and $\mathbf{c}_2$ [Shaded: Fixed genes].

4. Eliminate duplicates of chromosomes from *CP* (if any), sort the rest according to their fitness values, and select a number of fittest ones considering the population size and the crossover rate.

- *Path mutation*. Mutation is a key biological evolution mechanism, which makes spontaneous and random changes in the alleles to ensure diversity in offsprings. The developed path crossover mechanism—in spite of its efficiency—does not make any changes in the "consumed" capacity of J's, as the number of I's matched to each $j \in J$ is the same in both parents and offsprings. This may eliminate some potential solutions, especially if the total capacity of J's is larger than the number of I's (*i.e.*, at least one cluster with one idle capacity). To resolve this issue and ensure the diversity of the next generations in terms of capacity utilization, a *path mutation* mechanism is developed. The mutation operator, similar to the crossover operator, produces offsprings through moving along a path and making spontaneous changes in the genes in sequence. The path mutation operator is designed in a way that the capacity limits are not violated, while, near-optimal offsprings are generated. If there is no idle capacity in the entire network of clusters, however, the mutation operator is skipped. The mutation procedure is as follows (Figure 6.5):

1. Randomly select a parent chromosome **c** and a gene $k \in I$, set $Z = 0$, and generate a list of J's with free capacity, *i.e.*, $L = \{ j : M_j > |\{ i : \mathbf{c}(i) = j,\ i \in I \}|,\ j \in J \}$.
2. Replace the allele of the current gene of **c** (*i.e.*, k) with $l \in L$ where $\hat{P}_{kl} = \max_{j \in L} \{ \hat{P}_{kj} \}$.
3. Place the original parent and the offspring into the chromosome pool CP, and set $Z \leftarrow Z + 1$.
4. If $Z < |I|$, move to the next gene from left to right in a cyclic fashion (*i.e.*, update k), and then return to Step 2. Otherwise, go to the next step.
5. Eliminate duplicates of chromosomes from CP (if any), sort the rest according to their fitness values, and select a number of fittest ones according to the population size and the mutation rate.

6.3.5 Parameters Setting

The parameters of the GA are set as follows. Population size is equal to $2 \times |I|$, proportional to the size of the problem. The algorithm stops if there is no improvement in the best solution obtained in the last 100 generations (stopping criterion). The crossover rate is 0.7, *i.e.*, 70% of the population undergo the path crossover operation. The mutation rate is 0.3, *i.e.*, 30% of the population undergo the path mutation operation.

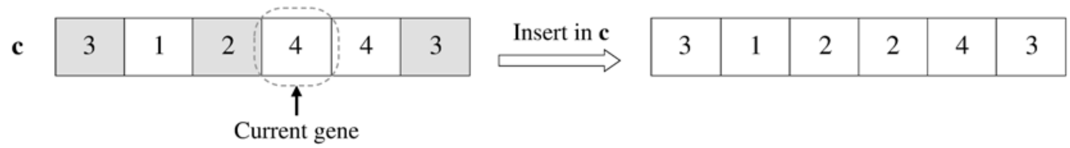


Figure 6.5. Example of path mutation operation on parent chromosomes **c**, where $\hat{P}_{4,2} > \hat{P}_{4,3}$ [Shaded: *J*'s with extra capacity; *i.e.*, $j \in L$].

6.4 Cluster Evolution: EA

Evolution, in the context of this case study, refers to the association/dissociation of individuals to/from the system over time. Due to the interdependencies between the preferences of *I*'s and the limited capacities of *J*'s, each single association of a new individual or dissociation of an existing one may significantly influence the optimal topology of the networked clusters. Examples include:

- Set *I*. Arrival/departure of a job to/from a production cell; Allocation of new interns/students to hospitals/schools; Introduction/declination of a product family in a market; Temporary or permanent loss of a team member or leader.
- Set *J*. Purchasing or salvaging facilities (*e.g.*, production equipment; transportation vehicles); Emergence/saturation of a new/an existing market; Opening or closing a line in a service system; Establishing a new school/hospital.

Figure 6.6 schematically illustrates the four types of evolution associated with the generalized matching. In a dynamic coalition of enterprises, for example, evolution can

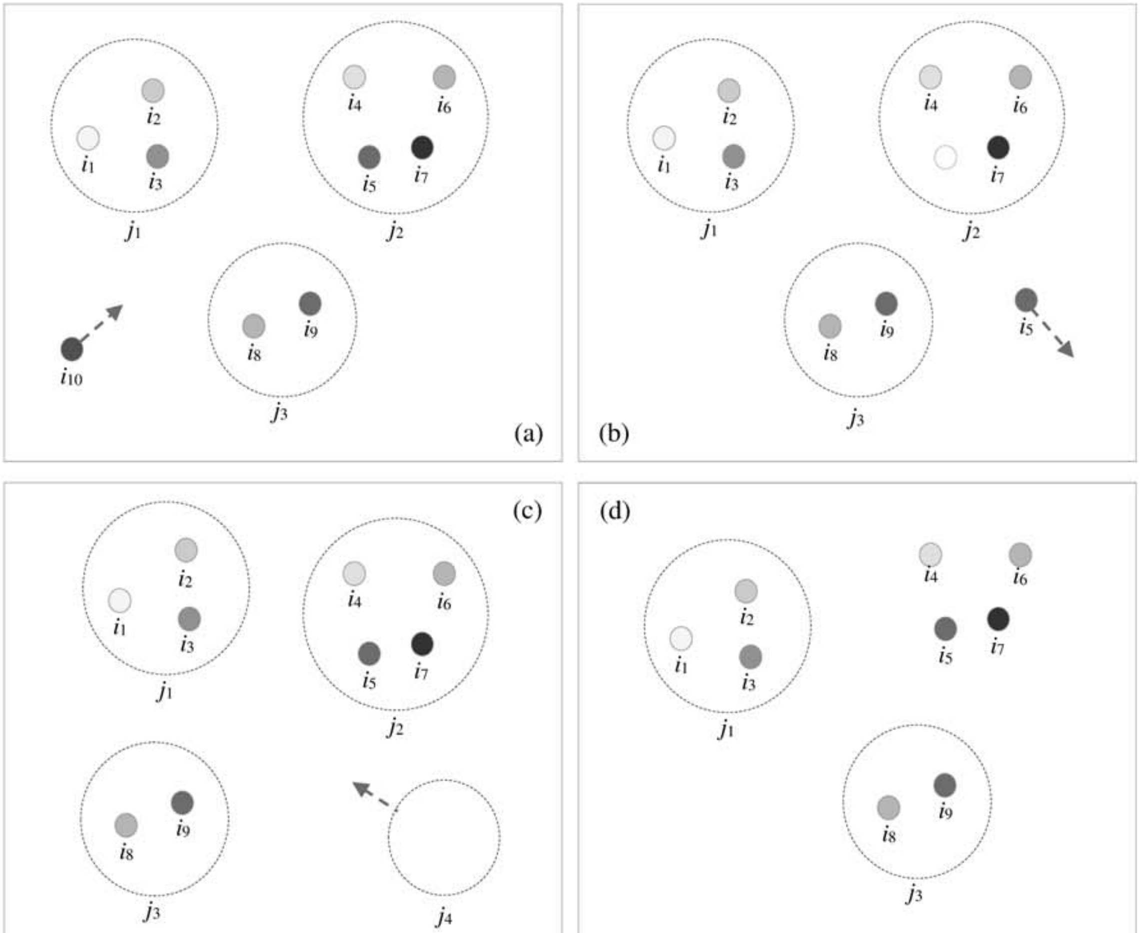


Figure 6.6. Evolution of a capacitated network of clusters. (a) Association of a new element to set I ($i_{10} \in I$); (b) Dissociation of an existing element from set I ($i_5 \in I$); (c) Association of a new element to set J ($j_4 \in J$); (d) Dissociation of an existing element from set J ($j_2 \in J$) [Arrow: Association/Dissociation].

take place through association of a new enterprise to the network (Figure 6.6.a), dissociation of an existing enterprise from the network (Figure 6.6.b), formation (Figure 6.6.c) or termination (Figure 6.6.d) of a coalition. In all cases, the entire network of individuals and clusters must adapt to the changes made through association/ dissociation of elements of either set I or set J in an optimal fashion. In the case of association, it is assumed that the information regarding the new individual (*i.e.*, original preferences and mutual influences) is known. In addition, only one association/dissociation is considered at a time. This assumption, however, is made for simplicity and the developed algorithms can be modified to handle simultaneous changes.

An EA is developed to enable optimal evolution and adaptation the networked clusters to changes. The EA follows the logic of the developed GA except for the initialization—after each single association/dissociation, the algorithm does not treat the problem as completely new. Instead, it *exploits* the good traits of the former network topology, which is assumed to be optimal, while *exploring* new solutions, in order to adapt to the respective change in a computationally efficient manner. Consider a network of capacitated clusters with optimal configuration, represented by chromosome $\mathbf{c}$. After any of the four types of changes (*i.e.*, association/dissociation of I's or J's) takes place, the EA incorporates that change in the original chromosome $\mathbf{c}$ through a set of heuristics, resulting in a *modified chromosome* $\mathbf{c^m}$. The modified chromosome then duplicates and reproduces itself through the path crossover and path mutation mechanisms until an initial population of chromosomes is generated. Afterwards, the first population of chromosomes evolves via the GA until the optimal configuration of networked clusters is obtained. This procedure is repeated after each single change occurs.

6.4.1 First Generation

The main assumption of the EA is that the original chromosome $\mathbf{c}$ (before any changes take place) represents the optimal configuration of the networked clusters. Hence, the modified chromosome must be generated in a way that upholds its optimality, *i.e.*, maintaining its good traits, while incorporating the new changes. A set of heuristics is developed to generate the modified chromosome under the four main types of evolution in the networked clusters:

1. *Association of I's*. After association of a new individual k to set I, *i.e.*, $I \leftarrow I \cup \{k\}$, a new *gene* is appended to the original chromosome $\mathbf{c}$. The new gene corresponds to k, and its allele l represents its match in set J, which satisfies the following condition in the modified chromosome $\mathbf{c^m}$:

   $$\hat{P}_{kl} = \max_{j \in L} \left\{ \hat{P}_{kj} \right\},$$

   where

   $$L = \left\{ j : M_j > \left| \left\{ i : \mathbf{c}(i) = j,\ i \in I \right\} \right|,\ j \in J \right\}.$$

   If $L = \varnothing$, set $\mathbf{c^m}(k) = 0$. The alleles of the other genes remain unchanged in the modified chromosome $\mathbf{c^m}$. The above procedure identifies the elements of set J with extra capacities, and then matches the new individual to the one with the maximum mutual preference score.

2. *Dissociation of I's*. After dissociation of an existing individual k from set I, *i.e.*, $I \leftarrow I \setminus \{k\}$, the gene of the corresponding individual is removed from the original chromosome $\mathbf{c}$, resulting in the modified chromosome $\mathbf{c^m}$.

3. *Association of J's*. After association of a new individual l to set J, *i.e.*, $J \leftarrow J \cup \{l\}$, the modified chromosome $\mathbf{c^m}$ is generated as follows:

   a) $\mathbf{c^m} = \mathbf{c}$. Sort the elements of set I in ascending order of $\hat{P}_{ij}$ and place them in the temporary set T.

b) Pick the first element of T and denote it by k.

c) If $\hat{P}_{kj} < \hat{P}_{kl}$, $j = \mathbf{c^m}(k)$, set $\mathbf{c^m}(k) = l$, and update $\hat{P}_{ij}$ is. Otherwise, go to Step 4.

d) $T \leftarrow T \setminus \{k\}$. If $M_l = \left|\{i : \mathbf{c^m}(i) = l,\ i \in I\}\right|$ or $T = \varnothing$ stop. Otherwise, return to Step (b).

The above procedure identifies the *least satisfied* elements of set I, *i.e.*, the ones with the lowest preference scores, and matches them, one by one, to l (the new element of Set J), in case this change increases their preference score. The procedure stops if l has no further capacity or there are no other elements in set I that prefer l to their current match in set J.

4. *Dissociation of J's*. After dissociation of an existing individual l from set J, *i.e.*, $J \leftarrow J \setminus \{l\}$, the modified chromosome $\mathbf{c^m}$ is generated as follows:

   a) $\mathbf{c^m} = \mathbf{c}$.

   b) Define sets L and T:

$$L = \left\{ j : M_j > \left|\{i : \mathbf{c^m}(i) = j,\ i \in I\}\right|,\ j \in J \setminus \{l\} \right\},$$

$$T = \left\{ i : \mathbf{c^m}(i) = l \right\}.$$

   If $L = \varnothing$ or $T = \varnothing$, go to Step (d). Otherwise, go to Step 3.

   c) Find $k \in T$ such that $\hat{P}_{kj} = \max_{i \in T,\ j \in L} \{\hat{P}_{ij}\}$, $\exists\ j \in L$, set $\mathbf{c^m}(k) = j$, and return to Step (b).

   d) If $T \neq \varnothing$, set $\mathbf{c^m}(i) = 0$, $\forall i \in T$.

After removal of l, the above procedure matches its corresponding elements in set I to other elements of set J with extra capacities (if any), in a way that the overall preference score is maximized (considering no other changes). In some cases, however, the number of elements of set I corresponding to l may be larger than the total available capacity of J's (after removal of l). In such cases, some elements of set I remain unmatched due to capacity shortage.

The above "adaptation heuristics" may not guarantee the optimality of the modified chromosome; however, they provide satisfactory quality. Each single change, however, may significantly influence the optimal topology of the networked clusters due to the interdependencies among preferences. Therefore, it may be difficult to directly identify the required modifications after each change takes place. The modified chromosome generated by the above heuristics must therefore be evolved to obtain the optimal configuration of the capacitated network of clusters. Optimization is handled by duplicating and reproducing the modified chromosomes through the reproduction operators of the GA, as described next.

6.4.2 Evolution

After the modified chromosome is generated, it duplicates and reproduces itself using the GA operators to generate new offsprings and eventually a population of chromosomes. The duplication and reproduction mechanism is similar to the *mitosis* process in cell division and duplication (Figure 6.7). Specifically, the modified chromosome first generates another chromosome through mutation. The two chromosomes then generate two new chromosomes through crossover. The four chromosomes then generate four new

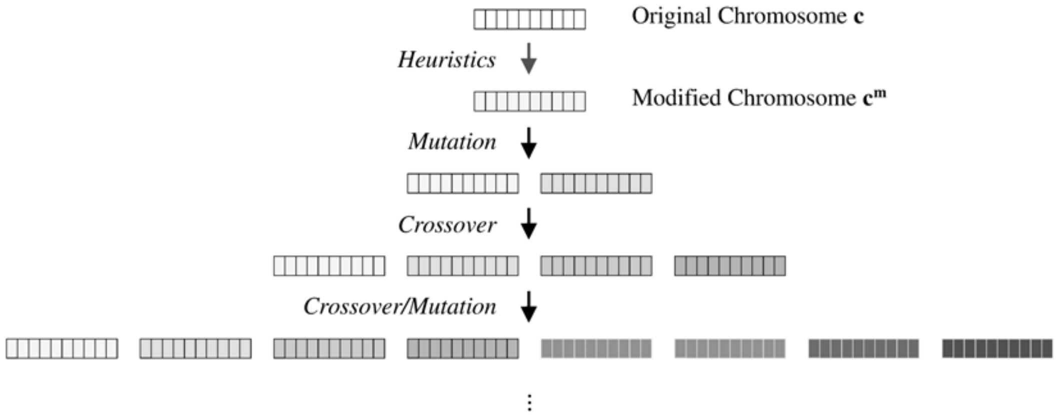


Figure 6.7. Mitosis-like duplication and reproduction of the modified chromosome and generation of the first population of chromosomes. In each reproduction step x, 2^x chromosomes are generated.

chromosomes, and the procedure continues until the number of generated chromosomes reaches a predefined population size. The evolution, from the modified chromosome toward the optimal one, is based on the path crossover and path mutation mechanisms, and the GA developed in the previous section. The EA is therefore composed of the following steps:

1. Define $C = \{\mathbf{c}^{\mathbf{m}}\}$ and PS as the set of chromosomes in the first population, and population size, respectively, and
   $L = \left\{ j : M_j > \left| \left\{ i : \mathbf{c}^{\mathbf{m}}(i) = j,\, i \in I \right\} \right|,\ j \in J \right\}$.
2. If $|L| > 0$, go to Step 3, otherwise, go to Step 4.
3.
    a) For each chromosome $\mathbf{c}$ in C do path mutation until an offspring chromosome $\mathbf{c}' \neq \mathbf{c}$ is generated. $C \leftarrow C \cup \{\mathbf{c}'\}$. If $|C| = PS$, go to Step 5. Otherwise, go to Step 3.b.

b) Randomly pair the chromosomes in C. For each pair $\mathbf{c}_1$ and $\mathbf{c}_2$, do path crossover until two offspring chromosomes $\mathbf{c}_1' \neq \mathbf{c}_2' \neq \mathbf{c}_1 \neq \mathbf{c}_2$ are generated. $C \leftarrow C \cup \{\mathbf{c}_1', \mathbf{c}_2'\}$. If $|C| = PS$, go to Step 5. Otherwise, return to Step 3.a.

4.

a) Perform swap mutation on $\mathbf{c^m}$ to generate another chromosome: $\mathbf{c}' = \mathbf{c^m}$, randomly select i, $k \in I$, $i \neq k$, and then $\mathbf{c}'(i) = \mathbf{c^m}(k)$ and $\mathbf{c}'(k) = \mathbf{c^m}(i)$. $C \leftarrow C \cup \{\mathbf{c}'\}$.

b) Randomly pair the chromosomes in C. For each pair $\mathbf{c}_1$ and $\mathbf{c}_2$, do path crossover until two offspring chromosomes $\mathbf{c}_1' \neq \mathbf{c}_2' \neq \mathbf{c}_1 \neq \mathbf{c}_2$ are generated. $C \leftarrow C \cup \{\mathbf{c}_1', \mathbf{c}_2'\}$. If $|C| = PS$, go to Step 5. Otherwise, return to Step 4.b.

5. Take C as the initial population, and do GA until the optimal configuration is obtained.

Steps 3 and 4 of the algorithm generate the first population of chromosomes. Step 3 deals with the situations where at least one element of set J has extra capacity, and applies path mutation and path crossover as reproduction mechanisms to guarantee the feasibility, diversity (through exploration of new solutions), and quality (through exploitation of good traits of parent chromosomes) of the generated chromosomes. Step 4, on the other hand, handles the situations where none of the element of set J has extra capacity. In such cases, the path mutation is not applicable. Therefore, the modified chromosome $\mathbf{c^m}$ first generates an offspring through a single swap mutation (Step 4.a), and then the rest of the initial population is generated through path crossover (Step 4.b). After generating the first population, the GA developed in Section 6.3 is applied to find the optimal configuration of the evolved networked clusters.

6.5 Numerical Experiments

A set of numerical experiments is conducted on a set of test-problems to analyze the impact of interdependencies between preferences on best matching, from both cluster formation and evolution perspectives. The numerical experiments are specifically intended to illustrate the role of IP in the formation of capacitated network of clusters, and the negative impact of disregarding them on the optimal configuration of clusters and the entire network. The experiments also analyze the sensitivity of cluster formation decisions to the intensity of mutual influences and level of interdependencies between preferences. Moreover, the experiments illustrate the way capacitated clusters evolve, and the impact of IP and capacity limits on their evolution. The following two scenarios are defined to perform the above analyses:

1. *Best matching without IP* (S_0). The same methodology is applied for cluster formation and evolution, where the influences of individuals on each other's preferences are ignored in the optimization process. That is, the optimization is performed without considering the actual values of α's, while they are ultimately incorporated in calculating fitness values.
2. *BMP-IP* (S_1).

