PLATFORM-DRIVEN CROWDSOURCED MANUFACTURING FOR MANUFACTURING AS A SERVICE

A Dissertation
Presented to
The Academic Faculty

by

Xuejian Gong

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2021

PLATFORM-DRIVEN CROWDSOURCED MANUFACTURING FOR MANUFACTURING AS A SERVICE

Approved by:

Dr. Roger J. Jiao, Co-Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Nagi Gebraeel, Co-Advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Seung-Kyum Choi
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Steven Y. Liang
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Donggang Yao
School of Materials Science & Engineering
Georgia Institute of Technology

Dr. Manpreet S. Hora
College of Business
Georgia Institute of Technology

Date Approved: December 1, 2021

ACKNOWLEDGEMENTS

It would have been impossible for me to finish my doctoral dissertation without the help and support of the people around me, the excellent research environment provided by, and the generous funding supported by Georgia Institute of Technology.

First and foremost, I would like to take this opportunity to express my sincerest gratitude and appreciation to my co-advisors, Dr. Roger J. Jiao and Dr. Nagi Gebraeel, at Georgia Institute of Technology, for their guidance, encouragement, and inspiration throughout Ph. D. study. I greatly appreciate Dr. Roger J. Jiao and Dr. Nagi Gebraeel, who help shape my research ability and carry me on through difficult times with their insights and suggestions.

Special thanks to my thesis reading committee, including Dr. Seung-Kyum Choi (Mechanical Engineering), Dr. Steven Y. Liang (Mechanical Engineering), Dr. Donggang Yao (Materials Science & Engineering), and Dr. Manpreet S. Hora (College of Business) for their precious time and suggestions to improve my dissertation in various ways.

I also would like to thank faculty members, staff, and students in Renewable Bioproducts Institutes, Georgia Institute of Technology, for their supportive conduit for industry-academia partnerships and their fellowship support for my Ph. D. research. I would like to thank Dr. Norman F. Marsolan, Dr. Chris Luettgen, Dr. J. Carson Meredith, Dione N Morton, and Virginia Howell for their help in my research and career advising.

I also would like to thank all the students in Room 264B GTMI building, staff at School of Mechanical Engineering at Georgia Institute of Technology who are always ready to help me in times of need. I would like to thank Glenda Johnson, Henry Camellia, Dr. Sungkun Hwang, Dr. Ruoyu Song, Dr. Jonas Landahl, Dr. Feng Zhou, Dr. Yitao Liu, Dr. Jun Du, Dr. Liang Hou, Dr. Wenchong Chen, Dr. Jiexiang Hu, Tianyi Lu, Shu Wang, Pan Zou, Mulang Song, David Mitchell, Dr. Hume Chad, and Yiyun Fei.

Last but not least, great thanks go to my family and friends for always being there when I need them most, and for supporting me through all these years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

O	Open innovator
C^0	Customer order
M	Manufacturing agents
α, A	Manufacturing agent cluster index and total number
$\mu_{n_\alpha}^{\alpha}$	Bidding manufacturing agent n_α in cluster α
n_α, N_α	Manufacturing agent index and total number in cluster α
φ_n	Non-bidding
D^0	Design specs
P^0	Process specs
$P_\alpha^I, \bar{P}^I$	Invitation broker in cluster α and its universe
$P_\alpha^E, \bar{P}^E$	Evaluation broker in cluster α and its universe
P^C	Project configuration manager
$\bar{P}$	Platform brokers
δ_k, Δ	Manufacturing subtask and its associated product
k, K	Index and total number of subtasks
Δ_α	RFQ for cluster α
B_α, B	Bids set in cluster α and its universe
$\mu^{\alpha *}$	Winning manufacturing agent in cluster α
S	Manufacturing supply contracts
B_α^*, B^*	Winning manufacturing bids and process spec set
λ, Λ	Initiated project index and total number of
π, Π	Manufacturing index and total number

$O_\lambda, \bar{P}_\lambda$	Open innovator and platform broker for project λ
$u(Pr)$	Preference function of expected function fulfillment range
$p(Pr)$	Probability distribution function of performance range
Pr	Performance variable
Fr^L, Fr^U	Upper and lower limits of requirement ranges
Pr^L, Pr^U	Lower and upper limits of performance range
$\mathbf{I}$	Information content
$P(Pr)$	Probabilities of successfully fulfill the expected performance
$c_{r_\alpha}^{\mathrm{F}_\alpha}, \mathbf{c}^{\mathrm{F}_\alpha}$	Evaluation criteria of r_α and its collective set
r_α, R_α	Index and total number of criteria
w_{r_α}	Weighting factor for criteria
A_p	Instance space with attributes p
x_n	Objects in the instance space A
D	Ordinal decisions or labels of these objects
$v(x_i, a)$	Value of the attribute or the decision related to x_i
f	Predicting function
T	decision tree
$R(\mathrm{T})$	prediction error of the tree
$\lvert\widetilde{\mathrm{T}}\rvert$	the number of leaf nodes
ϑ	regularization parameter
i, I	Index and number of market segments
j, J	Index and number of product variants in the product family
k, K	Index and number of modules including common modules and selective modules
l_k, L_k	Index and number of module instance for the k-th module

N^C	Number of competitive products in the market
U_{ij}	Conjoint utility of the j-th product variant for the i-th market segment
Q_i	Revenue of each product variant in the i-th market segment
u_{ikl}	Part-worth utility of the l-th module instance of the k-th module for the i-th market segment
r_{kl}	Cost for buying one unit of the l-th module instance of the k-th module
θ	A positive scaling parameter of the MNL model
U_{ij}^C	Conjoint utility of the j-th competitive product for the i-th market segment
t_{kl}	Manufacturing time for the l-th module instance of the k-th module
C^{fix}	Fixed cost for each manufacturer
C^{var}	Variable operation cost for each manufacturer per unit time
t_k^{PF}	Product family manufacturing time for the k-th module
T^O	Planned life of the crowdsourced manufacturing operations
$Pre(k)$	Set of all the direct and indirect predecessors of the k-th task
x_{jkl}	Binary decision variable such that $x_{jkl} = 1$ if the lth module instance of the kth module is selected for the jth product variant, and $x_{jkl} = 0$ otherwise
y_j	Binary decision variable such that $y_j = 1$ if the j-th product variant is selected, and $y_j = 0$ otherwise
$z_{k\pi}$	Binary decision variable such that $z_{k\pi} = 1$ if the k-th task is assigned to the π-th manufacturer, and $z_{k\pi} = 0$ otherwise
w_π	Integer decision variable which indicates the number of service providers for the π-th manufacturer
Z_1, Z_2, Z_3, Z_4	Common deviation function
$\mathcal{G}$	Directed graph for C-VRPCD
$\mathcal{V}$	All possible locations for C-VRPCD

$\mathcal{A}$	All possible trips as arcs for C-VRPCD
$\{0\}$	Cross-dock depot
$\{\vert 0\vert\}$	Dummy cross-dock
M^U	Upstream manufacturers
M^D	Downstream manufacturers
n^U	Total number of upstream manufacturers M^U
n^D	Total number of downstream manufacturers M^D
$\mathcal{A}^U$	Denotes all possible arcs connecting M^U, $\{0\}$, and $\{\vert 0\vert\}$
$\mathcal{A}^D$	Denotes all possible arcs connecting M^D, $\{0\}$, and $\{\vert 0\vert\}$
c_{ij}	Transportation cost
$\mathcal{S}$	Logistic service requests
K	Fleet size
R^P	Pickup routes
R^D	Delivery routes
β_r^k	Decision variable to indicate a pickup tour r utilize vehicle k
c_r	Transportation cost attached to pickup tour r
$\gamma_{r'}^k$	Decision variable to indicate delivery route use vehicle k
$c_{r'}$	Transportation cost attached to delivery route r'
$\tau_{i^U}^k$	Exchanging decision variable to indicate the load i^U in vehicle k is exchanged in cross-dock depot
$c_{i^U}^k$	Exchanging cost attached to the load i^U in vehicle k is exchanged in cross-dock depot
$a_r^{i^U}$	Binary parameter describes the pickup route r whether it visit upstream manufacturer i^U
$b_{r'}^{i^D}$	Binary parameter describes the delivery route r' whether it visit downstream manufacturer i^D

$\upsilon^k, \varphi^k, \nu_{i^U}, \mu_{i^D}, \pi_i^k, \chi_i^k$	Dual variables of constraints of RLMP of C-VRPCD
$\mathcal{P}$	Partial path of ESPPRC
$r(\mathcal{P})$	Cumulative reduced cost of the partial path $\mathcal{P}$
$q(\mathcal{P})$	Cumulative path load of the partial path $\mathcal{P}$
$t(\mathcal{P})$	Cumulative path time consumption of the partial path $\mathcal{P}$
$\mathcal{v}_s$	Start nodes
$\mathcal{v}_e$	End node
$\mathcal{A}_i^+$	The set of accessible nodes set of current one $\mathcal{v}_i$
$\mathcal{P}^*$	The best-performed path
$\underline{r}(\mathcal{v}_i, t(\mathcal{P}))$	The lower bounds for every node
$[\underline{t}, \bar{t}]$	Time bound of planning horizon
$\mathfrak{d}$	Bound step size
$\mathbf{B}$	Lower bound matrix
t_O	Time of real-time order O starts recommendation
$\kappa_{i^U}^k$	Branching uncertainty index
$\mathcal{M}$	Total number of machines in a flow line
Ω_1	Set of new generated orders
Ω_2	Set of existing orders
t_0	Past time horizon
q_j	Job quantity of order j
a_j	Earliest release date of order j
b_j	Latest release date of order j
p_{ij}	Processing time of order j on machine i

d_j	Due date of order *j*
e_j	Revenue of order *j*
$w_{1,j}$	Tardiness penalty coefficient of order *j*
$w_{2,j}$	Unit inventory cost of order j for a final product
$w_{3,j}$	Unit cost for WIP inventory
$t_i(k, k')$	Sequence-dependent setup time of job k immediately precedes job k'
π_j	Net revenue of an order *j*
$u_\tau(k_0 + k)$	Release time of part τ of job k
$\mathcal{E}_{m\times n}$	$m \times n$ max-plus algebraic zero matrix
E_n	$n \times n$ max-plus algebraic identity matrix
P_i	Places buffer of machine in TEG
B_i	Buffer of machine in TEG
P_i'	Real-time WIP in TEG
P_i''	The restart of the corresponding machine
N_i	Capacity of buffer
$x_i(k)$	Time instant at which the machine m_i begins working on the k-th job
$\sigma_i(k)$	Job processing time of machine i for the k-th job
M_τ	Machine quantity of the τ-th sub-line
M_{P+1}	Machine quantity of the main-line
C_j	Order completion time of order *j*
Ω	Set of all orders
$I_{current}$	Current number of iterations

LIST OF ABBREVIATIONS

ICT	Information and Communications Technologies
IoTs	Internet of Things
CIMS	Computer Integrated Manufacturing System
SCM	Supply Chain Management
MBSE	Model-Based System Engineering
CPS	Cyber-Physical Systems
MES	Manufacturing Execution System
ERP	Enterprise Resource Planning
CNs	Customer Needs
FRs	Functional Requirements
DPs	Design Parameters
PVs	Process Variables
MaaS	Manufacturing-as-a-Service
WIP	Work-In-Progress
F/LOSS	Free/Libre Open-Source Software
HOPE	Haier Open Partnership Ecosystem
MTO	Make-to-Order
OEM	Original Equipment Manufacturer
RFQ	Request for Quotation
m-VRPTW	Multiple Vehicle Route Planning with Time Window
FBS	Functional, Behavioral, And Structural
BMP	Best-Matching Problem

DoS	Degree of Satisfaction
MAS	Multi-agent System
DES	Discrete Event Simulation
PDF	Probability Distribution Function
RMI	Rank Mutual Information
PFP	Product Family Planning
MLB	Manufacturer Loading Balancing
LPP	Linear Physical Programming
MNL	Multinomial Logit
NBGA	Nested Bilevel Genetic Algorithm
GA	Genetic Algorithm
TBA	The Bilevel Approach
TSA	The Sequential Approach
C-VRPCD	Crowdsourcing Vehicle Routing Problem with Cross-Docking
MILP	Mixed-Integer Linear Programming
B&P	Branch-and-Price
VRPTW	Vehicle Routing Planning with Time Window
VRPCD	Vehicle Routing Problem with Cross-Docking
ESPPRC	Elementary Shortest Path Problem with Resource Constraints
RMP	Restricted Master Problem
r-OAS	real-time Order Acceptance and Scheduling
r-JRP	real-time Job Release Planning
BIO	Bilevel Interactive Optimization
OAS	Order Acceptance and Scheduling
r-OASR	Joint Optimization for r-OAS and r-JRP

TEG	Timed Event Graph
r-OAS-B	r-OAS Model with Bottleneck Machines
r-OAS-F	r-OAS Model for Permutation Flow Lines
r-OAS-B I	A Single r-OAS-B
r-JRP-B II	r-OAS-B with r-JRP
r-OAS-F I	A Single r-OAS-F
r-OAS-F II	r-OAS-F with r-JRP
BIO I	Does Not Cooperate with r-JRP
BIO II	Does Cooperate with r-JRP
PoW	Proof of Work
PoS	Proof of Stake
PoA	Proof of Authority
IPFS	Interplanetary File System
CID	Content Identifier
ECC	Evolutionary Competition-Cooperation
CUI	Capacity Unbalance Index
ESS	Evolutionary Stable Strategies
NE	Nash Equilibrium

SUMMARY

Platform-driven crowdsourced manufacturing is an emerging manufacturing paradigm to instantiate the adoption of the open business model in the context of achieving Manufacturing-as-a-Service (MaaS). It has attracted attention from both industries and academia as a powerful way of searching for manufacturing solutions extensively in a smart manufacturing era. In this regard, this work examines the origination and evolution of the open business model and highlights the trends towards platform-driven crowdsourced manufacturing as a solution for MaaS. Platform-driven crowdsourced manufacturing has a full function of value capturing, creation, and delivery approach, which is fulfilled by the cooperation among manufacturers, open innovators, and platforms. The platform-driven crowdsourced manufacturing workflow is proposed to organize these three decision agents by specifying the domains and interactions, following a functional, behavioral, and structural mapping model. A MaaS reference model is proposed to outline the critical functions and inter-relationships among them. A series of quantitative, qualitative, and computational solutions are developed for fulfilling the outlined functions. The case studies demonstrate that the proposed methodologies and can pace the way towards a service-oriented product fulfillment process.

This dissertation originally proposes a manufacturing theory and decision models by integrating manufacturer crowds through a cyber platform. This dissertation reveals the elementary conceptual framework based on stakeholder analysis, including dichotomy analysis of industrial applicability, decision agent identification, workflow, and holistic framework of platform-driven crowdsourced manufacturing. Three stakeholders require

three essential service fields, and their cooperation requires an information service system as a kernel. These essential functions include contracting evaluation services for open innovators, task execution services for manufacturers, and management services for platforms. This research tackles these research challenges to provide a technology implementation roadmap and transition guidebook for industries towards crowdsourcing.

Accordingly, mathematical and computational models are developed within the framework to support: 1) value capture function from a functional view for manufacturing resource selection and aggregation; 2) value creation function from a behavioral view for crowdsourcing task execution decision making; 3) value delivery function from a structural view for crowd dynamics modeling and operational protocol revision; 4) cyber kernel as a prerequisite for information services of platform-driven crowdsourced manufacturing. These coherent technical elements along the service reference model lay the theoretical foundation of this research, as described below.

First, in order to search and select manufacturers through a crowdsourcing process, a crowdsourcing contracting mechanism incorporating explicit and inexplicit criteria evaluation methods is proposed. It perceives crowdsourcing product fulfillment efforts through a cyber platform as tournament-based crowdsourcing, formulated with various activities and symbolic systems. The challenge of a crowdsourcing contracting evaluation mechanism can be further decomposed as engineering functional and business operation reputational evaluation, with explicit and inexplicit criteria, respectively. This research proposes a quantitative methodology of manufacturers evaluation for engineering functional requirements based on information-content measurements and a decision-tree learning algorithm for business operational reputational evaluation of manufacturers through monotonic classification. The results of various criteria are aggregated through multi-attribute utility theory.

Second, based on the platform-driven crowdsourced manufacturing, a crowdsourcing task derivation method is proposed to optimally solve the tradeoff between product family planning and manufacturing load balancing. It formulates the reconfiguration of a series of innovative products as a product family planning problem from the front-end, and a manufacturing load planning problem from the back-end, which pursuits maximum sale profits and minimum unbalanced task segmentation, respectively. It can be mathematically modeled as a bilevel programming problem, solved by a proposed nested-bilevel genetic algorithm.

Third, a networked material flow management service through cross-docking is proposed to serve manufacturers to peel off their peripheral activities and concentrate on their leading competing edges. As a platform-driven logistics solution, cross-docking divides service routes into pickup and delivery routes and enhances overall efficiency by exploring similarities among routes and minimizing inventory cost at the depot. A branch-and-price algorithm is proposed to solve this large-scale combinatorial optimization problem efficiently. This logistic service decision-making can be modeled as a crowdsourcing vehicle routing problem with cross-docking. A pulse algorithm is applied to solve the pricing problem, and a branching heuristic is applied to solve the problem effectively and exactly.

Fourth, to support the optimal decision-making of manufacturers on production planning, a real-time order acceptance and accommodation methodology is established. It aims to serve manufacturers with optimal decision making for accepting crowdsourcing orders and mixing the incoming and existing orders incorporating the real-time status of the manufacturing facility. This problem can be modeled following a bilevel architecture with order acceptance and scheduling on the leader level as well as job release planning on the follower level. A construction methodology of digital twins of manufacturers' shop floor is proposed, and the algorithmic solution of this bilevel problem incorporating real-time status is established.

Fifth, blockchain-based information service systems are proposed to serve all stakeholders for crowdsourcing contracting, real-time status monitoring, and product fulfillment data management. Adopting blockchain-based smart contracts is a key to managing distributive databases to ensure security. The proposed architecture also incorporates smart sensing technologies and enables real-time informed decision-making in platform-driven crowdsourced manufacturing.

Sixth, an evolutionary competition-cooperation game model is developed to find a robust revision protocol to sustain a prosperous manufacturer population. It formulates the cohort decision-making process as an evolutionary model. Thus, enable behavioral modeling of adoption and reversion of crowdsourcing strategy in a multi-cluster manufacturer population, which serves operation excellence of platforms.

CHAPTER 1. INTRODUCTION

Manufacturing companies are confronted with challenges for satisfying various individual customer needs while more efficiently managing product variety for product development than their competitors (Brettel et al., 2017, Jiao et al., 2003). The extent of market-of-one has been foreseen as a potential driving force for next transformation of the global economy, leading to the traditional mass production paradigm being shifted to mass customization and personalization (Pine, 1993, Tseng et al., 2010). This paradigm shift introduces a large variety to not only the product domain but also the production domain. Thus, this change implies a higher variance of demands and markets, as well as a complex technology portfolio and dynamic supply chain structure (ElMaraghy et al., 2013).

Manufacturing companies utilize information sharing and coordination technologies to deal with the variety, where companies cooperate with a peer of suppliers through a series of fabrications to finish final products in the time deadlines (Sahin and Robinson Jr, 2005). Manufacturing companies usually own excess production capacities to avoid the violation of the deadline, while approaching manufacturing capabilities and resources from a scalable and changeable production network is a more efficient way to adjust capacities (Freitag et al., 2015). The cloud-based manufacturing resource sharing can explore the value of idle resources, as well as utilize the excess capabilities from the cooperation of the global supply chain (Wu and Yang, 2010). Following the vision of industry 4.0, the synergy of the highly customized products and intense competition challenges the manufacturers with a decreased product lifecycle, quick response to emerging technologies, and agile organization structure (Brettel et al., 2017).

1.1 Open Business Model

The open business strategy has been recognized as an efficient way of constructing a quick response problem-solving system in terms of utilizing external assets and knowledge to develop its own capabilities (Chesbrough and Appleyard, 2007). An open business strategy can be defined as participating in an open initiative to capture value, which can be divided into the reliance of the external assets or knowledge and the access to project results by external partners (Appleyard and Chesbrough, 2017). It has been witnessed that a growing amount of companies have adopted open business strategies and geared towards the installation of open business models through their business activities, which have achieved an agile enterprise structure and more massive capability arsenal (Kortmann and Piller, 2016). The open business model allows the creation of whole new complementary links in a value chain, which explicitly arranges the stakeholders along with positions of value creation, deliveries, and capturing (ElMaraghy and ElMaraghy, 2014). This openness provides the structure to integrate the various external partners to ease the installation of new technologies and scaling up of capabilities.

In a manufacturing environment, the external suppliers act as a subcontractor who serves the value chain with its core competence, while the core company can outsource its peripheral economic activities and focus on its essential competitive edges (Trentin et al., 2012). This outsourcing can decompose the volatile yet complex value chains to simpler jobs, which potentially enables the subcontractors to handle the variety by exploring the commonalities among the jobs to maximize the reutilization of their competitive edges (Jiao et al., 2007a). The concentration of the competitive edges will enhance the production volume in companies' core manufacturing activities to achieve economies of scale in a high product and production variety as well as a significant variance of demand environment (ElMaraghy et al., 2009). The mass cooperation of the manufacturers entails a highly interactive manufacturing network that relies on the cooperative collaboration mechanism along the value chain (Tapscott and Williams, 2008).

1.2 Platform-Driven Crowdsourcing

Crowdsourcing has been recognized as a connecting approach to installing the open business model by transcending organizational boundaries in order to leverage resources and capabilities across distributed stakeholders (Kohler, 2015). Different from the conventional strategy of outsourcing in supply chain management that emphasizes how to assign a task to a designated agent, crowdsourcing utilizes an open call to a crowd for maximally exploiting the external resources (Bücheler and Sieg, 2011). Crowdsourcing entails a new value-based model as a social-economic computational platform in which products and services are created and delivered in an open, collaborative, and distributed manner (Green et al., 2017). As a computational production platform, crowdsourcing is a large problem-solving model that utilizes Internet technologies to coordinate, negotiate, and manage the crowds for performing the specific organizational tasks (Saxton et al., 2013). It implies a superior broker system to coordinate the information and material flow among the stakeholder crowds and therefore enable the companies to crowdsource their peripheral activities and concentrate on their core competitive edges (Redlich and Bruhns, 2008).

Among many perspectives of crowdsourcing, we approached from a platform-driven method to peel out the coordinating and negotiating responsibilities from the crowdsourcer and eases the way of applying the open business model as the foundation of product innovation and development. Following a "four pillars" taxonomy of the crowdsourcing, namely the crowd, crowdsourcer, crowdsourced task, and crowdsourcing platform, precisely, the crowdsourcing platform plays an intermediate role between the crowdsourcer and crowd (Hosseini et al., 2014). Thus, the crowdsourcing platform can help the crowdsourcer to explore external knowledge and resource by coordinating the activities of designers and manufacturers to achieve a collaborative product fulfillment network.

A stream of state-of-the-art information and communications technologies (ICT) and industry trends enable the platform to transcend the partners' borders extensively and build up information exchanging network as a driving power for the crowdsourcing product fulfillment process. This platform-driven and collaborative integration of various business and operation processes forge an extended enterprise, in which crowdsourcing and early involvement of partner crowds become new competitive edges for innovative products development (Füller, 2010). The integration of smart sensors and the networked manufacturing systems has established a cyber-physical manufacturing environment, where the synergy of Internet of Things (IoTs), big data analysis, machine intelligence and the conventional manufacturing technologies, like computer integrated manufacturing system (CIMS), supply chain management (SCM), production logistics has stimulated a gigantic manufacturing technology advancement for crowdsourcing product realization, which is collectively envisioned as Industry 4.0 (Schwab, 2017). Owing to the competitiveness in collaboration across multiple entities towards an enterprise with an open yet virtual architecture, decentralization has been recognized as one core characteristic of

Industry 4.0 (Schuh et al., 2014). Many advancing enterprise information technologies have been advocated, and continuously emerging, for the achievement of a digitalized enterprise. Specifically, the new advantages of discrete event simulation (DES), model-based system engineering (MBSE), and CAX software provides the possibility of modeling the manufacturing activities from a distance (Jahangirian et al., 2010). The synergy of cyber-physical systems (CPS) and manufacturing execution system (MES), enterprise resource planning (ERP) in a manufacturing environment vertically integrates the real-time data from the equipment fleet and the cyber architecture based on a digitalized platform (Weyer et al., 2015).

A successful application of the open business model requires collaboration among external partner crowds sharing resources and capabilities along with a coherent product fulfillment flow (Simard and West, 2006). The platform-driven method can be installed to penetrate the partners' boundaries and utilize the data stream to enhance the product fulfillment activities to a collaborative-crowdsourcing one. Thanks to the platform peel the peripheral activities of the manufacturers and links to a large population of the partners, it drives the collaborative-crowdsourcing product fulfillment workflow by formulating the functions, interactions, and processes (Gong, 2018).

1.3 Smart Manufacturing

The recent technological trends reshape the product realization facility to a smarter and more autonomous system, which enables the manufacturing companies to optimize material flow for large manufacturing network accommodation, implement predictive decision-making scheme for dynamic reaction, agile reconfiguration for end-to-end throughput improvement, as well as the optimal partner allocation for cost minimization (Crawford, 2018). These revolutions bring ubiquitous connectivity to the manufacturing environment, which allows the collection of significant volumes of dispersed information and leads to the support of distributed decision-making in the context of manufacturing (Monostori et al., 2016). The blockchain technology enables a distributed and decentralized knowledge management system to support the connecting among the manufacturer crowds in an open environment (Li et al., 2018a). Besides, to connect to the decentralized computing power, the synergy of cloud computing and edge computing explores the connections through a virtual platform (Li et al., 2018b). This synergy has transformed manufacturing into an agile and intelligent process, which paves the way for adopting open business models via autonomous reconfiguration of the distributed manufacturing resources (Rosen et al., 2015).

Such technological progress consolidates the foundation of the future manufacturing paradigm shifts. Open manufacturing explores the technical prerequisite and information architecture for a manufacturing company to install open business models, which aims to achieve an open yet agile enterprise architecture and integrates external resources to its own fulfillment workflow based on a crowdsourcing information platform (Li et al., 2018b). The achievement of an open architecture envisions a transformation towards a large-scale cooperative product fulfillment model, which connects a large manufacturer peer crowds and reconfigure a collaborative network to satisfy volatile customer needs. The requirement of accommodating a dynamic and collaborative network implies the adoption of a service-oriented paradigm which installs X-as-a-service to the manufacturing regime as service manufacturing (Kusiak, 2019). Social manufacturing studies the interactive relationship among the manufacturer crowds, which formulates the construction of a manufacturing network as an autonomous organizing process (Jiang et al., 2016a). The paradigm of cloud-based design and manufacturing offers the framework of connecting smart entities across a population of companies, thus, enable a demand-capacity matching mechanism to serve the collaboration for product realization (Wu et al., 2015).