In both scenarios, it is assumed that there are interdependencies between the preferences of the elements of set I for the elements of set J. S_0, however, disregards those interdependencies. Four test-problems TP (20, 8), TP (40, 18), TP (60, 25), and TP (80, 30) are generated, where the first and second entries respectively denote $|I|$ and $|J|$ (the proportion between $|I|$ and $|J|$ in the test-problems are selected arbitrarily). In each experiment on each test-problem, the preferences, the influences of I's the preferences of each other, and the capacities of the J's are generated randomly following uniform distribution. The preferences are uniformly distributed in [0, 1], where 0 and 1 indicate no and full preference, respectively. The influences of I's on each other's preferences are uniformly distributed in [–0.2, 0.2]. The capacity limits of J's are generated randomly, where the total capacity is uniformly distributed in $[|I|-0.1|I|, |I|+0.1|I|]$ (to ensure consistency and cover instances with limited and extra capacities). The reason behind using uniform distribution is to maintain the randomness of each independent experiment on each test-problem; that there is no a priori information on the exact values of the parameters.

6.5.1 Results

The first two goals of the numerical experiments are addressed in this section, *i.e.*, the impact of IP on cluster formation and the sensitivity of the results to the intensity of mutual influences between I's. Ten experiments are performed on each test-problem. All experiments are independent of each other in terms of preferences, influences of I's on each other's preferences, and capacities of J's. Table 6.1 shows the optimal solutions (chromosomes) for the last experiment on TP (20, 8), under S_0 and S_1. The positions (genes) and their respective values (alleles) correspond to the elements of sets I and J, respectively. The two scenarios yield different network configurations with a considerable gap between their optimal fitness values, as expected. Figure 6.8 shows the gap between the normalized optimal fitness values obtained through S_0 and S_1 for different test-problems. In all experiments, S_0 and S_1 yield completely different network configurations. (Details are omitted for the sake of brevity.) The results prove that if there are interdependencies between the preferences, disregarding them may lead to misleading and non-optimal decisions.

Table 6.1. Optimal chromosomes of S0 and S1; TP (20, 8).

S_0	2	3	4	6	8	5	3	2	2	8	2	7	5	2	8	3	2	1	2	2
S_1	3	3	4	6	8	2	1	5	2	8	2	7	5	3	2	2	2	2	2	8

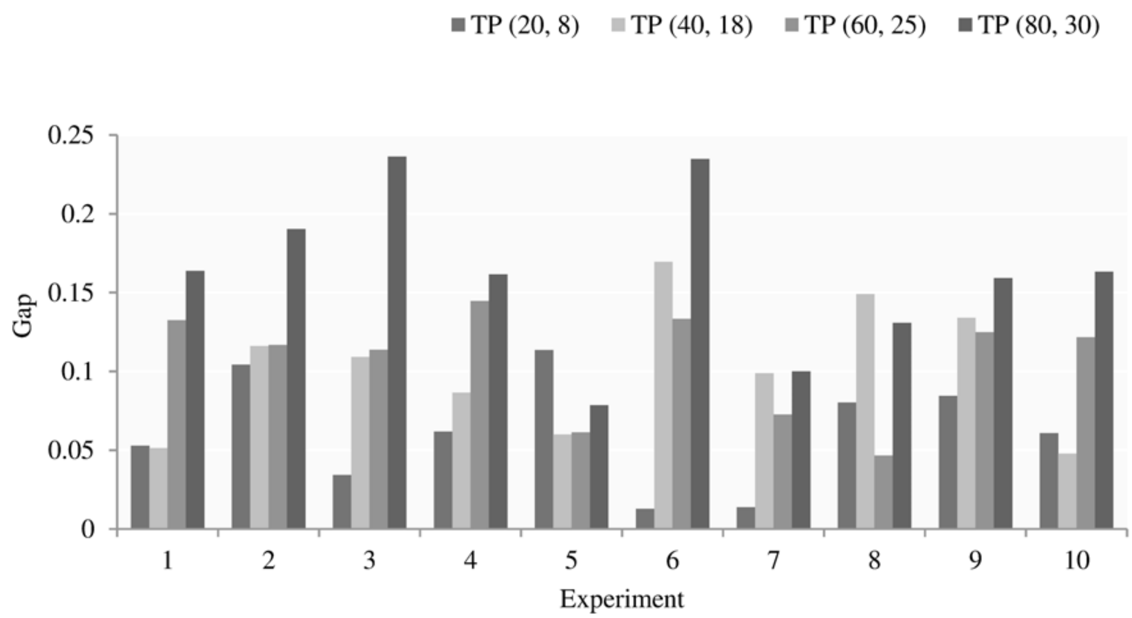


Figure 6.8. The gap between the optimal normalized fitness values of S_0 and S_1 [$\text{Gap} = \bar{F}_{S1} - \bar{F}_{S0}$].

A paired t-test is performed to analyze the significance of the gap between the optimal fitness values of S_0 and S_1. The test compares the means of the two treatments (*i.e.*, the fitness values of S_0 and S_1) based on the following hypotheses: H_0: $\overline{F}_{S_0} = \overline{F}_{S_1}$; H_1: $\overline{F}_{S_0} \neq \overline{F}_{S_1}$, where $\overline{F}$ denotes the fitness values. At the significance level of 0.05 and the freedom degree of 9, the paired t-test parameter is 2.26, which is lower than the t-test estimates obtained for different test-problems: 5.63, 7.77, 9.94, and 10.26 for TP (20, 8), TP (40, 18), TP (60, 25), and TP (80, 30), respectively. The null hypothesis is therefore rejected, which implies a significant gap between the fitness values of S_0 and S_1. Since the gap is positive in all cases, it is concluded that S1 significantly outperforms S_0, where the gap, according to the t-test estimates, is almost proportional to the size of the problem.

The gap between the results of S_0 and S_1 is rooted in the intensity of the influences of the I's on each other's preferences (*i.e.*, α's). The results shown in Figure 6.8 are based on the default values of α's (*i.e.*, [–0.2, 0.2]). To indicate the impact of α's on the gap between the results of S_0 and S_1, a sensitivity analysis is performed on each test-problem (one experiment on each), as shown in Figure 6.9. The results prove that higher mutual influences of I's on each other's preferences, positive or negative, lead to more dramatic changes in the optimal cluster formation, and disregarding such influences becomes substantially more detrimental to the optimality of matching.

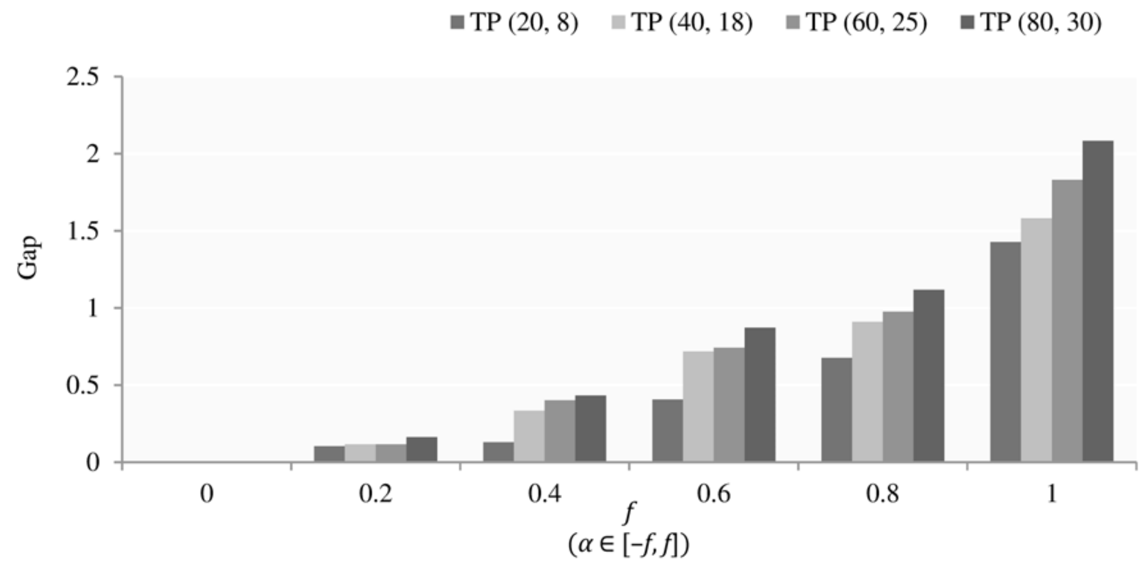
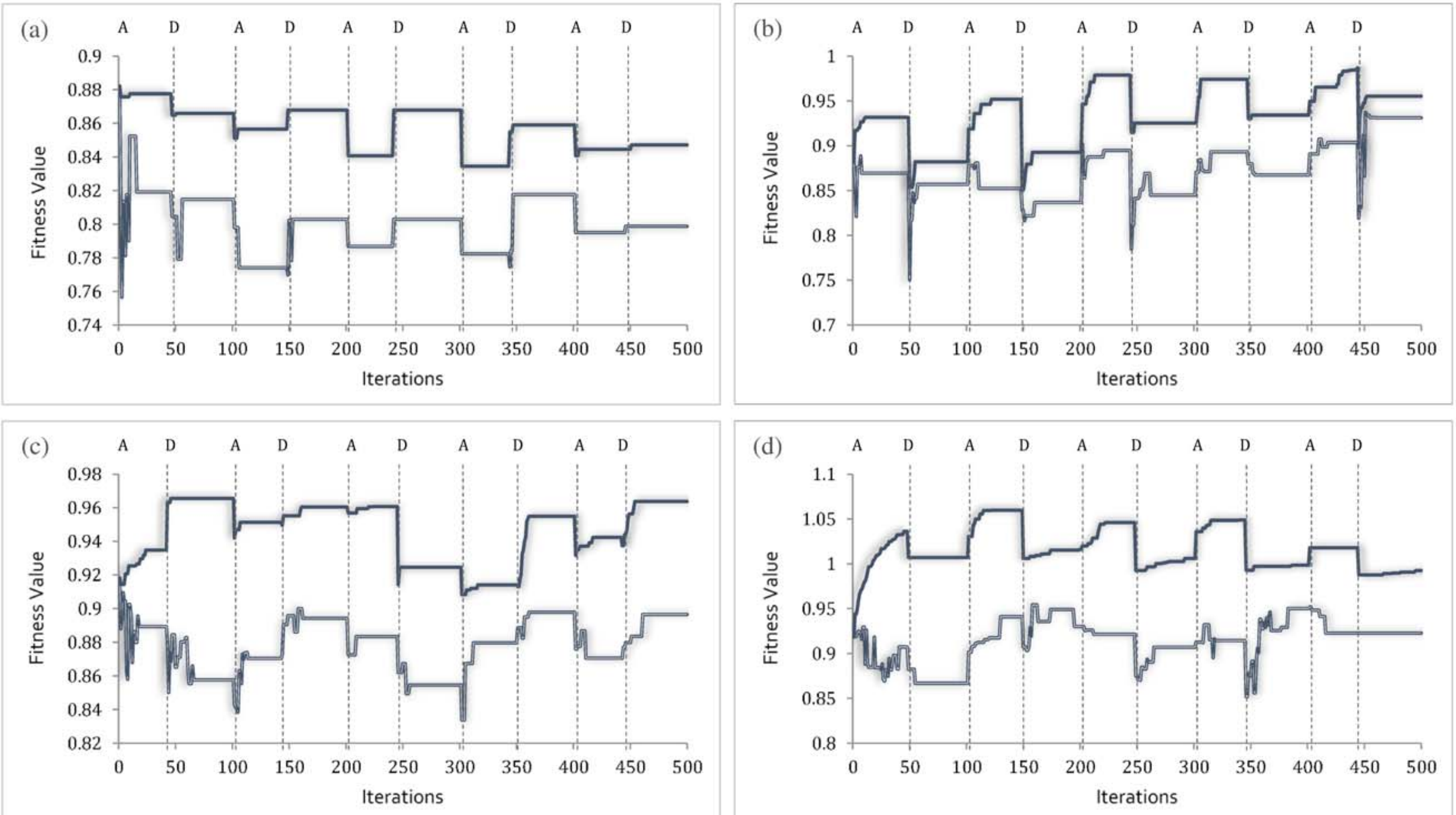


Figure 6.9. Sensitivity of the gaps between the optimal normalized fitness values of S_0 and S_1 to α's.

The optimal network configuration obtained from the last experiment through S_1 is selected as the initial state. Then, ten changes, five associations and five dissociations, are simulated and made in a cyclic fashion, *i.e.*, a new element associates to the network, then an existing element dissociates from the network, and so on. The changes are made in sets I and J in separate and independent experiments. All the input parameters corresponding to the associating individuals (*e.g.*, preferences, capacities) are generated in the same manner as described in Section 4.1. In all experiments, the default values of α's are used, *i.e.*, uniformly distributed in [–0.2, 0.2]. Figure 6.10 illustrate the evolution of the capacitated networks of clusters associated with TP (20, 8), TP (40, 18), TP (60, 25), and TP (80, 30), under S_0 and S_1. Note that the EA is executed once, simultaneously under S_0 and S_1 and on the same problem setting.



6.10. Evolution of capacitated clusters through association (A)/dissociation (D) of I's (a, c) and J's (b, d) in two instances of M:1 / RC, IP / +, OS / ES; (a, b): SZ (20, 8); (c, d): SZ (40, 18), where SZ ($|I|$, $|J|$). High impact of IP on the overall satisfaction and the possibility of "erroneous evolution", if disregarded (S_0); Rapid adaptation to changes via the adaptation heuristics and genetic algorithm; Lower overall impact of single changes as the system's size grows. (S_0: —; S_1: —)

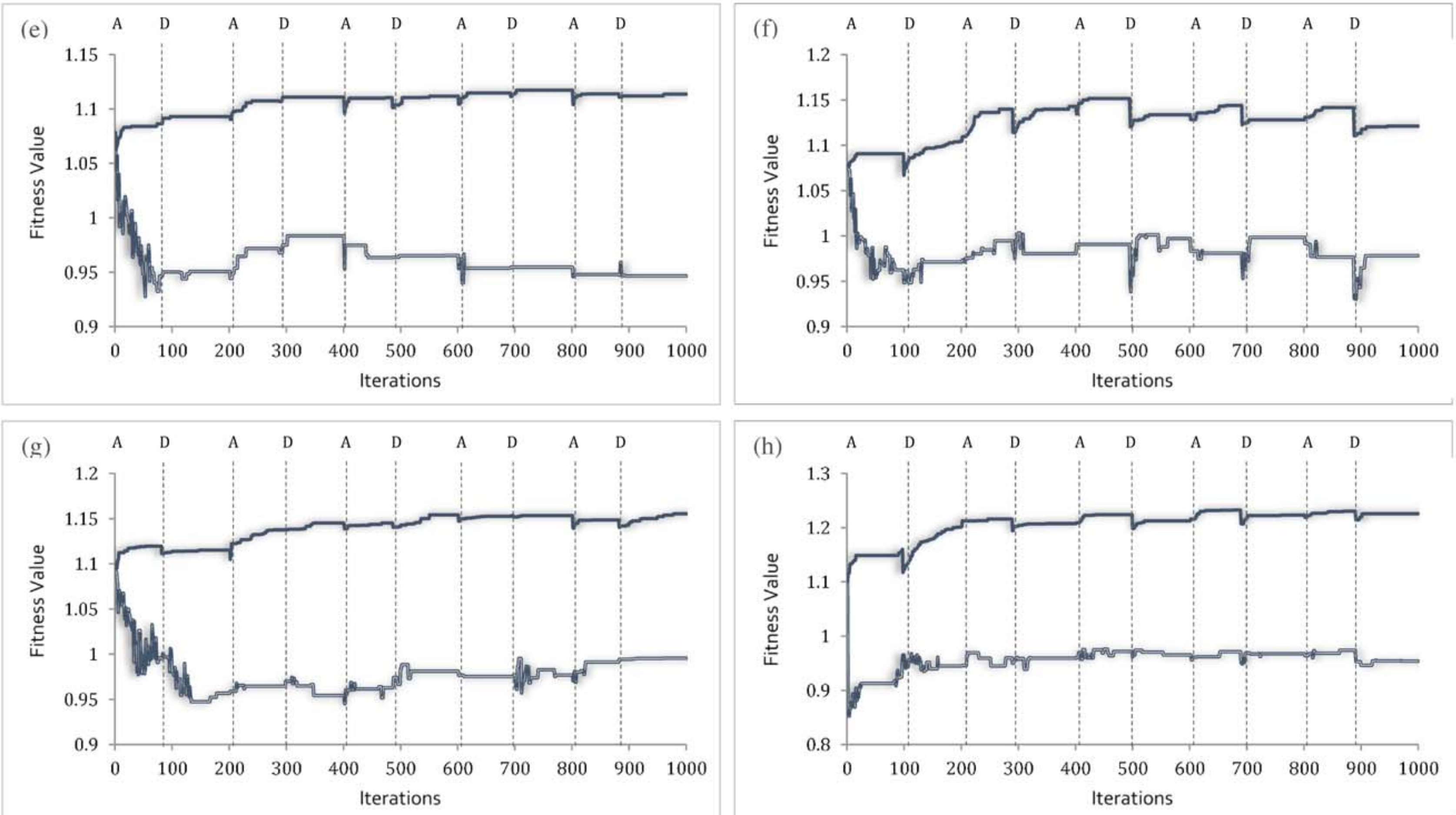


Figure 6.10. (*Continued*) Evolution of capacitated clusters through association (A)/dissociation (D) of *I*'s (e, g) and *J*'s (f, h) in two instances of *M*:1 / *RC*, *IP* / +, *OS* / *ES*; (a, b): *SZ* (60, 25); (c, d): *SZ* (80, 30). High impact of IP on the overall satisfaction and the possibility of "erroneous evolution", if disregarded (S_0); Rapid adaptation to changes via the adaptation heuristics and genetic algorithm; Lower overall impact of single changes as the system's size grows. (S_0: —; S_1: —)

6.5.2 Discussion

In most cases, association/dissociation of an element to/from set I decreases/increases the fitness value (see Figure 6.10; a, c, e, g). Set J shows an opposite behavior (Figure 6.10; b, d, f, h). The reason lies in the capacity limitations of J's, which may prevent some I's from being matched to their more preferable element in set J. Hence, associating an additional element to set I makes the capacity limits even more restrict, while association of a new element to set J provides additional capacities and more possibilities, and thus alleviates such limitations. The same interpretation can also be valid for the dissociation of individuals.

In spite of the above analyses, there are some cases with completely opposite behaviors, where, for example, dissociation of an element from set I decreases the overall fitness value (see, *e.g.*, Figure 6.10.c, Iteration ~250). Such behaviors may be due to the interdependencies between preferences, rather than the capacity limits. For instance, dissociation of an element from set I with high positive α values (*i.e.*, an element with highly positive influences on the preferences of the others) can considerably diminish the overall fitness value.

The "noisy" and "suboptimal" behavior of S_0 is due to disregarding IP (Recall that the main assumption is that there are interdependencies between the preferences). That is, under S_0, the clusters evolve without considering the actual values of the α's. Nevertheless, the values shown in Figure 6.10 are the "actual" fitness values of the "erroneously evolved" networks of clusters (considering the actual values of α's). As shown in Figure 6.10, disregarding such interdependencies also leads to sub-optimal evolutions, where the gap between the fitness values of the sub-optimal and optimal configurations is almost proportional to the size of the problem (see also Figure 6.9).

The developed adaptation heuristics for generating the modified chromosome and the EA for generating the initial population effectively handle the evolutionary behaviors in terms of association or dissociation. As shown in Figure 6.10, the first generations of chromosomes (*i.e.*, iterations) generated after each single change provide acceptable fitness values. Besides, the EA enables the clusters as well as the entire network to adapt, recover, and return to the optimal/near optimal state in a *timely efficient* manner, after a considerably small number of iterations. The computational efficiency of the developed algorithms is investigated next.