Specifically, the crowdsourced manufacturing has been proposed based on the application of cloud-based design and manufacturing. It has been further developed to organize a dynamic resource sharing mechanism across manufacturers in a crowd to achieve the production network construction from a large manufacturer population (Kaihara, 2001). The coordination mechanism, which is offered by the crowdsourcing platform, synchronizes manufacturing activities across the companies and lets the manufacturer own an excessive capacity from the cooperation of the partners in the production network (Freitag et al., 2015). The resource matching and pricing mechanisms enable a cloud-based capability and knowledge exchanging marketplace to accelerate the production network reconfiguration process (Kang et al., 2016). From the variety management perspective, crowdsourced manufacturing provides a rapidly responsive reconfiguration of the existed resources and knowledge to serve volatile customer needs.

1.4 Research Objectives

The primary objective is to investigate the platform-driven crowdsourced manufacturing to achieve Manufacturing as a Service (MaaS). The specific objectives and motivations are organized in Figure 1-1. Accordingly, the primary objective can be decomposed into the sub-objectives that are to answer the following research issues:

1) How to model and analyze crowdsourced manufacturing workflow across all decision agents systematically;

2) What are the models and mechanisms for crowdsourcing tasks execution;

3) How to serve the information flow across the decision making in crowdsourced manufacturing;

4) What operational protocols can sustain a long-time prosperity of the manufacturer crowds.

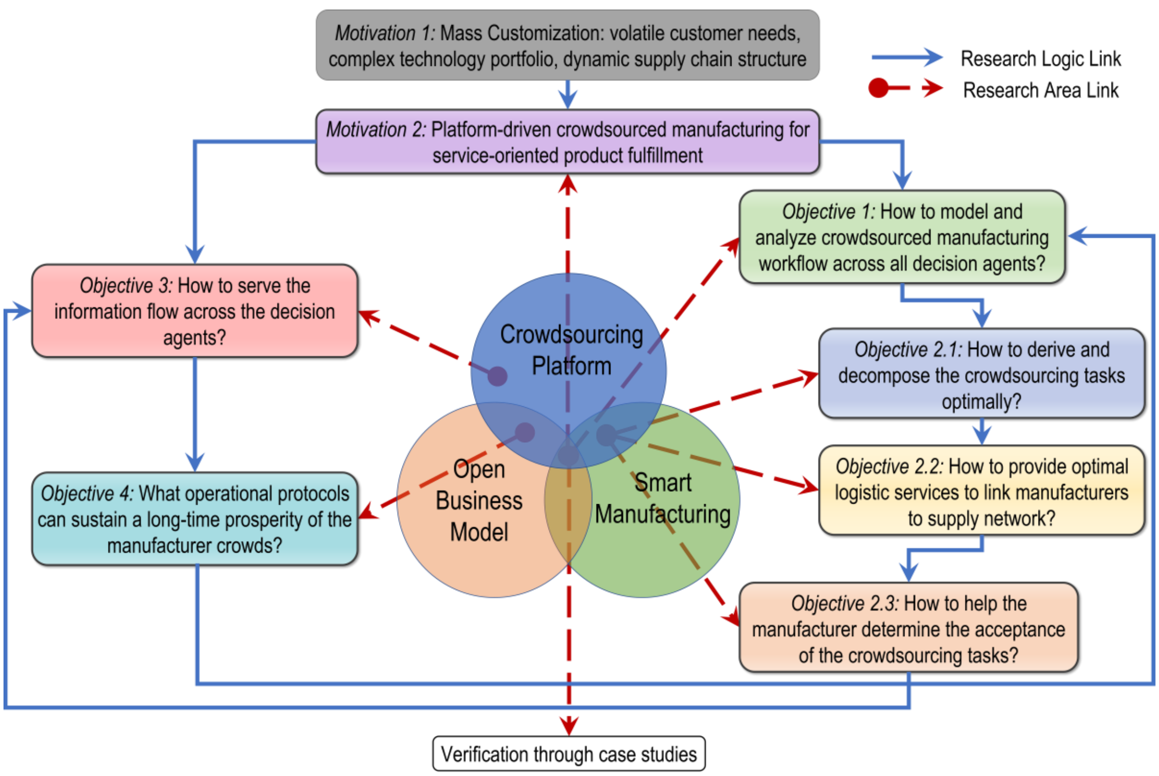


Figure 1-1 Research motivations and objectives

Towards this end, the corresponding objectives are proposed as follows.

1.4.1 *Model and analyse platform-driven crowdsourced manufacturing workflow for crowdsourcing supply network reconfiguration*

Objective 1 proposes a workflow to organize all decision agents in a crowdsourced manufacturing workflow. In line with the principles of axiomatic design (Suh, 1998), the product fulfillment process comprises a set of cascading mapping of “what-how” relationships across four consecutive domains, including the customer, functional, physical, and process domains. Systematic design decisions in each of the domains are characterized by the Customer Needs (CNs), Functional Requirements (FRs), Design Parameters (DPs), and Process Variables (PVs), respectively. Traditionally, the mapping decisions from one domain to another are processed centrally within one enterprise (Jiao et al., 2007a). With the adoption of the open business model, such decisions are becoming decentralized across many decision agents while transcending organizational boundaries in order to leverage resources and capabilities through crowdsourcing (Montes and Goertzel, 2019). The platform-driven crowdsourced manufacturing should ensure the responses from crowds can satisfaction of customers, which implies the requirement of robust and generic contracting evaluation mechanisms.

It is hypothesized that a proper crowdsourced manufacturing workflow can present the full function of crowdsourcing in the context of manufacturing, which specifies the interaction sequence, functional domains, information flow, material flow, and contracting evaluation mechanism among the decision agents.

1.4.2 *Investigate the methods and mechanism to support crowdsourcing task execution*

Objective 2 propose service-oriented systematic methods and mechanisms to ease the accommodation of crowdsourced manufacturing. The motivation of the manufacturer to participate in crowdsourced manufacturing is capturing the value of its abilities to manage manufacturing resources, planning the process, and executing the crowdsourcing tasks. It relies on the cooperation of the open innovators and platform to serve the manufacturer with a system of approaches to link the upstream and downstream partners, as well as to touch and connect the customers. This service system offers functions as planning an arsenal of manufacturing capabilities to fulfil a broader market, rapid configuration of the manufacturing network, and material flow management tools to accelerate the manufacturer's accommodation of crowdsourced manufacturing.

It is hypothesized that a service system can restructure the design of the innovative products to the executable crowdsourcing tasks, serve these tasks with logistic services, as well as the decision-support for accommodating these tasks in the model of crowdsourcing.

The objective can be further decomposed into three sub-objectives, which aim to solve problems in the product, platform, and manufacturing domain, respectively. The first sub-objective is crowdsourcing task derivation and decomposition to serve the transformation of the product design to executable crowdsourcing tasks. The subsequent sub-objective is planning the material for the task flow to transport the material and work-in-progress (WIP) in time based on the partner selection. The last sub-objective is developing a systematic method to help the manufacturers determine the acceptance of the tasks and mix the tasks to their shop floor, according to the real-time situation.

1.4.3 Crowdsourced manufacturing information service system analysis and architecture design

The third objective is the development of a decentralized information management system to serve the decision agents. Because the manufacturers are searched by crowdsourcing to serve the value chain fulfillment, these agents can be widely dispersed. Data management in crowdsourced manufacturing is challenged to handle the process data, which is generated by the dispersed partners. Moreover, these process data stream should be visited by both open innovators and platforms for execution quality monitoring and large-scale cooperation of the manufacturers, respectively. The synergy of the cloud database and blockchain technology offers the distributed query and retrieve services for the manufacturing data source from the machine level data, manufacturing resources data, production data, as well as logistic data (Li et al., 2018a). The platform-based blockchain structure for IoT can also ease the adoption of crowdsourced manufacturing since it enables interactions among manufacturers via smart contracts in a dispersed and peer-to-peer network without intermediary trust (Bahga and Madisetti, 2016).

It is hypothesized that an information service system can serve the decision agents in crowdsourced manufacturing with the required information and decision support, which can manage the distributed product fulfillment data, backtrack the fulfillment process, as well as provide optimal decision-support on the resource planning.

1.4.4 Crowdsourced manufacturing operational protocols optimization and simulation

The fourth objective is investigating managerial protocols to make optimal decisions on the operation of the two-folded demand-capacity marketplace to achieve long-time prosperity. Different from the traditional product manufacturing process, which plans the manufacturing processes inner an enterprise or outsources several peripheral activities to designated partners, the decision making in the crowdsourced manufacturing shows a collective and distributive characteristic. Thus, MaaS requires moving beyond exclusive use of hierarchical decision making, drawing on the power of crowdsourcing and markets wherever possible. Because the crowdsourced manufacturing entails competitive and collaborative workflows that relies on a group decision support system to facilitate the problem-solving process (Thuan et al., 2013a), a successful operational protocol indicates an understanding of the behaviour and evolution of not only manufacturer individual but also the crowd population. The existence of evaluation and awarding processes imply a natural competition inner a manufacturer cluster. Due to the product realization relies on the mass-collaboration across the manufacturer clusters, the inter-cluster cooperation is observed, which can enhance the capability of the manufacturer population to attract more open innovators.

It is hypothesized that it can consider the evolutionary competition-cooperation relationship in the manufacturer population and provide a robust “if-then” scenario to predict the evolution of the population.

1.5 Organization of This Dissertation

In this regard, this dissertation proposes platform-driven crowdsourced manufacturing as a systematic solution towards the installation of MaaS. Figure 1-2 presents the technical roadmap of this dissertation, including motivation & significance, problem formulation, technical approach, methodology & solution, and validation & application.

Chapter 1 discusses the motivation and significance of this research topic, along with a holistic view of research goals and scope. Chapter 2 provides a comprehensive review of various topics related to this research.

Chapters 3, 4, and 5 are responses to research objective 1. Chapter 3 proposes the conceptual framework through stakeholder analysis. It explores applicable industries, essential stakeholders, and their driven forces towards platform-driven crowdsourced manufacturing. A running case study of tank trailer crowdsourcing is presented to examine the potential of platform-driven crowdsourced manufacturing. This chapter sketches essential concepts to formulate crowdsourced manufacturing in terms of workflow on the project level, holistic framework, as well as networked information and material flow across decision agents.

Chapter 4 formulates the key research problems of this dissertation. It presents the fundamental issues underlying platform-driven crowdsourced manufacturing through a structural implication approach. These fundamental issues help provoke insights into how to solve them systematically.

Chapter 5 proposes a crowdsourcing contracting evaluation mechanism to select manufacturers considering the satisfaction of the customer. The evaluation process can be decomposed into engineering functional performance and business operational ranking to reflect the efficiency of performance delivery to customer expectation range and business reputation through historical review, respectively.

Chapters 6, 7, and 8 are devoted to gear forward research towards research objective 2. Chapter 6 reports the development of task derivation and decomposition of the executable crowdsourcing tasks based on the Stackelberg game-theoretical decision-making scheme. This chapter formulates the profit maximization problem in the front-end customer domain as a product family planning problem, which can be solved as a combinatorial problem to seek an optimal solution of a combination of function modules. Since the product functional modules have intrinsic connections to manufacturing processes, the decomposition of innovative products into crowdsourcing tasks entails a manufacturer load planning problem, which can be formulated as a separation optimization problem. A bilevel joint optimization algorithm is proposed and validated in Chapter 6.

Chapter 7 is devoted to logistic service modeling for manufacturers to handle a networked material flow across the agent crowd. A crowdsourcing environment introduces dynamic and networked material flow to logistic service in platform-driven crowdsourced manufacturing. Adopting an X-as-a-Service, MaaS requires the platform to provide logistic decision support services through vehicle route planning. A pick-up and delivery problem with crossdocking is formulated and solved through an effective branch-and-price algorithm.

Chapter 8 focuses on the interactive bilevel optimization formulation for crowdsourcing task acceptance and accommodation by investigating the interplay between task allocation on a supply chain perspective and order rescheduling on a factory perspective. It builds a digital twin of the manufacturer's shop floor based on a max-plus algebra model, which enables reflecting real-time data on a decision-making service. It formulates a real-time order acceptance and scheduling for data-enabled permutation flow shops, which is solved by a bilevel interactive programming algorithm. Finally, a case study and comparisons with prevailing approaches are reported.

Chapter 9 conducts an architecture design of a blockchain-based smart contracting and distributed data management system for information services to all stakeholders in platform-driven crowdsourced manufacturing. This chapter reflects research objective 3. It uses smart contracting technology to solidify a crowdsourcing supply cooperation network and use it as an encryption key to manage product fulfillment data. A blockchain-based database is established to manage wide dispersed product fulfillment data. The stakeholders in platform-driven crowdsourced manufacturing can stream product fulfillment data from Industrial IoTs and enterprise software. Finally, a case study is reported to reflect the potential of the proposed architecture.

Chapter 10 introduces optimal operational protocol derivations and adjustment mechanisms for platform-driven crowdsourced manufacturing. This chapter adopts a population dynamics perspective to model the behavioral interactions among manufacturer clusters and formulates an optimal operational protocol derivation problem as an evolutionary competition-cooperation game. An optimal operational protocol can be explored by finding an equilibrium point and the corresponding stability analysis.

The last chapter, Chapter 11 summarizes the achievements in addressing the research objectives and issues. A critical assessment is given to highlight the limitations and possible improvements of this research, along with recommendations for future work.

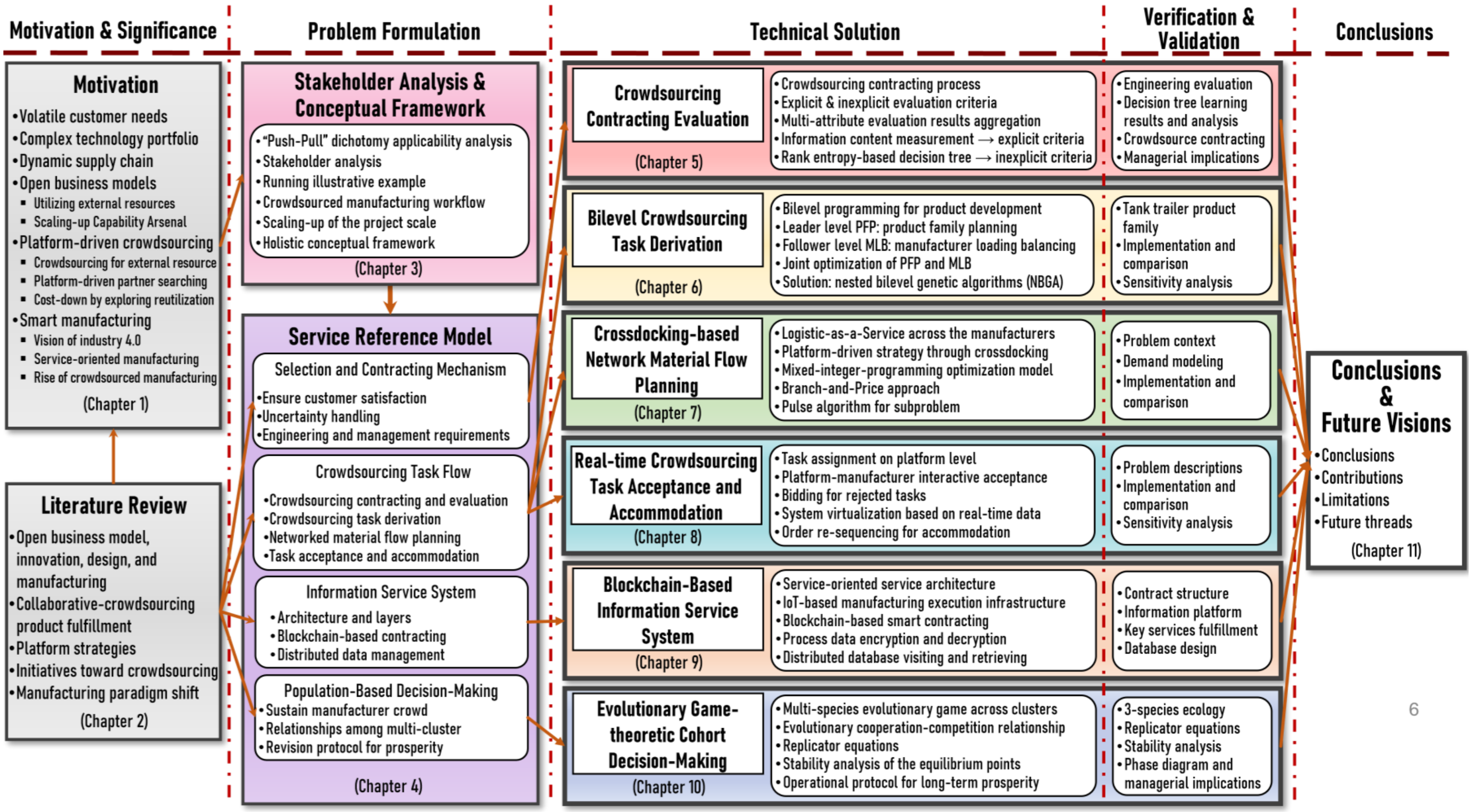


Figure 1-2 Organization of this dissertation

CHAPTER 2. LITERATURE REVIEW

2.1 From Open Business Model to Open Manufacturing

The business model is defined as a framework that consists of stakeholders gathering structures, as well as the methods of creating, delivering, and capturing the value (Zott et al., 2011). The purpose of a business model is to provide a set of heuristic logic to connect the technical ideas and realize the economic value (Chesbrough and Rosenbloom, 2002). With the opening of the conventional enterprise borders, the company can explore a larger volume of ideas and knowledge, as well as utilize a broad spectrum of external assets, resources, and positions for a more efficient value capturing (Chesbrough, 2007). The open business model is coined by Chesbrough to describe the linking organizations outside the company border to yield new products or services by using the power of division of labor (Chesbrough, 2006). Following the generic product development process in engineering design (Eppinger and Ulrich, 2015), the open business model provides the transition of conventional product fulfillment to a series of open activities. The mapping relationship of the open innovation, open design, and open manufacturing with generic product development process is shown in Figure 2-1.

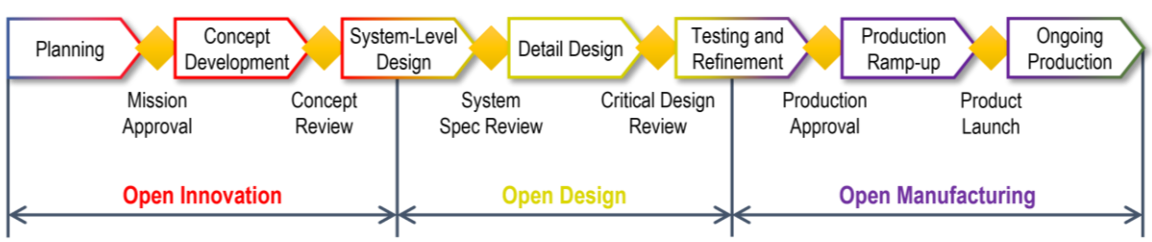


Figure 2-1 Generic product development process

As an earlier base of the open business model, open innovation is applied to depict the distributive innovation process based on purposively managed knowledge flows across the boundaries of organizations (Bogers et al., 2017). Companies have widely recognized the open innovation as an accelerator for the internal leap from research to development and market expansion of external acquisition of their knowledge (Vanhaverbeke and Chesbrough, 2014). In contrast to the traditional vertical integration model, the open innovation horizontally structures a dynamically interactive network for various clusters of autonomous firms throughout the innovation process (Dhanaraj and Parkhe, 2006). Moreover, from a value chain perspective, an increasing amount of the industries organize the firms as a central platform structure, where the core firm seeks the inflow of the external knowledge for their targeted markets, while the surrounding firms outflow their knowledge to help the core company to save the cost from do-it-all-yourself (Gawer and Cusumano, 2014). Through this innovation network, the participating firms of the open innovation can identify their market opportunities, link to the advanced research and technology, collect a variety of product concepts, as well as initiate the configuration of product family architecture (Grönlund et al., 2010). From a product development perspective, open innovation provides a systematical method to install the open business model to cover product planning, along with the concept development, and end at the transition stage of system-level design.

Targeted at the later product innovation and development process, the open design depicts the installation of the open business model by design communities to open the border of the company, collaborate with the external designer crowds, and achieve a flexible design capability (Boisseau et al., 2018). The concept of the open design originates from the free/libre open-source software (F/LOSS), which has created legends like Linux and Wikipedia (Weber, 2004). The open design enables a decentralized development process, a dynamic development structure, as well as the involvement of a crowd of developers (Wheeler, 2011). Since the physical products are increasingly data-driven and digitalized, the open business model propagates from digital information processing to the design of tangible products (Raasch et al., 2009). The concept of the open design has been defined as the designers allowing external partners to access, modify, and derive from the product design (Micklethwaite, 2012). Based on the designer crowds and advanced Internet access, open design structures a collaborative design team from external designer crowds and harmoniously integrates the design outcomes (Koch and Tumer, 2009). Open design is capable of parametric design of modularized design tasks based on a given product architecture (Vallance et al., 2001), as well as aggregate design results with a systematical computation mechanism to coordinate the numerical conflicts (Binnekamp et al., 2006). These developments of open design imply that it can be served as a transforming approach to gear the transition stage of system-level design to detail design and the start of product testing and refinement towards more open activities.

The rise of smart manufacturing enables a highly democratized manufacturing network, which is characterized as decentralized, service-oriented, and easy to access (Bull and Groves, 2009). The democratization of the manufacturing will lead to the installation of the open business model, which is empowered by a dynamic network of agents who are acquiring technologies and resources in a self-directed and ad hoc way (Richardson, 2016) This post-Fordism sociotechnical trends can be summarized as open manufacturing to depict this manufacturing ecosystem. The open manufacturing integrates the manufacturing resources and knowledge from the distributed manufacturer community by a decentralized network to support the manufacturing operation (Li et al., 2018b). The collaboration of the manufacturer crowds utilizes cloud computing technologies to access the manufacturing agents, as well as blockchain for production knowledge and information exchanging (Li et al., 2018a). An open-source exchanging marketplace will provide a variety of external manufacturing technologies and sharing excessive manufacturing resources and capabilities, which will ease the configuration of manufacturer crowds to a collaborative team for fulfilling various product orders (Banerjee et al., 2015).

2.2 Collaborative-Crowdsourcing Product Fulfillment

Among the approach of accessing external knowledge and resource to implement an open business model, crowdsourcing has been highlighted as an ICT-enabled and social media-based innovation tool (Kittur et al., 2008, Martini et al., 2014). This concept has been introduced to describe the utilization of open calls to form a peer-production for a task from a crowd of undefined people (Howe, 2006). Thanks to the wisdom of crowds, the collection of intelligence from a large group of heterogeneous participants are believed to show the superiority of a limited group of elites (Leimeister et al., 2009). Several factors have been highlighted as the impulses of the participants, includes self-market or promotion, tangible or intangible compensation, social fames, and reputation, to name but a few (Bayus, 2010).

Since the crowdsourcing mechanism has been recognized as beneficial to problem-solving for the technical tasks, a stream of research has geared forward the formulation of crowdsourcing. Surowiecki (2005) identifies four essential prerequisites to ensure the successful crowdsourcing decision-making: 1) diversity, each participant can offer unique knowledge or capabilities; 2) independence, to avoid the influence from peers; 3) decentralization, the information is sharable for locally process by participants; 4) aggregation, the fulfilled crowdsourcing tasks can be collectively aggregated. Bonabeau (2009) divides the crowdsourcing decision-making processes into two stages, the generation of possible solutions and the evaluation of these solutions. The essential cornerstones of crowdsourcing have been summarized as: 1) the application of open calls to explore the crowd; 2) a task set that needs to be fulfilled; 3) compensation of the contribution (Allon and Babich, 2020). Considering the complexity of the crowdsourcing task set, the success of crowdsourcing emphasizes independence and decentralization in the solution generation process to ensure the cognitive diversity, as well as the semantic coherence of the most successful solution sets to ensure the aggregation (Rosen, 2011).

The advancement of ICT brings ubiquitous connectivity to the decision-making entities worldwide through the mobile network and social media. Meanwhile, the synergy of industrial IoTs and CPS paves the way for an extensive collaboration among the practitioners from industry. Several industry pioneers have started the installation of crowdsourcing. These practices can be generally divided into two categories, existed giants operate a designated platform to reach the external resources via crowdsourcing to majorly serve their market, like MyStarbucks operated by Starbucks and Haier Open Partnership Ecosystem (HOPE) by Haier, as well as the third-party company operate a crowdsourcing platform to serve their customer’s market, like Amazon Mechanical Turk and ZBJ.com in China.

A stream of researchers summarizes the classification methodology to analyze the type of these crowdsourcing practices. Based on the types of requirements and collecting methods of contributions, two dimensions has been summarized as subjective or objective contents, as well as aggregated or filtered contributions (Prpić et al., 2015). The research of crowdsourcing has revealed a series of crowdsourcing model from the industry application based on the characteristics of the demands, which includes "winner-takes-all" or multiple responses, defined task or unresolved problem, individual efforts or collaborative manners, crowds qualification, activity-targeted or fundraising, requirements on response qualities, activeness of participation, to name but a few (Grewal-Carr and Bates, 2016). Since collaborative-crowdsourcing product fulfillment is a process to realize the innovative product planning, the crowdsourcing tasks can be identified as an explicitly defined design or manufacturing requirement. Thus, it requires qualified designers and manufacturers to fulfill the tasks through a series of design solution derivations and fabrications, respectively. The crowdsourcing models for the collaborative-crowdsourcing product fulfillment task are sketched in Figure 2-2.

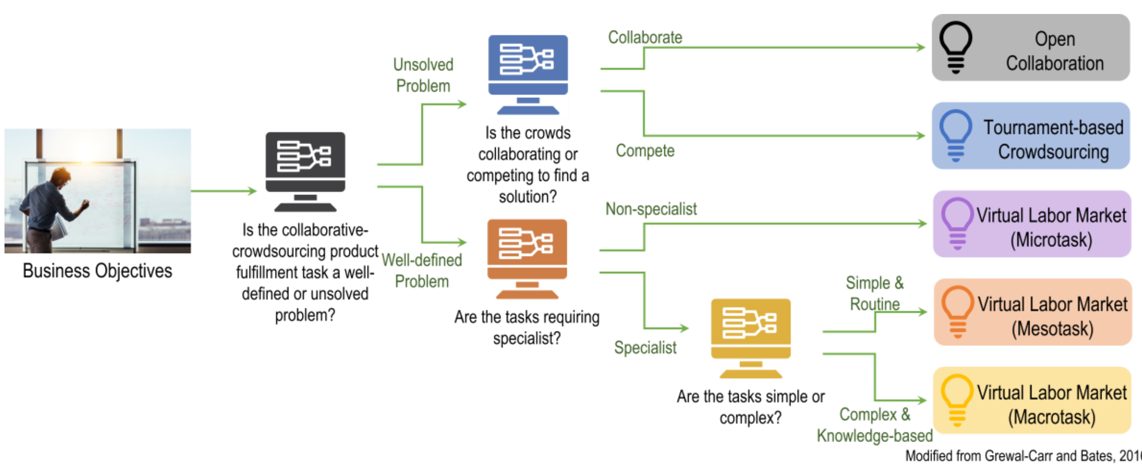


Figure 2-2 Crowdsourcing models for product fulfillment tasks

The crowdsourcing models for defined crowdsourcing tasks can be classified into three categories based on their openness, business motivations, and operation management issues: 1) Open collaboration, 2) Tournament based crowdsourcing, and 3) Virtual labor market (Allon and Babich, 2020). Open collaboration is a social media-based idea searching approach, where the organizations publish crowdsourcing tasks to a community of agents and aggregate the responses in the community to serve the decision making. In this approach, the crowdsourcing task can be identified as an unsolved problem, which has no sophisticated problem definition but expects innovative contributions after a mass collaboration of the community. This crowdsourcing type usually has no monetary incentives for the agent crowds, and the agents are not expecting to be qualified as an entering barrier in most cases. Tournament based crowdsourcing formulate a series of activities to let the seeker connect to the solvers and select the winner after idea competition. The existence of the tournament implies more objective incentives and a relatively open environment, as well as the stimulation of competition among solver crowds. In contrast, the virtual labor market entails a web-based platform that plays an intermediate role between the crowdsourcer and a crowd with required qualifications. The platform can match the capabilities in the crowd and the requirement of a crowdsourcing task. It has also been observed that the platform can play an evaluation role to test the program (i.e., Upwork) or service monitoring (i.e., Uber and DiDi).

The crowdsourcing tasks in the virtual labor market can be further classified by the problem scale, the wellness of problem definition, and the specialty of required skills. The microtask model serves a well-defined problem structure, which is easy enough to be classified as an everyday task. This model can be identified as an extension of the traditional subcontracting, which explores the natural resources external from the company. The emerging Instacart and Uber is a microtask crowdsourcing application in the service industry sector. The Mechanical Turk by Amazon can serve the user-involved product design by hiring a crowd of potential customers for the prototype survey. The mesotask is a well-defined task that expects the specialist to provide routine and straightforward solutions. It has an explicit expectation on solution quality, delivery quantity, and methods, as well as the violation terms. This model can be used for software development as well as mechanical fabrication, which has a clear process routine and a set of specified qualifications. The efforts for the mesotask are less complicated and innovative comparing with the macrotask model. The macrotask serves the complex crowdsourcing objective, which is often installed on the research and development of the product or strategic consultant. The efforts for macrotask are more knowledge-based and subjective, which can help the crowdsourcer to expand their knowledge and resource arsenal without owning a designated department or sign a long-term subcontract. For example, the HOPE platform enables Haier to develop broad connections to an ocean of research groups to develop new products.

Although recognized as an emerging paradigm of product development by both industries and academia, there is a lack of systematical installation roadmap of crowdsourced product development (Shergadwala et al., 2020). Crowdsourcing can be utilized for product idea generation, concept design, detailed design, physical prototyping, and design evaluation, which are the essential product innovation and design stages in the new product development (Tran et al., 2012). The generic crowdsourcing process for new product development can be summarized as five consecutive stages, namely task definition, task broadcasting, response collection, response evaluation, and winner awards (Qin et al., 2016). After the product planning and design stage, crowdsourcing has been explored to install on the production stage. The concept of crowdsourced manufacturing originates from the cloud-based manufacturing system, which reflects a manufacturing capability sharing and production organizing mechanism among the cloud-based and widely connected manufacturing network (Wu et al., 2015).

2.3 Industry Initiatives Toward Crowdsourcing

Crowdsourcing has been widely applied to a spectrum of industries worldwide. Recognizing the power of "wisdom of crowds," rapid growth in the crowdsourcing sector is observed to enable a broad application to a spectrum of industries. Figure 2-3 shows the annual number of investments and the corresponding amount on crowdsourcing companies in China from 2006 to 2018. It shows that a large amount of investment has been devoted to incubating the funding of crowdsourcing companies, which implies a rapid growth in the past decades.

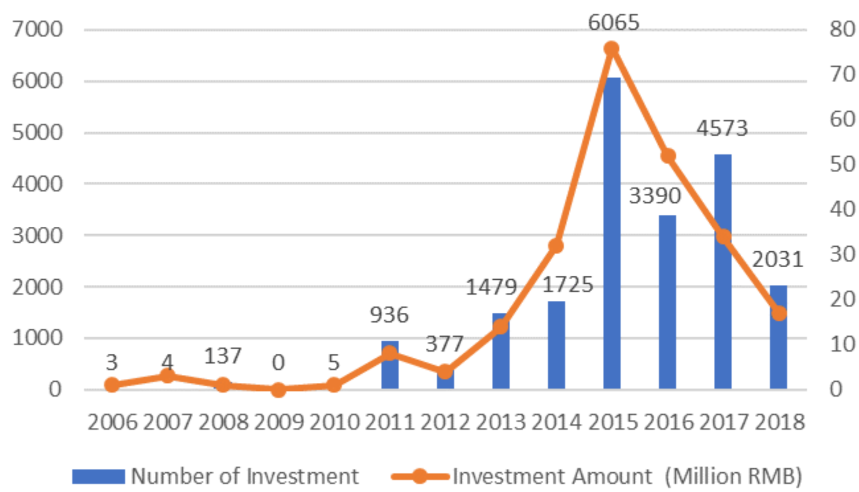


Figure 2-3 Investment on crowdsourcing in China

Figure 2-4 summarizes the representative crowdsourcing platform companies in terms of their targeted markets, founded time, and the latest monthly page views of their domain page. A sizeable monthly page view number implies a prosperous crowd ecosystem and substantial fulfillment capability. The average monthly page view number of the selected representative companies is around 300 thousand.

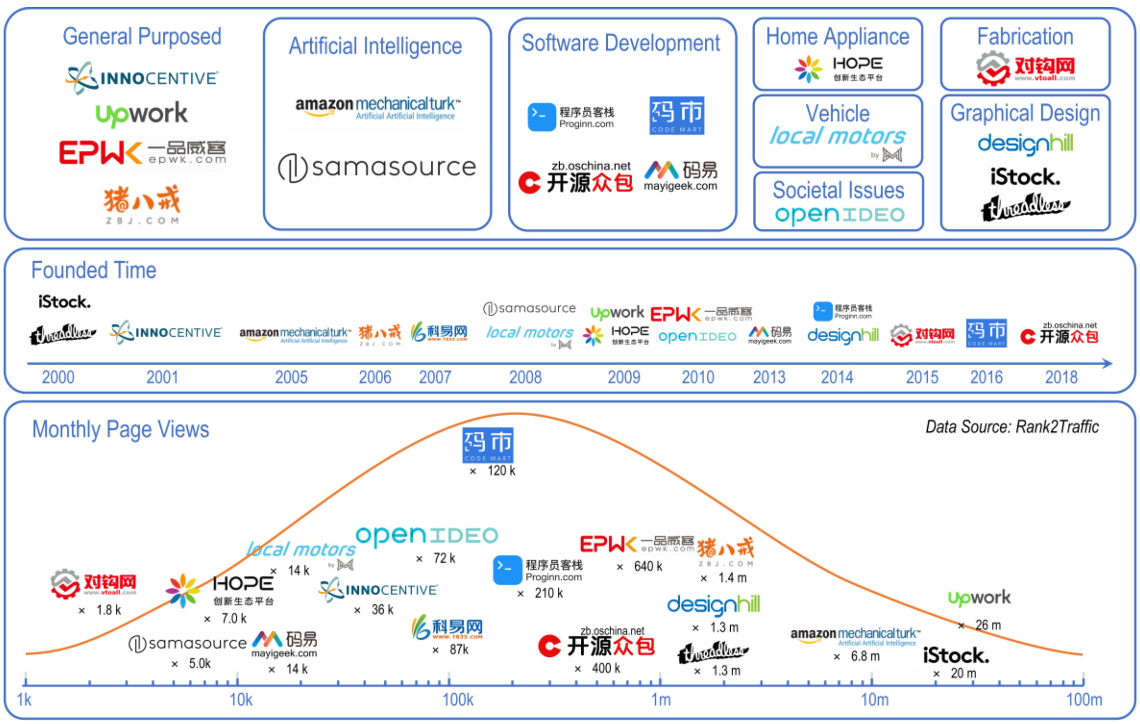


Figure 2-4 Representative Crowdsourcing Platform Companies

The crowdsourcing platform companies like Amazon Mechanical Turk and Samasource paves the way of connecting a large amount of labor to the companies to fulfill micro-tasks at a relatively low cost. Thanks to the massive amount of the Internet users, Amazon Mechanical Turk can sustain a large yet diversified crowd to serve the human-subject survey, data annotation, as well as the data cleaning and verification, to accelerate the development of artificial intelligence-related projects (Buhrmester et al., 2016). Samasource, a company established in 2008, utilizes the exponentially growing Internet users globally by employing low-income workers in the developing countries to providing high-quality, large-scale training data for profit (Ojanperä et al., 2018).

Compared to the micro-task, which only requires the crowds to have general capabilities, crowdsourcing can also link the expert crowds. The crowdsourcing platform companies in China have linked a large population of developers and serves the ICT giants like Alibaba and Huawei with heterogeneous capabilities and rapid response, which includes proginn.com, Code Mart, zb.oschina.net, mayigeek.com, to name but a few. Out of the IT-related industries, the leading home appliances and consumer electronics manufacturer Haier has installed the open business model and established HOPE to restructure the centralized enterprise to “a sea of entrepreneurs” and sustain a vast innovation ecosystem to serve the product research and development for their appliance sector (Chen, 2016). The HOPE platform provides crowdsourcing services for both internal development teams and external partners, which achieve a series of successful products by gathering the “wisdom of crowds” (Lewin et al., 2017). The local motors, which is founded in 2009 as a US vehicle manufacturing company, explores the design ideas via crowdsourcing and manufacturing the products by the cooperation of the manufacturer network (Norton and Dann, 2011). Crowdsourcing can also serve the mechanical fabrication, by linking the demands and fabrication capabilities, like machining, metal sheet forming, heat treating, along with others. Vtoall.com is a Chinese fabrication crowdsourcing platform that serves a two-sided marketplace for demands-capability matching.

Since the graphical design relies less on the physical assets like manufacturing facilities and equipment than IT and manufacturing industries, crowdsourcing can establish the approaches to the designer crowds. Designhill serves the design service with a broad spectrum of logos, websites, apparel, and other projects by linking a large crowd of designers. Threadless is a more expertized company that operates an online artist community and e-commerce to fulfill volatile t-shirt and other apparel customer needs. The stock photography company iStock provides photos and video transaction services in a crowdsourcing way, which serves a large crowd of creators. The societal issue is a less objective and complex problem that influences a large number of individuals in a society. Crowdsourcing company OpenIDEO is a social impact platform that builds connections to a crowd of companies to solve the tough social problem (Fuge et al., 2014).

Thanks to the development of crowdsourcing and related Internet technologies, some crowdsourcing companies broaden their targeted market sectors and restructure their platform to a general purposed platform. InnoCentive is funded by Eli Lilly and Company to accelerate the research internal of the companies. However, after a series of partnerships and acquiring, the InnoCentive is a general purposed open innovation and crowdsourcing company to allow the organization to publish the problem as well as the problem solvers to earn monetary rewards and reputations. ZBJ.com is a Chinese crowdsourcing company that started from targeting industrial design to coverage of legal services, marketing services, ICT development, software engineering, engineering design, and graphical design by utilizing problem solver crowds. Upwork is a virtual labor marketplace that enables demand-expertise matching and remote collaboration to serve a broad spectrum of industries. EPWK.com is a creative crowdsourcing service company in China, which serves small and medium companies with project planning and marketing services, knowledge, and software development transactions and services.

These successive industry initiatives verify the feasibility of crowdsourcing and value of "wisdom-of-crowds," thus, pave the way of installing crowdsourcing to the manufacturing environment by forging a collaborative and coordinated manufacturing network.

2.4 Emerging Consensus on MaaS

As an emerging technology vision, smart manufacturing reshapes the landscape of manufacturing industries with sensors, computing platforms, communication technologies, as well as data-intensive modeling and predictive engineering (Kusiak, 2018). Driven by the advancement of the knowledge exchange marketplace, sharing economies on the manufacturing shop floor, as well as increasingly democratizing and opening trends, smart manufacturing is characterized as decentralized, service-oriented, and platform-based (Kusiak, 2019). Originates from network manufacturing that uses centralized crowdsourcing, the open manufacturing adopts decentralization ideology. It utilizes blockchain and edge computing to construct a cross-enterprise knowledge and service sharing framework (Agostinho et al., 2016, Li et al., 2018b). The open manufacturing can be viewed as an incubator for small and medium manufacturers since it regulates the knowledge and service sharing standards and protocols. It can support manufacturers to develop scalable and flexible business scale at a lower cost and eventually improves the overall quality, efficiency, and effectiveness of manufacturing services. From a supply chain aspect, the open manufacturing decouples design, logistics, and service layers from physical assets (Kusiak, 2020). Targeted at achieving this capability, the open manufacturing enterprises will be amenable to the X-as-a-service mode, where X represents, e.g., manufacturing, supply chain, and logistics. This manufacturing paradigm is also called service manufacturing. Recognizing the great benefit of resource and service sharing, several technical challenges have been highlighted as the bottlenecks towards an extensively opened manufacturing environment, includes crowdsourcing contract design, diversified supply chain reconfiguration, distribution coordination mechanisms, to name but a few.

The implications of social media and the Internet change the consumption-manufacturing relationship in industries from four aspects: production socialization, role shift of consumers, the driving force of production innovation, and virus-like information propagation in social media (Hamalainen and Karjalainen, 2017, Jiang et al., 2016b). These aspects drive the current manufacturing paradigm to a more decentralized, open, adaptive, and socialized one, which is coined as social manufacturing. Focus on this new manufacturing paradigm, a large volume of research has geared forward the content of social manufacturing, which includes, the blockchain-based tracking system for the self-organizing process (Leng et al., 2019), RFID-based execution systems for inter-enterprise monitoring and dispatching (Ding et al., 2016), the socialized production network generation framework (Jiang and Ding, 2018), and outsourcer-supplier coordination mechanisms (Guo and Jiang, 2019, Leng et al., 2017), to name but a few. These research streams hold the opinion that small and medium service-oriented enterprises can be aggregated into different kinds of horizontal manufacturing communities to enlarge their bargain power and common profits through initial clustering and self-organization. As a systematical software solution, social manufacturing provides a series of demands-capability matching functions, includes requirement and capacities releasing, intelligent matchmaking, production monitoring, and participator collaboration management (Ding et al., 2016). However, current related research has a limited exploration of the coordination for manufacturing networks, collaborative-negotiation contracting among stakeholders, and an evaluation for solution validation and quality control.

Cloud manufacturing follows the successful application of cloud computing, in which diversified resources and abilities are intelligently sensed and connected into the broader Internet, and automatically managed and controlled using IoT technologies (Wu et al., 2013, Tao et al., 2014). In this manufacturing environment, users search and invoke the qualified manufacturing cloud services from a related cloud based on their requirements and organize them to be a virtual manufacturing environment or solution to complete their manufacturing tasks (Tao et al., 2011). Based on this concept, Wu et al. (2012) propose a cloud-based design and manufacturing model which identifies cloud consumer, cloud provider, cloud broker, and cloud carriers as its stakeholders and a distributed infrastructure with an interfacing system. Thanks to the dynamic characteristic of cloud manufacturing, the manufacturing equipment across multiple dispersed manufacturing sites can be rapidly reconfigured and repurposed (Schaefer et al., 2012). Thus, the significance of automation and digitization of manufacturing operations is highlighted in cloud manufacturing, which implies a widely connected manufacturer community (Wu et al., 2013). The researchers in cloud manufacturing pay more attention to the cloud-based technologies, includes a consumer-provider interactive framework, cloud-based equipment automation, and web-based service-oriented system for resource monitoring and controlling, but minimal research has focused on the construction of manufacturing network to serve the targeted market and cooperation architecture for different business entities along the product fulfillment workflow.

The cloud-based framework enables the manufacturing companies widely connected while transcending the conventional enterprise borders. This manufacturing network paves the way for installing negotiation-based cooperation between the factories to share their excessive capabilities and outsource their peripheral manufacturing capabilities. The reconfigurable supply chain system can support the quickly responsive construction of a product fulfillment chain for both the production network and the knowledge marketplace (Chida et al., 2019). This dynamic production network enables the realization of a highly personalized product family (Tan et al., 2017), and achieves efficient manufacturing resource sharing among federated production networks (Kadar et al., 2018).

2.5 Chapter Summary

The topics reviewed in this chapter offer guidance to examines stakeholder analysis and conceptual framework of platform-driven crowdsourced manufacturing in the Chapter 3. Considering the complexity of the platform-driven crowdsourced manufacturing, Chapter 4 examines fundamental issues and reference model as a research agenda. Considering the limitations of various topics reviewed here, I propose methodologies that can overcome their respective limitations in Chapters 5, 6, 7, 8, 9, 10 to address a cornerstone of crowdsourced manufacturing.

CHAPTER 3. STAKEHOLDER ANALYSIS AND CONCEPTUAL FRAMEWORK OF PLATFORM-DRIVEN CROWDSOURCED MANUFACTURING

Crowdsourced manufacturing forms a dynamic supply chain with a trichotomy of its stakeholder roles, namely client, requester, and provider (Chida et al., 2019). This concept describes a broadcasting and searching process based on the crowdsourcing model. Considering the adoption of open business model, it entails a “maker-platform operator” business model as an upgrade of the current peer-to-peer manufacturing network based on the maker to platform model (Kortmann and Piller, 2016). The client serves as an open innovator in the open business model as well as a crowdsourcer in the crowdsourcing model, who installs the open business model, designs a new product, and seeks a structured supply chain to fulfill the corresponding product. The requester serves as the operator of the crowdsourcing platform, who broadcasts the crowdsourcing tasks, collects the responses, as well as evaluates the response, and awards the winner with contracts. The provider serves as a participant of the manufacturer crowd who shares their expertized manufacturing capabilities to capture value.

In line with the principles of axiomatic design (Suh, 1998), the product fulfillment process comprises a set of cascading mapping of “what-how” relationships across four consecutive domains, including the customer, functional, physical, and process domains. Systematic design decisions in each of the domains are characterized by the CNs, FRs, DPs, and PVs, respectively. Traditionally, the mapping decisions from one domain to another are processed centrally within one enterprise (Jiao et al., 2007a). With the adoption of the open business model, such decisions are becoming decentralized across many decision agents while transcending organizational boundaries in order to leverage resources and capabilities through crowdsourcing (Montes and Goertzel, 2019).

The differences among driven forces in various industries implies different ways of adapting to open business models, as well as adaptabilities to the crowdsourcing model. Section 3.1 will focus on the applicable analysis of the innovative products and variant product industries. The differences among decision agents imply different expected benefits from the participation of crowdsourced manufacturing, which significantly diversify the decision-making processes among the agents in crowdsourced manufacturing. To explore the driving force of decision agents in crowdsourced manufacturing in the applicable industries, section 3.2 after the next examines their current challenges, potential contributions to the entire community, and specific needs from the crowdsourced manufacturing paradigm. In section 3.3, a running illustrative case of tank trailer is kicked-off to demonstrate the installation map of crowdsourced manufacturing on the existed production network.

Moreover, this paper develops the conceptual framework of crowdsourced manufacturing in detail. Section 3.4 proposes the crowdsourced manufacturing workflow along a value chain. It explains the fundamental mechanism underlying the workflow, which includes the decision agents, domain, processes, as well as crowdsourcing contracting mechanism. Section 3.5 presents a holistic framework of crowdsourced manufacturing workflow in the case that multiple value chains link customer clusters and manufacturing agent crowds through the same platform companies. It also examines the networked flow in crowdsourced manufacturing, which includes information flow and manufacturing flow among the customers, open innovators, platforms, and manufacturing agent crowds.

3.1 Dichotomy of Industrial Applicability

As an emerging manufacturing paradigm, various industries show different accordance with the distinctive context of company and compatibilities to crowdsourced manufacturing. Generally, there is a common dichotomy to analyze the pulse of innovative product development, namely, market-pull versus technology-push (Brem and Voigt, 2009). Market-pull industries innovate the value chain from the inadequate satisfaction from current customer needs, as well as a variant product by rearranging existed value chain. In contrast, technology-push industries to invent new technologies when the target market is ambiguous then finds paths to target markets to commercialize the new technical know-how (Lubik et al., 2013).

Because the new technical invention is reckoned to be the prerequisite and core competitive edges in the companies by technology push industries, the benefits of large manufacturer crowds and a consequent quickly responsive manufacturing network show less attractiveness to these companies. The effect of intellectual property as a barrier to market entry will filter a large proportion of manufacturers to participate in crowdsourced manufacturing. Meanwhile, the risk of mismanagement of intellectual property may harm these companies. For example, Gore-Tex is a famous technology push innovative products, which is an expanded Teflon sheet made by W. L. Gore and Associates. The commercialization of Gore-Tex starts with the successful development of a porous form of polytetrafluoroethylene with a microstructure characterized by nodes interconnected by fibrils (Gore, 1976). Utilizing this innovative material and related process, W.L. Gore developed a series of products, including apparel fabrics, medical devices, insulation layers for advanced electric cables, to name but a few. Several apparel companies use Gore-Tex material and core processes to accelerate the commercialization by collecting customer needs and provide access to the market. In this case, due to W.L. Gore has ownership of the intellectual property of Gore-Tex and lacks access to the market. It acts as a material supplier and process solution provider. Thus, the company entails a manufacturer in crowdsourced manufacturing and serves the other innovators like apparel companies, surgical apparatus start-ups, or electric device manufacturers. These industries have a different innovation impulse and involve Gore-Tex as an incremental improvement.

Different from technology push, market pull industries based on the existed connections to the customers. The companies sense the volatile customer needs from their customer in targeted markets, translate these customer needs to specific requirements for new function realizations, then finding an appropriate technology to achieve maximum satisfaction. Market-pull industries create value by bringing reconfigured available technologies and capabilities to customers to achieve “Make-to-Order” (MTO). In this regard, it provides a reconfiguration roadmap for a crowdsourcing platform to arrange manufacturers into a network. Since the product development decisions in market pull industries are customer-driven, these companies provide the directions and evaluation criteria for the contributions of manufacturer crowds.

3.2 Decision Agents in Crowdsourced Manufacturing

3.2.1 Open Innovators

The open innovators are transformed from the conventional Original Equipment Manufacturer (OEM), who collects customer needs and serve the customer after-sales, develop the product concept, as well as the design of the products. The open innovators have a large peer population with highly specialized targeted markets and corresponding highly customized orders in an industry sector. Thanks to the mass customization paradigm, the combination of platform-based design and modular design can serve a highly responsive product configuration using existed technologies and sub-systems (Simpson et al., 2006).

One of the motivations of participating in the crowdsourcing manufacturing paradigm is to seek a larger population of heterogeneous manufacturers to realize product innovation. Although the existence of a crowdsourcing platform creates the approach to manufacturer crowds, the set-up time and risk of initiating crowdsourced manufacturing with the platform are still barriers to open innovators. That set-up time can be decomposed to the negotiation between open innovators and platform, the crowdsourcing contracting between platform and manufacturers, as well as the lead time to fulfilling the crowdsourcing tasks. The risks for open innovators lie on the leaking of intellectual property of innovative products, the involvement of under-qualified manufacturers, and the failure of crowdsourcing task aggregation. These requirements imply a systematical product variety coordination system to serve the crowdsourcing product information management, and an information system to monitor the manufacturing process.

Moreover, the transformation of product innovation to fulfilled products requires the collaboration of platforms and manufacturers per se. The platform decomposes the product design to subtasks and packages to crowdsourcing tasks for broadcasting. This restructuring process relies on the semantic coherence from product design to crowdsourcing tasks, and a guideline for open innovators can accelerate this transition process. Moreover, since the essentiality of the review and evaluation in the product development process, the task fulfillment evaluation criteria set struggles for the evaluator. The targets are not always explicit, while the criteria are dependent (Jiao and Tseng, 1998). A system of methods to ease the monitor of the manufacturing activities and evaluation of contribution from the manufacturer crowds is essential.

3.2.2 Platforms

A platform can be considered as a multi-sided market, which serves distinct crowds of third-party users and provides each other with network benefits (Eisenmann et al., 2011). It is operated as a marketplace to enable the match of demands and supplies. From a manufacturing network perspective, a crowdsourcing platform is a two-sided market, where the open innovators publish their innovations as a set of manufacturing demands and manufacturers publish their manufacturing capabilities as supplies. Following a platform model, the crowdsourcing platform company integrates external open innovators and manufacturers, thereby create and capture value from that manufacturing network.

This integrating process entails a reconfiguration of the manufacturing capabilities to serve the innovations. The crowdsourcing platform performs as an intermediator among the manufacturing activities. It reallocates the existed skills in the manufacturer crowds to accommodate the emerging innovating value chains based on a system of reconfiguration mechanism. It can help the open innovator peel the process-related and organization-related configuration activities and utilize the service from the platform and manufacturing capabilities from manufacturer crowds instead. On the other side of the platform, manufacturer crowds can get rid of finding target markets as well as coordinating with upstream and downstream manufacturer partners.

In the crowdsourcing model, the expansion of manufacturers and open innovator population is the platform's continual pursuit. A larger open innovator population leads to an increasing number of paths to the market and a wider variety of customer needs. At the same time, more manufacturers linked to the platform implies a bigger arsenal of capability and knowledge to fulfill the customers. Since the expansion of the open innovator scale leads to a rising number of value chains, the platform is challenged by installing a system of configuration strategies to allocate crowdsourcing tasks as a result of the intermediator the platform played. A population of manufacturers and dynamic manufacturing networks significantly raises the complexity of the coordination decision-making process in the platform. The manufacturing activity synchronization and conflict solving policies for partner crowds are required by the platform under the paradigm of crowdsourced manufacturing.

3.2.3 Manufacturers

The manufacturers are advanced from the supplier tiers in the current supply chain configuration, who operate the factories and provide materials, sub-assembly, and products according to the orders from open innovators. The manufacturer crowd can be divided into several sub-clusters according to the position along the value chain, for instance, raw material providers, secondary suppliers, direct suppliers, assemblers, and so on. The primary pursuit of the manufacturers is the approaches to the broader market via the collaboration between open innovator crowds and crowdsourcing platform. With a broader explored market, a larger volume of the value chains can be brought to the crowdsourcing manufacturing network. These value chains constructed a cross-link relationship, while a manufacturer can achieve maximum reusability based on the commonality of the process. Thus, with the expansion of the customer clusters, manufacturers can focus on their core competitive edges and achieve economies of scale.