6.5.3 Computational Efficiency

The *scalability* of the developed algorithms in terms of variations in the computational time for different combinations of $|I|$ and $|J|$ values is analyzed here. Since the experiments are based on randomly generated parameters, the time required for collecting and cleaning input data is not included in the overall computational time. In some real-life applications, however, the data may be incomplete, fuzzy, or subjective, and the parameter values may need to be defined based on expert knowledge. For example, the actual gains or losses of an enterprise from joining, remaining in, or leaving a coalition is a function of various factors and dynamics such as market behavior, lifecycle of products, and political interactions with certain collaborators/competitors, and their calculation require compilation of various qualitative data such as surveys, interviews, and expert judgements into quantifiable preference scores. In such cases, however, the time required for preparing the input parameters grows polynomially with the problem size. That is, considering $T(|I|, |J|)$ as the time required to prepare the input parameters for BMP-IP between two sets I and J, it can be shown that $T(|I|, |J|) = O(|I|^2 + |I|\times|J|)$, which is clearly polynomial in $I\times J$. In experiments on actual data, the data must be cleaned in a way that provides the following necessary and sufficient data-points to the models: mutual preferences; influences of individuals on each other's preferences; capacities of clusters.

Figure 6.11 shows the growth rate of the computational time of the GA as a function of different values of $|I|$ and $|J|$. The experiments are performed by changing the value of $|I|$ for different values of $|J|$, and the recorded times represent the time laps from starting to run the algorithm until the stopping criterion is reached. The values represent the (approximate) average running time of the GA program in MATLAB on an Intel® Core™ i7 processor. The obtained experimental curves indicate a polynomial growth in the computational time, which implies feasibility of solving large-sized instances using the

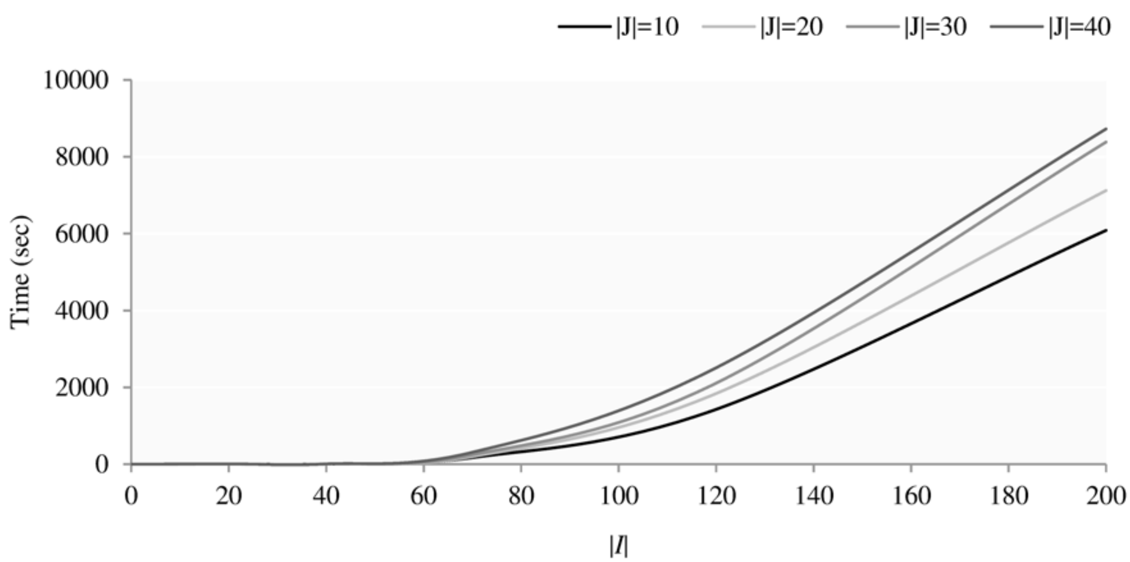


Figure 6.11. The experimental (approximate) curve of the polynomial growth of the GA's computational time as a function of $|I|$ and $|J|$.

developed algorithms. The computational efficiency of the GA represents that of the EA, since the EA uses the same mechanism except for the mitosis-like duplication and reproduction of the modified chromosome for generating the first population.

6.6 Concluding Remarks

In this case study, the BMP-IP is defined and analyzed, as a new instance of generalized matching problem where the mutual influences on and interdependencies between preferences are incorporated as determinant decisional factors. A QAP formulation is developed to mathematically formulate the BMP-IP. A set of evolutionary algorithms is developed to effectively handle the complexity of the problem with relatively polynomial growth in the computational time as a function of problem size, and enable self-adaptation and self-evolution of networked capacitated clusters. It is shown through several experiments that IP along with capacity limits dramatically influence generalized best matching decisions—the gaps between the fitness values of S_0 and S_1 in optimal cluster formation, association/dissociation of individuals and clusters are ~5%-15%, ~7%-11%, and ~15%-26%, respectively.

The observations of this work indicate that, given interdependencies between preferences, the developed methodology substantially improves the performance of any networked system with analogous features and structures—from homogenous teams of humans (*e.g.*, pilots; soldiers; students; workers; technicians; doctors; interns; roommates) to heterogeneous teams of humans and/or machines, with applications in production, manufacturing, supply, logistics, healthcare, and transportation. The main idea is that IP, driven by interpersonal emotions among humans, technical/technological specifications and affinity attributes of machines/components, or both, is an influential and inevitable characteristic of any generalized matching problem.

The results obtained through our experiments indicate that the BMP-IP, besides all of its known inherent complexities and dynamics, is sensitive to the mutual *influences* between every single pair of individuals. Although this phenomenon is more common in social networks compared to other types of systems, similar behaviors are expected from teams of artificial entities (*e.g.*, sensors, robots, enterprises), based on factors analogous to emotion in humans (*e.g.*, operational compatibility; task requirements). For example, *product affinity* is a practical example of IP in storage allocation (Li *et al.*, 2015), where products with high affinity (*e.g.*, being ordered together) are located in the same/closest possible aisles of the warehouse so that the material handling equipment take shorter routes while placing or picking up products[1].

[1] Note that this is a 1:1 matching problem between products and storage locations. See Li *et al.* (2015) for detailed description and formulation of affinity-based storage allocation.

CHAPTER 7. CASE 4—COLLABORATIVE SERVICE ENTERPRISES☆

An organization can be described as a system, or a system of systems, of homogeneous/ heterogeneous resources deployed for processing and synthesizing a set of tasks, in order to accomplish a set of goals (Ko and Nof, 2012). Examples include a procurement system with a network of order, product, and resource agents; an intelligent warehouse system with a network of sensors/RFID (Radio Frequency Identification) tags, readers, and antennas; a virtual factory composed of a set of reference models, decoupled software tools, shop floor devices, communication and computational elements, middleware, knowledge bases, servers; a manufacturing network with a set of suppliers, distributers, and manufacturing equipment. In line with the growing complexity and dynamicity of markets, businesses, and processes, and the emerging needs for higher flexibility, scalability, and resilience (Putnik *et al*., 2013), traditional organizations have gradually transformed from monolithic and self-reliant systems into highly-distributed and interconnected networks. In such emerging networks, the key to sustain and evolve is to engineer, improve, systematize, and automate *collaboration* among distributed entities (Nof *et al*., 2015; Moghaddam and Nof, 2015[d]). In this context, a Collaborative Networked Organization (CNO) refers to a

☆ The preliminary versions of this case study was presented at the *Industrial and Systems Engineering Research Conference*, Montréal, Canada, 2014, and the *INFORMS Annual Meeting*, Philadelphia, USA, 2015. The materials presented in this case study are adapted from two works of the author published in the *International Journal of Production Research* (DOI: 10.1080/00207543.2015.1125544) and under review in the *Computers and Industrial Engineering* journal.

network of autonomous, distributed, and heterogeneous entities (organizations and people) that collaborate through sharing information, resources, and responsibilities to achieve common/compatible goals (Nof, 2007; Camarinha-Matos *et al.*, 2009).

7.1 Motivation

Advances in collaborative e-Work theories and technologies have facilitated computer-supported and communication-enabled collaboration among geographically dispersed organizations, regarding *e-Activities* such as e-Business, e-Commerce, e-Logistics, and e-Manufacturing. A CNO, besides its *physical dimension*, represents a *virtual* (v-)Organization as well; a distributed network of independent organizations that collaborate to achieve a set of individual and common goals (Camarinha-Matos *et al.*, 2009). The virtual dimension of the CNO, *i.e.*, the v-Organization, enables collaboration among physical networked organizations, from minimal file exchange to direct access to resources such as computers, machines, software, and databases (Foster *et al.*, 2001). Specifically, while some tasks require physical proximity to resources (*e.g.*, sensing by distributed sensors; environmental monitoring or rescue by mobile robots; cross-docking in a distribution network), some other tasks can be processed remotely through information and communication channels (*e.g.*, remote control of tele-robots; virtual manufacturing; remote modeling/simulation/problem-solving). In this case study, the tasks in the former and the latter classes are respectively referred to as physical (p-)Tasks and electronic (e-)Tasks. Table 7.1 summarizes examples of location-allocation decisions in various types of CNO, from micro-scale sensor networks to extended enterprises.

Table 7.1. Examples of collaborative location-allocation decisions in various cyber-supported CNO.

CNO	Organizations	Resource sharing	e-Task sharing	Example refs.
Wireless sensor networks	Sensor clusters	Dynamic clustering of sensor nodes, and/or reconfiguration of sensor network	Collaborative info transmission to base-station under certain communication protocols	Ko *et al.* (2010); Kulkarni *et al.* (2011)
Swarm robotics	Teams of robots	Dynamic formation of teams of mobile robots; Dynamic sharing of data, code, memory, computational tools, *etc.*	Dynamic (re-)allocation of tasks (*e.g.*, sensing; picking/ placing; carrying; rescuing; cleaning) under unforeseen situations	Grabowski *et al.* (2000); Nouyan *et al.* (2009)
Collaborative R&D Institutes	Universities; Laboratories	Dynamic sharing of equipment, facilities, models, platforms, researchers, *etc.*	Collaborative design, programming, decision-making, problem solving	Cummings and Kiesler (2005)
e-Manufacturing networks	Manufacturing sites; Suppliers; Distributers	Dynamic allocation of decoupled software tools, shop floor agents, knowledge bases, servers, communication elements, *etc.*	Collaborative process planning/scheduling, and factory design; Remote monitoring and control of agents, robots, machines, distribution networks	Nof (2007); Shen *et al.* (2007); Camarinha-Matos *et al.* (2009)
Multinational corporations	Regional headquarters; Subsidiaries	Dynamic allocation of employees, departments, authorities, planning, marketing, finance, IT resources, *etc.*	Remote meeting/ conferences; Collaborative planning, scenario evaluation, analysis, visualization, and crisis management	Kostova *et al.* (2008); Singh (2012)

This case study rethinks the design of CNO by incorporating both the physical and virtual dimensions of interaction and collaboration in the decisions. Without loss generality, CNO is abstracted as a network of organizations, each with a certain set of heterogeneous *resources* used for processing a set of *tasks* (p-Tasks and e-Tasks). Recently, various classes of the CNO have been inspired by the notion of cloud computing and its extension, from sharing merely computing resources (Mell and Grance, 2009) to other types of resources on the cloud, with applications in areas such as manufacturing (Xu, 2012), education (Sultan, 2010), and healthcare (Hood *et al.*, 2012). Due to certain physical limitations associated with the p-Tasks and some resources, collaboration is enabled and augmented, in both physical and virtual dimensions, through two distinct but interrelated types of decisions, as follows (Figure 7.1):

1. *Allocation of tasks* (virtual dimension). Each organization owns a certain set of resources and is assigned a certain set of tasks to process. Due to the dynamic variations in demand and capacity, organizations may encounter capacity shortage/surplus over time. Given the "e" and "p" classifications of tasks, organizations can collaborate by sharing their e-Tasks in the case of insufficiency of local resources. Collaborative processing of e-Tasks balances the overall workload of the network, and minimizes idle resources, congestions, and delays.
2. *Location of resources* (physical dimension). Unbalanced distribution of resources among organizations may lead to considerable inefficiency and nullify the positive impacts of e-Task sharing. For instance, some organizations may lack enough resources for processing their assigned p-Tasks, which require physical proximity

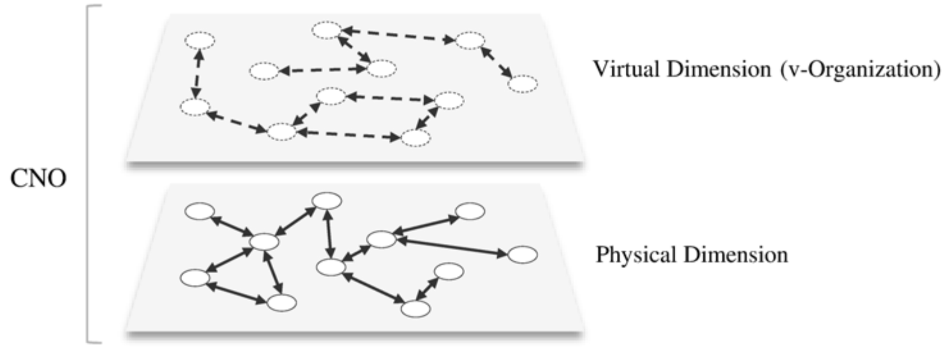


Figure 7.1. Collaboration in physical (location of resources) and virtual (allocation of tasks) dimensions of a CNO.

to the respective resources. Moreover, high rates of e-Task sharing among organizations, due to lack of local processing resources, leads to higher communication network load, congestion, delays, and cost, and increases the possibilities of errors and conflicts. Such problems can be avoided by enabling the organizations to dynamically share their (shareable) resources.

7.2 Outline

The problem under study is defined as Collaborative Location-Allocation Problem (CLAP) with two conflicting objectives: (1) Maximizing tasks fulfillment rate; (2) Minimizing (unnecessary) collaboration rate. The objectives are achieved by concurrent best matching between the sets of tasks and resources, and the sets of resources and organizations (see Cases 1 and 2). For mathematical representation of the CLAP, a Bi-Objective Mixed-Integer Programming (BOMIP) formulation is developed, which can also be used for solving small-sized instances. Due to the computational complexity of the CLAP (Owen and Daskin, 1998), a novel tabu search algorithm is developed where the neighborhood search mechanism follows a novel best matching heuristic for neighborhood search inspired by the natural *justice rule* (henceforth, the algorithm is called TS-Jr.). The TS-Jr. algorithm enables optimal (re)configuration of CNO (Chituc and Nof, 2007), given the (potential) dynamic changes in the characteristics of the tasks and/or resources along with the topology of the network. Several experiments are conducted to illustrate the impact of collaboration in CNO through CLAP on task fulfillment, utilization of resources, and stability of the entire network, as well as the algorithmic efficiency and quality of TS-Jr. It is shown that the achieved improvements are in line with the emerging requirements for flexibility, scalability, and resilience of CNO. The CLAP, according to the PRISM taxonomy of best matching, is

- A *three-sided* best matching problem between the sets of organizations O, resources R, and tasks T, where typically $|O| \leq |R| \leq |T|$. The pairwise relations between the sets, *i.e.*, O:R, O:T, and R:T, are *one-to-many*, *one-to-many*, and *many-to-many*, in that order (Figure 7.2; D1), which imply that each organization can be assigned multiple resources and tasks, each resource can process multiple tasks, and each task can be processed by multiple resources.
- A *resource-constrained* best matching problem (limited resources for processing the tasks), where *collaboration* is enabled among organizations through sharing resources and e-Tasks (Figure 7.2; D2).
- A *multi-criteria* best matching problem (Figure 7.2; D3)—the objectives are to maximize task fulfillment, and minimize collaboration rate, which in turn lead to higher utilization of resources and stability of CNO.

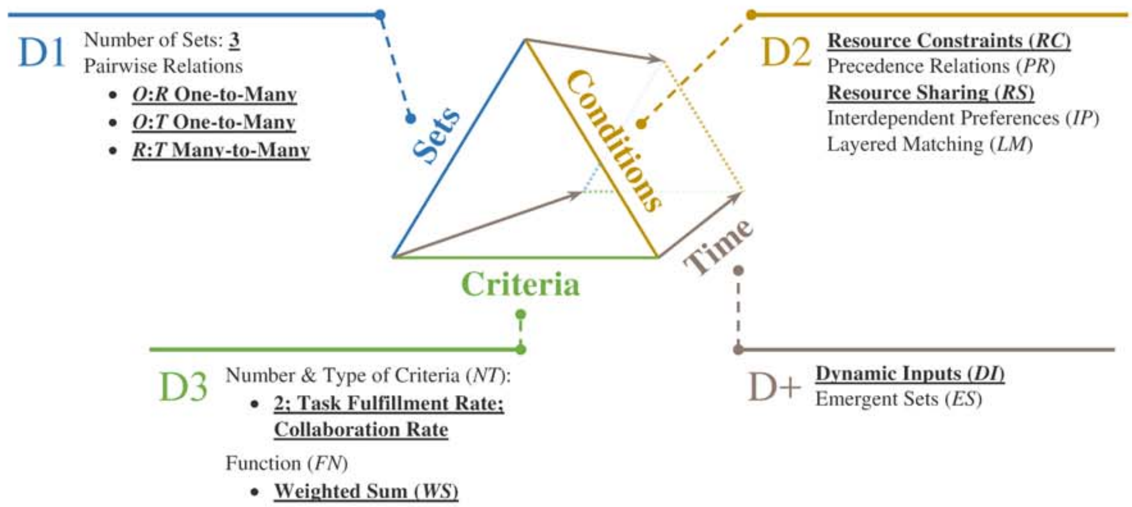


Figure 7.2. Classification of the CLAP according to the PRISM taxonomy of best matching.

- A *dynamic* best matching problem where the characteristics of the system and its components (*e.g.*, tasks requirements; resources configuration; association/ dissociation of tasks and/or resources) may vary over time (Figure 7.2; D+).

7.3 Mathematical Formulation

Let $R_t \subseteq R$ denote the set of resources eligible for processing task $t \in T$, L_r denote the capacity limit of resource $r \in R$, C_{rt} denote the capacity required by task $t \in T$ if processed by resource $r \in R_t$, T^p and T^e respectively denote the sets of p-Tasks and e-Tasks ($|T^p| + |T^e| = |T|$), and $T_o \subseteq T$ denote the set of tasks associated with organization $o \in O$. It is assumed that all tasks, resources, and their respective characteristics are known. It is also assumed that there are no precedence relations between the tasks. The problem is then to find the best three-sided match between the elements of sets O, R, and T with respect to two main criteria; task fulfillment rate and collaboration rate.

A BOMIP formulation is developed for mathematical representation of the CLAP. A three-index formulation is applied, which represent best matching between the sets *O*, *R*, and *T*. The binary variables χ_{ort} are defined, where $\chi_{ort} = 1$, if task *t* is allocated to resource *r*, and resource *r* is located in organization *o*, and $\chi_{ort} = 0$, otherwise. The three-index formulation enhances the representation of the dependencies between the location and allocation decisions. In the following, the task fulfillment rate and collaboration rate objectives are presented, along with the constraints of the BOMIP model[1]:

1. *Objective 1: Maximize task fulfillment rate*.

$$\max\ Z_1 = \sum_{r \in R_t} \sum_{t \in T} \alpha_{rt}. \tag{7.1}$$

In the first objective function, α_{rt} denotes the amounts of resources $r \in R_t$ consumed by task $t \in T$. Accordingly, the fulfillment rate (FR) of each individual task can be estimated using the following formula:

$$FR_t = \sum_{r \in R_t} \left(\frac{\alpha^*_{rt}}{C_{rt}} \right), \qquad \forall\, t \in T, \tag{7.2}$$

where superscript * denotes optimality.