Under a paradigm shift to the open business model, these manufacturing enterprises participate in the crowdsourced manufacturing model in two different scenarios. The first scenario entails a group of manufacturers take crowdsourced manufacturing as their primary economic activities and plan their schedule in the center of crowdsourced tasks. They rely on the assignment of crowdsourcing tasks from the platform and the access to target markets from the open innovators. An inferior task allocation solution will lead to inefficiency supply chain configuration in platform level and order congestion or avoidable production line idle at the manufacturer level. A successful task allocation system should base on the modeling of the manufacturer's plant and the logistic system, as well as a global production planning for platform and manufacturers. Because of the participation to crowdsourced manufacturing implies an increasing number of value chains going through the manufacturer, the material flow coordination among the upstream and downstream partners is increasingly complex. As the approach of realizing material flow, the logistics issue is a rising challenge for manufacturers. The coordination along the material flow is established on the exchange the information on manufacturing activities, logistics, and inventory among the manufacturing networks. Thus, a fusion of the current ERP/MES system which accommodates the open structure is needed in the crowdsourced manufacturing paradigm.

In the second scenario, the manufacturers have their value chain to serve their major manufacturing activities while sharing their excess manufacturing capabilities along with the manufacturing network. Manufacturers in this scenario operate production line with a mix of existed tasks from their own value chain and crowdsourcing tasks from the platform. In addition to the common challenges which are mentioned in the first scenario, manufacturers in the second scenario struggle with the balance of the existed tasks and the crowdsourcing tasks. The introduction of crowdsourcing tasks is from a high-frequency negotiation contracting process among the manufacturing network, which entails a resource matching process of innovation demands and sharable capabilities. This two-sided matching process requires the manufacturers to discover their shareable capabilities in terms of compatible products, time windows, and quantity. The way of sharable capability calculation is manipulating the order sequencing, batching, and balancing on the shop floor. A system of methods to help manufacturers maximize their shareable capabilities can be recognized as a significant barrier to participating in the paradigm of crowdsourced manufacturing.

3.3 Case Study of Tank Trailer Crowdsourcing

The running example in this paper is the transformation of the tank trailer industry to the paradigm of crowdsourced manufacturing. The tank trailer industry is driven by customers and rearranges manufacturing capabilities and technologies to achieve maximum satisfaction. Most tank trailer companies connect to the market by themselves and plan the products considering the customer needs they collected. A large volume of essential parts and system is relying on the corporation with external partners, including axles, braking system, pump assembly, accessories, to name but a few. From the after-sale service, the tank trailer companies can sense some customer needs, for instance, the need for minimizing maintenance efforts, meters of trailer mileage, tire pressure adjustment, etc. A broad search for solution suppliers can achieve the innovation of the tank trailer by defining a series of requirements to collect the solution. For example, the automatic tire inflation system can be a solution for tire pressure adjustment, and the corresponding manufacturer can serve the realization of this new tank trailer. The synergy of the existed capabilities and technologies builds up the backbone of the reconfiguration of manufacturers.

The volatile customer needs and a large population of suppliers bring a large amount of product variety to the tank trailer industry, as shown in Figure 3-1. The different fluids in various market sectors will lead to an extensive product variety, some of the fluids are flammable or explosive, some fluids may cause the fouling issues, some chemicals are erosional, some fluids require edible safe through the transportation, to name but a few. The specific customer has their personalized requirements on the accessories, includes but not limited to ladders, pump systems, toolboxes. Due to the laws and regulations are distinctive in different markets, the products vary in length and tonnage, cross-section shape, as well as the end shape. Currently, configure-to-order is a prevailing strategy to handle this variety. The practitioners integrate the modules and organize the manufacturing according to the customer orders.

This paper specifically focuses on the tank trailer industry in Mainland China, which shows a firm reliance on the manufacturing network. Nearly a hundred tank trailer companies have located in Quanpuzhen, Shandong Province in the past three decades and the total output of this industry cluster takes about two-thirds of the national market share (Gringer, 2018). The satellite map of this industrial cluster is shown in Figure 3-2, in which the red label represents a related manufacturer.

Figure 3-1 Variety of tank trailers

Figure 3-2 Map of Tank Trailer Industrial Cluster (Created from: Google Earth, 2018)

The cooperation of these manufacturer crowds relies on the long-term outsourcing and personal relationship of the owners. This paper will show a transformation roadmap of the installation of the crowdsourced manufacturing paradigm to the conventional manufacturing industries. By applying state-of-the-art technologies, the construction of the cooperative manufacturing relationship can be fully digitalized. At the same time, the coordination among the manufacturers can be achieved by the pervasive connections and transferable engineering software.

3.4 Crowdsourced Manufacturing Workflow

The paradigm shift to crowdsourced manufacturing implies offering the integration path of external partners into all activities in the value creation and capture along the value chain. Thus, the product manufacturing is fulfilled based on the collaboration of multi-parties in three physical domains: open innovation domain, crowdsourced manufacturing platform domain, and open manufacturing domain.

The open innovation domain is the front-end domain, which brings connections to the customers, as the transportation companies. The open innovator O has been identified as the primary decision agent in the open innovation domain, who takes in charge of collecting the CNs, sketching product design, as well as sales and aftersales service of the final product. Following the tank trailer example, O is a tank trailer manufacturing company which adopts the open business model and installs the crowdsourcing model as a crowdsourcer. The O collects the CNs and saves them into customer orders C^0 as the start of the product fulfillment process. After the completion of product design, O initiates the crowdsourced manufacturing process with the platform by delivering the product design files. The finish of product manufacturing will lead to the handover of final products to O, which enables the final sales of products to the customers.

The manufacturing agents M are the decision agents in the back-end of crowdsourced manufacturing, which is collected in the manufacturing agent domain. The M own the knowledge of generating manufacturing plans considering their processes constraints and resource utilization limitation, as well as the fulfillment capabilities of the actual manufacturing tasks. In the tank trailer example, M represents suppliers in the trailer industry cluster, which are divided into various clusters based on their specialty. According to the inherent properties of the value chains and industry, the cluster can be indexed with α, $\alpha \in [1, \mathrm{A}]$, where A is the total number of the clusters. For instance, the manufacturers in tank trailer industries can be divided into frame suppliers, axle companies, steel sheet factories, welding workshops, to mention just a few. These bidding agents can be denoted as μ, which is a collective set of the bidding manufacturer in every agent cluster α. The individual manufacturer can be denoted as $\mu_{n_\alpha}^{\alpha}$, $\bar{\mu}^{\alpha} = \{\mu_{n_\alpha}^{\alpha}\}|_{N_\alpha}$, where n_α is the index of bidding manufacturer in cluster α, $n_\alpha \in \mathbb{N}^{+}$, and N_α is the total number of manufacturers in the cluster α. The unification of the bidding manufacturers in all clusters is the bidding agent cluster, $\bar{\mu} = \{\bar{\mu}^{\alpha}\}|_{\mathrm{A}}$. Meanwhile, due to the heterogeneity of the operating environment, some M may determine not to participate in crowdsourced manufacturing., which are collected in the non-bidding agent cluster $\bar{\varphi}$. The individual manufacturer is denoted as φ_n, $\forall \varphi_n \in \bar{\varphi}$, where n is the index of agents in the non-bidding cluster and N is the total size of φ_n.

The crowdsourced manufacturing platform domain is the intermediate domain, which builds up the bridges between the front-end open innovation domain and the open manufacturing domain. The platform brokers P are the primary agents in the crowdsourced manufacturing platform domain, who take in charge of tasks processing and deliveries to O, contracting and coordination with M, as well as the submission of crowdsourcing contracting results. There are two virtual fields in the crowdsourced manufacturing platform domain as input and output, namely crowdsourcing information management and crowdsourcing contracting broker. The field of crowdsourcing information management is the interface to the open innovation domain, which has two databases to save DPs as design specs D^0 and PVs as process specs P^0. The other virtual field is the crowdsourcing contracting broker, which sends open calls to open manufacturing domain to invite M, collect responses from manufacturer crowds, and award the preferred M with supply contracts. The open call broadcasting and response collection is realized by two brokers. Invitation broker $\bar{P}^I$ realizes the invitation function of the crowdsourcing contracting mechanism. P_α^I follows the index of manufacturing cluster α, $\forall P_\alpha^I \in \bar{P}^I$. Similarly, the collecting and evaluation function is fulfilled by manufacturing evaluation brokers $\bar{P}^E$, where the individual evaluation broker is P_α^E, $\forall P_\alpha^E \in \bar{P}^E$. The index α follows the manufacturing cluster α, which indicate the accountability of the P_α^I and P_α^E. Project configuration manager P^C achieves the coordination of the front-end and back-end interfaces, who receives product design specs D^0 and restructures the product design to crowdsourcing tasks, as well as summarize the manufacturing contracts and save the process specs to P^0. The union of P^C, $\bar{P}^I$, and $\bar{P}^E$ is the platform brokers $\bar{P}$, $\bar{P} = P^C \cup \bar{P}^I \cup$

$\bar{P}^E$. The workflow of crowdsourced manufacturing, along with the example of tank trailer industries, is shown in Figure 3-3.

The workflow of crowdsourced manufacturing is started with the project initiation process by O after they finish the product design. The deliverables of the initiation process are the saved product design files. The second process is the accessing of the product structure from D^0 by project configuration manager P^C. The product structure is denoted as $\Delta = \delta_1 \times \dots \delta_k \times \dots \delta_K$, where δ_k, $k \in [1, K]$, depicts a specific manufacturing subtask like a trailer frame or pump system. The k is the subtask index, and K is the total number of the subtasks. Following a platform-based product development approach, δ_k can also be perceived as a product module. The structured Δ depicts the internal relationships of the product, e.g., the assembly structure. After Δ is retrieved, P^C restructures it to manufacturing request for quotation (RFQ) $\Delta_\alpha \in \Delta$, where $\alpha \in [1, \mathrm{A}]$, includes a set of the manufacturing subtask δ_q. The number of requests for quotation Δ_α follows the number of manufacturing agent clusters, thus Δ_α shares the cluster index α with $\mu_{n_\alpha}^\alpha$. The broadcasting of Δ_α is done by P_α^I as an invitation. M receives the Δ_α, analyzes the requirements, and makes the participating decisions. The participating agents $\mu_{n_\alpha}^\alpha$ respond with manufacturing bids. The manufacturing bids from each cluster α is collected in $B_\alpha, \alpha \in \mathbb{N}$. All the B_α are collected by $\bar{P}^E$ in the manufacturing bids set $B = \{B_1, \dots, B_\alpha, \dots, B_\mathrm{A}\}$. The $\bar{P}^E$ also evaluate these bids, thus select the preferred bids B_α^* and the corresponding winner μ^{α^*}. The winner agents are rewarded by manufacturing supply contracts $S = \mu^{1^*} \times \dots \mu^{\alpha^*} \times \dots \mu^{A^*}$, where the winner μ^{α^*} in each cluster α is organized by a cartesian product to entail a manufacturing network. The manufacturing supply contracts S will also be sent to P^E and the corresponding manufacturing bids B_α^* are saved to P^0 as process spec sets $B^* = \{B_\alpha^*\}|_{\mathrm{A}}$. After the execution of crowdsourced manufacturing, the final products can be manufactured by M and delivered to customers. As the final stage of crowdsourced manufacturing, the product-related information will be sent to O. The information of products serves the sale of the products and also provide aftersales services.

Different from the “cascading” model in axiomatic design, the product fulfillment process in the crowdsourced manufacturing paradigm is shown as “zigzagging.” The reason for this change is the involvement of external partners. Thus, innovative product fulfillment is achieved by the collaboration of all the decision agents in the fulfillment process. However, this kind of collaboration is forged in the form of contracting, and the coordination of product material flow (Jiao et al., 2006). Thus, the workflow in crowdsourced manufacturing can be characterized as a collaborative-negotiation based supply contracting process.

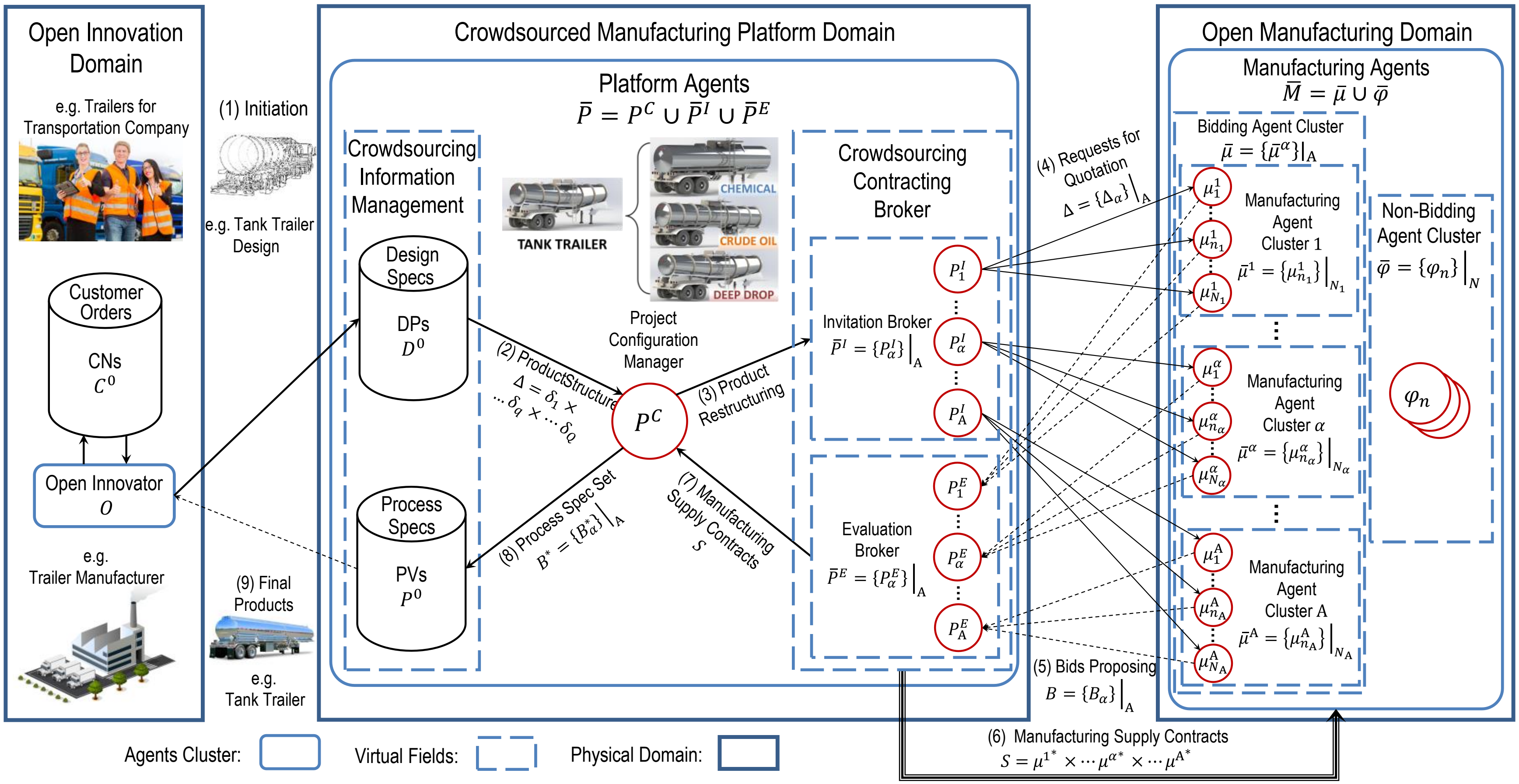


Figure 3-3 Crowdsourced manufacturing workflow

3.5 Holistic Frameworks of Crowdsourced Manufacturing

As an intermediate marketplace for open innovators and manufacturers, the platform serves a population of distinctive open innovators to connect a broader market, meanwhile invites extensive manufacturers to expand the capability arsenal and gain the network benefits. This impulse of expansion leads to a large number of value chains going through the platform company, which can be treated as a series of projects. The platform company uses crowdsourced information management fields to serve open innovators in the front-end, which provides databases to manage design and process specs. The project configuration manager P^C serves as a coordinator to manage the project workflow. The manufacturers are accommodated by the platform by the crowdsourcing contracting mechanism fields, where they broadcast the crowdsourcing tasks and collect the responses. A synergy of these three constructs a module to serve a value chain. The increasing number of value chains requires the platform to scaling up an appropriate amount of serving modules to serve the manufacturing of the products. The conceptual framework to demonstrate the platform-driven crowdsourced manufacturing workflow, which accommodates multiple projects, is shown in Figure 3-4.

A larger population of open innovators O bring a variety of customers, for example, tank trailers, container trailers, refrigerated trailers, along with others. These open innovators initiate various value chains as product design projects, which is indexed by $\lambda, \lambda \in [1, \Lambda]$, where Λ is the total number of initiated projects. The platform develops the corresponding interfaces to serve these open innovators and corresponding manufacturing agents clusters. The corresponding open innovator and platform brokers of project λ can be represented as O_λ and $\bar{P}_\lambda$, respectively. Each $\bar{P}_\lambda$ has a broadcasting output and bid collecting interfaces to connect the manufacturers M in manufacturer agent population, which is indexed by π, $\pi \in [1, \Pi]$. The total number of the manufacturer agent population is denoted as Π. The scaling up of the platform company leads a stacking of the serving modules which transforms the two-dimensional perspective of crowdsourced manufacturing workflow to a three-dimensional holistic framework.

The horizontal dimension is the direction of the workflow. The front-end is the open innovators, while the back-end is the manufacturing agent crowds. This dimension realizes the crowdsourced manufacturing workflow with all the processes shown in Figure 3-3. The vertical dimension is the functions of the platform. The upper interfaces serve the workflow from front-end to back-end, and the lower interfaces serve the opposite direction. At the front-end, the upper interfaces are the data management module to receive the project initiation from O_λ and save the product design specs into D_λ^0. The upper interfaces at the back-end let the $\bar{P}_\lambda^I$ to send the request for quotation to connect the M. The lower interfaces towards back-end handle the bids proposing from the M to enable the evaluation by $\bar{P}_\lambda^E$. The lower interfaces towards the front-end manage the process specs in P_λ^0 and interact with O_λ to inform the product delivery. The third dimension represents the depth, which entails a variety of value chains in this holistic view. The corresponding open innovator O_λ and a serving module $\bar{P}_\lambda$ in crowdsourced manufacturing platform P are arranged along each value chain, where $\bar{\bar{P}} = \{\bar{P}_\lambda\}|_\Lambda$.

Based on the holistic conceptual framework of crowdsourced manufacturing, Figure 3-5 illustrates the information flow and the networked material flow in the paradigm of crowdsourced manufacturing. The open innovator establishes the information connections to the customers to serve the collection of the CNs for product innovation, as well as sales and aftersales service for the final product. Meanwhile, the platform leads the manufacturing agent crowds to realize product manufacturing. After the final assembly, the products can be sent from the last manufacturing agents in the network to the customers. Therefore, two flows build up the linkage to the customer, the information flow from and to open innovators I_λ as the stimulus of the crowdsourced manufacturing paradigm, as well as the material flow from manufacturing agent crowds as the physical delivery of the products. In this case, the platform company acts as a bridge to link the information flow and material flow. The information flows between O_λ and $\bar{P}_\lambda$ initiate the crowdsourcing projects and set the product configuration to serve the crowdsourcing product fulfillment. Moreover, it also establishes the monitoring approach for the I_λ to supervise the manufacturing process.

At the back end of the platform company, the contracting information flow enables invitation of the M_π and allocate the tasks to forge of the manufacturing network. In the following manufacturing execution stage, the information flow also serves as the handler to coordinate the material flow inner the manufacturer agent crowds. Because the manufacturers utilize their specialties to maximize the economies of scale, one M_π can participate in multiple value chains. For instance, because of the commonalities between the value chains, a trailer axle company can participate in three value chains to connect the tank trailer, container trailer, and refrigerate trailer market, respectively. Moreover, various value chains imply different process precedence, and a manufacturer can serve distinct positions.

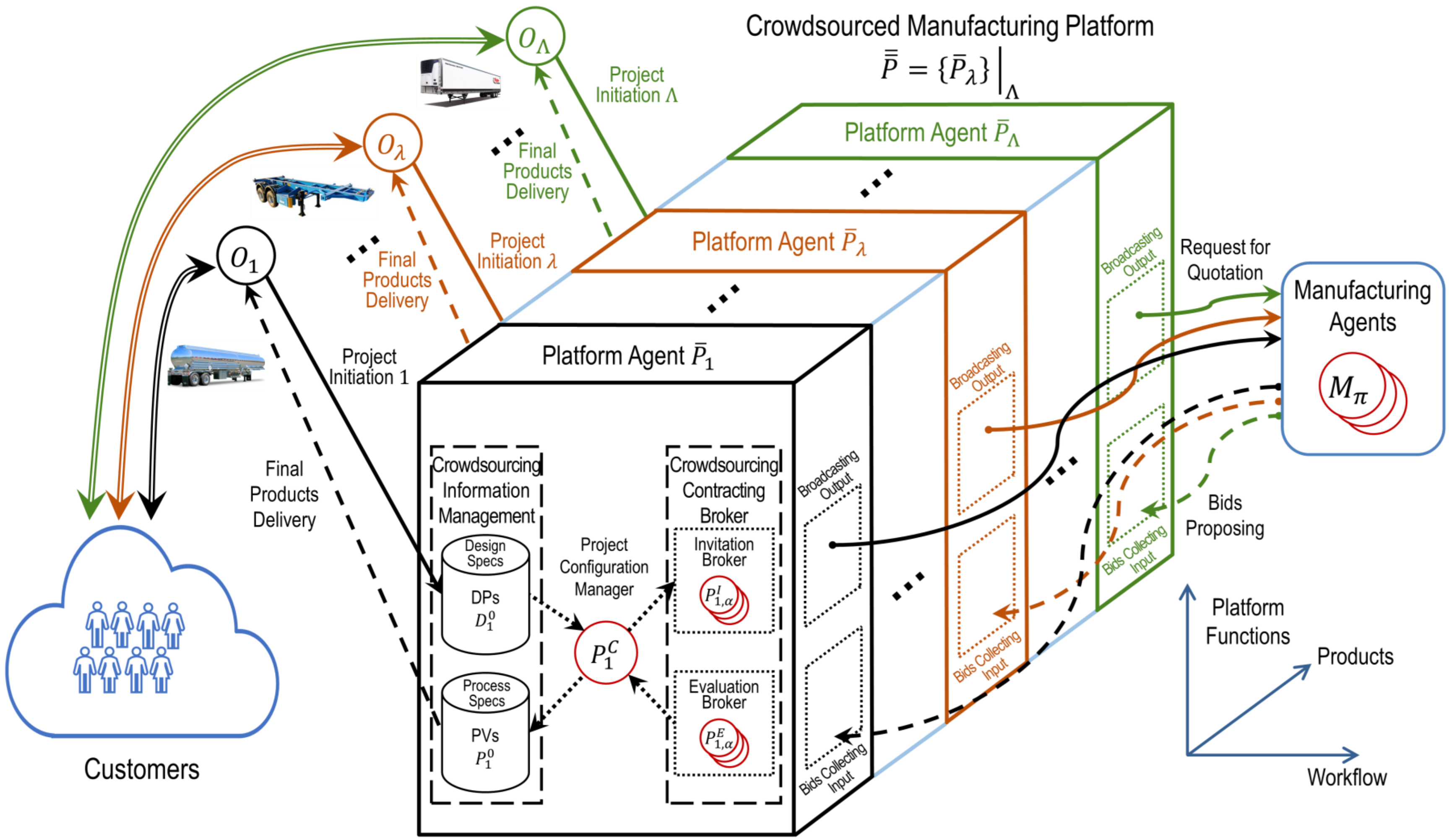


Figure 3-4 A holistic conceptual framework of the crowdsourced manufacturing workflow

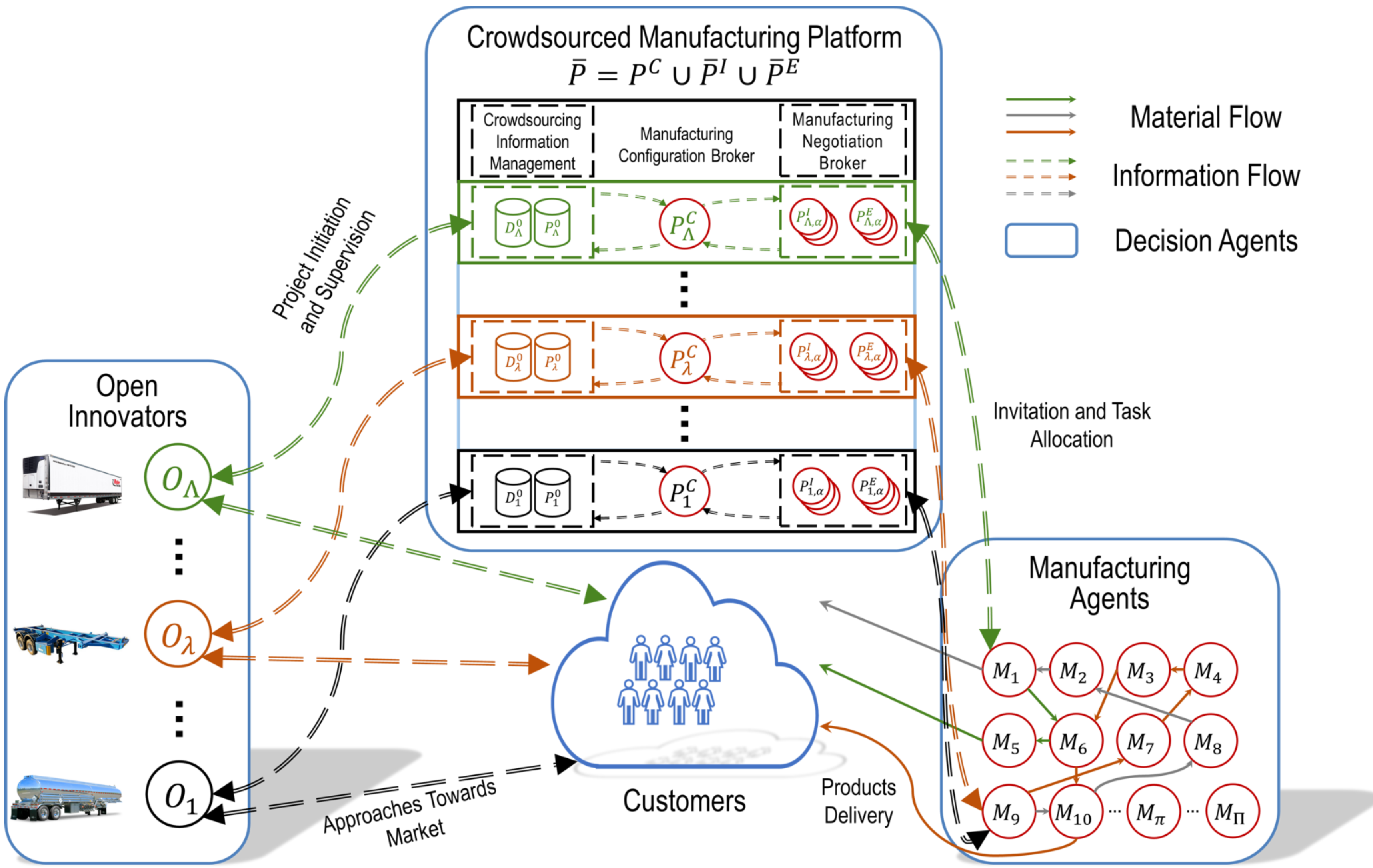


Figure 3-5 Information and material flow in crowdsourced manufacturing

3.6 Chapter Summary

This chapter proceeds dichotomy analysis of industrial applicability and stakeholder analysis of the platform-driven crowdsourced manufacturing, which identifies targeted industries and three critical stakeholders. Based on a running illustrative examples of tank trailer industries, the workflow of platform-driven crowdsourced manufacturing is proposed. The holistic framework is sketched to demonstrate the scaling up of the platform company, as well as information and material flow in platform-driven crowdsourced manufacturing. Such a profound understanding of these analysis and models provides a clear direction for a research agenda in the next chapter.

CHAPTER 4. FUNDAMENTAL ISSUES AND A REFERENCE MODEL TOWARDS MANUFACTURING AS A SERVICE

Recognize the paradigm shift towards crowdsourced manufacturing and the corresponding driving forces, this chapter implements a structural implication on platform-driven crowdsourced manufacturing and outlines fundamental issues from multiple views to summarize a MaaS reference model. From the stakeholder analysis in Chapter 3, these views are from open innovators, manufacturers, and platforms. The views from open innovators and platforms sketches functional requirements of a contracting evaluation services to ensure customer satisfactions and management optimal decision-making service to ensure prosperous manufacturer population, respectively. The view from manufacturers outlines the functional requirements of a series of task execution services along with the workflow in Chapter 3, which include task derivation and decomposition in product domain, logistic route planning in platform domain, as well as task acceptance and accommodation to link the external and internal material flow in manufacturing domain. In the end, section 4.6 proposes a MaaS reference model as a research agenda for critical technical elements to gear forward the development of platform-driven crowdsourced manufacturing for MaaS.

4.1 Structural implications of crowdsourced manufacturing

From the analysis of information and material flow in crowdsourced manufacturing, the operation of companies in one decision agent cluster is influenced by the collaboration with companies in the rest two decision agent clusters. For instance, the operation of the platform companies relies on the characteristics of the participated open innovators, as well as the capabilities and the variety of manufacturer crowds it linked. The performance of these external partners restricts the economic behaviors of the companies itself, like the platform can only link to the targeted market with the collaboration with the related open innovators and manufacturers. Meanwhile, the capabilities of external partners can technically support the expansion of the companies. The rest two crowds of partner structure a two-dimensional decision-making plane, which presents a decision space to choose a value chain and a position along with it. The company can explore the portfolio of collaboration potentials in that decision space, where the abundant and diversity of the partner crowds will determine the limits of participating in crowdsourced manufacturing. From a service-oriented perspective, the rest two crowds collaboratively construct a service system to serve the company as a user. This two-dimensional decision-making scheme generally exists in all three decision agent clusters, namely open innovators, platforms, and manufacturers.

From the trichotomy analysis of decision agents in crowdsourcing, each three decision agent clusters have their own standpoints as well as the motivations. Thus, the views of each decision agent cluster originate from different contexts of companies and seek various operational objectives. The distinctiveness of decision agents implies perpendicular relationships among the resulted views. In this regard, a cubic structure is proposed to represent this system of perpendicular view and corresponding decision-making planes, as shown in Figure 4-1.

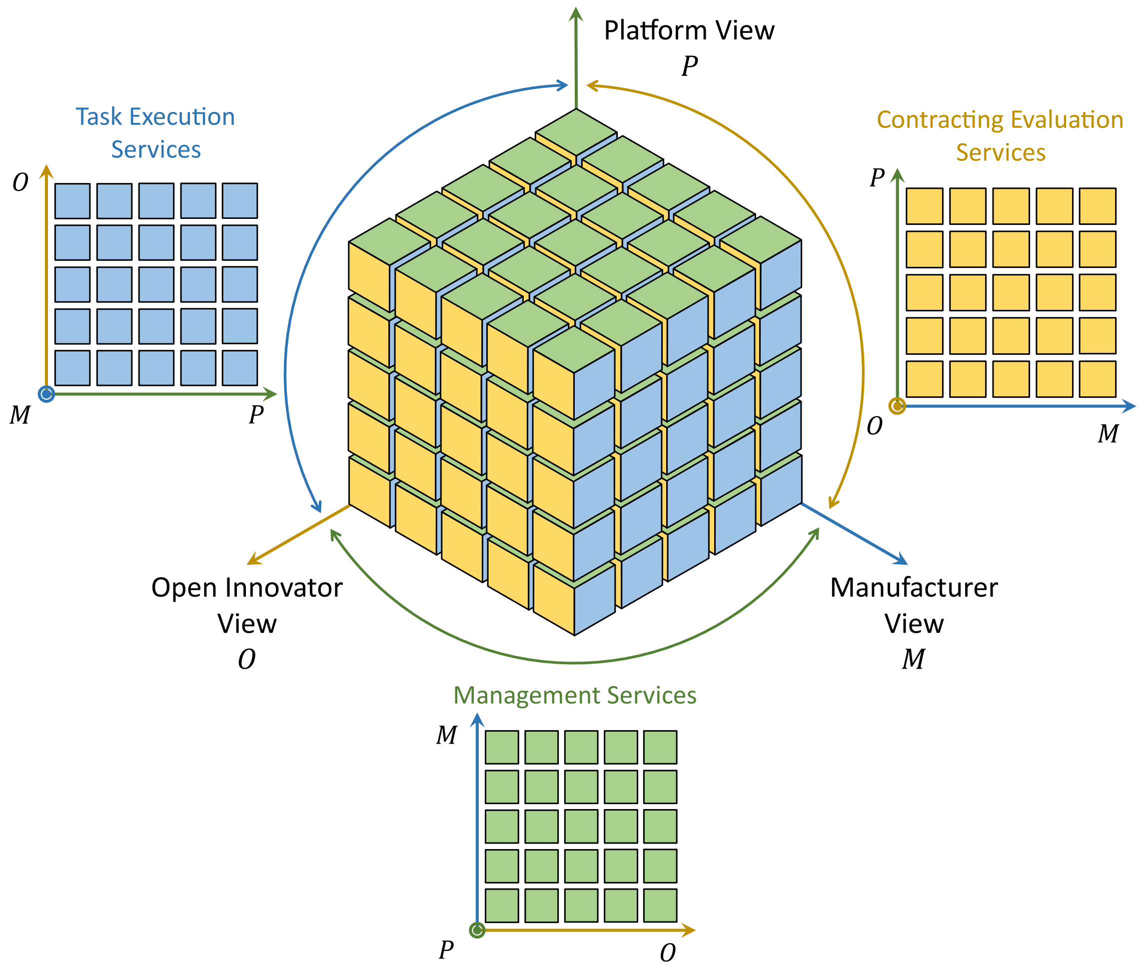


Figure 4-1 Structural implications of platform-driven crowdsourced manufacturing

There are three dimensions to represent the views of open innovators, platforms, and manufacturers. From each view, the rest two structure a discretized decision-making plane and acts as a portfolio of product fulfillment functions of a service system. The synergy of three views of the decision agent cluster leads the two-dimensional discretized decision-making planes to a three-dimensional block array to represents the selection and collaborative relationships among the decision agents. The block represents an engagement of three specific decision agent to construct a collaborative manufacturing relationship for innovative product fulfillment. The mapping of each pair of decision agent clusters implies a collaboration relationship to form a functional area for accommodating the external partners. This collaboration can achieve the acceleration of the execution of the value chain for targeted users. These three manufacturing functional areas entail the X-as-a-service paradigm, where each functional area aims to lead the crowdsourced manufacturing to be user-friendly to be quickly responsive and accommodate a variety of partners. In this regard, three service systems has been highlighted to serve views from decision agents in crowdsourced manufacturing.

4.2 Contracting Services for Open Innovators

From the position of the value chain, the open innovator holds a functional view that describes the functional requirements of a crowdsourcing value chain and the corresponding manufacturing network. This view is the result of the capability of connecting to the customers, selling products, as well as aftersales services. The impulse of participating in crowdsourced manufacturing is seeking the cooperation of platforms and manufacturers to fulfill the innovative products. Specifically, in an X-as-a-service paradigm, the demands of the open innovators require a service system that can provide service-oriented solutions of product manufacturing resources, capabilities, as well as the supply chain reconfiguration mechanisms with a maximized solution delivery efficiency and minimized deviation from customer expectations. Platform-driven crowdsourced manufacturing formulates a digitalized platform-manufacturer plane as a decision space of various capabilities and integrating methods.

As shown in Figure 4-2, platform-driven crowdsourced manufacturing leads the open innovator O to a broker-based dispersed manufacturing system, where selects the platforms P and manufacturers M as partners to realize their innovative product plans. The selection of the manufacturer is based on a robust evaluation mechanism to reflect their performance on customer requirements for maximum utility delivery. On the other hand, the platforms are selected according to their aggregation method of evaluation results for a better reflection on open innovator's preference. From the product fulfillment perspective, the MaaS acts like an e-commerce platform that offers access to the market of manufacturing capabilities customize the most appropriate product fulfillment services through crowdsourcing contracting methods. It should integrate evaluation mechanisms and aggregation methods into a crowdsourcing contracting services to configure a supply network considering engineering functional and business operational performance.

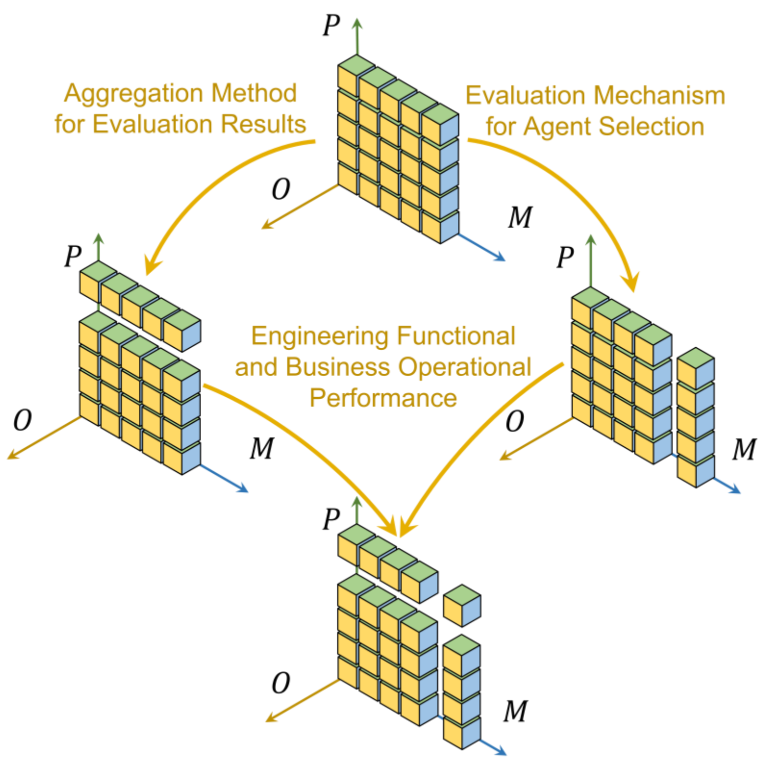


Figure 4-2 Functional view from open innovator

4.3 Task Execution Services for Manufacturers

The manufacturer holds a behavioral view which reveals the applications of a set of manufacturing technologies by managing manufacturing resources, process planning, and crowdsourced manufacturing tasks execution. Following MaaS strategy, task execution services aim to help manufacturers peeling peripheral activities by offering substitutive services. As shown in Figure 4-3, it forges the manufacturing network for manufacturer by providing logistic services, which mobilizes manufacturing resources and WIP according to the precedence relationship.

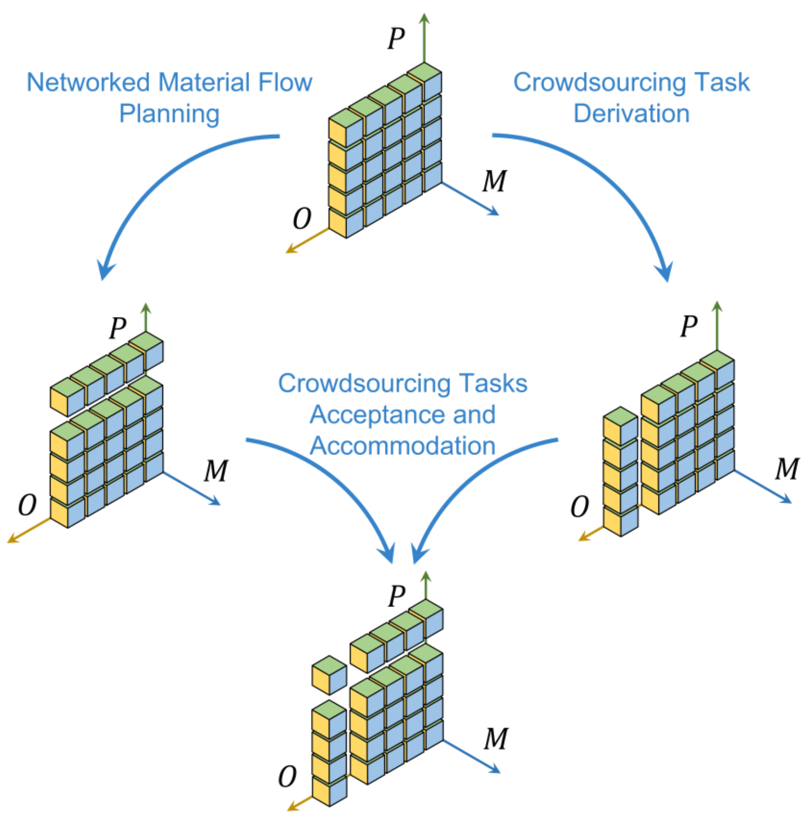


Figure 4-3 Behavioral view from manufacturer

A manufacturer describes the subdivision of the product realization process along the value chain. A crowd of them outline the technological possibilities for the value chain, as well as define the limits from the physical world. The impulse of concentrating on core competitive edges to achieve the economics of scales implies the manufacturers to require complementary cooperation among the manufacturer crowds. The manufacturer's purpose of participating the crowdsourced manufacturing is capturing the value it created, which in terms of finding a system of approaches to link the upstream and downstream partners, as well as to touch and connect the customers. Crowdsourced manufacturing formulates a platform-open-innovator plane as a decision space of various value chains and coordination mechanisms. The collaboration of open innovators and platform entails a task execution service system, which offers decision making support functions as optimal task derivation and decomposition mechanisms, material flow management tools, as well as task acceptance and accommodation interactive models. The synergy of these functions can accelerate the manufacturer's accommodation of crowdsourced manufacturing. The task execution service acts like an MES/ERP system on a large scale. The research tasks in the crowdsourcing task handling area can be shown in Figure 4-4.

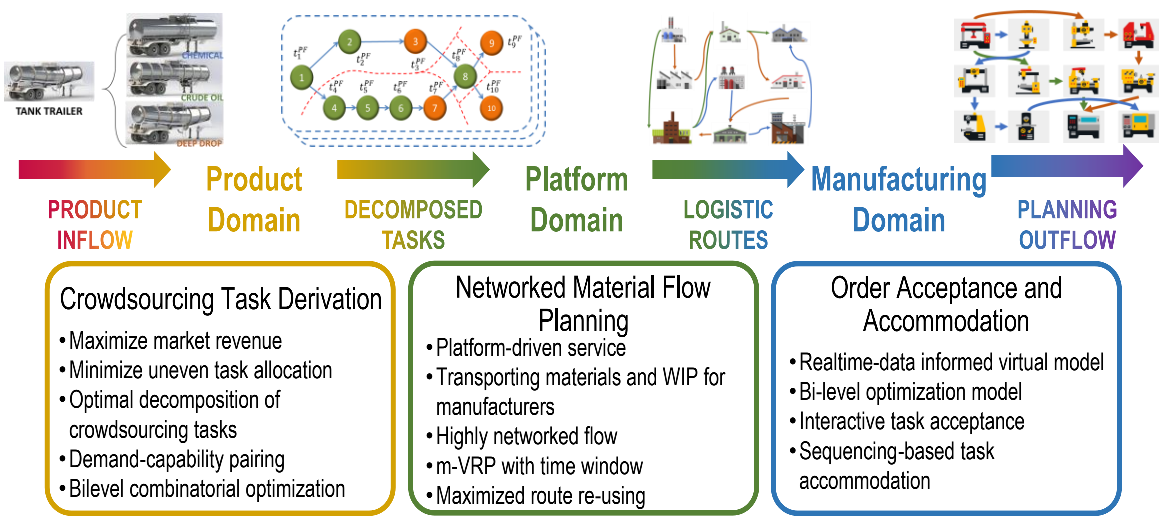


Figure 4-4 Task executions in crowdsourced manufacturing

4.3.1 Crowdsourcing Task Derivation

Crowdsourcing task derivation is embedded in the start phase of an innovative product fulfillment task. It should balance the requirement from customer side, which seeks a maximized market share, and constraints from manufacturer sides, which seeks a minimum manufacturing cost. It entails an optimal planning of the product family and manufacturer portfolio. These problems are linked by the product and process structure. The selection of the product modules will generate a BOM, which serves as an input of the manufacturer allocation problem. On the manufacturer side, an optimal decomposition of a product into crowdsourcing task. This process is essentially a combinatorial optimization problem which cluster several processes into a crowdsourcing task. The decomposition result will constrain the minimum manufacturing cost in the product family planning side. This interactive decision-making problem should be solved for crowdsourced manufacturing.

4.3.2 Networked Material Flow Planning

The material flow management domain in crowdsourced manufacturing aims to send as well as pick up the required material, WIP, subassemblies, or final products on time. Due to the large variety of value chains and the corresponding process variety, a manufacturer can be downstream partners for a set of upstream partners, since it is a vertex in a networked material flow network. Thus, the process variety will propagate from process domains to the logistics domain, therefore challenges the companies with keeping a reasonable cost as well as aligning customers, products, processes, and logistics for delivering an increasing product variety. From a platform-based perspective, a resource platform can collect the information from the manufacturer crowds, formulate the origins and destinations of the service demands, find the common routes in the corresponding transportation service tasks, and synchronize the manufacturing activities to achieve just-in-time. Thus, this logistic service function can be modeled as multiple vehicle route planning with time window (m-VRPTW) to handle a highly networked material flow and seeks a maximized route re-using for a crowdsourcing environment.

4.3.3 Crowdsourcing Task Acceptance and Optimal Accommodation

Since the crowdsourcing decision-making is summarized as a two-step process, solution generation by the manufacturer crowds and evaluation by the platform, the manufacturing task acceptance and accommodation is targeted to serve this interactively optimal decision making between these two decision agents. The searching for the sharable capabilities entails an order re-arranging process to allocate the existing orders on the shop floor to determine the acceptance of the orders. After the awarding process with manufacturing supply contracts to select the preferred manufacturers, the resources re-planning serves the management the mix of orders on the shop floor. The results of resource re-planning decision making are delivered through a re-sequencing process of the newly assigned crowdsourcing task orders and existed task orders. It can be summarized as an accelerator for manufacturers to better explore their manufacturing potentials and utilize them to fulfill open innovator’s demands through crowdsourcing.

4.4 Management Service for Platform

As an intermediate role played in the crowdsourcing value chains, the platform company holds a structural view that reconfigures the manufacturers to a supply network for product fulfillment. The platform operates a two-sided marketplace as its primary economic activities, which can link demands from open innovators and capabilities from manufacturers. Crowdsourced manufacturing presents an open-innovator-manufacturer plane to serve the platform, which provides various product fulfillment demands and a large volume of different capabilities. The intermediate role between open innovators and manufacturers of a platform implies a requirement of monitoring and management tool. Its function can be further decomposed to model the dynamics of manufacturer population and derive a set of revision protocols for optimal revenue sharing among manufacturers.

Figure 4-5 shows the structural view from platform. The manufacturer crowd is naturally divided into various manufacturing clusters according to their competitive edges. Thus, the manufacturers who are affiliated to one cluster are confronted with a massive impact of competition. Because of the existence of the awarding process by the manufacturing evaluation broker in the platform, only the best-performed manufacturer in each cluster can be selected and awarded with one contract. A robust management strategy should be derived for balancing and stimulating manufacturing capacity. Besides, the realization of the value chains requires a broad spectrum of competitive edges and a large volume of capacity, which is essentially a multi-party process. From this perspective, the relationships among the manufacturers are not only competition but also cooperation. A game theoretic model for describing this complex relationship is essential for platform. Maintaining an active and prosperous manufacturer crowd lead to a management service system as a solution for the platform company.

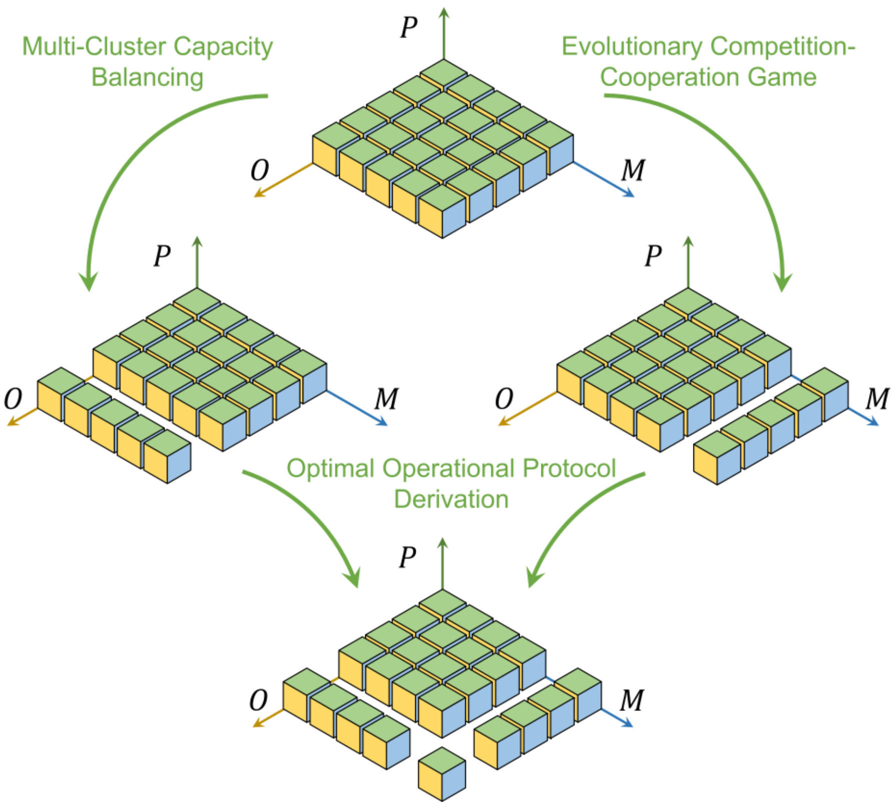


Figure 4-5 Structural view from platform

4.5 Information Service System as the Kernel

The collaboration among open innovators, manufacturer, and platform company is enabled by an information service system which can serve the interactions among these decision agents. It serves a prerequisite role in crowdsourced manufacturing, which solidifies the value chains in the form of contract and provides information exchanging functions for stakeholders. Different from the conventional outsourcing, crowdsourced manufacturing involves a larger number of external partners, which has high variety and geographically disputed. A contracting function to serve a such complex population along crowdsourced manufacturing. The information management is another essential function in this service system. It should allow the stakeholders access to and stream the product fulfillment data without security concern. The synergy of these functions implies a blockchain-based contracting and distributed information management service system which accommodate streaming data from Industrial IoTs.

4.6 A Crowdsourced Manufacturing MaaS Reference Model

Utilizing the functional, behavioral, and structural (noted as FBS) views, the crowdsourced manufacturing integrates several business functions following a coherent structure. This integration work is realized by the mappings from views from the three-dimensional structure of crowdsourced manufacturing, as shown in Figure 4-1. The mapping from one view to another implies a service system to serve the rest view. Thus, the combination of mapping relationships among open innovator's functional view, manufacturer's behavioral view, and platform's structural views sketches a cyclic MaaS reference model as a research agenda for crowdsourced manufacturing, as shown in Figure 4-6.

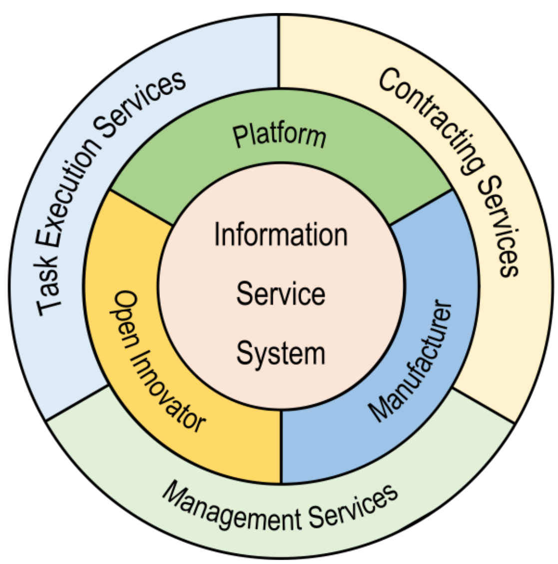


Figure 4-6 A MaaS reference model of platform-driven crowdsourced manufacturing

The view mapping between the behavioral view from manufacturers to a structural view from platform reflects digitalized contracting evaluation function module of MaaS system, where ensures the quality of product fulfillment service that open innovator received. Chapter 5 proposes a contracting mechanism incorporating with explicit and inexplicit criteria evaluation for quality assurance of the response selection and customer satisfaction.

The mapping from a functional view and a structural view reflects reconfiguration and operation of a manufacturing network, which enables the manufacturer to utilize core competing capabilities and link other manufacturers to outsource the peripheral activities. This material network requires crowdsourcing tasks execution function modules of MaaS system. Recognizing manufacturers as the targeted user, Chapter 6, Chapter 7, and Chapter 8 propose the task execution service by sketching a manufacturer-friendly service architecture which includes task derivation and decomposition methods, inter-manufacturer material flow planning, as well as task acceptance and shop floor scheduling interactive framework, respectively.

The mapping between the functional view from open innovators and behavioral one from manufacturers entails a large volume of digitalized product fulfillment demands, manufacturers, and the corresponding clusters. This mapping implies a management service functional module of MaaS for the platform to monitor and derive an optimal managerial protocol. Chapter 10 proposes a population-dynamics-based model for multi-cluster manufacturer crowds, as well as a protocol deriving methods based on evolutionary game. Chapter 9 proposes the information service system by implementing blockchain-based smart contracts and a distributed database. It provides the technical cornerstones for platform, manufacturer, and open innovator to solidify their collaboration relationship and information exchanging platform without security concerns.