2. *Objective 2: Minimize collaboration rate.*

$$\min\ Z_2 = \sum_{o \in O} \sum_{r \in R} \sum_{t \in T \setminus \{T_o\}} \chi_{ort} \alpha_{rt}. \tag{7.3}$$

The second objective function minimizes the overall amount of task sharing—the usage of external resources associated with other organizations for processing a task. Naturally, the value of Objective Function (7.3) is zero for all p-Tasks. This

---

[1] This model is an extension of Model (M3.3), *M*:1 / *RC*, *RS* / +, *OS*, presented in Chapter 3

requirement is addressed by imposing proper constraints on α_{rt}, as indicated in the following subsections. Given Objective Function (7.3), the collaboration rate (CR) can be estimated using the following formula:

$$CR = \frac{Z_2^*}{\sum_{r \in R} \sum_{t \in T} \alpha_{rt}^*}, \tag{7.4}$$

where superscript * denotes optimality. Note that, in this context, CR represents the ratio of the total *collaboratively* processed (consumed) tasks (resources) to the overall processing (consumption) of tasks (resources).

3. *Constraints 1: Resource limits.*

$$\sum_{t \in T} \alpha_{rt} \chi_{ort} \leq L_r, \qquad \forall\, o \in O,\ r \in R. \tag{7.5}$$

Constraints (7.5) ensure that the capacity limits of individual resources are not violated—the total capacity used by different tasks (including p-Tasks along with the original and shared e-Tasks) allocated to each resource must be less than or equal to its limit.

4. *Constraints 2: Task fulfillment.*

$$FR_t \leq 1, \qquad \forall\, t \in T. \tag{7.6}$$

Constraints (7.6) ensure that the fulfillment rate of each task (see Eq. (7.2)) is bounded to one—the sum of the relative amounts of resources used by each task must not exceed one.

5. *Constraints 3: Allocation of tasks.*

$$\alpha_{rt} \leq C_{rt} \sum_{o \in O} \chi_{ort}, \qquad \forall\, r \in R,\ t \in T. \tag{7.7}$$

$$\chi_{ort} = 0, \qquad \forall o \in O,\ r \in R,\ t \in T^{p},\ t \notin T_{o}, \quad \text{or} \quad \forall o \in O,\ r \notin R_{t},\ t \in T. \qquad (7.8)$$

Constraints (7.7) ensure that if task t is not assigned to resource r at any organization, the respective consumption value, *i.e.*, α_{rt} must be zero. Otherwise, that value is limited to C_{rt}, *i.e.*, the capacity required by task t if processed by resource r. Constraints (7.8) ensure that (i) all p-Tasks can only be processed by their respective organization (are not shareable), and (ii) tasks (including all e-Tasks and p-Tasks) are not assigned to *ineligible* resources for processing.

6. *Constraints 4: Location of resources.*

$$\sum_{\substack{o' \in O \\ o' \neq o}} \sum_{t' \in T} \chi_{o'rt'} \leq \zeta\left(1 - \chi_{ort}\right), \qquad \forall o \in O,\ r \in R,\ t \in T. \qquad (7.9)$$

Constraints (7.9) satisfy the condition that each individual resource cannot be *physically* located in multiple organizations (ζ represents a sufficiently large positive number).

7. *Constraints 5: Feasibility of decision variables.*

$$\alpha_{rt} \geq 0,\ \ \chi_{ort} \in \{0, 1\}, \qquad \forall o \in O,\ r \in R,\ t \in T. \qquad (7.10)$$

Constraints (7.10) ensure that the resource consumption variables are nonnegative, and the location-allocation variables are binary.

7.4 Optimization: TS-Jr.

The CLAP is a multidimensional (three-sided) generalized best matching problem, known as an NP-hard problem (Owen and Daskin, 1998). For solving the generalized best matching problem, various exact (*e.g.*, Savelsbergh, 1997; De Farias *et al.*, 2000; Haddadi and Ouzia, 2004), approximation (*e.g.*, Cohen *et al.*, 2006; Jeet and Kutanoglu, 2007), relaxation (*e.g.*, Benders and van Nunnen, 1983; Lorena and Narciso, 1996; Yagiura *et al.*, 1999), and metaheuristic (*e.g.*, Osman, 1995; Lorena *et al.*, 1999; Lourenc and Serra, 2002; Yagiura *et al.*, 2004[a]) algorithms have been proposed in literature. Among all, a good candidate in terms of both solution quality and computational efficiency is tabu search (Yagiura *et al.*, 2004[b]), which is also known as a powerful metaheuristic for solving multidimensional best matching problems (Laguna *et al.*, 1995).

Tabu search, a memory-based metaheuristic introduced by Fred W. Glover in 1986, is based on neighborhood search with *prohibition* strategies, which mark previously visited solutions (temporarily or permanently) as forbidden or *tabu* in order to prevent cycling and improve the efficiency of search mechanism (see CHAPTER 2). The algorithm starts with an initial solution *S*, and repeatedly *explores* through its neighborhood, while *exploiting* the knowledge of tabu points/areas, until a stopping criterion is reached. The moves are directed by various mechanisms, among which *shift*, *swap*, and *ejection chains* (Glover, 1996) are the most common. For the multidimensional best matching problem, specifically, Laguna *et al.* (1995) recommend simple, double, and circular ejection chains as efficient mechanisms for movement in the neighborhood. Besides the movement mechanism, however, a critical step in tabu search is to find the *best move* in each iteration.

In the CLAP, due to the interdependencies among task sharing and best matching decisions, identification of the best move may not be possible via simple infeasibility or cost-improvement measures (Laguna *et al.*, 1995). Thus, a novel TS-Jr. algorithm is developed, which is composed of two main stages as described next: (1) Generating a feasible, high-quality initial solution S; (2) Identifying and applying the best next move to generate an updated solution S'; and calculating/updating the *tabu time* for previously visited solutions.

7.4.1 Initialization: The Rock and Sand Heuristic

The performance of tabu search depends highly on the initialization procedure, the *feasibility* and *quality* of the initial solution S. In the CLAP, feasibility implies no violation of Constraints 1 to 5. Hence, an encoding scheme is applied (Figure 7.3; matrix **M**), which guarantees satisfaction of Constraints 3 (allocation of tasks), 4 (location of resources), and 5 (binary best matching variables). In addition, a heuristic procedure—inspired by the *rock*

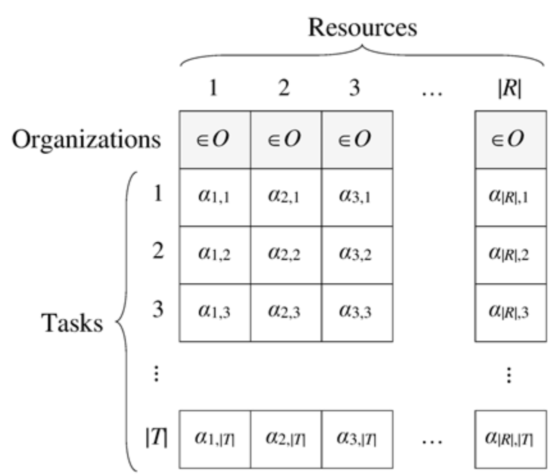


Figure 7.3. Encoding matrix **M**; *e-Tasks*: Multiple columns can take positive values, representing their allocation to the respective resource and organization; *p-Tasks*: Only the columns with their first row equal to their respective organization can take positive values; *Resources*: The entries of the first row represent their respective organization.

and sand analogy in time management—is developed, which besides satisfaction of the rest of the constraints (*i.e.*, nonnegative task fulfillment; resource limits), provides acceptable quality for the initial solution *S*. The idea of the rock and sand initialization heuristic is to allocate larger resources (≡ rocks) first to the organizations with higher demand, and then allocate smaller resources (≡ sands) to enhance the balance between resources and demand, and thus, reduce the need for task sharing (*i.e.*, the CR objective). Allocation of resources to different tasks is then performed in a way that maximizes task fulfillment rates (*i.e.*, the FR objective), while avoiding violations from the resource limits. The rock and sand initialization heuristic is as follows:

0. Generate a $(|T|+1)\times|R|$ zero matrix $\mathbf{M}$.

Location of Resources

1. For all organizations $o \in O$, estimate (average) the overall resource requirement:

$$ER_o = \sum_{t\in T_o}\sum_{r\in R_t}\left(\frac{C_{rt}}{|R_t|}\right), \qquad \forall\, o \in O.$$

2. Locate the largest resource in the organization with maximum estimated overall resource requirement (break ties arbitrarily):

$$\mathbf{M}[1,\hat{r}] = \hat{o}, \qquad L_{\hat{r}} = \max_{r\in R}\{L_r\};\ ER_{\hat{o}} = \max_{o\in O}\{ER_o\}.$$

3. Set $ER_{\hat{o}} \leftarrow ER_{\hat{o}} - L_{\hat{r}}$ and $R \leftarrow R\setminus\{\hat{r}\}$. If $R = \varnothing$, go to Step 4; otherwise, return to Step 2.

Allocation of Tasks

4. Given the location decisions, calculate the average resource requirement for all tasks:

$$AR_t = \sum_{r \in X_t} \left(\frac{C_{rt}}{|R_t|} \right), \qquad \forall t \in T,$$

where $X_t = \{ r | r \in R_t;\ \mathbf{M}[1,r] = o;\ t \in T_o \}$, $\forall t \in T^p$, and $X_t = R_t$, $\forall t \in T^e$.

5. If $T = \varnothing$ or $R = \varnothing$, go to Step 7; otherwise, identify task $\hat{t}$ where (break ties by prioritizing p-Tasks over e-Tasks; if the tasks are the same type, break the tie arbitrarily)

$$AR_{\hat{t}} = \min_{t \in T} \{ AR_t \},$$

and set $z_{\hat{t}} = 1$.

6. Identify resource $\hat{r} \in X_{\hat{t}}$ where $C_{\hat{r}\hat{t}} = \min_{r \in X_{\hat{t}}} \{ C_{r\hat{t}} \}$ (break ties arbitrarily). Calculate $\alpha_{\hat{r}\hat{t}} = z_{\hat{t}} \cdot C_{\hat{r}\hat{t}}$:
   - If $\alpha_{\hat{r}\hat{t}} < L_{\hat{r}}$, set $\mathbf{M}[\hat{t}+1, \hat{r}] = \alpha_{\hat{r}\hat{t}}$; $L_{\hat{r}} \leftarrow L_{\hat{r}} - \alpha_{\hat{r}\hat{t}}$; $T \leftarrow T \setminus \{\hat{t}\}$; and return to Step 5.
   - If $\alpha_{\hat{r}\hat{t}} \geq L_{\hat{r}}$, set $\mathbf{M}[\hat{t}+1, \hat{r}] = L_{\hat{r}}$, $L_{\hat{r}} = 0$, $R \leftarrow R \setminus \{\hat{r}\}$, $X_{\hat{t}} \leftarrow X_{\hat{t}} \setminus \{\hat{r}\}$ and $z_{\hat{t}} \leftarrow z_{\hat{t}} - (L_{\hat{r}} / C_{\hat{r}\hat{t}})$. If $X_{\hat{t}} \neq \varnothing$, redo Step 6; otherwise, return to Step 5.

7. Output matrix **M** as the initial solution *S*.

The rock and sand heuristic is composed of two phases, for (i) locating the resources in the organizations, and (ii) allocating the p-Tasks and e-Tasks to the combinations of organizations and resources. This is done by estimating the amount of resources required by each organization, considering its respective tasks (Step 1), and locating the largest resources, one at a time, in the organization with the highest demand (Steps 2 and 3), following the rock and sand analogy. This heuristic procedure is intended to minimize the CR by distributing the resources with respect to the levels of demand. The idea behind the task allocation heuristic is that (i) tasks with lower average resource requirements, in general, have higher priority for allocation, and (ii) in case of ties, p-Tasks are prior to e-Tasks. This is done by prioritizing the tasks based on their average resource requirements (given the allocation of resources) (Steps 4 and 5), and then allocating them, one at a time, to the most efficient resource(s) (Step 6). This procedure continues until all tasks and/or resources are allocated.

The fitness of matrix **M** is calculated using the following formula:

$$F(\mathbf{M}) = w_{FR} \cdot \left(\sum\nolimits_{t \in T} FR_t / |T|\right) + w_{CR} \cdot (1 - CR), \tag{7.11}$$

where w_{FR} and w_{CR} respectively denote the weights of the FR and the CR objectives, and the values of FR_t and CR are obtained through Eq. (7.2) and (7.4), respectively, where

$$\hat{\alpha}_{rt} = \mathbf{M}[t+1, r], \qquad \forall\, r \in R,\, t \in T,$$

$$\hat{\chi}_{ort} = \begin{cases} 1, & \text{if } \mathbf{M}[1, r] = o \text{ and } \mathbf{M}[t+1, r] > 0, \\ 0, & \text{Otherwise,} \end{cases} \qquad \forall\, o \in O,\, r \in R,\, t \in T.$$

7.4.2 Neighborhood Search: The Natural Justice Rule

Various theories and philosophies stress, in some way, the existence of *justice* in nature—that "*you gain what you deserve*". The theory of Social Darwinism, an extension of Charles

Darwin's law of *natural selection* to sociology, states that stronger/weaker individuals or groups have higher/lower chances of survival, growth, and dominance in society. Oxford Dictionaries define *justice* as "*just behavior or treatment*" or "*the quality of being fair and reasonable*", which is, to a great extent, reflected by the notion of natural justice rule—an individual's "*gain*" is an indication of their "*power*". In the context of best matching problems, an individual's power may be represented by their overall popularity, associated costs/benefits, resources, fitness, dimensional tolerance, and so on. Accordingly, following the natural justice rule, the *deviation* between the power and the gain of individuals (*e.g.*, tasks; resources; organizations) can be viewed as an indication of the quality of a solution. This is the basis for the TS-Jr. algorithm developed in this work.

In tabu search, the moves in each iteration are applied in the neighborhood by making slight changes in the current solution *S*; *e.g.*, changing the assignment of one or more tasks in the best matching between tasks and agents (Yagiura *et al.*, 2004). The main challenge, however, is to find the *best move* (*e.g.*, the best alternative tasks and the respective changes in their assignments). The developed neighborhood search heuristic addresses this issue—based the natural justice rule—by defining a generic *power-gain deviation function*, as follows:

$$D_i = P_i - G_i, \quad \forall\, i, \tag{7.12}$$

where P_i and G_i respectively represent the levels of power and gain of individual *i*. Accordingly, individuals with larger values of $|D_i|$ have higher likelihood of being selected for the next move in the neighborhood. For instance, if the actual processing time of a task on its assigned machine (*i.e.*, gain) is considerably higher/lower than its average processing time on all machines (*i.e.*, power), there is a high likelihood that this is not an optimal match—neither for the task nor for the machine—and thus, there is high chance that the task/machine is selected for the next move. The key point, however, is to properly define the power and gain values in different best matching instances.

In the CLAP, the natural justice rule is applied to the individuals in set O (organizations), regarding their matching to the elements of sets R (resources) and T (tasks). The power-gain deviation function for the elements of set O is then calculated using the following formula:

$$D_o = \underbrace{\sum_{t \in T_o} \sum_{r \in R_t} \left(\frac{C_{rt}}{|R_t|} \right)}_{P_o} - \underbrace{\sum_{r \in R} L_r \max_{t \in T} \left\{ \hat{\chi}_{ort} \right\}}_{G_o}, \quad \forall o \in O. \tag{7.13}$$

That is, D_o is a function of the deviation between the total *average* resource requirements of the tasks attributed to organization o (*i.e.*, P_o), and the total resources assigned to organization o (*i.e.*, G_o). Thus, by minimizing $|D_o|$ in an iterative fashion, the neighborhood search is automatically directed towards minimizing the CR objective (improving the distribution of resources among the organizations). In order to maximize the FR objective (better allocation of resources to the tasks), the rock and sand heuristic is applied for reallocation of the tasks in each iteration. The neighborhood search heuristic is as follows:

0. Set matrix $\mathbf{M}_0 = \mathbf{M}$ as the initial solution S, $TL = \varnothing$ as the initial tabu list, $IT = 0$ as the counter for stopping the algorithm, and $\tau(o, r, t) = 0$, $\forall o \in O, r \in R, t \in T$, as the tabu time of (o, r, t).

Selection

1. Randomly generate $\gamma \in [0, 1]$. If $\gamma < \gamma_0$, select organizations $\hat{o}$ and $\breve{o}$ where:

$$\left|D_{\hat{o}}\right| = \max_{o \in O} \left\{\left|D_o\right|\right\}, \tag{7.14.a}$$

$$\left|D_{\breve{o}} - D_{\hat{o}}\right| = \max_{o \in O} \left\{\left|D_o - D_{\hat{o}}\right|\right\}. \tag{7.14.b}$$

Otherwise, select organizations $\hat{o}$ and $\breve{o}$ randomly from O, where $\hat{o} \neq \breve{o}$.

Ejection Chains

2. If $D_{\hat{o}} > D_{\breve{o}}$ identify resource $\hat{r}$ where $\mathbf{M}\left[1, \hat{r}\right] = \breve{o}$ and $\left|L_{\hat{r}} - \left|D_{\hat{o}}\right|\right| = \min_{r \in R;\, \mathbf{M}[1,r] = \breve{o}} \left\{\left|L_r - \left|D_{\hat{o}}\right|\right|\right\}$, and update $\mathbf{M}\left[1, \hat{r}\right] = \hat{o}$. Otherwise, identify resource $\hat{r}$ where $\mathbf{M}\left[1, \hat{r}\right] = \hat{o}$ and $\left|L_{\hat{r}} - \left|D_{\breve{o}}\right|\right| = \min_{r \in R;\, \mathbf{M}[1,r] = \hat{o}} \left\{\left|L_r - \left|D_{\breve{o}}\right|\right|\right\}$, and update $\mathbf{M}\left[1, \hat{r}\right] = \breve{o}$.
3. Follow Steps 4-6 of the rock and sand heuristic to reallocate all the tasks based on the updated location of the resources. For each task $\hat{t}$, remove resource $\hat{r}$ from the list of eligible resources, *i.e.*, $X_{\hat{t}} \leftarrow X_{\hat{t}} \setminus \left\{\hat{r}\right\}$, if their combination belongs to the tabu list, *i.e.*, $\left(o,\, \hat{r},\, \hat{t}\right) \in TL$, $\mathbf{M}\left[1, \hat{r}\right] = o$.

Fitness and Stopping Criterion

4. Output matrix **M** as the updated solution *S'*. If $0 < F\left(\mathbf{M}\right) - F^* < \delta$, $\delta > 0$, update $IT \leftarrow IT + 1$; otherwise, set $IT = 0$ (F^* denotes the best solution so far). If $IT > IT_{\max}$, stop and output **M** as the optimal solution; otherwise, go to Step 5.

Tabu List Update

5. Update all positive tabu times $\tau\left(o,\, r,\, t\right) \leftarrow \tau\left(o,\, r,\, t\right) - 1$, $\forall o \in O,\, r \in R,\, t \in T$, $\tau\left(o,\, r,\, t\right) > 0$, and

$$\tau(o, r, t) = \max\left\{\left\lceil \theta \cdot \left(F(\mathbf{M}) - F(\mathbf{M}_0)\right)\right\rceil, 0\right\},$$
$$\forall (o, r, t) \in \{(o, r, t) \mid \mathbf{M}[t+1, r] = 0; \tag{7.15}$$
$$\mathbf{M}_0[t+1, r] > 0; \mathbf{M}_0[1, r] = o\},$$

where $\theta > 0$ is a predefined parameter, and $\lceil \cdot \rceil$ denotes the ceiling function.

6. Update tabu list $TL \leftarrow TL \cup \{(o, r, t)\}$, $\forall o \in O, r \in R, t \in T$, $\tau(o, r, t) > 0$, set $\mathbf{M}_0 = \mathbf{M}$ as the initial solution *S*, and return to Step 1.

The neighborhood search heuristic starts by *probabilistically* selecting two organizations in two consecutive rounds, following the natural justice rule: In the first round, the organization with the largest deviation—based on the justice rule (either positive or negative)—is selected (Eq. (7.14.a)), while in the second round, an organization is selected, which has the most opposite situation of the first organization (Eq. (7.14.b)). The use of parameter γ_0 enables *exploitation* of the knowledge provided by the natural justice rule (*i.e.*, neighborhood search on the two extremely opposite cases of capacity-demand imbalance) as well as *exploration* of new solutions in each iteration. The selection is followed by an extended version of the ejection chains proposed by Laguna *et al.* (1995), where the organization with extra capacity shares one of its resources that best fits the resource requirement of the organization with capacity shortage (Step 2). Following this update, the tasks that are not in the tabu list (exploitation of former knowledge on previously visited low-quality points) are reallocated to the organizations and resources via the task allocation stage of the rock and sand heuristic presented earlier (Steps 3).

The fitness of the updated solution is calculated using Eq. (7.11). If no improvement is made (*i.e.*, the gap between the fitness values of the current and the best solutions is smaller than a predefined parameter $\delta > 0$) over the last $IT_{\max}$ iterations, the algorithm stops. Otherwise, it updates the tabu times and the tabu list, and resumes the neighborhood search. For simplicity and without loss of generality, the tabu times are considered proportional to the improvements made in the fitness values after each iteration (Eq. (7.15)). That is, if adjustment of a certain set of variables results in a larger improvement in the fitness value, those variables remain in the tabu list for a longer period. If there is no/negative improvement, the leaving variables do not enter the tabu list. Parameter θ is defined by the user, based on the fitness values, to adjust the performance of the algorithm. Depending on the application and computational requirements of specific problems, however, other (more advanced) functions can be applied for this purpose.

The main purpose of the developed rock and sand heuristic and the neighborhood search heuristic based on natural justice rule is to enhance the quality and efficiency of tabu search in finding the optimal solution to the CLAP.

7.5 Numerical Experiments

The CLAP is an extension of the original LAP (Location Allocation Problem) with additional considerations on capacity limits and collaboration through resource and task sharing. Hence, in order to highlight the impact of collaboration on the quality of location-allocation decisions in collaborative CNO in terms of task fulfillment and collaboration efficiency, the results of the CLAP obtained from the TS-Jr. algorithm (as scenario S_1) are compared with the results of a non-collaborative scenario (S_0), which involves all the features of the CLAP and TS-Jr. (including optimal resource location decisions) exclusive of the possibility of dynamic task sharing (virtual collaboration) among organizations.

Several test-problems, denoted by $|O|:|R|:|T|$, are generated through a random procedure for relative comparison between S_1 and S_0 as well as performance analysis of the TS-Jr. It is assumed that all resources are eligible for processing all tasks, *i.e.*, $R_t = R$, $\forall\ t \in T$. The capacity requirements by individual tasks are randomly generated following Uniform distribution for all problem instances: $C_{rt} \in$ Uniform [50, 150], $\forall\ r \in R$, $t \in T$. The capacity limits of the resources are generated with respect to the capacity requirements (90%-110% of the mean): L_r = Uniform [0.9, 1.1] × $\text{mean}_t\ (C_{rt})$, $\forall\ r \in R$. The type of the tasks (T^p or T^e) and their respective organizations (T_o) are determined randomly in a way that each organization is attributed to at least one task. The algorithmic parameters are also set as follows: $\gamma_0 = 0.30$ $\delta = 0.01$; $\theta = |T|$; $IT_{\max} = 500$; $w_{FR} = w_{CR} = 0.5$.

7.5.1 Results and Analyses

20 independent small- to large-sized CLAPs are generated following the experimental design procedure explained earlier. For each problem, the TS-Jr. algorithm is capable of suggesting the (near) optimal decision regarding the best location of resources, allocation of tasks to resources/organizations, and collaboration among organizations through resource sharing and task sharing. All this information is integrated and reported—for decision-maker/user of the TS-Jr.—through matrix **M** of the optimal solution. Figure 7.4 presents the results obtained from the TS-Jr. algorithm for scenarios S_0 and S_1, in terms of

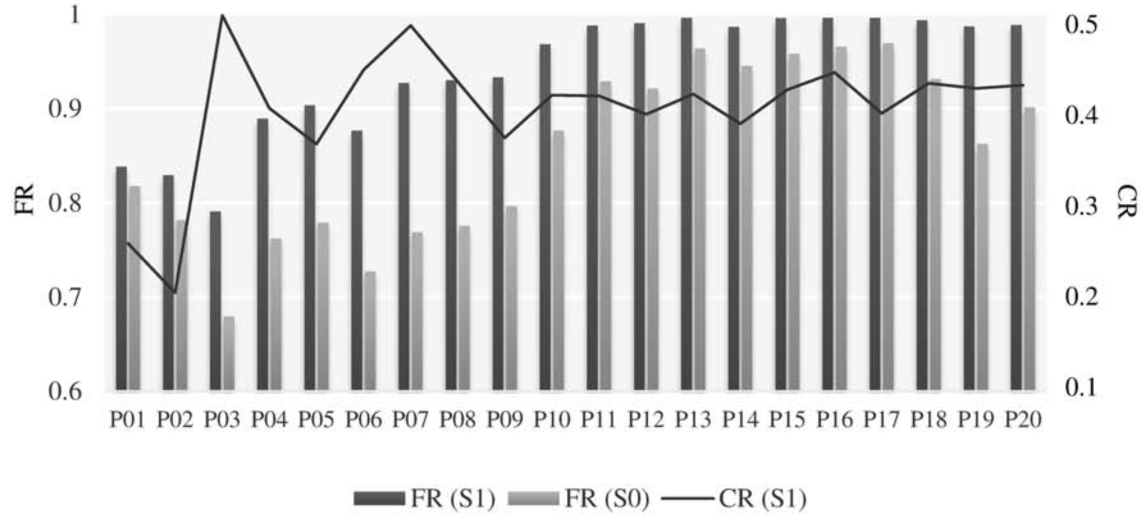


Figure 7.4. Comparison between S_0 and S_1 in terms of FR and CR.

FR and CR (only for S_1). Table 7.2 presents detailed information on the problem size and computational time for each CLAP. (Note that $|O|$, $|R|$, and $|T|$ in the test-problems are selected arbitrarily and can take any other values.)

Table 7.2. Details of the CLAP test-beds.

Problem ID	Size ($\|O\|$:$\|R\|$:$\|T\|$)	$\|T^e\|/\|T^p\|$	Run time (sec.)*
P01	5:12:20	1.00	7.05
P02	10:15:25	0.56	8.12
P03	15:20:35	1.06	11.11
P04	20:30:50	1.17	17.25
P05	25:35:60	0.88	20.11
P06	30:45:80	1.22	27.19
P07	35:55:90	1.31	31.45
P08	40:70:120	1.07	43.14
P09	45:90:150	0.80	55.45
P10	50:120:180	1.00	70.01
P11	55:150:200	1.02	79.45
P12	60:180:250	1.03	107.22
P13	70:240:300	1.06	129.73
P14	80:300:400	0.93	191.74
P15	90:380:500	1.31	262.97
P16	100:450:600	0.99	313.45

Table 7.2. Details of the CLAP test-beds.

Problem ID	Size ($\|O\|:\|R\|:\|T\|$)	$\|T^e\|/\|T^p\|$	Run time (sec.)*
P17	120:600:800	0.90	452.23
P18	150:800:1200	0.99	797.11
P19	180:900:1500	0.98	1031.90
P20	200:1200:2000	0.97	1429.91

* The TS-Jr. algorithm is programmed in MATLAB R2014a executed from the Purdue ITaP's Go Remote Cloud on an Intel® Core™ i7 Processor.

The experimental results reveal that, in all cases, scenario S_1 outperforms scenario S_0 in terms of FR thanks to the possibility of collaborative processing of e-Tasks by organizations. This indicates the significant impact of collaboration on the service level and utilization of distributed resources, where less than 20% of the e-Tasks, on average, are processed locally at their respective organizations. This, in turn, increases the flexibility and stability of the CNO in handling demand disruptions with the same level of resources (Moghaddam and Nof, 2014). The experiments also show that as the number of resources and tasks—and thus their *variety*—increases, become more capable of fulfilling the tasks, due to access a larger and more diverse pool of resources and higher likelihood of finding their best match. This phenomenon can be explained, more technically, by the collaborative fault tolerance principle of the collaborative control theory (Nof, 2007), where larger networks of agents/resources are proven to result in more resilient network—and thus higher FR—through *teaming* (Reyes Levalle and Nof, 2015).

A paired t-test is conducted to evaluate the significance of the gap between the results of S0 and S1. The test compares the mean values of two treatments, the FR of S_0 and S_1 obtained for P01 to P20, according to the following hypotheses; H_0: $FR_{S0} = FR_{S1}$; H_1: $FR_{S0} \neq FR_{S1}$. At the significance level of 0.05 and freedom degree of 19, the paired *t*-

test parameter is 2.09, which is lower than the t-test estimates obtained based on the results of S_0 and S_1: 3.87. Hence, the null hypothesis is rejected, implying a significant gap between the FR values of S_0 and S_1. Consequently, because the gap is positive in all test problems, it can be concluded that S_1 significantly outperforms S_0.

7.5.2 Computational Efficiency

The TS-Jr. algorithm is experimentally proven to be able to yield near-optimal solutions to the CLAP with significantly short computational time (see Table 7.2). However, since both the initialization and the neighborhood search mechanisms are the first of their kind, the performance of the TS-Jr. algorithm must be compared with a benchmark in order to ensure its computational efficiency and solution quality. This is typically performed by comparing the performance of the developed algorithm with other similar/peer exact/heuristic/ metaheuristic algorithms (*e.g.*, genetic algorithms; simulated annealing; branch-and-bound), or its solution with a lower/upper bound. Since CLAP is a new problem and none of the aforementioned (and similar) algorithms has been used for solving it/its extensions before, the following two benchmarks are used for performance evaluation of the TS-Jr. algorithm:

1. *Tabu search* (B_1) is considered as the first benchmark, with all the features and mechanisms of TS-Jr. for initialization and neighborhood search (*i.e.*, the rock and sand heuristics) except for the ejection chains for neighborhood search, which are based on random selection of organizations (rather than selection based on the natural justice rule).

2. *Upper bound* (B_2) for the CLAP is generated through a heuristic procedure based on the rock and sand analogy, and considered as the second benchmark. The upper bound is obtained by relaxing the condition on local processing of p-Tasks at their respective organization, and setting their resource requirements equal to their minimum value, *i.e.*,

$$C_{rt} = \min_{r \in R}\{C_{rt}\}, \quad \forall t \in T.$$

The optimal solution to the modified CLAP (see Moghaddam and Nof, 2016[b]) provides an upper bound for the FR objective. The experimental results presented in the previous section indicate that CR is approximately equal to 80% of the ratio of e-Tasks. Hence, 80% of this value is simply considered as an approximate experimental lower bound for CR. Accordingly, the upper bound for the CLAP is calculated as follows:

$$UB_F = w_{FR} \cdot \max\left\{ \frac{\sum_{r \in R} L_r}{\sum_{t \in T} \min_{r \in R}\left\{C_{rt}\right\}}, 1 \right\} + w_{CR} \cdot \left(1 - 0.7 \cdot \frac{\left|T^e\right|}{\left|T^e\right| + \left|T^p\right|} \right). \qquad (7.16)$$

Figure 7.5 summarizes the performance comparison between the TS-Jr. algorithm and the two benchmarks, tabu search and the upper bound. Both the TS-Jr. and the tabu search algorithms are executed 5 times, for 500 iterations, and on four small- to large-sized CLAPs. (Note that the test-problems generated in this section are different from the test-problems used earlier.) The presented results illuminate the computational efficiency and quality provided by the developed neighborhood search mechanism based on the natural justice rule—as the only distinction between the TS-Jr. algorithm and the applied tabu

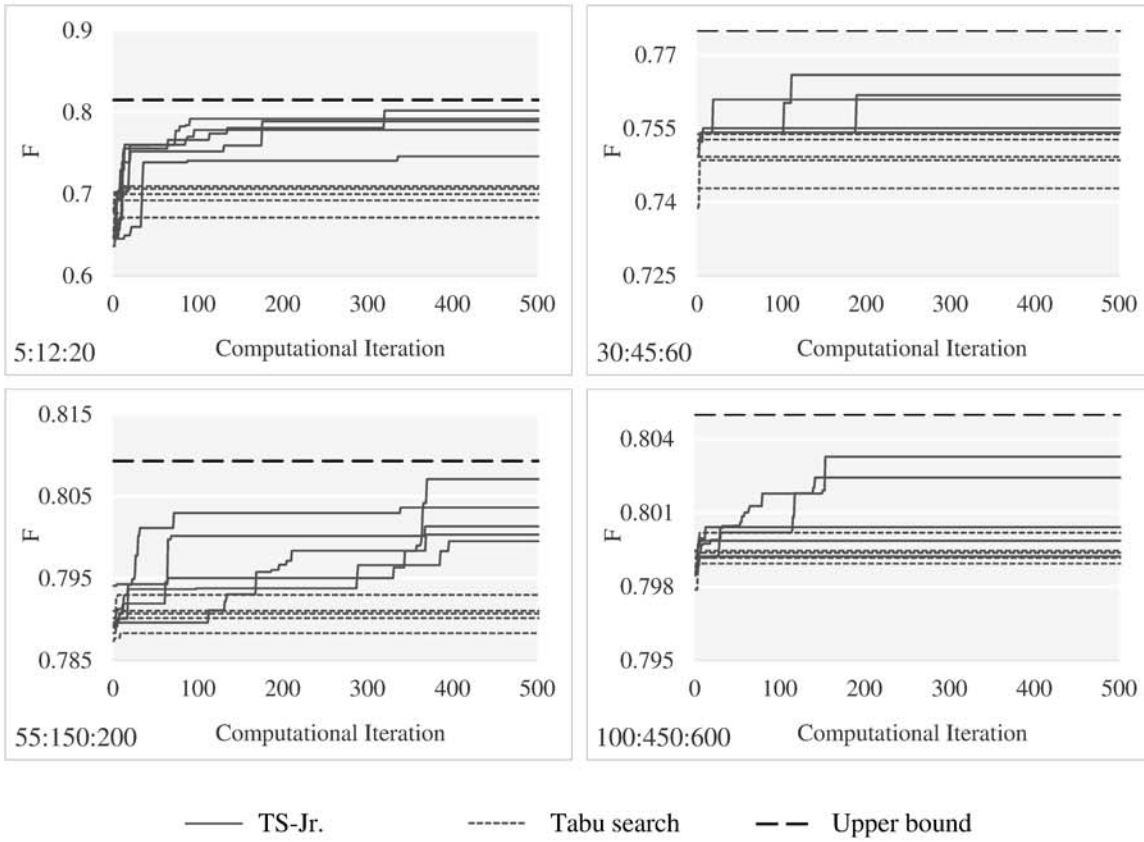


Figure 7.5. Analysis of the efficiency and quality of the TS-Jr. algorithm compared to tabu search and upper bound.

search. The TS-Jr. algorithm show remarkable computational merits by outperforming the “efficient” tabu search and converging to the relatively “loose” upper bound after a small number of iterations. This proves that the probabilistic process of exploring new solutions (as performed in the applied tabu search) and exploiting the knowledge of capacity-demand imbalance (based on the natural justice rule) enables stronger and more focused neighborhood search mechanism as opposed to traditional approaches based on random selection.

7.6 Concluding Remarks

This case study formalizes an emerging problem in the design of cyber-supported and communication-enabled CNO, and develops a computationally efficient and high-quality algorithm for optimization of the problem. The CLAP offers improvements in the utilization, service level, and stability of any CNO, from sensor networks and teams of robots to large-scale enterprises and multinational corporations, through dynamic collaboration and best-matching of distributed tasks and resources. Although certain modifications may be required in each case, the overall mechanics of the process are similar and in line with the fundamentals of the CLAP. The TS-Jr. algorithm, an improved tabu search with a powerful neighborhood search mechanism inspired by the natural justice rule, is proven as an effective tool for solving large-scale CLAPs. The computational efficiency of the TS-Jr. enables fast reconfiguration, adaptation, and evolution of CNO under various dynamic changes in the characteristics/domain of the problem; *e.g.*, association/ dissociation of organizations, resources, and/or tasks to/from the CNO.

The main assumption in defining the CLAP and developing the TS-Jr. algorithm is that the goals of all individuals are in line with the global goals of the entire CNO—that all interactions are collaboration-based rather than competition-based. In some systems, however, individuals may be biased, untrustworthy, or have conflicting goals with the goals of other individuals or the entire CNO. That is, not every individual is *incented* to collaborate. In such situations, the CNO is prone to substantial conflicts and instability; therefore, the collaboration and best-matching decisions must be made through centralized negotiation protocols or delegated to distributed intelligent and autonomous agents.

Accordingly, distributed and agent-based CLAP together with the issues of trust, security, and incentive-based collaboration constitute the main stream of research on the CLAP in the future.

CHAPTER 8. CONCLUSIONS

8.1 Summary of Original Contributions

Advances in service-orientation and collaborative e-Work have gradually increased the need for scalability, integrability, and resilience of systems, and enabled concurrency in distributed operations—from minimal file exchange to large-scale inter-organizational collaborations. The escalated rates of interactions in such complex networks, however, have brought up a new cohort of challenges in terms of flexibility and scalability along with optimality and timeliness of decisions. In this context, the hypothesis is that if every single element of a distributed system is matched to its best peer(s) at the right time (*e.g.*, suppliers--retailers--customers in enterprise networks, jobs--machines--sensors in factories, vehicles--routes in transportation systems, robots--humans--orders in warehouses), the outcome will be a better system that addresses (at least partially) those emerging challenges. The complex, dynamic, and uncertain nature of modern manufacturing and service systems, however, has made such *best matching* processes very difficult to accomplish; in terms of both distribution of decisional capabilities (*e.g.*, hierarchy/heterarchy) and nature of interactions among distributed individuals (*e.g.*, competitive/cooperative).

This dissertation investigates the problems of mismatch and best matching in distributed manufacturing, supply, and service systems. In this context, three research questions were outlined, and addressed as follows (Table 8.1):

RQ1-1: *What is a good taxonomic framework for systematic syntheses, identification, and specification of matching problems in different areas?*

Answer: The PRISM taxonomy of best matching is proposed as a systematic framework for synthesis, identification, and specification of matching processes with respect to 3+1 dimensions (see CHAPTER 3). The PRISM framework provides a general mechanism for classification of matching processes with respect to three main dimensions: D1, the characteristics of the individuals/sets to be matched; D2, specific conditions/requirements of the matching process; D3, the criteria and procedures by which the best match is evaluated and determined. The additional dimension (D+) addresses the progression in the three main dimensions of matching, that may or may not take place over time.

RQ1-2: *What are the most important characteristics of such framework?*

Answer: The developed (and similar) taxonomic framework must be simple and comprehensive. Its simplicity enhances mapping of different matching processes to a general structure, which in turn enables analogical reasoning and comparison between similar but unrelated matching processes. Its comprehensiveness enables covering every aspect and feature of various matching processes in one unified and efficient structure.