4.7 Chapter Summary

This chapter examines the fundamental issues and proposes a MaaS reference model for platform-driven crowdsourced manufacturing. Based on the structural implications of platform-driven crowdsourced manufacturing, these fundamental issues include contracting evaluation services for open innovators, task executions services for manufacturers, management services for platform, and information service systems as prerequisites. A MaaS reference model is proposed as a research agenda of the following studies, which also elaborates the interrelationships underlying the following chapters.

CHAPTER 5. CROWDSOURCING CONTRACTING INCORPORATING EXPLICIT AND INEXPLICIT CRITERIA EVALUATION

MaaS through platform-driven crowdsourced manufacturing offers new opportunities for reaching external partner's knowledge and resources while allowing companies to focus on their core competencies. This chapter envisions a collaborative organization scenario of a crowdsourcing supply network, in which tournament-based crowdsourcing entails contracting decisions among design and manufacturing agents as a best-matching problem (BMP). The most important activity of the crowdsourced manufacturing process is the selection of the self-interested agents and organizations to dynamically form and configure a crowdsourcing network with sharable manufacturing capabilities. A robust agent selection mechanism relies on an effective mix of explicit and inexplicit criteria evaluation, which reflect engineering functional requirements and business operational expectations. This chapter develops a quantitative evaluation of manufacturers for engineering functional requirements based on information-content measurements. The preference on business operational reputation of manufacturer is achieved by decision-tree learning for monotonic classification. The evaluation results of different criteria are aggregated through multi-attribute utility theory. The proposed method determines which agent best satisfies the pre-defined engineering functional and business operational requirements from customers, which enables a better matching of fulfilling agents with customers from a manufacturer crowd. A case study of tank trailer mass customization through crowdsourcing is reported to illustrate the potential of MaaS through crowdsourced manufacturing.

5.1 Contracting Mechanism for Crowdsourced Manufacturing

Successful instantiation of MaaS requires collaboration among external partners, for which product fulfillment flow management is of primary importance. The open innovators and open manufacturers from various clusters are all engaged through an inter-organizational network and their crowdsourcing relationships are contractually tied to collaboration for fulfilling different knowledge and capabilities along with a coherent product fulfillment flow (Simard and West, 2006). Such a crowdsourcing contracting mechanism is akin to traditional supply contracting that formally formulates the transactions between the stakeholders to pursue the coordination of diverse decision makers and organize them into supply chain networks (Giannoccaro and Pontrandolfo, 2004). Together with the advancement of a collaborative product fulfillment process, the negotiation system is proposed to coordinate distributed enterprises (Mansouri et al., 2012). A negotiation contracting system entails a bilateral negotiation scheme coincides with a supply contract with an emphasis on the design of the efficient negotiation mechanisms, protocols, and strategies (Shin and Jung, 2004). In practice, every organization and entities in the supply chain networks are operating in heterogeneous environments with different objectives and constraints (Swaminathan et al., 1998). Since it is observed that a successful crowdsourcing decision-making process requires diversity and independence of the individuals in the crowds (Surowiecki, 2005), the crowdsourcing contracting is more challenging than conventional supply contracting.

The collaborative-negotiation process is generally divided into three consecutive phases, namely inviting, bidding, and awarding. Following the crowdsourced manufacturing workflow described in Figure 3-3, these three phases coincide with the interactions between platform agents P and manufacturing agents M. The inviting process acts as an RFQ, where the platform agents act as crowdsourcer to send an open call for solutions or capabilities of an independent crowdsourcing subtask. Subsequently, the crowds will solve the subtasks and response with a bid. The collection of bids from crowds can be described as a tournament for a reward, under a scheme of highest-bids-wins, considering the performance or efforts to the original subtask. The evaluation broker awards the best manufacturing agent based on the evaluation result. A sophisticated collaborative-negotiation contracting scheme should serve not only the interactions among crowdsourcing entities but also the motivation of the crowds and the quality of the final products. Such requirement implies an effective contracting evaluation mechanism to explore the maximum satisfaction from the perspective of customers.

The customer satisfaction of a crowdsourcing task is determined by the evaluation mechanism from platform companies, which is challenged by three aspects:

1) The crowdsourcing contracting evaluation is characterized as a large-scale multi-criteria decision-making problem. Different from the traditional outsourcing which invites designated partners to participate in the product fulfillment process, crowdsourcing relies on the wisdom of crowds, which implies the crowd can generate a large volume of solutions (Lakhani and Panetta, 2007). Constructing a supply network in a crowdsourcing environment implies a cooperation with new partners, which requires evaluation mechanisms to incorporate classification results of their business operational level. This

process should review their historical performance and exploit their reputation among customers. It addresses the necessity of an evaluation mechanism incorporating explicit and inexplicit criteria.

2) A stream of uncertainty is inevitable along the crowdsourcing product fulfillment workflow. From a design perspective, this uncertainty can be traced from the subjectiveness lying in the evaluation process and the variation of system performance (Jiao and Tseng, 1998, Siskos et al., 1984). In practice, the experts conduct evaluation based on their heuristic "rule of thumb", which has been historically done on an ad hoc basis (Thurston and Crawford, 1994). Establishing a model of the preference of the bids and the decision-making in the evaluation process to serve the contracting mechanism is critical to the realization of collaborative-crowdsourcing product fulfillment. From the manufacturing perspective, the performance of the production system shows strong dynamic and stochastic characteristic in the real manufacturing environment. Such characteristics are shown in the fluctuation of the throughput time, tolerance, and rejection rate. In addition, the evaluation of the contracting is in the early stage, which implies the design and manufacturing solutions are subjected to slight changes in later process. A method to mimic the uncertainty of the performance is critical in the development of evaluation mechanism.

3) Since the crowdsourcing is aiming to fulfill the diverse requests, the evaluation is a two-fold process. It is observed that the crowds in the crowdsourcing activities show a return of the vast amount of noise (Andrew, 2007). An evaluation mechanism should ensure the performance of the delivered solution can target the subtasks' requirements. From the product fulfillment perspective, the evaluation should pursue a maximized degree

of satisfaction (DoS) of customers, as well as a minimized deviation of the system performance to the requirements. Meanwhile, the trade-offs of the crowdsourcing tasks are reflected by various of conflicting criteria in evaluation. After single-criterion evaluation, the result should be ready to be aggregated for comparison. To sum up, the evaluation is a complex problem, a generic and formulated evaluation scheme is essential to handle the scale of evaluation.

5.2 Crowdsourcing Evaluation for MaaS

From the systematical perspective, the crowdsourcing system is an artificial and collaborative system, which has three interactive components: crowdsourcer, platform, and participants (Zhao and Zhu, 2014). For the purpose of product innovation, Lüttgens et al. (2014) categorize the interactions inner the crowdsourcing into six stages which are allocated in two sides of the platform. Following this conceptual framework, the flow of achieving an innovative product development is separated into two sides of the platform. The activities between crowdsourcer and the platform include initiating the project, contract negotiation with the platform, and reintegration of the subtasks to a collaborative product fulfillment network. These activities are essentially intermediary finding and authorization processes. In contrast, the activities fall between platform and participants can be perceived as a “tournament-based crowdsourcing” (Afuah and Tucci, 2012). There are three stages are summarized in the tournament, namely request for proposal formulating, open calls for solutions, and bids evaluation. The platform is authorized by the crowdsourcer to hold a tournament to broadcast the subtasks and rewards the corresponding participants which submit the best performance bids. After the best bids are evaluated, the preferred participants are selected from the crowds. The crowdsourcer and the selected participants will formulate their collaboration interrelationship in a fold of contracts and turn the crowdsourcing process to an integrated problem-solving process (Lüttgens et al., 2014). This conceptual framework gears forward the crowdsourcing from the perspective of a crowdsourcer in the context of an innovation project. Gong et al. (2019) gear forward this framework with a formulated product fulfillment process in an engineering context.

The spirit of the diverse participant population and decentralized problem-solving implies a collaborative crowdsourcing system can be viewed as a multi-agent system (MAS). MAS technology is a paradigm for the researching of the organizational architecture, decision-making process and coordination mechanism for distributed, knowledge-based, and autonomous problem-solving modules (Brenner et al., 2012, Gupta et al., 2001). MAS collects a set of agents as an agent population; each agent has their perspective and incentives to maximize its utility in a dynamic circumstance (Wooldridge, 2009). The agents work independently or cooperatively to solve the problem, and their local goals and objectives can be integrated by the negotiation of the supply contracts to achieve the system's overall goals (Kaihara, 2003). The MAS can be applied to analyze the supply chain coordination issue considering information, material, and financial flow, respectively (Dudek and Stadtler, 2005, Gaonkar and Viswanadham, 2001, Govindan and Popiuc, 2014). Jiao et al. (2006) propose a MAS to explore the collaborative negotiation product fulfillment contracting mechanism in a global network. Besides, MAS enables the modeling of coordination and behavior mechanism in a dynamic environment (Xiao et al., 2007).

The evaluation methods for crowdsourcing tasks have been studied heavily. The fuzzy analysis can represent and manipulate the imprecise evaluation criteria through the product fulfillment process (Ragin et al., 2006). To ensure the fulfillment of crowdsourcing tasks, the fuzzy ranking method can evaluate high variety product fulfillment plans by maximizing the overlap between the expectation of request for proposal and performance of the responses (Jiao and Tseng, 1998). The multi-utility theory provides the aggregation of results of intercorrelated multi-criteria evaluation problems under uncertainty to handle the subjectivity of the customer preference lying under the evaluation criteria (Claudio et al., 2014, Keeney et al., 1993).

The high variety product fulfillment seeks an efficient information delivering to the customers and avoid the deviation and redundant efforts of the system performance (Du et al., 2006). The evaluation methods based on the information theory have been developed to accommodate large variety in a crowdsourcing era (Zhao et al., 2016). By measuring the effective information delivery from the crowds, the candidates can be selected by fuzzy information axioms (Akay et al., 2011).

5.3 Engineering Functional Evaluation with Explicit Criteria

Contracting evaluation measures the performance of proposed solutions according to their capabilities of fulfilling various requirements. This section treats the crowdsourcing contracting evaluation as a multi-criteria decision-making process, which combines inexplicit business operation and explicit engineering requirements into the solution selection. In this regard, DoS function is introduced to quantify the evaluation result to a value between 0 and 1. And a multiplicative multi-attribute utility theory is applied to aggregate the DoS vector. The evaluation of design and manufacturing bids has different characteristics, but information content theory can evaluate their effective deliveries to the requirements.

5.3.1 Information Content Measure and DoS Formulation

Manufacturer $\mu_{n_\alpha}^{\alpha}$ will configure a production system and propose it as a manufacturing bid B_{α,n_α}, where n_α is the index of manufacturer in each cluster α. This bid acts as a manufacturing plan for fulfilling RFQ Δ_α along its process routes. B_{α,n_α} can be evaluated based on the performance of the configured production system. Discrete-event simulation (DES) has been widely used to imitate the operations of a real-world agent-based production system by modeling the changes of state variables at a discrete set of points in time (Borshchev and Filippov, 2004). The stochastic model of the manufacturing system can be established based on the output analysis of the DES (Alexopoulos et al., 1998).

In this case, the preference of a RFQ for a design bid can be modeled by a preference function, which is represented in a form of utility function of the system performance. The utility function models the range of performance as perception from function domain. The preference function for a performance of system is represented as $u(Pr)$. The probability of a manufacturing bid can fulfill the corresponding RFQ depends on the performance range it achieved, which can be represented in the form of probability distribution function (PDF) $p(Pr)$. The calculation of precepted utility of a bid's performance can be quantified based on the product of preference $u(Pr)$ and PDF $p(Pr)$ of performance variable Pr in the fulfillment range. Moreover, the aggregation of evaluation results can follow the similar multi-attribute utility scheme. In this regard, this section will explore the evaluation mechanism for manufacturing bids in a generic form.

To evaluate a bid's performance on fulfilling a certain requirement, the method of information content measurement is used. In the original formulation of the information content, the preference of the design range is assumed as uniform. However, because of the preference of the requirements, a triangular preference function shows the superiority in modeling the expected design and manufacturing performance range (Jiao and Tseng, 2004). The preference function of the fulfillment range can be generalized as $u(Pr)$. The lower and upper limits of these requirement ranges are defined as: $[Fr^L, Fr^U]$. The PDF of a bid's system performance can be generally represented by $p(Pr)$. The lower and upper limits of these performance range can be generally defined as: $\forall Pr \in [Pr^L, Pr^U]$. The information content **I** is a measurement of $P(Pr)$, which is defined as equation (5.1).

$$\mathbf{I} = \log_2 P(Pr) \tag{5.1}$$

As illustrated in Figure 5-1, the probabilities of successfully fulfill the expected performance $P(Pr)$ can be calculated by the integration of the precepted probability of success, which is the integral of the product of preference $u(Pr)$ and performance $p(Pr)$. It models the perceived system performance over the fulfillment range from the customer, which is shown in equation (5.2).

$$P(Pr) = \mathrm{E}[\, u(Pr)] = \int_{Fr^L}^{Fr^U} u(Pr) \cdot \, p(Pr)\, dPr \tag{5.2}$$

And the DoS of a bid towards a certain requirement is formulated in equation (5.3).

$$DoS = \frac{1}{1-\mathbf{I}} = \frac{1}{1-\log_2 \int_{Fr^L}^{Fr^U} u(Pr)\cdot\ p(Pr)\,dPr} \tag{5.3}$$

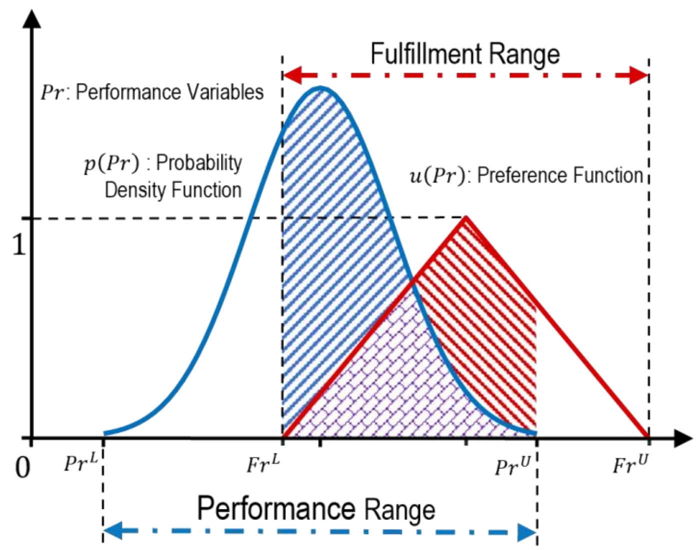


Figure 5-1 Preference function and performance distribution

5.3.2 Multi-criteria Contracting Evaluation Representation

This research views bid evaluation as a series of multi-criteria decision-making problems in the context of collaborative-crowdsourcing product fulfillment, thus ensuring the fulfillment of the requirements regarding DPs and with a coordinated quantity and lead time. The manufacturing bids are collected in the finite set B_α. Each bid set B_α will be evaluated by the evaluation criteria set $\mathbf{c}^{\mathrm{F}_\alpha} = \{c_1^{\mathrm{F}_\alpha}, \dots, c_{r_\alpha}^{\mathrm{F}_\alpha}, \dots, c_{R_\alpha}^{\mathrm{F}_\alpha}\}$, where r_α and R_α are the index and total number of criteria in $\mathbf{c}^{\mathrm{F}_\alpha}$. The performance of a design bid $B_{\alpha,m_\alpha}$is measured by the criteria $c_{r_\alpha}^{\mathrm{F}_\alpha}$ and noted as the $DoS\left(c_{r_\alpha}^{\mathrm{F}_\alpha}, B_{\alpha,m_\alpha}\right)$where $B_{\alpha,m_\alpha} \in \mathbf{B}_\alpha$, $c_{r_\alpha}^{\mathrm{F}_\alpha} \in \mathbf{c}^{\mathrm{F}_\alpha}$. In a multi-criteria evaluation condition, the evaluation result of a bid B_{α,m_α} can be represented by a R_α-dimensional vector:

$$DoS\left(B_{\alpha,m_\alpha}\right) = \left[DoS\left(c_1^{\mathrm{F}\alpha}, B_{\alpha,m_\alpha}\right), \ldots, DoS\left(c_{r_\alpha}^{\mathrm{F}\alpha}, B_{\alpha,m_\alpha}\right), \ldots, DoS\left(c_{R_\alpha}^{\mathrm{F}\alpha}, B_{\alpha,m_\alpha}\right)\right]. \quad (5.4)$$

To model the relative importance among the vector, a weighting factor is introduced and noted as w, for a R_α -dimensional vector. The evaluation representation of manufacturing bids is formulated similarly. The evaluation result of a bid can be aggregated to total DoS, which is denoted as $TDoS\left(B_{\alpha,m_\alpha}\right)$ and defined as:

$$TDoS\left(B_{\alpha,m_\alpha}\right) = \sum_{r_\alpha=1}^{R_\alpha} w_{r_\alpha} \cdot DoS_{c_{r_\alpha}^{\mathrm{F}\alpha}}\left(B_{\alpha,m_\alpha}\right). \quad (5.5)$$

However, in practice, the DoS for different requirements are heterogeneous and correlated per se. Moreover, the multi-attribute utility theory is proposed to handle the underlying correlation (Ji et al., 2013):

$$UDoS\left(B_{\alpha,m_\alpha}\right) = \frac{1}{K}\left[\left[\prod_{r_\alpha=1}^{R_\alpha}\left(K \cdot w_{r_\alpha} \cdot DoS_{c_{r_\alpha}^{\mathrm{F}\alpha}}\left(B_{\alpha,m_\alpha}\right) + 1\right)\right] - 1\right], \quad (5.6)$$

where the $UDoS\left(B_{\alpha,m_\alpha}\right)$ is normalized $DoS\left(B_{\alpha,m_\alpha}\right)$ with $\mathrm{F}_\alpha^{\mathrm{D}}$. K is a normalizing constant which scales $UDoS$ from 0 to 1. K can be derived from the equation (5.7).

$$1 + K = \prod_{r=1}^{R_\alpha}\left(1 + K \cdot w_{r_\alpha}\right) \quad (5.7)$$

Moreover, in the multiplicative form in equation (5.5), w_{r_α} is different from the additive form. It is not viewed as a weight, but rather an attribute-scaling parameter for accurate trade-off making (Claudio et al., 2014). The sum of the weights should not require to be exactly 1 in additive form equation (5.4) (Lewis et al., 2006). If the K is 0, it indicates there is no preference of the attributes, and the equation (5.5) is equivalent to the equation (5.4) (Krishnamurty, 2006). After the evaluation, the results can be collected in a finite set with M_α elements, and the most preferable bid can be selected by finding the minimum or the maximum value in the set:

$$\max/\min\left(\{UDoS(B_\alpha)\}_{M_\alpha}\right) \rightarrow B_\alpha^* \in \mathbf{B}_\alpha. \tag{5.8}$$

5.4 Decision Tree-based Evaluation for Inexplicit Criteria with Monotone Ordinal Measures

5.4.1 Intangible Criteria Evaluation

Based on the characteristics of the criteria, evaluation problems can be divided into both tangible criteria evaluation and intangible criteria evaluation. The former indicates approaches to quantify the object performance according to the given criteria, like most of the evaluation for engineering performance. However, not all performance can be quantified with numeric values based on the criteria requirements. In this scenario, imprecise linguistic words can be used to fuzzily evaluate the performance, like in some criteria for business performance.

Because the objective of evaluation is to compare the performance of different objects from a certain perspective, to replace absolute numeric measurement with ordinal measurement for describing the DoS is an approach to intangible criteria evaluation. By doing so, objects can be ranked and ordered based on their performance. This will not violate the inexplicitness or fuzziness regarding the intangibility of a criterion. Therefore, as long as the contracting brokers keep the evaluation mechanism consistent, the ranking or ordering of the object performance will be based on the same logic, making the objects at different time points comparable. Furthermore, such evaluation is supposed to be monotone, which means when one object dominates another, the evaluation of the dominator will not be worse than the dominated one.

In this regard, the evaluation of intangible criteria can be modeled as a monotonic classification problem. To address this problem, the rank entropy-based decision tree is used to learn the evaluation mechanism under intangible criteria, in which rank entropy is used to branch the decision tree based on the data monotonicity.

5.4.2 Definition of Monotonic Classification

Monotonic classification refers to ordinal classification problems with the monotonic constraint. Let A be an instance space where p is the number of attributes, which can be noted as $A = A_1 \times A_2 \times \ldots \times A_p$. Let $U = \{x_1, x_2, \ldots, x_n\}$ be a set of objects in the instance space A, with D being the ordinal decisions or labels of these objects. The value of the attribute or the decision related to x_i can be expressed as $v(x_i, a)$ or $v(x_i, D)$, where $a \in A$. In the ordinal relation, $\leq$ is used to describe no worse than between two objects. For example, x_i is no worse than x_j in terms of D can be noted as $v(x_j, D) \leq v(x_i, D)$ or $x_j \leq_D x_i$. Usually, x_i dominates x_j refers to that every attribute value of x_i is no worse than x_j. Based on this concept, a predicting function f that relates A to D can be expressed as below:

$$f: U \rightarrow D \tag{5.9}$$

The monotonic constraint is then defined as below, which should always be satisfied in monotonic classification:

$$x_i \leq x_j \Rightarrow f(x_i) \leq f(x_j), \forall x_i, x_j \in U \tag{5.10}$$

In other words, if x_j is dominated by x_i, the decision of x_j will not be worse than that of x_i, not vice versa.

5.4.3 Rank Entropy-based Decision Tree

Rule extraction from monotonic data attracts some attention from the domains of machine learning and decision analysis. Decision tree induction is an efficient, effective, and understandable technique for rule learning and classification modeling (Quinlan, 2014), where a function is required for evaluating and selecting features to partition samples into finer subsets in each node. The rank entropy measure originates from Shannon's information entropy, which is robust in evaluating features of the monotone dataset (Hu et al., 2011). Also, this measure reflects the ordinal structures in monotonic classification. Therefore, a decision tree algorithm based on the rank entropy measure is used in this study.

Some preliminary definitions are given for introducing the rank entropy-based decision tree.

$$[x_i]_B^{\leqslant} = \{x_j \in U \middle| x_i \leq_B x_j\}, \text{where } B \subseteq A \tag{5.11}$$

$$[x_i]_D^{\leqslant} = \{x_j \in U \middle| x_i \leq_D x_j\} \tag{5.12}$$

Equation (5.11) and (5.12) describe objects no worse than x_i in terms of attributes or decision. Similar to the concept of the information entropy, the ascending rank entropy of an object set U is defined as below:

$$RH_B^{\leq}(U) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{|[x_i]_B^{\leq}|}{n} \tag{5.13}$$

The ascending rank joint entropy of an object set U is defined as

$$RH_{B\cup D}^{\leq}(U) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{|[x_i]_B^{\leq} \cap [x_i]_D^{\leq}|}{n}. \tag{5.14}$$

The ascending rank mutual information (RMI) of an object set U is defined as

$$RMI^{\leq}(B,D) == -\frac{1}{n}\sum_{i=1}^{n} \log \frac{|[x_i]_B^{\leq}| * |[x_i]_D^{\leq}|}{n * |[x_i]_B^{\leq} \cap [x_i]_D^{\leq}|}. \tag{5.15}$$

The rank entropy-based decision tree uses the RMI value for the branch operation. Essentially, RMI describes the degree of monotonicity between the attribute set B and the decision set D. The workflow of the rank entropy-based ordinal decision tree is shown in Table 5-1.

5.4.4 Monotone Decision Tree Pruning

Like other supervised machine learning techniques, decision tree learning also faces the overfitting problem during its training process. Given a certain dataset, without predefined limitations on the tree structure, the decision tree will grow larger until till it fits all the data in the dataset. This can guarantee the prediction performance with enough specialization on the existing dataset by increasing the model complexity, but the trained decision tree may perform badly with poor generalization on unseen data. On the other hand, a small tree model may not express important data characteristics due to the scale of its complexity. Therefore, a balance can be made between model generalization and specialization to address the overfitting problem, and pruning is the technique for overfitting in decision trees.

Table 5-1 Algorithm workflow of rank entropy-based ordinal decision tree

Step 1: Generate the root node with input sample data.
Step 2: If this node satisfies the stopping criterion:
make this node as a leaf.
else
branch this node.
Step 3: for each attribute $a_i \in Criteria$
for each value $c_j \in a_i$
divide samples into two subsites with c_j.
If $v(a_i, x) \leq c_j$
$v(a_i, x) = 1$
else
$v(a_i, x) = 2$
calculate $RMI_{c_j} = RMI(a_i, D)$.
end j
end i
Step 4: select a_i and c_j such that $RMI = \underset{i}{\arg\max} \max_{j} RMI(a_i, c_j, D)$.
Step 5: If $\max RMI \geq \varepsilon$
branch this node using the attribute a_i and the value c_j.
else
stop branching this node.

There are two pruning approaches based on when pruning happens: pre-pruning and post-pruning. Pre-pruning is to set stopping criteria during the training process. For instance, the maximum tree depth or minimum information gain can be specified to stop the tree from growing deeper or splitting before the decision tree fits the whole dataset.

Pre-pruning is a fast and efficient method to avoid overfitting. Post-pruning approach refers to pruning the tree after the training process is complete. The idea is to substitute some subtree with a leaf node to reduce the model complexity based on some measures. A common method is the cost-complexity pruning (Breiman et al., 2017), which proposes a cost-complexity function to optimize on subtrees:

$$R_{\vartheta}(\mathrm{T}) = R(\mathrm{T}) + \vartheta \cdot |\widetilde{\mathrm{T}}| \tag{5.16}$$

where T is a decision tree, $R(\mathrm{T})$ is the prediction error of the tree, $|\widetilde{\mathrm{T}}|$ is the number of leaf nodes, and ϑ is a regularization parameter. The use of this evaluation function is to calculate its value for every node in the tree, considering that node is being pruned. And the node with minimum function value should be the node to prune. By adding a penalty cost on model complexity, this evaluation approach can find a smaller decision tree with better generalization ability.

Different from non-monotone decision trees, monotonic constraints must be satisfied in the tree leaf nodes, while direct post-pruning on a monotone decision tree may make the tree nonmonotone. Therefore, pruning monotone decision trees needs to guarantee the monotonicity while reducing its complexity. To address this issue, several fixing methods are proposed to make a nonmonotone decision tree monotone through minimal adjustments. In this way, the balance between model complexity and prediction performance on unseen data can be made by continuously pruning and fixing the tree.

In this work, the best fix method is used for monotone tree pruning (Feelders and Pardoel, 2003). This pruning method prunes the parent node of a nonmonotone leaf that brings the largest decrease in the number of nonmonotone leaf pairs. To avoid prune parent nodes generated in an early stage, the pruning process is conducted bottom-up and firstly deals with nodes with minimum number of descendants. If multiple nodes have the same number of nonmonotone pairs, the node with the least number of observations is selected to be pruned. This process is conducted until the tree is monotone again.