RQ2-1: *What are the best approaches for structuring and formulating matching problems and processes?*

Answer: Matching is a binary decision, and, in mathematical terms, is represented by binary variables. Best matching is an optimization problem; the problem of finding the best match out of a set of potential alternatives. Hence, best matching problems are naturally formulated as binary programs or mixed-integer programs, when other dependent (and not necessarily binary) decision variables are involved. In a best matching model, the three dimensions of the PRISM taxonomy may be formulated as objective function(s), constraints, or both. For example, resource sharing decisions are represented in capacity constraints, while IP (Interdependent Preferences) are incorporated in the objective function of matching problems, leading to nonlinear (*e.g.*, quadratic) objective functions. Although binary/mixed-integer programming are efficient methods for mathematical formulation of matching problems, they are limited to centralized and static settings (*i.e.*, no D+). In dynamic and distributed environments where decision-making authority is delegated to decentralized individuals, matching processes must be formulated using interaction mechanisms and protocols.

RQ2-2: *What algorithms and protocols can be developed to efficiently solve those best matching problems?*

Answer: Matching is known as an NP-hard problem. Various exact and heuristic approaches have been proposed in literature for handling its computational complexity. In this dissertation, various heuristic and metaheuristic tools, including genetic and evolutionary algorithms and tabu search, are developed for solving the new (and relatively unstructured) best matching problems in classic centralized and static environments. In addition, several interaction mechanisms and protocols including task administration and predictive best matching protocols and collaborative multi-agent systems are developed for design and execution of matching processes in distributed and dynamic environments.

RQ3-1: *How can the developed best matching algorithms and protocols be validated?*

Answer: The developed concepts, models, algorithms, and protocols are validated mathematically (*e.g.*, network stability—Case 1; line balanceability—Case 2) and/or through numerical experiments. The new concepts (*e.g.*, best matching with resource sharing; IP) are validated through comparison with the existing concepts (*e.g.*, non-collaborative matching; no IP). The new optimization and control mechanisms are also compared with the existing equivalent methodologies in terms of both computational efficiency and decision-making capabilities.

RQ3-2: *What case studies, experiments, scenarios, and statistical analysis methods must be deployed to test and highlight the relative impact of those methodologies?*

Answer: Four case studies are conducted on recent and emerging instances of matching in supply networks, manufacturing systems, social networks, and service systems. The case studies provide detailed descriptions on various extensions of matching problems, in terms of definition, impact, and solution methodologies. The case studies were selected according to the mission of the PRISM Center as well as the expertise and interests of the author.

Table 8.1. Relationship between research questions and dissertation structure.

	Research Question	Concepts and Methodologies
RQ1	*What is a good taxonomic framework for systematic syntheses, identification, and specification of matching problems in different areas?*	3.1 The PRISM Taxonomy of Best Matching 3.1.1 D1: Sets 3.1.2 D2: Conditions 3.1.3 D3: Criteria 3.1.4 D+: Time, Progression
	What are the most important characteristics of such framework?	
RQ2	*What are the best approaches for structuring and formulating matching problems and processes?*	2.1 Matching Problem Structures and Characteristics 4.3 Optimization: MIP and CPLEX 4.3.1 Mathematical Formulation 5.3 Optimization: MOMIP and Goal Programming 5.3.1 Mathematical Formulation 6.3 Cluster Formation: QAP and GA 7.3 Mathematical Formulation
	What algorithms and protocols can be developed to efficiently solve those best matching problems?	2.2 Methodologies 4.4 Control: TAP and PBMP 4.4.1 General Logic 4.4.2 TRAP—Task Requirement Analysis 4.4.3 SRAP—Shared Resource Allocation 4.4.4 STOP—Synchronization and Time-Out 5.4 Control: CMAS 5.4.1 Tool Sharing-Best Matching Protocol 6.3 Cluster Formation: QAP and GA 6.3.1 Encoding 6.3.2 Initialization 6.3.3 Fitness Evaluation 6.3.4 Reproduction 6.3.5 Parameters Setting 6.4 Cluster Evolution: EA 6.4.1 First Generation 6.4.2 Evolution 7.4 Optimization: TS-Jr. 7.4.1 Initialization: The Rock and Sand Heuristic 7.4.2 Neighborhood Search: The Natural Justice Rule

Table 8.1. Relationship between research questions and dissertation structure.

	Research Question	Concepts and Methodologies
RQ3	*How can the developed best matching algorithms and protocols be validated?*	4.3 Optimization: MIP and CPLEX 4.3.2 Numerical Experiments 4.4 Control: TAP and PBMP 4.4.5 Numerical Experiments 5.3 Optimization: MOMIP and Goal Programming 5.3.2 Numerical Experiments 5.4 Control: CMAS 5.4.2 Numerical Experiments 6.5 Numerical Experiments 6.5.1 Results 6.5.2 Discussion 6.5.3 Computational Efficiency 7.5 Numerical Experiments 7.5.1 Results and Analyses 7.5.2 Computational Efficiency
	What case studies, experiments, scenarios, and statistical analysis methods must be deployed to test and highlight the relative impact of those methodologies?	CHAPTER 4 Case 1—Collaborative Supply Networks CHAPTER 5 Case 2—Collaborative Assembly Lines CHAPTER 6 Case 3—Clustering with Interdependent Preferences CHAPTER 7 Case 4—Collaborative Service Enterprises

8.2 Future Research Directions

Future research must investigate the limitations of this work regarding key success factors in interconnected and vibrant manufacturing, supply, and service networks, develop solutions for real-time optimization and control of interactions, and enhance the individuals' intelligence for timely decision-making regarding of *what to do*, *with whom to interact*, *how* and *when*. Realization of this goal indeed requires (1) comprehensive characterization and taxonomy of distributed manufacturing and service systems, their requirements, constraints, and objectives; (2) assessment of the nature of interactions (*e.g.*, competitive/cooperative), and development of proactive mechanisms for handling controllable and uncontrollable behaviors; (3) design of effective decision-making networks through optimal distribution of decisional capabilities (*e.g.*, hierarchy/ heterarchy); and (4) development of models and algorithms based on operations research, artificial intelligence, and information technology (*i.e.*, analytics and informatics) for design, administration, and feedback.

The existing knowledge on the theories of interaction, distributed control, and matching is not sufficient for addressing such challenges in modern manufacturing and service systems, and that motivates my research on interaction engineering through best matching. This PRISM model formalizes such processes with respect to 3+1 dimensions, and provides a comprehensive and standardized framework for identification, specification, and design, in various domains such as supply (Case 1), manufacturing (Case 2), social (Case 3), and service (Case 4) networks. This dissertation sheds light on the significant impact of best matching on the competitive performance of distributed systems. Realization of such design guidelines in practice, however, requires tremendous efforts in both basic and applied research in this area. Moreover, the following directions are recommended with respect to the specific problems addressed throughout the case studies:

1. *Variability and uncertainty*. The uncertainties associated with costs of matching/sharing, information exchange, and possibility of acceptance/rejection of collaboration proposals during negotiations are practical scenarios that are worthy of attention in future studies.

2. *Steady state matching*. Optimal cluster design (*e.g.*, for supplier-customer/tool-workstation pairs) in the steady state, and inter- and intra-collaboration decisions and protocols are other directions to be addressed in future research.
3. *Real-time decisions*. All case studies present, to some certain extent, a *macro* view of matching processes. Implementation of such algorithms and protocols in real time, however, requires certain information exchange and negotiation procedures between different individuals. This issue must be addressed in future research.
4. *Communication and informatics*. Future research also must address the design of middleware architecture and the supportive components, decision support systems, modeling tools, and database systems associated with the presented best matching processes.
5. *Conflict and error detection and prevention* (Chen and Nof, 2007). Future research must address the issues related to the detection, resolution, and prevention of potential errors and conflicts in matching; *e.g.*, delayed response/sharing; conflict of interest for collaboration.
6. *Group- vs. self-orientation.* It is assumed in all case studies that the entire network of individuals is *incented* to collaborate. That is, all elements are *group-oriented*, seeking a set of common objectives that are necessarily in line with their local objectives. In some cases, however, the individuals may be *self-oriented* and their individual goals may contradict the common goal of the network. The network is then prone to instability, and the developed methodologies may require substantial modifications in order to be applicable.

7. *Variable preferences and perceptions*. The preferences of individuals (*e.g.*, cost; time) are modeled as functions of the mutual *influences* of individuals on each other's preferences, where each pair of individuals is assumed to have fixed *perceptions* about each other (Case Study 3). This, however, may not be true in many cases where two individuals may change their perceptions about each other after a period or dynamically. This issue increases the complexity of the match and requires modifications in the definitions, formulations, and methodology.
8. *Multidimensional matching*. The basic instance of generalized matching was considered for studying the BMP-IP for the sake of simplicity in definitions and formulations. In a similar manner, the methodology can be extended to more comprehensive and realistic instances of the BMP-IP.
9. *Social networks and emotion*. An important trait that differentiates networks of humans (social networks) from other types of networks is *emotion*. The IP was formulated as a linear (increasing, altruism; decreasing, envy) function of the mutual influences of individuals on each other's preferences. In social networks, however, such influences may not be easily quantifiable and may have nonlinear relation with IP, due to the complexity and dynamicity of humans' emotion in their mutual communications and interactions.
10. *Emergence of "coopetition"*. In spite of the benefits of collaboration, competition is an inevitable behavioral pattern in almost every system, from microorganisms to multinational corporations. Altruistic and selfish behaviors always go side by side and give each other meaning. The notion of "coopetition" is then coined in game theory to represent situations where competitors prefer to both compete and

collaborate with each other. In the context of matching, this is an important topic to address in the future, where the definition of “the best match” is dynamic and depends on the mutual interaction and local benefits of individuals.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abdulkadiroğlu, A., Pathak, P.A., Roth, A.E., 2005. The New York City High School Match. American Economic Review, Papers and Proceedings 95 (2), 364-367.

Ahujaa, R.K., Orlinb, J.B., Tiwaric A., 2000. A greedy genetic algorithm for the quadratic assignment problem. Computers and Operations Research 27 (10), 917-934.

Aköz, O., Petrovic, D., 2007. A fuzzy goal programming method with imprecise goal hierarchy. European Journal of Operational Research 181, 1427–1433.

Anuar, R., Bukchin, Y., 2006. Design and operation of dynamic assembly lines using work-sharing. International Journal of Production Research 44 (18–19), 4043–4065.

Askin, R.J., Chen, J., 2006. Dynamic task assignment for throughput maximization with work sharing. European Journal of Operational Research 168 (3), (2006) 853–869.

Avery, C., Jolls, C., Posner, R.A., Roth, A.E., 2007. The New Market for Federal Judicial Law Clerks. University of Chicago Law Review 74, 447-486.

Axsäter, S., Howard, C., Marklund, J., 2013. A distribution inventory model with transshipments from a support warehouse. IIE Transactions, Special Issue on Operations Engineering and Analysis 45 (3), 309-322.

Baker, M.B. and Sheasby, J., 1999. Extensions to the generalized assignment heuristic for vehicle routing. European Journal of Operational Research, 119, 147–157.

Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. Science 286 (5439), 509-512.

Barbas, J. and Marin, A., 2004. Maximal covering code multiplexing access telecommunication networks. European Journal of Operational Research, 159, 219–238.

Barbati, M., Bruno, G., Genovese, A., 2012. Applications of agent-based models for optimization problems: A literature review. Expert Systems with Applications 39, 6020–6028.

Bartholdi J.J., Eisenstein D.D., 1996. A production line that balances itself. Operations Research 44 (1), 21–34.

Baybars I, 1986. A survey of exact algorithms for the simple assembly line balancing problem. Management Science 32 (8), 909–932.

Becker C., Scholl A., 2006. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research 168 (3), 694–715.

Bell, A.M., 2002. Locally interdependent preferences in a general equilibrium environment. Journal of Economic Behavior and Organization 47 (3), 309–333.

Benders, J.F. and Van Nunen, J.A., 1983. A property of assignment type mixed linear programming problems. Operations Research Letters 2, 47–52.

Benjaafer, S., El-Hafsi, M., and de Vericourt, F., 2004. Demand allocation in multiple-product, multiple-facility, make-to-stock systems. Management Science, 50 (10), 1431–1448.

Boffey, T.B., 1989. Location problems arising in computer networks. Journal of the Operational Research Society, 40 (4), 347–354.

Bokhari, S.H., 1987. Assignment Problems in Parallel and Distributed Computing, Kluwer Academic Publishers, Norwell, MA.

Boysen, N., Fliedner, M., Scholl, A., 2007. A classification of assembly line balancing problems. European Journal of Operational Research 183 (2), 674–693.

Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: Which model to use when? International Journal of Production Economics 111 (2), 509–528.

Bressoud, T.C., Rastogi, R., and Smith, M.A., 2003. Optimal Configuration for BGP Route Selection, in Proceedings of IEEE INFOCOM' 2003, 2nd Annual Joint Conference on the IEEE Computer and Communications Societies, San Francisco, 2, 916–926.

Bukchin Y., Rabinowitch I., 2006. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. European Journal of Operational Research 174 (1), 492–508.

Bukchin, Y., Sofer, T., 2011. Bi-directional work sharing in assembly lines with strict and flexible assembly sequences. International Journal of Production Research 49 (14), 4377–4395.

Burkard, R., Dell-Amico, P., Martello, S., 2009. Assignment Problems. SIAM, ISBN 978-1-611972-22-1.

Burkard, R.E., Rendl, F., 1991. Lexicographic bottleneck problems. Operations Research Letters 10 (5), 303–308.

Burton, J., Banerjee, A., 2005. Cost-parametric analysis of lateral transshipment policies in two-echelon supply chains. International Journal of Production Economics 93-94, 169-178.

Bussmann S., Sieverding J., 2001. Holonic Control of an Engine Assembly Plant – An Industrial Evaluation. Proceedings of the 2001 IEEE SMC Conference, Arizona, 169-174.

Cabrales, A., Calvó-Armengol, A., 2008. Interdependent preferences and segregating equilibria. Journal of Economic Theory 139 (1), 99–113.

Camarinha-Matos, L.M., Afsarmanesh, H., Galeano, N., Molina, A., 2009. Collaborative networked organizations - Concepts and practice in manufacturing enterprises. Computers and Industrial Engineering 57, 46-60.

Campbell, J.F., Langevin, A., 1995. The snow disposal assignment problem. Journal of the Operational Research Society 46 (8), 919–929.

Caridi, M., Sianesi, A., 2000. Multi-agent systems in production planning and control: An application to the scheduling of mixed-model assembly lines. International Journal of Production Economics 68 (1), 29–42.

Caron, G., Hansen, P., Jaumard, B., 1999. The assignment problem with seniority and job priority constraints, Operations Research 47 (3), 449–454.

Carraresi, P., Gallo, G., 1984. A multi-level bottleneck assignment approach to the bus drivers_ rostering problem. European Journal of Operational Research 16 (2), 163–173.

Cattrysse, D.G., Salomon, M., and Van Wassenhove, L.N., 1994. A set partitioning heuristic for the generalized assignment problem. European Journal of Operational Research, 72, 167–174.

Cattrysse, D.G., Van Wassenhove, L.N., 1992. A survey of algorithms for the generalized assignment problem. European Journal of Operational Research 60 (3), 260–272.

Chan, F.T.S., Wong, T.C., Chan, L.Y., 2009. An evolutionary algorithm for assembly job shop with part sharing. Computers and Industrial Engineering 57 (3), 641–651.

Chang, G.J., Ho, P.-H., 1998. The b-assignment problems, European Journal of Operational Research 104 (3), 593–600.

Chen X.W., Nof S.Y., 2007. Error detection and prediction algorithms: Application in robotics. Journal of Intelligent and Robotic Systems 48 (2), 225–252.

Chen X.W., Nof S.Y., 2010. A decentralized conflict and error detection and prediction model. International Journal of Production Research 48 (16), 4829–4843.

Cheng, C.H., Goh, C-H., and Lee, A., 1996. Solving the generalized machine assignment problem in group technology. Journal of the Operational Research Society, 47, 794–802.

Chituc, C.M., Nof, S.Y., 2007. The Join/Leave/Remain (JLR) decision in collaborative networked organizations. Computers and Industrial Engineering 53, 173–195.

Cohen, R., Katzir, L., and Raz, D., 2006. An efficient approximation for the generalized assignment problem. Information Processing Letters 100, 162–166.

Cox, K.B., 2001. The effects of unit morale and interpersonal relations on conflict in the nursing unit. Journal of Advanced Nursing 35 (1), 17-25.

Cromley, R.G. and Hanink, D.M., 1999. Coupling land use allocation models with raster GIS. Journal of Geographical Systems, 1, 137–153.

Cuevas, E., Echavarría, A., Zaldívar, D., Pérez-Cisneros, M., 2013. A novel evolutionary algorithm inspired by the states of matter for template matching. Expert Systems with Applications 40 (16), 6359–6373.

Cummings, J.N., Kiesler, S., 2005. Collaborative research across disciplinary and organizational boundaries. Social Studies of Science 35, 703-722.

De Farias, I.R., Johnson, E.L., Jr., and Nemhauser, G.L., 2000. A generalized assignment problem with special ordered sets, a polyhedral approach. Mathematical Programming Ser. A, 89, 187–203.

Dell'Amico, M., Martello, S., 1997. The k-cardinality assignment problem, Discrete Applied Mathematics 76 (1–3), 103–121.

Dobson, G. and Nambimadom, R.S., 2001. The batch loading and scheduling problem. Operations Research, 49 (1), 52–65.

Dong, F., Akl, S.G., 2006. Scheduling Algorithms for Grid Computing, State of the Art and Open Problems. Technical Report No. 2006-504, School of Computing, Queen's University, Kingston, Ontario.

Drexl, A., 1991. Scheduling of project networks by job assignment. Management Science, 37 (12), 1590–1602.

Drezner, Z., 2003. A new genetic algorithm for the quadratic assignment problem. INFORMS Journal on Computing 15 (3), 320–330.

Drezner, Z., 2008. Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem. Computers and Operations Research 35 (3), 717 – 736.

Duin, C.W., Volgenant, A., 1991. Minimum deviation and balanced optimization, A unified approach. Operations Research Letters 10 (1), 43–48.

Elias J.J. and Roth A.E., 2007), "Econ One on One, Kidney Transplantation" WSJ. Online, forthcoming (Roth, 2007).

Esfarjani K., Nof S.Y., 1998. Client-server model of integrated production facilities. International Journal of Production Research 36 (12), 3295–3321.

Foster, I., Kesselman, C., Tuecke, S., 2001. The anatomy of the grid: Enabling scalable virtual organizations. International Journal of High Performance Computing Applications 15, 200-222.

Foulds, L.R., Wilson, J.M., 1999. On an assignment problem with side constraints, Computers and Industrial Engineering 37 (4), 847–858.

Franz, L.S., Miller, J.L., 1993. Scheduling medical residents to rotations, Solving the large-scale multi-period staff assignment problem, Operations Research 41 (2), 269–279.

Freling, R., Romeijn, H.E., Romero-Morales, D., and Wagelmans, A.P.M., 2003. A branch-and-price algorithm for the multi-period single-source problem. Operations Research, 51 (6), 922–939.

Gaertner, W., 1974. A dynamic model of interdependent consumer behavior. Zeitschrift für Nationalökonomie 34 (3-4), 327-344.

Gale, D., Shapley, L.S., 1962. College admission and stability of marriage. The American Mathematical Monthly 69 (1), 9-15.

Gambardella, L.M., Taillard, E.D., Dorigo, M., 1999. Ant colonies for the quadratic assignment problem. Journal of the Operational Research Society 50 (2), 167-176.

Gamberini R., Grassi A., Rimini B., 2006. A new multi-objective heuristic algorithm for solving the stochastic assembly line rebalancing problem. International Journal of Production Economics 102 (2), 226–243.

Gass S.I., 1987. The setting of weights in linear goal-programming problems. Computers and Operations Research 14, 227–229.

Gavirneni S., 2002. Information flows in capacitated supply chains with fixed ordering costs. Management Science 48 (5), 644–651.

Geetha, S., Nair, K.P.K., 1993. A variation of the assignment problem, European Journal of Operational Research 68 (3), 422–426.

Geetha, S., Vartak, M.N., 1994. The three-dimensional assignment problem with capacity constraints, European Journal of Operational Research 73 (3), 562–568.

Ghosh S., Gagnon R.J., 1989. A comprehensive literature-review and analysis of the design, balancing and scheduling of assembly systems. International Journal of Production Research 27 (4), 637–670.

Glover, F.W., 1986. Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research 13, 533–549.

Glover, F.W., 1994. Genetic algorithms and scatter search: unsuspected potential. Statistical Computing 4 (2), 131-40.

Glover, F.W., 1996. Ejection chains, reference structures and alternating path methods for traveling salesman problems. Discrete Applied Mathematics 65, 223–253.

Grabowski, R., Navarro-Serment, L.E., Paredis, C.J.J., Khosla, P.K., 2000. Heterogeneous teams of modular robots for mapping and exploration. Autonomous Robots 8, 293-308.

Grygiel, G., Algebraic Rk-assignment problems. Control and Cybernetics 10 (3&4), 155–165.

Gupta, S.K., Punnen, A.P., 1988. Minimum deviation problems. Operations Research Letters 7 (4), 201–204.

Haddadi, S. and Ouzia, H., 2004. Effective algorithm and heuristic for the generalized assignment problem. European Journal of Operational Research 153, 184–190.

Harvey, N.J.A., Ladner, R.E., Lovasz, L., and Tamir, T., 2006. Semi-matchings for bipartite graphs and load balancing. Journal of Algorithms, 59 (1), 53–78.

Hood, L., Balling, R., Auffray, C., 2012. Revolutionizing medicine in the 21st century through systems approaches. Biotechnology Journal 7, 992-1001.

Hopp, W.J., Van Oyen, M.P., and Tekin, E., 2004. Benefits of skill chaining in serial production lines with cross-trained workers. Management Science 50 (1), 83–98.

Huang C.Y., Wu Y.W., 2003. Decision model for partnership development in virtual enterprise. International Journal of Production Research 41 (9), 1855–1872.

Jagdev H.S., Thoben K.D., 2001. Anatomy of enterprise collaborations. Production Planning and Control 12 (5), 437–451.

Jalisi, Q.W.Z. and Cheddad, H., 2000. Third party transportation, a case study. International Journal of Industrial Engineering, Theory Applications Practice, 7 (4), 348–351.

Jamison, J., 2012. Games with synergistic preferences. Games 3 (1), 41-55.

Jeet, V. and Kutanoglu, E., 2007. Lagrangean relaxation guided problem space search heuristic for generalized assignment problems. European Journal of Operational Research 182, 1039–1056.

Jennings N.R., Sycara K., Wooldridge M., 1998. A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1, 275-306.

Jiménez M., Arenas M., Bilbao A., Rodrıguez M.V., 2007. Linear programming with fuzzy parameters: An interactive method resolution. European Journal of Operational Research, 177, 1599–1609.

Kalagnanam, J.R., Davenport, A.J., and Lee, H.S., 2001. Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. Electronic Commerce Research, 1, 221–238.

Kang H., 1994. Development of Information Exchange Protocols for Distributed Inspection Integration. MS thesis, School of Industrial Engineering, Purdue University (Thesis advisor: Nof, S.Y.), West Lafayette, IN, USA.

Kennington, J., Wang, Z., 1992. A shortest augmenting path algorithm for the semi-assignment problem. Operations Research 40 (1), 178–187.

Kim Y.K., Song W.S., Kim J.H., 2009. A mathematical model and a genetic algorithm for two-sided assembly line balancing. Computers and Operations Research 36 (3), 853 – 865

Klastorin, T.D., 1979. On the maximal covering location problem and the generalized assignment problem. Management Science, 25 (1), 107–112.

Klusch M., 2001. Agent-mediated trading: Intelligent agents and e-Business. In: Agent Technology for Communication Infrastructure (Ed.: Hazelden A.L.G., Bourne R.A.), Wiley, New York, pp. 59–76.

Ko, H.S., Nof, S.Y., 2012. Design and application of task administration protocols for collaborative production and service systems. International Journal of Production Economics 135, 177-189.

Koçkesen, L., Ok, E.A., 2000. Evolution of interdependent preferences in aggregative games. Games and Economic Behavior 31 (2), 303-310.

Koopmans, T.C., Beckman, M., 1957. Assignment problems and the location of economic activities. Econometric 25 (1), 53-76.

Kostova, T., Roth, K., Dacin, M.T., 2008. Institutional theory in the study of multinational corporations: A critique and new directions. Academy of Management Review 33, 994-1006.

Kouvelis, P., Yu, G., 1997. Robust Discrete Optimization and Its Applications, Kluwer Academic Publishers, Dordrecht, Netherlands.

Kuhn, H.W., 1955. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 (1&2), 83–97.

Kuhn, H.W., 1955. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 (1-2), 83–97.

Kulkarni, R.V., Förster, A., Venayagamoorthy, G.K., 2011. Computational intelligence in wireless sensor networks: A survey. IEEE Communications Surveys and Tutorials 13, 68-96.

Kutanoglu E., Mahajan M., 2009. An inventory sharing and allocation method for a multi-location service parts logistics network with time-based service levels. European Journal of Operational Research 194 (3), 728–742.

Laguna, M., Kelly, J.P., González-Velarde, J., Glover, F., 1995. Tabu search for the multilevel generalized assignment problem. European Journal of Operational Research 82, 176-189.

LeBlanc, L.J., Shtub, A., and Anandalingam, G., 1999. Formulating and solving production planning problems. European Journal of Operational Research, 112, 54–80.

Lee, Y.H., Jung, J.W., Jeon, Y.S., 2007. An effective lateral transshipment policy to improve service level in the supply chain. International Journal of Production Economics 106 (1), 115-126.

Lee, D., Stajkovic, A.D., Cho, B., 2011. Interpersonal trust and emotion as antecedents of cooperation: Evidence from Korea. Journal of Applied Social Psychology 41 (7), 1603-1631.

Lee, D.-H., Kim, Y.-D., 1998. A multi-period order selection problem in flexible manufacturing systems. Journal of the Operational Research Society 49 (3), 278–287.

Lee, M-K., 1992. A storage assignment policy in a man-on-board automated storage/retrieval system. International Journal of Production Research, 30 (10), 2281–2292.

Lee, S.M., Schniederjans, M.J., 1983. A multicriteria assignment problem, A goal programming approach. Interfaces 13 (4), 75–81.

Leitão P., 2009. Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence 22 (7), 979–991.

Li, J., Moghaddam, M., Nof, S.Y., 2015. Dynamic storage assignment with product affinity and ABC classification—a case study. International Journal of Advanced Manufacturing Technology. DOI:10.1007/s00170-015-7806-7.

Ling, F.Y.Y., Tran, P.Q., 2012. Effects of interpersonal relations on public sector construction contracts in Vietnam. Construction Management and Economics 30 (12), 1087-1101.

Lingard, L., Reznick, R., DeVito, I., Espin, S., 2002. Forming professional identities on the health care team: Discursive constructions of the 'other' in the operating room. Medical Education 36 (8), 728-734.

Littschwager, J.M. and Tcheng, T.H., 1967. Solution of a large-scale forest scheduling problem by linear programming decomposition. Journal of Forestry, 65, 644–646.

Lorena, L.A. N. and Narciso, M.G., 1996. Relaxation heuristics for a generalized assignment problem. European Journal of Operational Research, 91, 600–610.

Lorena, L.A.N., Narciso, M.G., and Beasley, J.E., 1999. A constructive genetic algorithm for the generalized assignment problem. Evolutionary Optimization, (2nd version).

Lourenco, H.R.D. and Serra, D., 2002. Adaptive approach heuristics for the generalized assignment problem. Mathware and Soft Computing, 9, 209–234.

Malhotra, R., Bhatia, H.L., Puri, M.C., 1985. The three dimensional bottleneck assignment problem and its variants, Optimization 16 (2), 245–256.

Maniezzo, V., Colorni, A., 1999. The ant system applied to the quadratic assignment problem. IEEE Transactions on Knowledge and Data Engineering 11 (5), 769-778.

Manning, T., Pogson, G., Morrison, Z., 2008. Interpersonal influence in the workplace - Part one: An introduction to concepts and a theoretical model. Industrial and Commercial Training 40 (2), 87-94.

Marler R.T., Arora J.S., 2004. Survey of multi-objective optimization methods for engineering. Structural Multidisciplinary Optimization 26, 369-395.

Martello, S., Pulleyblank, W.R., Toth, P., de Werra, D., 1984. Balanced optimization problems, Operations Research Letters 3 (5), 275–278.

Martello, S., Toth, P., 1995. The bottleneck generalized assignment problem. European Journal of Operational Research 83 (3), 621–638.

Mazzola, J., Neebe, A., and Dunn, C., 1989. Production planning of a flexible manufacturing system in a material requirements planning environment. International Journal of Flexible Manufacturing Systems, 1 (2), 115–142.

Mazzola, J.B., Neebe, A.W., 1986. Resource-constrained assignment scheduling. Operations Research 34 (4), 560–572.

Mell, P., Grance, T, 2009. Perspectives on cloud computing and standards. National Institute of Standards and Technology (NIST). Information Technology Laboratory.

Miettinen, K. 1999: Nonlinear Multiobjective Optimization. Boston: Kluwer Academic Publishers.

Moghaddam M., Nof S.Y., 2013[a]. Best Matching and Task Administration Protocols for Effective Demand and Capacity Sharing. Proceedings of the 22nd International Conference on Production Research, Brazil.

Moghaddam M., Nof S.Y., 2013[b]. Dynamic Tool Sharing with Best Matching Protocols for Efficient Assembly Line Balancing. Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems, Brazil.

Moghaddam M., Nof S.Y., 2014. Combined demand and capacity sharing with best matching decisions in enterprise collaboration. International Journal of Production Economics 148, 93–109.

Moghaddam M., Nof S.Y., 2015[a]. Balanceable assembly lines with dynamic tool sharing and best matching protocols—a collaborative assembly framework. IIE Transactions 47, 1363–1378.

Moghaddam M., Nof S.Y., 2015[b]. Real-time administration of tool sharing by best matching to enhance assembly lines balanceability and flexibility. Mechatronics 31, 147–157 (Special Issue: "New Trends in Intelligent Manufacturing").

Moghaddam M., Nof S.Y., 2015[c]. Best matching with interdependent preferences—implications for capacitated cluster formation and evolution. Decision Support Systems 79, 125–137.

Moghaddam M., Nof S.Y., 2015[d]. The collaborative factory of the future. International Journal of Computer Integrated Manufacturing (Special Issue: "Factories of the Future: Challenges and Leading Innovations in Intelligent Manufacturing"). DOI:10.1080/0951192X.2015.1066034.

Moghaddam M., Nof S.Y., 2016[a]. Real-time optimization and control mechanisms for collaborative demand and capacity sharing. International Journal of Production Economics 171, 495–506 (Special Issue: "Challenges for Sustainable Operations – Selected papers of ICPR 2013").

Moghaddam M., Nof S.Y., 2016[b]. Collaborative location-allocation decisions in networked v-Organizations. Computers and Industrial Engineering. Under review.

Moghaddam M., Nof S.Y., Menipaz E., 2016. Design and administration of collaborative networked headquarters. International Journal of Production Research (Special Issue: "Distributed Manufacturing to Enhance Productivity"). Accepted.

Mohanty, B.K., Vijayaraghavan, T.A.S., 1995. A multi-objective programming problem and its equivalent goal programming problem with appropriate priorities and aspiration levels: A fuzzy approach. Computers and Operations Research 22, 771–778.

Monfared, M.A.S. and Etemadi, M., 2006. The impact of energy function structure on solving generalized assignment problem using Hopfield neural network. European Journal of Operational Research, 168 (2), 645–654.

Monostori, L., Vánczaa, J., Kumara, S.R.T., 2006. Agent-Based Systems for Manufacturing. CIRP Annals - Manufacturing Technology 55 (2), 697–720.

Morita, P.P., Burns, C.M., 2014. Trust tokens in team development. Team Performance Management 20 (1), 39-64.

Niculescu, M., Payne, C.R., Krishnan, V., 2013. One-to-One and One-to-Many Business Relation Marketing: Toward a Theoretical Framework. Journal of Business-to-Business Marketing 20 (2), 51-64.

Niederle, M., Proctor, D.D., Roth, A.E., 2006. What will be needed for the new GI fellowship match to succeed? Gastroenterology 130, 218-224.

Nof, S.Y., 2003. Design of effective e-work: review of models, tools, and emerging challenges. Production Planning and Control 14 (8), 681–703.

Nof, S.Y., 2007. Collaborative control theory for e-Work, e-Production, and e-Service. Annual Reviews in Control 31, 281-292.

Nof, S.Y., Wilhelm, W.E., Warnecke, H.J., 1997. Industrial Assembly. Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK.

Nof, S.Y. (Ed.), 2009. Springer Handbook of Automation. Springer Publishers, Hidleberg-New York.

Nof, S.Y., 2007. Collaborative control theory for e-Work, e-Production, and e-Service. Annual Reviews in Control 31 (2), 281-292.

Nof, S.Y., Ceroni, J.A., Jeong, W., Moghaddam, M., 2015. Revolutionizing collaboration through e-Work, e-Business, and e-Service. Springer series in ACES, Automation, Collaboration & E-Service.

Nouyan, S., Groß, R., Bonani, M., Mondada, F., Dorigo, M., 2009. Teamwork in self-organized robot colonies. IEEE Transactions on Evolutionary Computation 13, 695-711.

Nowakovski, J., Schwarzler, W., Triesch, E., 1999. Using the generalized assignment problem in scheduling the ROSAT space telescope. European Journal of Operational Research 112 (3), 531–541.

Olsson, F., 2009. Optimal policies for inventory systems with lateral transshipments. International Journal of Production Economics 118 (1), 175-184.

Öncan, T., 2007. A Survey of the Generalized Assignment Problem and Its Applications. INFOR 45 (3), 123–141.

Osman, I.H., 1995. Heuristics for the generalized assignment problem, simulated annealing and tabu search approaches. OR Spectrum 17, 211–225.

Osman, I.H., Christofides, N., 1994. Capacitated clustering problems by hybrid Simulated Annealing and Tabu Search. International Transactions in Operational Research 1 (3), 317–336.

Owen, S.H., Daskin, M.S., 1998. Strategic facility location: A review. European Journal of Operational Research 111, 423–447.

Paterson, C., Kiesmüller, G., Teunter, R., Glazebrook, K., 2011. Inventory models with lateral transshipments: A review. European Journal of Operational Research 210 (2) 125-136.

Paterson, C., Teunter, R., Glazebrook, K., 2012. Enhanced lateral transshipments in a multi-location inventory system. European Journal of Operational Research 221 (2), 317-327.

Pentico, D.W., 2007. Assignment problems: A golden anniversary survey. European Journal of Operational Research 176 (2), 774–793.

Pigatti, A., de Aragao, M.P., and Uchoa, E., 2004. Stabilized branchand- cut-and-price for the generalized assignment problem. Electronic Notes in Discrete Mathematics, 19, 389–395.

Pollak, R.A., 1976. Interdependent preferences. American Economic Review 66 (3), 309-320.

Postlewaite, A., 1998. The social basis of interdependent preferences. European Economic Review 42 (3-5), 779-800.

Privault, C. and Herault, L., 1998. Solving a real-world assignment problem with a metaheuristic. Journal of Heuristic, 4 (4), 383–398.

Punnen, A.P., Aneja, Y.P., Categorized assignment scheduling, A Tabu search approach. Journal of the Operational Research Society 44 (7), 673–679.

Putnik, G., Sluga, A., ElMaraghy, H., Teti, R., Koren, Y., Tolio, T., Hon, B., 2013. Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap. CIRP Annals - Manufacturing Technology 62, 751–774.

Rabbani, M., Moghaddam, M., Manavizadeh, N., 2012. Balancing of mixed-model two-sided assembly lines with multiple U-shaped layout. The International Journal of Advanced Manufacturing Technology 59 (9-12), 1191–1210.

Rajan, V.N., Nof, S.Y., 1996. Cooperation requirement planning (CRP) for multi-processors: optimal assignment and execution planning. Journal of Intelligent and Robotic Systems 15 (4), 419–435.

Ravindran, A., Ramaswami, V., 1977. On the bottleneck assignment problem, Journal of Optimization Theory and Applications 21 (4), 451–458.

Rekiek, B., Delchambre, A., 2005. Assembly line design: the balancing of mixed-model hybrid assembly lines with genetic algorithms. Springer series in advanced manufacturing, 1st ed. Springer, London.

Reyes Levalle, R., Nof, S.Y., 2015. Resilience by teaming in supply network formation and re-configuration. International Journal of Production Economics 160, 80–93.

Risse, S., 2011. Two-stage group rent-seeking with negatively interdependent preferences. Public Choice 147 (3-4), 259-276.

Romanowski, A., Wozniak, P., Jaworski, T., Fiderek, P., Kucharski, J., 2013. Modelling interpersonal relations in surgical teams with fuzzy logic. Lecture Notes in Computer Science 7629 LNAI (1), 469-479.

Ross, G.T. and Soland, R.M., 1977. Modeling facility location problems as generalized assignment problems. Management Science, 24, 345–357.

Roth, A.E., 2007. What have we learned from market design? Working paper, National Bureau of Economic Research 13530.

Roth, A.E., Xing, X., 1994. Jumping the Gun, Imperfections and Institutions Related to the Timing of Market Transactions. American Economic Review 84, 992-1044.

Roth, A.E., 2008. Deferred acceptance algorithms, history, theory, practice, and open questions. International Journal of Game Theory 36 (3-4), 537–569.

Ruland, K.S., 1999. A model for aeromedical routing and scheduling. International Transactions in Operational Research, 87, 57–73.

Sahni, S., Gonzalez, T., 1976. P-complete approximation problems. Journal of the ACM 23 (3), 555-565.

Sarin, S.C., Erel, E., Dar-El, E.M., 1999. A methodology for solving single-model, stochastic assembly line balancing problem. Omega 27 (5), 525–535.

Savelsbergh, M.W.P., 1997. A branch-and-price algorithm for the generalized assignment problem. Operations Research 45, 831–841.

Scarelli, A., Narula, S.C., 2002. A multicriteria assignment problem. Journal of Multi-Criteria Decision Analysis 11 (2), 65–74.

Seliger, G, Krützfeldt, D., 1999. Agent-Based Approach for Assembly Control. Annals of the CIRP 48 (1), 21-24.

Seok, H., Nof, S.Y., 2013. Collaborative capacity sharing among manufacturers on the same supply network horizontal layer for sustainable and balanced returns. International Journal of Production Research. In print

Seok, H., Nof, S.Y., 2014. Collaborative capacity sharing among manufacturers on the same supply network horizontal layer for sustainable and balanced returns. International Journal of Production Research 52 (6), 1622-1643.

Shen, W., Hao, Q., Yoon, H.J., Norrie, D.H., 2006. Applications of agent-based systems in intelligent manufacturing: An updated review. Advanced Engineering Informatics 20 (4), 415–431.

Shen, W., Lang, S., Wang, L., 2005. iShopFloor: An internet-enabled agent-based intelligent shop. IEEE Transactions on Systems, Man, and Cybernetics, Part C 35 (3), 371–381.

Shen, W., Hao, Q., Wang, S., Li, Y., Ghenniwa, H., 2007. An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. Robotics and Computer-Integrated Manufacturing 23, 315-325.

Shieh, H-M. and May, M-D., 2001. Solving the capacitated clustering problem with genetic algorithms. Journal of the Chinese Institute of Industrial Engineers, 18 (3), 1–12.