In conclusion, pruning avoids overfitting from the training dataset by reducing the model complexity, so that it predicts better on unseen dataset. Pruning will be beneficial when the objects in the instance space are difficult to be enumerated.

5.5 Aggregate Business Ranking Result with Functional Preference

A comprehensive agent comparison needs the aggregation of both business ranking and engineering functional preference. In this thesis, the multi-attribute utility theory is used for aggregation, considering equation (5.6) provides an architecture to accommodate ranking result and the ordinal results from decision tree classification can be defined as a number from 0 to 1. In this sense, the business ranking result can be viewed as another evaluation attribute. The aggregation can be formulated as below:

$$UDoS(\mu^{\alpha}) = \frac{1}{K}\left[(K \cdot w_{eng} \cdot UDoS + 1) \cdot (K \cdot w_{bus} \cdot DoS_{bus} + 1) - 1\right], \tag{5.17}$$

where DoS_{bus} is a predefined value based on the business ordinal ranking, and K is derived from:

$$1 + K = (1 + K \cdot w_{eng}) \cdot (1 + K \cdot w_{bus}) \tag{5.18}$$

5.6 Case Study on Crowdsourcing Contracting Evaluation

Tank trailer manufacturing is a typical manufacturing process that involves a variety kinds of assembly processes and parts. To meet the purpose of carrying different kinds of chemical substances, varieties of tank trailers are designed which also form a huge tank trailer family. A case study of tank trailer through platform-driven crowdsourced manufacturing is used to illustrate the proposed theory. Through the platform-driven crowdsourced manufacturing flow, the illustration of the contracting mechanism and the implementation of bid evaluation and agent selection have been explored.

Figure 5-2 shows the structure of the primary objects of a tank trailer family which combines the product variants and process variants. The name of each assembly operation and manufacturing operation in the Generic Product and Process Structure (GPPS) are shown in Figure 5-2. In a generalized tank trailer manufacturing process, there are at least two manufacturing operations and eight assembly operations. Each raw material, purchased part, manufacturing operation and assembly operation contains amounts of alternatives. Therefore, to design a proper tank trailer that meets individual customer needs, a massive number of alternative parts and operations need to be evaluated in multiple design criteria before making decisions. This evaluation process can be extremely complex due to massive number of alternatives. DoS is employed to evaluate design alternatives in the illustrative case.

Based on the GPPS of the tank trailer family, an ontology model could be built as shown in Figure 5-2. In the ontology model, purchased parts, raw material and assembly are class type data. Different from the GPPS, process operation is not shown in Figure 5-2. Arrows in the figure show the subordinate relationship between classes. For instance, Insulation, ‘Out Layer’ and ‘Vessel Assembly’ are subclasses of ‘Tank Sub-assembly 2’.

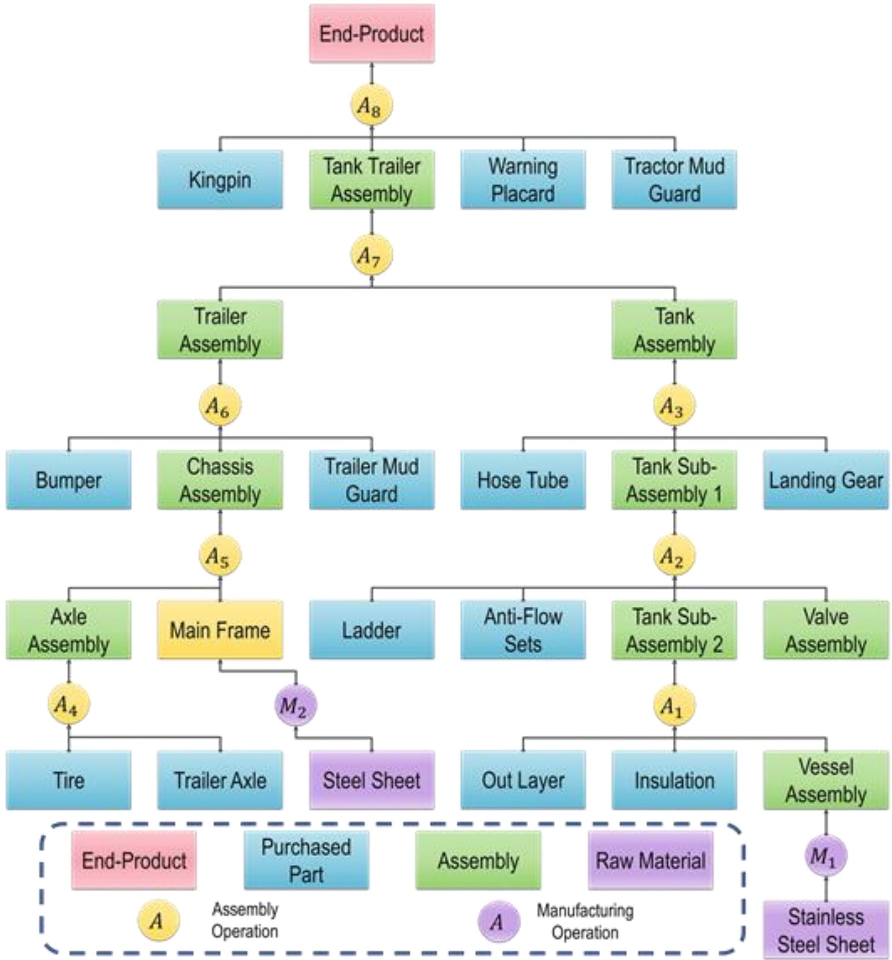


Figure 5-2 Generic product and process structure of a tank trailer family

5.6.1 Engineering Performance Evaluation with Multi-attribute Utility Theory

As the product fulfillment process and the crowdsourcing supply contracting mechanism depicted previously, O takes in charge of the generation of C^0. This C^0 is a series of the customer orders, specified the expectations for a trailer to fulfill the tank usage process. These orders are sent to virtual field of crowdsourcing information management and saved as design specs D^0. These design specs have internal hierarchical and precedence relationship, and the products are structured to $\Delta = \delta_1 \times \dots \delta_q \times \dots \delta_Q$. Before sending these products to the crowdsourcing invitation broker $\bar{P}^I$, project configuration manager P^C restructures Δ to RFQ Δ_α, according to the number of manufacturing agent clusters. In our case, invitation broker P_2^I sends the RFQ $\Delta_2$to the crowd as an open call and specify the evaluation criteria to guide the evaluation process. After receiving RFQ, 4 agents in μ^2 decide to bid, which are noted as $\{\mu_1^2, \mu_2^2, \mu_3^2, \mu_4^2\}$. Accordingly, 4 manufacturing bids are collected by the P_2^E in a finite set B_2, where $B_2 = \{B_{2,1}, B_{2,2}, B_{2,3}, B_{2,4}\}$. In this case, the bids are described in the Table 5-2.

P_2^E evaluates these bids based on their functional performance. Based on the mechanism established in Section 5.3, the evaluation methods can be decomposed into the following steps. Firstly, P_2^E specifies the corresponding range parameters of P_{Fr} and P_{Pr} for every criterion and bid, thus the preference function of expected performance and the PDF of achieved performance is established. Firstly, P_2^E specifies the corresponding range parameters of P_{Fr} and P_{Pr} for every criterion and bid, thus the preference function of expected performance and the PDF of achieved performance is established. Secondly, using equation (5.1) and equation (5.2), the information contents I can be derived. Thirdly, calculating the DoS using equation (5.3) and aggregate these DoS. At last, the preferred bid can be selected by the rule which is depicted in the equation (5.6). The evaluation process for the illustrative example is demonstrated in the Table 5-2. The engineering evaluation results can then be aggregated with agent business performance for final agent selection.

Table 5-2 Evaluation process for manufacturing bids

Evaluation Criteria of M_2 Frame Welding for a Chemical Tank Trailer																
Evaluation Criteria		Strength						Dimension						Estimated Cost (USD)		
		Tensile Strength (MPa)			Flexural Strength (MPa)			Tolerance of Upper Surface (mm)			Verticality of Main Frame (MPa)					
Fulfillment Range		$u(Pr)$			$u(Pr)$			$u(Pr)$			$u(Pr)$			$u(Pr)$		
		Fr^L	Fr^P	Fr^U	Fr^L	Fr^P	Fr^U	Fr^L	Fr^P	Fr^U	Fr^L	Fr^P	Fr^U	Fr^L	Fr^P	Fr^U
		250	500	500	550	800	800	-14	0	14	-2	0	2	1000	1000	2000
Attribute Scaling Constant	w_a	0.2			0.1			0.25			0.1			0.3		
Performance Range		Pr^L	$p(Pr)$	Pr^U	Pr^L	$p(Pr)$	Pr^U	Pr^L	$p(Pr)$	Pr^U	Pr^L	$p(Pr)$	Pr^U	Pr^L	$p(Pr)$	Pr^U
Bids Performance: SHS	$B_{2,1}$	240	$N(280,20^2)$	280	550	$N(630,40^2)$	710	-14	$N(0,10^2)$	14	-2	$N(0,1.5^2)$	2	1500	$N(1900,200^2)$	2300
Bids Performance: GMAW	$B_{2,2}$	220	$N(250,15^2)$	280	520	$N(600,40^2)$	680	-14	$N(0,7^2)$	14	-2	$N(1,1^2)$	2	1100	$N(1200,50^2)$	1300
Bids Performance: GTAW	$B_{2,3}$	220	$N(250,15^2)$	280	520	$N(600,40^2)$	680	-14	$N(0,6^2)$	14	-2	$N(0,0.8^2)$	2	1400	$N(1500,50^2)$	1600
Bids Performance: Stir Welding	$B_{2,4}$	320	$N(350,15^2)$	380	600	$N(700,50^2)$	800	-14	$N(0,4^2)$	14	-2	$N(0,0.5^2)$	2	1300	$N(1400,50^2)$	1500
Evaluation Results: SHS	$B_{2,1}$	0.2481			0.3791			0.4874			0.4753			0.2603		
Evaluation Results: GMAW	$B_{2,2}$	0.1566			0.3063			0.5834			0.5834			0.7564		
Evaluation Results: GTAW	$B_{2,3}$	0.1566			0.3063			0.6260			0.6447			0.5		
Evaluation Results: Stir Welding	$B_{2,4}$	0.4307			0.5565			0.7282			0.7570			0.5757		
Normalized DoS					$UDoS(B_{2,1})$						0.3411					
					$UDoS(B_{2,2})$						0.5048					
					$UDoS(B_{2,3})$						0.4425					
					$UDoS(B_{2,4})$						0.5901					

5.6.2 Business Performance Evaluation through Decision Tree Learning

In this chapter, business performance evaluation is done based on the learning of history design agent performance with the rank entropy-based decision tree. Due to the lack of ordinal monotonic classification benchmark data sets for machine learning, to show the effectiveness of the decision tree-based business performance evaluation, both an artificial data set of agent business performance and a real-world regression data set are used, where the latter is further transformed into a monotone one.

The history evaluation is collected as a rated class which results from ordinal measures in four perspectives: support responsiveness during the design, delivery punctuality, design concept reliability, and customer service after delivery. Because of the implicitness of these criteria, evaluation of each single criterion is done by assigning an integer value from 0 to 3, where a higher value represents higher satisfaction. The rate of the overall business performance is based on evaluation of the four criteria, which has four levels: "Wonderful", "Good", "Ordinary", and "Bad". The evaluation process should strictly satisfy the monotonic constraint, meaning one object will not be rated as a lower class than the other object who has equal or worse evaluation results on four criteria. The evaluation process from four attributes to the final rate is to be learned by the rank entropy-based decision tree.

Considering the attribute space only contains 4^4 unique instances, pruning operations can do more harm than good, because the model will not be overfitted if the attribute universal space is not far larger than the space provided by the training dataset. Therefore, in this illustrative case, the decision tree is not pruned after training.

Based on the above formulation, the training dataset is displayed in Table 5-3. The dataset is created with 60 samples by an algorithm that generates unstructured monotone ordinal data (Potharst et al., 2009).

Table 5-3 The monotone dataset for business performance evaluation

Wonderful	0332, 1213, 2133, 2222, 2233, 2330, 2333, 3203, 3213, 3231, 3232, 3311, 3313, 3321, 3322, 3323
Good	0213, 0230, 0320, 1132, 1303, 1311, 2122, 2131, 2203,2221, 2311, 3122, 3200, 3212, 3221
Ordinary	0111,0202,0212,1102,1121,1131,1211,2013, 2023,2103,2111,2120,2130,3033,3121
Bad	0013,0020,0022,0120,0200,0201,1003, 1011,1210,2011,2020,3001,3012,3100

After the training of the rank entropy-based decision tree, the result is shown in Figure 5-3. To make the figure readable, the evaluation values of four criteria are noted as x_1, x_2, x_3, x_4, and the overall service evaluation is noted as y, where {3,2,1,0} represents $\{wonderful,\ good,\ ordinary,\ bad\}$, respectively. The final decision tree has 31 leaves, corresponding to 31 rules for customer service classification.

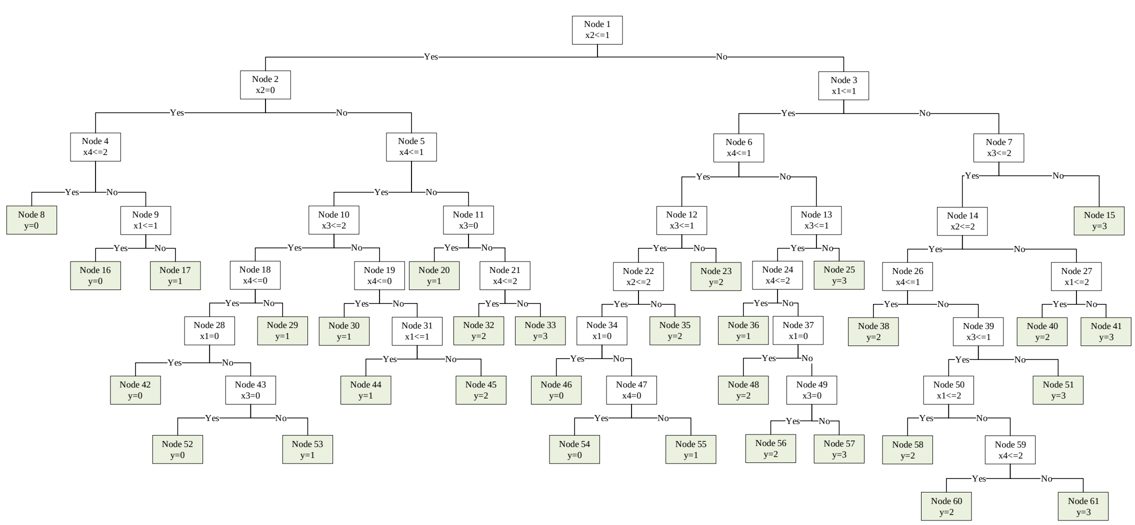


Figure 5-3 Decision tree for customer service evaluation

The existing business performance of the four bidding agents $\mu_1^2, \mu_2^2, \mu_3^2, \mu_4^2$ are displayed as Table 5-4. The results of agent business performance evaluation are then aggregated with engineering functional evaluation for the final agent selection.

Table 5-4 The existing business performance of four agents

Manufacturing agent	Responsiveness	Punctuality	Reliability	Customer Service	Rate
μ_1^2	0	2	0	3	Good
μ_2^2	1	3	3	2	Wonderful
μ_3^2	2	1	3	2	Good
μ_4^2	2	0	3	1	Bad

The validation of the algorithm on the real-world data set is constructed from a computer hardware data set (Frank, 1987). This data set contains 209 records, and each record contains 6 integer attributes and 2 integer responses. In this study, the 6 attributes and the estimated relative performance are used to construct the monotone data set.

Firstly, to make the data set monotone, MYCT (machine cycle time) is transformed as equation (5.19), since this attribute originally has a negative correlation with the response. Secondly, the response is discretized into 6 classes through k-means clustering as the evaluation result. After process, there are 190 unique records, and the discretized ranking results are shown in Table 5-5.

To train the monotone decision tree, the data set is split into a training data set consisting of 141 records and a validation data set with 49 records. The trained decision tree is illustrated in Figure 5-4. And the accuracy on the validation data set is 93.88%.

Table 5-5 The discretized ranking results

Ranking	1	2	3	4	5	6
Range of Estimated Relative Performance	$[15, 50]$	$[52, 107]$	$[113,199]$	$[220,290]$	$[341,603]$	$[749,1238]$
Number of Records	107	38	24	9	8	4

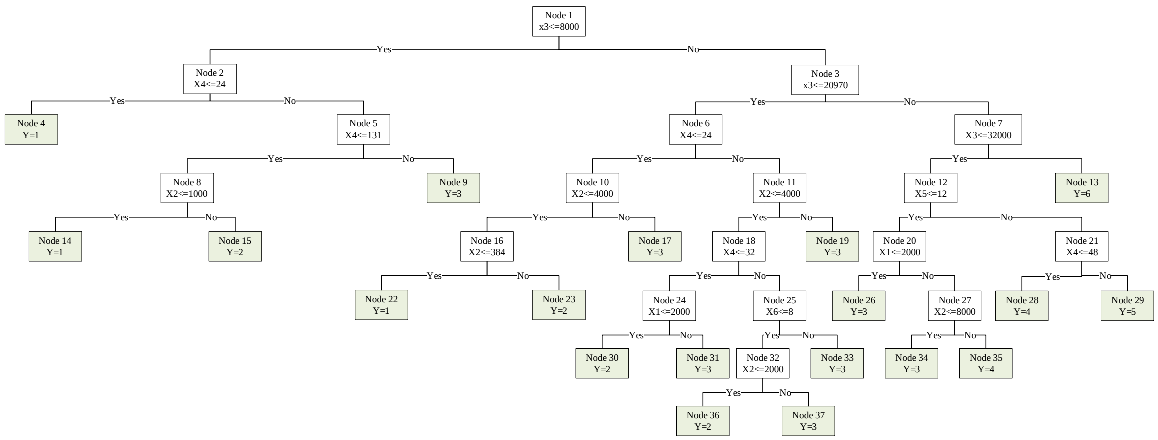


Figure 5-4 Trained decision tree for validation

5.6.3 Evaluation Aggregation of Engineering and Business Performance

After obtaining the engineering and business evaluation, they will be aggregated through the multi-attribute utility theory, as described in Section 5.5. In this illustrative case, DoS for four different business rates “Wonderful”, “Good”, “Ordinary”, and “Bad” are given 1.0, 0.8, 0.6, 0.3 separately. The weight for engineering performance evaluation w_{eng} and for business performance evaluation w_{bus} are given 0.5 and 0.2 separately.

Considering $K = 3$, the final evaluation process is shown in Table 5-6. After the evaluation, μ_2^2 shows superiority and $B_{2,2}$ is selected to be B_2^*. Thus, the corresponding bidding agent μ_2^2 will be evaluated as μ^{2^*} and awarded by the supply contract S. This evaluation and contracting results will be sent to the crowdsourcing information management field and saved as process specs P^0.

Table 5-6 The final evaluation result

Final aggregation result	μ_1^2	μ_2^2	μ_3^2	μ_4^2
$UDoS(\mu^\alpha)$	0.4124	0.6038	0.4875	0.4082

5.6.4 Managerial Implications

The presenting case study of tank trailer manufacturing service process shows an example of the implementation of proposed crowdsourcing workflow. It has examined the feasibility of the platform-driven crowdsourced manufacturing for MaaS. Following the method of tournament-based crowdsourcing, this case study shows the process of reaching external manufacturer crowds via a platform and instantiates the tournament to search the best-performed manufacturing solution. In addition, the proposed evaluation mechanism provides an approach to constructing the collaborative-crowdsourcing team. The contracting evaluation mechanism has taken into account the uncertainty of the manufacturing process, the aggregation of the various criteria from engineering functional requirements and business operational requirements.

From the managerial perspective, the trailer company can focus on its core competitiveness by applying crowdsourced manufacturing with platform. Meanwhile, the main frame company can manufacture the part of trailer assemblies without direct contact with customers. The platform-driven crowdsourced manufacturing offers a bridge to link the manufacturing solution provider crowds for sharing their core competitiveness and capabilities complimentarily. Therefore, they have higher possibility to achieve economies of scale and ease the application of the emerging technologies.

5.7 Chapter Summary

Crowdsourcing contracting entails a BMP problem, for which a best-matching protocol plays a critical role in evaluating and selecting appropriate design or manufacturing agents for crowdsourcing. The information content measure performs as a neutral indicator for unifying diverse domain-specific metrics of manufacturing capabilities within a coherent evaluation framework. Decision tree learning facilitates selection decision making by incorporating past performance data of crowdsourcing agents. This data-driven method suggests good opportunities to extend crowdsourcing contracting to considering a large smart service network over time across a planning horizon.

The evaluation mechanism is the backbone of the contracting mechanism since it serves the selection of the best solution. It will not only determine the customer's satisfaction of final products, but also the coordination of the crowds' cohort decision making. An information contents measurement-based engineering functional evaluation mechanism is proposed. Such evaluation implies the satisfaction is determined by the overlap of the expectation and performance. This two-folded evaluation scheme handles the uncertainty originating from the design and manufacturing domains. The multi-attribute utility theory supports the aggregation of the evaluation. The various criteria can be applied to the evaluation of a solution. This scheme serves the simplification of the evaluation process and allows the scaling of the tournament.

Decision tree-based evaluation mechanism is proposed to explore the inexplicit evaluation of the candidate manufacturer, which highlights the potential of incorporating the service level based on the history records. Monotonic classification rules are proposed for regulating the dominance relationship among the manufacturer in crowds. A synergy of rank entropy-based decision tree learning, and monotone decision tree pruning can extract the business operational ranking of the manufacturers with a minimum overfitting from training dataset. The aggregation method of engineering and business evaluation results are proposed for final selection.

As a technical solution that serves open innovator a contracting evaluation mechanism, this chapter combines quantitative analysis of a solution's performance on functional engineering requirements and the qualitative analysis of the business operation requirements. By aggregating evaluation results through multi-attribute theory, the open innovator can ensure that the selected solution is not only a functional satisfied approach, but also is fulfilled through a reputational solution provider.

CHAPTER 6. CROWDSOURCING TASK DERIVATION AND DECOMPOSITION THROUGH GAME THEORETICAL DECISION-MAKING: A BILEVEL JOINT OPTIMIZATION MODEL FOR EQUILIBRIUM SOLUTIONS

The conversions of manufacturing functional areas towards services imply a transformation of product fulfillment process to a distributed one via a service-oriented cyber platform. As multiple value chains are executed, the volatility and complexities of the customer needs are increasing, resulting in a high production variety and risk in the open manufacturing domain (Gupta et al., 2000). Product differentiation in crowdsourced manufacturing can be achieved by integrating external partners from a platform-driven perspective. The crowdsourcing supply contracting implies a collaborative product fulfillment by a co-creation process of decision agents along the value chain based on the crowdsourcing task allocation and derivation (Shen et al., 2019). Successful product fulfillment operations planning must be coordinated with the product family planning (PFP) at the frontend of open innovation domain. These changes challenge the traditional PFP decision-making considering its manufacturer loading balancing (MLB) problem.

This chapter proposes a leader-follower interactive decision-making mechanism for crowdsourced manufacturing of PFP and MLB based on Stackelberg game. A bilevel optimization model with linear physical programming is developed and solved, comprising an upper-level PFP optimization problem and a lower-level MLB optimization problem. The upper-level PFP determines the optimal configuration of product variants with the objective of maximizing the market share and the total profit in the product family. The lower-level MLB seeks for the optimal partition of manufacturing processes among manufacturer clusters in order to minimizing the operation cost of product variants and balancing the loads among manufacturers. A case study of is reported to demonstrate the feasibility and potential of the proposed bilevel interactive optimization approach.

6.1 Crowdsourced Manufacturing Task Derivation and Decomposition

Owing to the ability to fulfill diversified customer needs with high resources utilization efficiency, the platform-driven strategy explores the common modules among the products and processes to enable mass customization (Park and Simpson, 2008). The instantiation of the platform-driven strategy is product family design and development, which involves multiple domains such as marketing, engineering, and supply chains, and there are specific decision-making problems in each domain (Pirmoradi et al., 2014). PFP is at the stage of product definition in the front-end open innovation domain, which determines product variants with their configuration in the product family according to the customer needs (Jiao and Zhang, 2005). Since it will be the input of the crowdsourcing task decomposition to generate tasks in crowdsourced manufacturing platform domain, the front-end PFP decision-making result will bring an inevitable impact.

Successful crowdsourced manufacturing implementation must include a coordinated decision-making process between PFP and MLB. Some key technical challenges should be addressed to achieve a systematic planning:

1) Interactive product fulfillment. Future manufacturing is equipped with ubiquitous connectivity in the manufacturing environment, allowing collection of significant volumes of dispersed information to support distributed decision making to fulfill manufacturing

tasks (Monostori et al., 2016). New open structure of cyber platform will create opportunities for transforming and expanding the manufacturing sector by developing intelligent cognitive assistants to perform as decision support systems to facilitate the crowdsourcing product fulfillment (Li et al., 2018b). Thus, multiple agents will be involved in PFP for crowdsourced manufacturing. It is necessary to develop decision-making approaches for this kind of multi-agent online interactive product fulfillment scheme.

2) Conflicting objectives. In a distributed yet collaborative product fulfillment process, PFP and MLB have different decision objectives (Du et al., 2019b, Medeiros et al., 2020). For example, the decision objectives of PFP are usually maximizing the customer perceived utility, maximizing the market share, minimizing the product development time, and so on (Kwong et al., 2010). The decision objectives of MLB for crowdsourcing task decomposition are usually minimizing the total manufacturing costs, minimizing the load indices, maximizing the relevance of manufacturing tasks, to name by a few (Kusiak, 2019). These decision objectives are interrelated, mutually restrictive and even conflicting with each other. Therefore, it is necessary to establish an effective decision-making mechanism to analyze and coordinate the interests of PFP and MLB decision-makings.

3) Goal preferences. As previously described, PFP and MLB for crowdsourcing task decomposition are essentially multi-objective optimization problems respectively. In the traditional weight-based techniques for this kind of multi-objective optimization problem, the process of determining appropriate weights or priorities is uncertain and time-consuming, and thus the practicality of these approaches is damaged (Hernandez et al., 2002). In addition, the decision-makers can not represent their preferences on each goal

using more physically meaningful preference ranges in these approaches (Ilgin et al., 2017). Thus, it needs to adopt a more flexible approach with physically meaningful formulation of targets for eliminating PFP and MLB decision-makers from subjective weight setting process.

In this regard, this chapter formulates MLB for crowdsourcing task decomposition as a distributed networked MLB problem. A leader-follower interactive decision-making mechanism for distributed collaborative design of PFP and MLB in crowdsourced manufacturing is proposed. A bilevel optimization model with linear physical programming is developed and solved, in which PFP plays as a leader and MLB acts as a follower.