Shigeno, M., Saruwatari, Y., Matsui, T., 1995. An algorithm for fractional assignment problems. Discrete Applied Mathematics 56 (2–3), 333–343.

Shtub, A., 1989. Modelling group technology cell formation as a generalized assignment problem. International Journal of Production Research, 27 (5), 775–782.

Singh, K.V., 2012. Impact of Cyber-Supported Collaboration on the Significance of Decision Factors of Regional Headquarters. Technical Report, Purdue University.

Sobel, J., 2005. Interdependent preferences and reciprocity. Journal of Economic Literature 43 (2), 392–436.

Sokkalingam, P.T., Aneja, Y.P., 1998. Lexicographic bottleneck combinatorial problems. Operations Research Letters 23 (1), 27–33.

Song, B.L., Wong,, W.K., Fan, J.T., Chan, S.F., 2006. A recursive operator allocation approach for assembly line-balancing optimization problem with the consideration of operator efficiency. Computers and Industrial Engineering 51 (4), 585–608.

Sprechter, A, 1999. A competitive branch-and-bound algorithm for the simple assembly line balancing problem. International Journal of Production Research 37 (8), 1787–1816.

Sultan, N., 2010. Cloud computing for education: A new dawn? International Journal of Information Management 30, 109-116.

Taillard, E.D., 1995. Comparison of iterative searches for the quadratic assignment problem. Location Science 3 (2), 87-105.

Talbi, E.-G., Roux, O., Fonlupt, C., Robillard, D., 2001. Parallel ant colonies for the quadratic assignment problem. Future Generation Computer Systems 17 (4), 441–449.

Tamiz, M., Jones, D.F., 1996. Goal programming and Pareto efficiency. Journal of Information and Optimization Sciences 17 (2), 291-307.

Tamiz, M., Mirrazavi, S.K., Jones, D.F., 1999. Extensions of Pareto efficiency analysis to integer goal programming. Omega 27, 179-188.

Tiacci, L., Saetta, S., 2011. Reducing the mean supply delay of spare parts using lateral transshipments policies. International Journal of Production Economics 133 (1), 182–191.

Tomes, N., 1986. Income distribution, happiness, and satisfaction: A direct test of the interdependent preferences model. Journal of Economic Psychology 7 (4), 425-446.

Torabi, S.A., Moghaddam, M., 2012. Multi-site integrated production-distribution planning with transshipment: a fuzzy goal programming approach. International Journal of Production Research 50 (6), 1726-1748.

Trentesaux, D., 2009. Distributed control of production systems. Engineering Applications of Artificial Intelligence 22 (7), 971–978.

Trick, M.A., 1992. A linear relaxation heuristic for the generalized assignment problem. Naval Research Logistics, 39, 137–152.

Tsui, L.Y., Chang, C.-H., 1992. An optimal solution to a dock door assignment problem. Computers and Industrial Engineering 23 (1–4), 283–286.

Valckenaers, P., Bonneville, F., Van Brussel, H., Bongaerts, L., Wyns, J., 1994. Results of the Holonic System Benchmark at KULeuven. Proceedings of the Fourth Int. Conf. on CIM and Automation Technology, New York, 128-133.

Vartak, M.N., Geetha, S., 1990. Specially structured precedence constraints in three-dimensional bottleneck assignment problems, Journal of the Operational Research Society 41 (4), 339–344.

Velásquez, J.D., Nof, S.Y., 2007. A Best Matching Protocol for Supplier Selection in e-Work Networks. Proceedings of the 19th International Conference on Production Research.

Velásquez, J.D., Nof, S.Y., 2008[a]. A best-matching protocol for collaborative e-Work and e-Manufacturing, International Journal of Computer Integrated Manufacturing 21(8), 943–956.

Velásquez, J.D., Nof, S.Y., 2008[b]. Integration of machine vision inspection information for best-matching of distributed components and suppliers, Computers and Industrial Engineering 59(1), 1–69.

Volgenant, A., 1996. Linear and semi-assignment problems, A core oriented approach. Computers and Operations Research 23 (10), 917–932.

Votaw, D.F., Orden, A. 1952. The personnel assignment problem. Symposium on Linear Inequalities and Programming, SCOOP 10, US Air Force, 155–163.

Wang, C.H., Paredis, C.J.J., 1999. On-Line Planning of Flexible Assembly Systems: An Agent-Based Approach. The 15th ISPE/IEE International Conference on CAD/CAM, Robotics, and Factories of the Future.

Wilson, J.M., 1997. A simple dual algorithm for the generalized assignment problem. Journal of Heuristics, 2 (4), 303–311.

Xu, X., 2012. From cloud computing to cloud manufacturing. Robotics and Computer-Integrated Manufacturing 28, 75–86.

Yagiura, M., Iwasaki, S., Ibaraki, T., and Glover, F., 2004[a]. A very large-scale neighborhood search algorithm for the multi-resource generalized assignment problem. Discrete Optimization 1, 87–98.

Yagiura, M., Ibaraki, T., Glover, F., 2004[b]. An ejection chain approach for the generalized assignment problem. INFORMS Journal on Computing 16, 133-151.

Yagiura, M., Yamaguchi, T., and Ibaraki, T., 1999. A variable depth search algorithm for the generalized assignment problem. In Metaheuristics, advances and trends in local search paradigms for optimization; Kluwer Academic Publisher Boston, MA, 459–471.

Yang, S., Allenby, G.M., 2003. Modeling interdependent consumer preferences. Journal of Marketing Research 40 (3), 282-294.

Yilmaz, G., Peña, J., 2014. The Influence of Social Categories and Interpersonal Behaviors on Future Intentions and Attitudes to Form Subgroups in Virtual Teams. Communication Research 41 (3), 333-352.

Yoon, S.Y., Nof, S.Y., 2010. Demand and capacity sharing decisions and protocols in a collaborative network of enterprises. Decision Support Systems 49 (4), 442–450.

Yoon, S.Y., Nof, S.Y., 2011. Affiliation/dissociation decision models in demand and capacity sharing collaborative network. International Journal of Production Economics 130 (2), 135–143.

Yu, Y. and Prasanna, V.K., 2003. Resource allocation for independent real-time tasks in heterogeneous systems for energy minimization. Journal of Information Science and Engineering, 19 (3), 433–449.

Yuan, Y., Mehrez, A., Gafni, A., 1992. Reducing bias in a personnel assignment process via multiplicative utility solution. Management Science 38 (2), 227–239.

Zhao, X., Atkins, D., 2009.Transshipment between competing retailers. IIE Transactions 41 (8), 665-676.

Zimokha, V.A. and Rubinstein, M.I., 1988. R & D planning and the generalized assignment problem. Automation and Remote Control, 49, 484–492.

VITA

VITA

MOHSEN MOGHADDAM

EDUCATION

Purdue University, West Lafayette, IN, USA

Ph.D., Industrial Engineering (Major Advisor: Prof. Shimon Y. Nof) – 4.0 GPA 2016

University of Tehran, Tehran, Iran

M.S., Industrial Engineering (Major Advisor: Prof. Ali Azadeh) – 4.0 GPA 2011

University of Tehran, Tehran, Iran

B.S., Industrial & Systems Engineering – 3.8 GPA 2009

PROFESSIONAL EXPERIENCES

Researcher, PRISM Center, Purdue University 2012-Present

Design of models, algorithms, and protocols based on operations research, computer and information technology, and artificial intelligence for distributed and collaborative control of e-Work, e-Manufacturing, and e-Service systems.

Consultant, Technical Assistance Program, Purdue University 2015-Present

Development and implementation of systems engineering solutions to enhance the productivity of private and public manufacturing, healthcare, and service industries in Indiana.

Business Engineering Analytics Developer, Kimberly-Clark (KC) Corp. 2012-2013

Design of algorithms and proof-of-concept software implementation in KC-LAO TEMPOS projects on e-Commerce, e-Procurement, supplier portfolio selection, distribution network design for resilience, warehouse and logistics control.

Sales Manager, German-Iranian Chamber of Commerce (AHK-Iran) 2011-2012
Marketing, advertisement, operations management; interaction, negotiation, and interview with customers at the Education department of AHK-Iran.

TEACHING EXPERIENCES

Purdue University – TA and mentor for Senior Design (IE431) ◊ Mentor for BS and MS students in summer internships and thesis (IE698) ◊ TA and lecturer for Design for e-Work & e-Service (IE588) ◊ TA and lecturer for Design & Control of Production and Manufacturing Systems (IE579)

University of Tehran – TA for Production Planning, Transportation Planning, Maintenance Planning, Fundamentals of Management

RESEARCH EXPERIENCES

Grant Proposal Preparation. Boeing Analytics Research Scholars (Supplier Selection) ◊ Office of Naval Research (Advances in Collaboration Sciences) ◊ Department of Defense (Intelligent and Adaptive Teaming in Social Networks) ◊ National Science Foundation (Brain-Inspired Distributed Control Architecture) ◊ Purdue Center for Global Food Security (Systems Science for Food Production and Supply Security)

Reviewer. Int. J. Production Economics ◊ Int. J. Production Research ◊ Int. J. Computer Integrated Manufacturing ◊ IEEE Transactions on Systems, Man, and Cybernetics ◊ Int. J. Information Technology & Decision-Making ◊ Neural Computing & Applications

Citation Indices (Google Scholar)**.** *Citations*: 307 ◊ *h-index*: 11 ◊ *i10-index*: 13

AWARDS

Travel Grant for Future Eng. Faculty and Professionals, Purdue University, 2015

BIW EB-2 USA Immigration – Department of Homeland Security, 2015

Second Place – Purdue University Intramural Indoor Soccer League, 2015

Best Paper – The 22nd International Conference on Production Research, Brazil, 2013

Best Graduate Researcher – University of Tehran, Iran, 2012

Graduate School Entrance Exam Waiver – University of Tehran, Iran, 2009

Faculty of Engineering (FoE) Award – University of Tehran, BS & MS, 2009 & 2012

PUBLICATIONS

PUBLICATIONS

Books

B2. **Moghaddam M.**, Nof S.Y. Best Matching Theory & Applications. Springer. *Forthcoming* (2016).

B1. Nof S.Y., Ceroni J., Jeong W., **Moghaddam M**. Revolutionizing Collaboration through e-Work, e-Business, and e-Service. Springer, 2015.

Journal Articles (PhD)

J9. **Moghaddam M.**, Nof S.Y., 2015. Best matching with interdependent preferences—implications for capacitated cluster formation and evolution. *Decision Support Systems 79*, 125–137.

J8. **Moghaddam M.**, Nof S.Y., 2015. The collaborative factory of the future. *Int. J. Computer Integrated Manufacturing* (Special Issue: “Factories of the Future: Challenges and Leading Innovations in Intelligent Manufacturing”). DOI:10.1080/0951192X.2015.1066034.

J7. **Moghaddam M.**, Nof S.Y., 2015. Balanceable assembly lines with dynamic tool sharing and best matching protocols—a collaborative assembly framework. *IIE Transactions 47*, 1363–1378.

J6. **Moghaddam M.**, Nof S.Y., 2015. Real-time optimization and control mechanisms for collaborative demand and capacity sharing. *Int. J. Production Economics 171*, 495–506 (Special Issue: “Challenges for Sustainable Operations – Selected papers of ICPR 2013”).

J5. **Moghaddam M.**, Nof S.Y., Menipaz E., 2015. Design and administration of collaborative networked headquarters. *Int. J. Production Research* (Special Issue: "Distributed Manufacturing to Enhance Productivity"). Accepted.

J4. **Moghaddam M.**, Nof S.Y., 2015. Real-time administration of tool sharing by best matching to enhance assembly lines balanceability and flexibility. *Mechatronics 31*, 147–157 (Special Issue: "New Trends in Intelligent Manufacturing").

J3. **Moghaddam M.**, Nof S.Y., 2014. Combined demand and capacity sharing with best matching decisions in enterprise collaboration. *Int. J. Production Economics 148*, 93–109.

J2. Li J., **Moghaddam M.**, Nof S.Y., 2015. Dynamic storage assignment with product affinity and ABC classification—a case study. *Int. J. Advanced Manufacturing Technology*. DOI:10.1007/s00170-015-7806-7.

J1. Zhong H., Reyes Levalle R., **Moghaddam M.**, Nof S.Y., 2015. Collaborative intelligence—definition and measured impacts on internetworked e-Work. *Management and Production Engineering Reviews 6*, 67–78.

Under Review

J11. **Moghaddam M.**, Nof S.Y., Collaborative location-allocation decisions in networked *v*-Organizations. *Computers and Industrial Engineering*.

J10. **Moghaddam M.**, Nof S.Y., Parallelism of pick-and-place operations by multi-gripper robotic arms. *Robotics and Computer Integrated Manufacturing*.

Journal Articles (Other)

J23. Azadeh A., Tohidi H., Zarrin M., Pashapour S., **Moghaddam M.**, 2016. An integrated algorithm for performance optimization of neurosurgical ICUs. *Expert Systems with Applications 43*, 142–153.

J22. Azadeh A., **Moghaddam M.**, Nazari T., Sheikhalishahi M., 2015. Optimization of facility layout design with ambiguity by an efficient fuzzy multivariate approach. *Int. J. Advanced Manufacturing Technology*. DOI:10. 1007/s00170-015-7714-x.

J21. Azadeh A., **Moghaddam M.**, Nazari-Doust B., Jalalvand F., 2015. Fuzzy and stochastic mathematical programming for optimization of cell formation problems in random and uncertain states. *Int. J. Operational Research 22*, 129–147.

J20. Azadeh A., Rahimi-Golkhandan A., **Moghaddam M.**, 2014. Location optimization of wind power generation-transmission systems under uncertainty using hierarchical fuzzy DEA: A case study. *Renewable and Sustainable Energy Reviews 30*, 877–885.

J19. Azadeh A., Maleki-Shoja B., **Moghaddam M.**, Asadzadeh S.M., Akbari A., 2013. A neural network meta-model for identification of optimal combination of priority dispatching rules and makespan in a deterministic job shop scheduling problem. *Int. J. Advanced Manufacturing Technology 67*, 1549–1561.

J18. Azadeh A., **Moghaddam M.**, Mahdi M., Seyedmahmoudi S.H., 2013. Optimum long-term electricity price forecasting in noisy and complex environments. *Energy Sources 8*, 235–244.

J17. Torabi S.A., **Moghaddam M.**, 2012. Multi-site integrated production-distribution planning with transshipment: A fuzzy goal programming approach. *Int. J. Production Research 50*, 1726–1748.

J16. Rabbani M., **Moghaddam M.**, Manavizadeh M., 2012. Balancing of mixed-model two-sided assembly lines with multiple U-shaped layout. *Int. J. Advanced Manufacturing Technology 59*, 1191–1210.

J15. Azadeh A., **Moghaddam M.**, Khakzad M., 2012. A flexible neural network-fuzzy mathematical programming algorithm for improvement of oil price estimation and forecasting. *Computers and Industrial Engineering 62*, 421–430.

J14. Azadeh A., Ziaei B., **Moghaddam M.**, 2012. A hybrid fuzzy regression-fuzzy cognitive map for forecasting and optimization of housing market fluctuations. *Expert Systems with Applications 39*, 298–315.

J13. Azadeh A., Negahban A., **Moghaddam M.**, 2011. A hybrid computer simulation-artificial neural network algorithm for optimization of dispatching rule selection in stochastic job shop scheduling problems. *Int. J. Production Research 50*, 551–566.

J12. Azadeh A., **Moghaddam M**., Asadzadeh S.M., Negahban A., 2011. An integrated fuzzy simulation-fuzzy data envelopment analysis algorithm for job-shop layout optimization: The case of injection process with ambiguous data. *European Journal of Operational Research 214*, 768–779.

J11. Azadeh A., Ghaderi S.F., Mirjalili M., **Moghaddam M**., 2011. Integration of analytic hierarchy process and data envelopment analysis for assessment and optimization of personnel productivity in a large industrial bank. *Expert Systems with Applications 38*, 5212–5225.

J10. Azadeh A., **Moghaddam M**., Geranmayeh P., Naghavi A., 2010. A flexible artificial neural network-fuzzy simulation algorithm for scheduling a flow shop with multiple processors. *Int. J. Advanced Manufacturing Technology 50*, 699–715.

J9. Azadeh A., Eivazy H., **Moghaddam M**., 2010. A flexible and dynamic algorithm for assessment and optimization of utility sectors. *Advanced Engineering Informatics 24*, 498–509.

J8. Azadeh A., Saberi M., Anvari M., **Moghaddam M**., 2010. An integrated artificial neural network K-means algorithm for performance assessment of electricity distribution units. *Journal of Scientific and Industrial Research 69*, 672–679.

J7. Azadeh A., Nouri M., **Moghaddam M**., 2013. Optimization of complex and large-sized single-row facility layout problems with a unique hybrid meta-heuristic framework. *Int. J. Operational Research 16*, 38–67.

J6. **Moghaddam M**., Rabbani M., Maleki-Shoja, B., 2012. Integrating lateral transshipment to aggregate production-distribution planning considering time value of money and exchange rate. *Int. J. Operational Research 13*, 439–464.

J5. Azadeh A., Naghavi A., **Moghaddam M**., 2011. A hybrid artificial neural network computer simulation approach for optimization of flow shop with multiple processors problem. *Int. J. Industrial and Systems Engineering 7*, 66–89.

J4. Azadeh A., Sharifi S., Izadbakhsh H., **Moghaddam M**., 2011. Integration of expert system and integer programming for optimization of strategic planning. *Int. J. Industrial and Systems Engineering 7*, 110–133.

J3. Azadeh A., Beheshtipour M., **Moghaddam M**., 2011. An artificial neural network approach for improved demand estimation of a cool-disk manufacturer. *Int. J. Industrial and Systems Engineering 7*, 357–380.

J2. Azadeh A., Ghaderi S.F., Mirjalili M., **Moghaddam M**., 2010. A DEA approach for ranking and optimization of technical and management efficiency of a large bank based on financial indicators. *Int. J. Operational Research 9*, 160–187.

J1. Azadeh A., Karimi A., Keramati, A., **Moghaddam M**., 2010. A multi-objective genetic algorithm for scheduling optimization of M job families on a single machine. *Int. J. Industrial and Systems Engineering 6*, 417–440.

Conferences (PhD)

C7. **Moghaddam M**., Nof S.Y., 2015. Collaborative networked v-Organizations: design and integration. *INFORMS Annual Meeting*, Philadelphia, USA.

C6. **Moghaddam M**., Silva J.R., Nof S.Y., 2015. Manufacturing-as-a-service—from e-Work and service-oriented architecture to the cloud manufacturing paradigm. *IFAC Symposium on Information Control in Manufacturing*, Ottawa, Canada.

C5. **Moghaddam M**., Nof S.Y., 2015. Interdependent preferences—definitions and impacts on team formation. *The 2015 Industrial and Systems Engineering Research Conference*, Nashville, USA.

C4. **Moghaddam M**., Nof S.Y., 2014. Location-allocation decisions in collaborative networks of service enterprises. *The 2014 Industrial and Systems Engineering Research Conference*, Canada.

C3. **Moghaddam M**., Nof S.Y., 2013. Best matching and task administration protocols for effective demand and capacity sharing. *Proceedings of the 22nd International Conference on Production Research*, Brazil.

C2. **Moghaddam M**., Nof S.Y., 2013. Dynamic tool sharing with best matching protocols for efficient assembly line balancing. *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems*, Brazil.

C1. Gnanasekaran A.M., Moshref-Javadi M., Zhong H., **Moghaddam M**., Lee S., 2013. Impact of patients priority and resource availability in ambulance dispatching. *The 2013 Industrial and Systems Engineering Research Conference*, Puerto Rico.