6.2 Bilevel Programming for Product Design and Development

Conjoint analysis, as a mainstream customer choice simulation technology, has been widely used for predicting customer preferences, and a large number of optimization models and intelligent algorithms have been developed for characterizing and solving the PFP problem (Pirmoradi et al., 2014). In the process of PFP, it is necessary to consider the influence from the manufacturing system factors synchronously (Michalek et al., 2011, Xiao et al., 2018). However, although the above streams of research consider the impact of manufacturing factors on PFP, the corresponding design decision-makings of manufacturing systems are not involved.

Xu and Liang (2006) proposed an integrated approach to plan product module selection and assembly line design with the objective of minimizing the total cost including quality loss, the assembly line reconfiguration and material cost, and the assembly operation related cost. Xu and Liang (2006) also established a multi-objective model to deal with this problem and solved it by adopting the modified Chebyshev goal programming. Objectives of their model are to minimize the total costs, minimize the product performance index, and minimize the assembly line smoothness index. Bryan et al. (2007) considered the concurrent design of product portfolio planning and mixed product assembly line balancing to develop a multi-objective model for minimizing the oversupply optional modules and maximizing the assembly line efficiency. However, the market demands of product variants are determined before optimization in all the above three models, and thus the effect of consumer preferences and purchase behaviors in marketing are not considered. Bryan et al. (2013) formulated a mixed integer non-linear programming model for the product family design with reconfigurable assembly systems considerations. Bryan et al. (2007b) further proposed co-evolution of product families and assembly systems over generations, and introduced a two-phase method based on the model developed by Bryan et al. (2013) for evaluating the co-evolution effectiveness. Deterministic choice rule was employed to simulate the consumer purchase behavior in the above two models. Since this rule is based on the assumption that each consumer will select the product that provides his or her maximum utility surplus, it will overestimate the market share for the most attractive product and underestimate it for other products (Cao et al., 2012). Hanafy and ElMaraghy (2017) formulated a mixed integer programming model for integrating assembly line planning with modular product platform configuration. Abbas and ElMaraghy (2018) introduced an integrated methodology for synthesizing assembly systems for customized products by co-platforming of products and assembly systems. However, all the above research are under the traditional integrated product fulfillment, in which a manufacturer implements a series of activities to develop the product and meet the customer needs. In addition, the constraint satisfaction approach can be used to coordinate the decisions across the product, process, and the supply chain to derive an effective manufacturer load planning result (Jiao et al., 2009). The cloud-based cyber platform synergizes the product and process information and enables the interactive optimization of product design and process setup (Fatahi Valilai and Houshmand, 2014).

Collaborative design decision-making for different decision-making problems across different domains in product engineering has attracted more and more attentions in recent years (Du et al., 2019a). These engineering decision-making issues using Stackelberg game include joint design of technical system modularity and material reuse modularity (Ji et al., 2013), joint optimization of product family modularity and material reuse modularity (Ma et al., 2016), joint optimization of product family module configuration and scaling design (Yu et al., 2016), coordinated configuration of product families and supply chains (Yang et al., 2015, Wang et al., 2016, Pakseresht et al., 2020), joint design of product portfolio planning and viral marketing (Zhou et al., 2015), coordinated configuration of service and product modules in the product-service systems (Li et al., 2015), coordinated optimization of product line planning and product platform configuration (Miao et al., 2017), collaborative design of modular product platforming and supply chain postponement (Xiong et al., 2018), and etc. All these joint decision-makings are dealt with by bilevel optimization based on the Stackelberg game theory from the perspective of distributed collaborative design. In addition, Liu (2016) developed game theoretic optimization models and algorithms for high variety assembly system design. Du et al. (2019b) reviewed this kind of leader-follower joint optimization problems and models for product design and development.

6.3 Coordinated Bilevel Optimal Decision-Making in Crowdsourced Manufacturing

A motivating example of tank trailer product family is considered to illustrate the problem setting. Considering dynamic market demands and short product lifecycles, tank trailer company plans to provide products and services for customers by adopting crowdsourced manufacturing. Tank trailer company has been connected to a crowdsourced manufacturing platform, which provides a cyber platform to small and medium-sized automotive and manufacturing factories in customized trailer industry. Following a crowdsourced manufacturing workflow, a wide spectrum of manufacturing services is connected and aggregated through cyber platform. Thanks to the adoption of platform-strategy in crowdsourced manufacturing, volatile product fulfillment demands can be accessed and the similarity among them can be explored. The platform can assign similar tasks to a manufacturer, which can allow the manufacturer achieving a maximized reusability of the related resources. Thus, with the expansion of the customer clusters, the manufacturers can focus on their core competitive edges and achieve economies of scale. Assume a tank trailer company plans to develop a family of custom trailer to meet customer needs in different market segments. Tank trailers can be considered as modular products, and each module required in the product family can be designed and manufactured by manufacturer crowds through the service-oriented manufacturing platform.

6.3.1 Crowdsourcing Task Derivation

The decision-making of PFP and MLB crowdsourcing task decomposition for tank trailer company is shown in Figure 6-1.

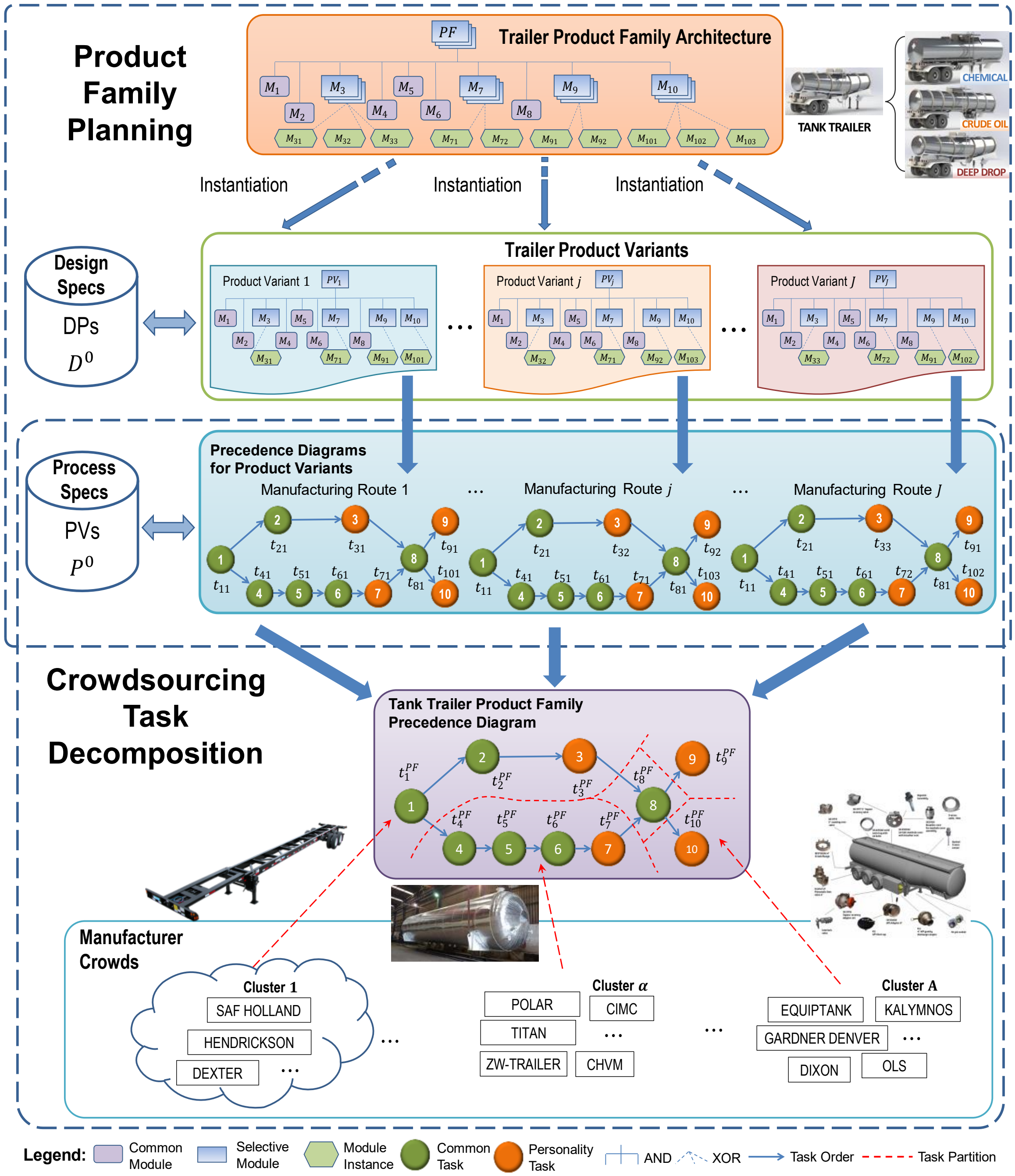


Figure 6-1 Operation planning in crowdsourced manufacturing

The first layer is the developed WS product family modular architecture, which contains six common modules, i.e., axle assembly (M1), tire (M2), steel sheet (M4), hose tube (M5), anti-flow sets (M6), landing gear and pin assembly (M8), and four selective modules, i.e., main frame (M3), insulation layer (M7), warning placard (M9), valve assembly (M10). The number of module instances for selective modules M3, M7, M9, and M10 are 3, 2, 2, and 3, respectively. For example, there are three module instances for M10, i.e., safety valve, emergency & flush valve, and API valve. A total number of J different product variants are configured in the process of PFP based on the trailer product family architecture, as shown in the second layer. For each product variant in the product family, one precedence diagram of the manufacturing rout is determined, as shown in the third layer. The node represents a manufacturing task that joints one model to the previously completed sub-manufacturing, the number outside the node is the manufacturing time for the corresponding task, and the arc indicates the precedence order. The fourth layer obtains the product family precedence diagram and the manufacturing task assignment. The last layer shows manufacturer clusters linked to the crowdsourced manufacturing platform, and each manufacturer cluster includes a few related manufacturing agents.

6.3.2 Coordinated Bilevel Optimal Decision-Making

A bilevel taxonomy of the problem of coordinated decision-making of PFP and MLB for crowdsourced manufacturing is shown in Figure 6-2. It can be described as follows: the company has established a modular product family architecture comprising several common modules and selective modules as a product development platform. Based on that platform, it plans to design a family of new product variants by combining different module instances of selective modules to satisfy diversified needs of customers.

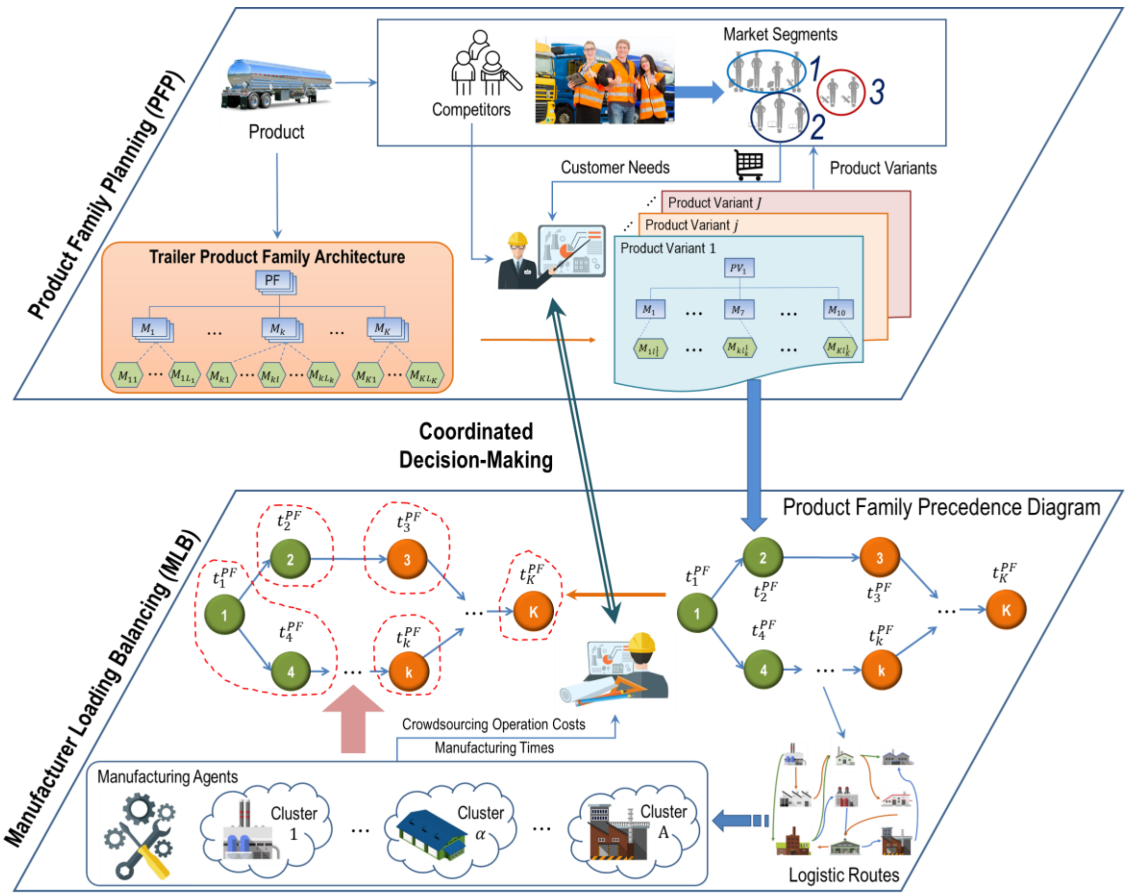


Figure 6-2 Coordinated bilevel optimal decision-making in crowdsourced manufacturing

The manufacturing services of these products are provided by the external suppliers, manufacturing agent crowds, through certain cyber service-oriented manufacturing platform. Given manufacturing capability requirement for all modules, the precedence diagrams for product variants in the product family can be derived to represent the order of task completion. The goal of the coordinated decision-making problem is to simultaneously determine the optimal configuration of each product variant in the product family, obtain the product family precedence diagram, as well as partition the manufacturing tasks among manufacturers according to competitive products, customer needs, crowdsourcing operation costs and manufacturing times with the objective of maximizing the market share and the total profit of the planned product family.

6.3.3 Solution algorithms

Since crowdsourced manufacturing fulfills products through collaboration of crowds, the traditional product fulfillment process is distributed based on collaborative negotiation (Gong et al., 2021). Following this scenario, the task derivation can be summarized as a two-folded decision-making process. Firstly, the platform predicts manufacturing capabilities, which formulates the portfolio planning problem as a combinatorial optimization problem based on the contract pricing, sharable time window, as well as the logistic cost of the manufacturers. This plan outlines the scale and combination of the manufacturers to accumulate sufficient production capability for various supply chains. On the other hand, it also serves the platform with a way of specifying a set of crowdsourcing tasks to minimize the resources idling and inventory. This two-folded service process requires the decision-making solution provides a systematic approach to solve the trade-off between the supply configuration to achieve global satisfaction and the specification of crowdsourcing tasks for local efficiency. Because a crowd of companies is required in crowdsourced manufacturing, the optimal configuration problem shows an interactive decision among the decision agents (Wu et al., 2021). As the crowdsourcing workflow described in chapter 3, the task allocation through a negotiation process can be decomposed into the requesting for quotation, propose of bids, as well as the awarding with supply contracts. This process entails an iteratively multi-level decision-making process, where the manufacturers respond with their manufacturing plan according to the limits of their shop floor and the platform responds with crowdsourcing task allocation to seek an optimal production portfolio plan.

Figure 6-3 illustrates the leader-follower interactive decision-making mechanism for joint design of PFP and MLB for crowdsourced manufacturing. The PFP design decision-maker plays a leader's role and handles the upper-level decision-making problem, which can be formulated as PFP optimization. The goal of the upper-level problem is to optimize the selection and configuration of product variants for maximizing the market share and total profit of the product family, in which considerations about market segments, customer preferences, and crowdsourcing operation costs should be incorporated. The MLB design decision-maker acts as a follower and deals with the lower-level decision-making problem, which can be formulated as MLB design optimization. After obtaining the PFP decisions derived from the upper-level optimization, the lower-level MLB aims to partition the manufacturing tasks among manufacturers for minimizing the crowdsourcing operation costs and the load index. During the formulation of the lower-level problem, the main influencing factors are operation costs of manufacturer crowds and manufacturing times of module instances. The total crowdsourcing operation cost obtained in the lower-level optimization will be fed back to the upper-level, and then the leader will adjust the PFP decisions for maximizing his own interest according to these cost figures. This distributed bilevel collaborative optimization of PFP and MLB proceeds in an interactive manner until the leader-follower equilibrium solution is achieved based on Stackelberg game. In addition, instead of assigning subjective weights, the preferences on each goal using physically meaningful preference ranges through linear physical programming (LPP) should be considered for both the PFP and MLB design decision-makers.

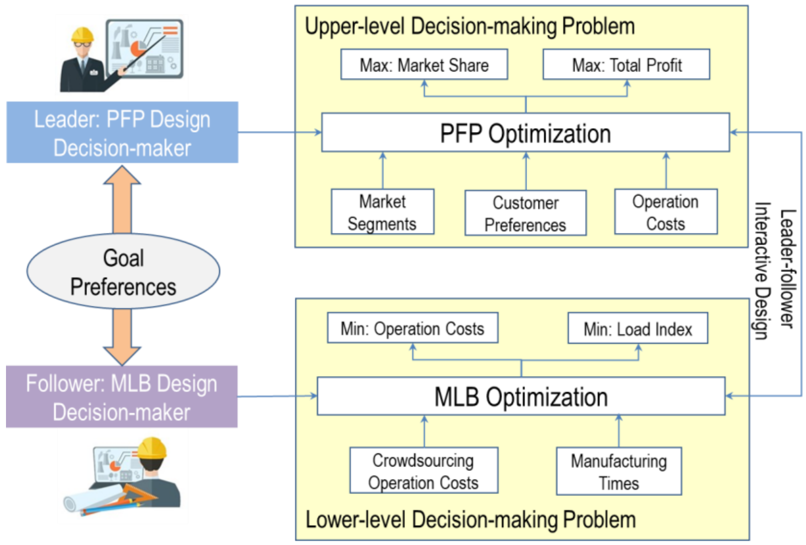


Figure 6-3 Leader-follower interactive decision-making mechanism

6.4 Joint Optimization of PFP and MLB

In this section, the distributed collaborative decision-making of PFP and MLB for crowdsourced manufacturing is formulated into a bilevel optimization model with LPP. Assumptions and nomenclature of this joint decision-making problem are given in Section 6.4.1. The upper-level PFP decision-making is modeled in Section 6.4.2, and the lower-level MLB design decision-making is modeled in Section 6.4.3. Section 6.4.4 lists other necessary constraints. Finally, the bilevel optimization model with LPP is presented in Section 6.4.5.

6.4.1 Model Assumption and Nomenclature

The basic assumptions in this research include:

1) One common module can be viewed as a selective module that has only one module instance, and a null module instance represents the absence of the corresponding module from one product variant.

2) All the module instances are technological feasible, and there is no compatible restriction on the combinations of these module instances.

3) The manufacturing of all the product variants in the product family are provided through the service-oriented manufacturing platform linked with various service pools, and each service pool consists of a few related candidate manufacturers.

4) Only the operation costs for manufacturers are considered in variable costs in this research, and the unit operation costs for different manufacturers are the same.

6.4.2 Upper-Level PFP Model

The upper-level PFP design decision-making aims to select the optimal product portfolio and determine the optimal configuration of each product variant with the objective of maximizing the market share and the total profit in the product family.

Suppose that the product market has been divided into I market segments through conducting a market survey and adopting a proper clustering technology, and consumers in each market segment have the same purchase preference. Consumers make purchase decisions based on the perceived utility surplus obtained from the corresponding product. Following the commonly used linear-additive part-worth utility model in conjoint analysis, the utility U_{ij} that one consumer in the i-th market segment can obtain by choosing the j-th product variant can be formulated as:

$$U_{ij} = \sum_{k=1}^{K} \sum_{l=1}^{L_k} (u_{ikl} - r_{kl}) x_{jkl}, \quad i = 1, 2, \ldots, I, \;\; j = 1, 2, \ldots, J \tag{6.1}$$

where u_{ikl} is the part-worth utility of the l-th module instance of the k-th module for the i-th market segment, r_{kl} is the cost for buying one unit of the l-th module instance of the

k-th module, and x_{jkl} is the decision variable which indicates whether (1) or not (0) the l-th module instance of the k-th module is assigned to the j-th product variant.

According to the multinomial logit (MNL) choice model for product family positioning in marketing, the probability P_{ij} that one consumer in the i-th market segment purchases the j-th product variant in the product family can be formulated as:

$$P_{ij} = \frac{exp\ (\theta U_{ij})}{\sum_{j=1}^{J} y_j exp\ (\theta U_{ij}) + \sum_{j=1}^{N^C} exp\ (\theta U_{ij}^C)}, \qquad i = 1, 2, \dots, I, \ \ j = 1, 2, \dots, J \qquad (6.2)$$

where θ is a positive scaling parameter of the MNL model. If θ goes to infinity, the MNL behaves like a deterministic model; and if θ approaches zero, it becomes a uniform distribution (Steiner and Hruschka, 2002).

The total market revenue R of the product family can be computed by multiplying the revenue of each product variant by the demand for this product variant firstly, and then summing all the product variant revenue, which can be formulated as:

$$R = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{l=1}^{L_k} Q_i P_{ij} r_{kl} x_{jkl} y_j\ , \qquad (6.3)$$

where y_j is the decision variable which indicates whether (1) or not (0) the j-th product variant is selected in the planned product family.

One objective of the upper-level PFP optimization problem is to maximize the total profit T of the product family, which is the difference between the total market revenue R and the total crowdsourcing operation cost OC of the product family, i.e.,

$$TP = R - OC = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L_k} Q_i P_{ij} r_{kl} x_{jkl} y_j - OC. \tag{6.4}$$

The other objective is to maximize the market share MS of the product family, which can be formulated as

$$MS = \frac{1}{\sum_{i=1}^{I} Q_i}\sum_{i=1}^{I}\sum_{j=1}^{J} Q_i P_{ij} y_j\,. \tag{6.5}$$

6.4.3 Lower-Level MLB Model

When the upper-level PFP decisions are determined, the lower-level MLB optimization decision-making seeks for the optimal partition solution of the manufacturing tasks among manufacturers in order to minimizing the crowdsourcing operation cost of product variants and balancing the loads among manufacturers.

Following the approach based on fixed costs and variable costs, the overall crowdsourcing operation costs OC of the product family can be formulated as:

$$OC = \sum_{\pi=1}^{\Pi}\left(C^{fix} + C^{var} T^{O}\right) w_{\pi} \tag{6.6}$$

where C^{fix} is the fixed cost for each MaaS service provider, and C^{var} is the variable operation cost for each manufacturer per unit time.

The load of the whole manufacturing process should be balanced, that is to say, the total manufacturing time allocated to each manufacturer should be as equal as possible. The load index among manufacturers LI can be defined by the standard deviation of manufacturing loads, i.e.,

$$LI = \left[\frac{1}{\Pi - 1}\sum_{\pi=1}^{\Pi}\left(\sum_{k=1}^{K} t_k^{PF} z_{k\pi} - \frac{T^O}{\sum_{i=1}^{I}\sum_{j=1}^{J} Q_i P_{ij}} w_\pi\right)^2\right]^{\frac{1}{2}} \tag{6.7}$$

where T^O is the planned life of the MaaS operations, t_k^{PF} is the product family manufacturing time for the k-th module, which can be computed as the weighted sum of the manufacturing task times for each product variant in the product family, i.e.,

$$t_k^{PF} = \sum_{j=1}^{J}\sum_{l=1}^{L_k} t_{kl} x_{jkl} \frac{\sum_{i=1}^{I} Q_i P_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} Q_i P_{ij}}, \quad k = 1, 2, \dots, K \tag{6.8}$$

6.4.4 Constraint Modelling

To establish the bilevel optimization model, some additional constraints about relationships among decision variables are required to be analyzed and formulated as below.

1) For each module of one product variant, exactly one and only one module instance can be selected. The exclusiveness conditions can be described as

$$\sum_{l=1}^{L_k} x_{jkl} = 1, \quad j = 1, 2, \dots, J, \;\; k = 1, 2, \dots, K \tag{6.9}$$

2) Since tasks are indivisibility work elements, each task is assigned to exactly one manufacturer. The occurrence constraints can be described as

$$\sum_{\pi=1}^{\Pi} z_{k\pi} = 1, \quad k = 1, 2, \dots, K \tag{6.10}$$

3) For each manufacturer, the total manufacturing time for tasks assigned to this manufacturer does not exceed the time available at this manufacturer. The time constraints can be described as

$$\sum_{k=1}^{K} t_k^{PF} z_{k\pi} \leq \frac{T^O}{\sum_{i=1}^{I}\sum_{j=1}^{J} Q_i P_{ij}} w_\pi, \quad \pi = 1, 2, \dots, \Pi \tag{6.11}$$

Thus, by equation (6.8), the time constraints can be rewritten as

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L_k} t_{kl} x_{jkl} Q_i P_{ij} z_{k\pi} \leq T^O \cdot w_\pi, \quad \pi = 1, 2, \dots, \Pi \tag{6.12}$$

4) Owing to technological and organizational conditions, the tasks must be assigned to manufacturers according to the precedence graph, i.e., the resulting sequence of manufacturing tasks cannot violate the precedence constraints among these tasks. The precedence constraints can be described as:

$$x_{k\pi} \leq \sum_{n=1}^{\pi} x_{hn}, \quad k = 1, 2, \dots, K, \quad \pi = 1, 2, \dots, \Pi, \quad h \in Pre(k), \tag{6.13}$$

where $Pre(k)$ is the set of all direct and indirect predecessors of the k-th manufacturing task.

6.4.5 Bilevel Optimization Model

Based on the objective functions and constraints, the bilevel joint optimization model of PFP and MLB can be formulated as below.

$$\text{Min } Z_1 = \sum_{i=1}^{2}\sum_{k=1}^{4} (w_{ik}^- d_{ik}^- + w_{ik}^+ d_{ik}^+) \tag{6.14.0}$$

$s.t.$

$$\frac{TP - \min(TP - TP \cdot Target_{m+1}, 0)}{TP \cdot Target_m} + d_{1m}^{-} - d_{1m}^{+} = 1 \quad m = 1,2,\dots,4 \tag{6.14.1}$$

$$\frac{MS - \min(MS - MS \cdot Target_{m+1}, 0)}{MS \cdot Target_m} + d_{2m}^{-} - d_{2m}^{+} = 1 \quad m = 1,2,\dots,4 \tag{6.14.2}$$

$$\sum_{l=1}^{L_k} x_{jkl} = 1, \quad j = 1,2,\dots,J, \;\; k = 1,2,\dots,K \tag{6.14.3}$$

$$U_{ij} = \sum_{k=1}^{K}\sum_{l=1}^{L_k}(u_{ikl} - r_{kl})x_{jkl}, \quad i = 1,2,\dots,I, \;\; j = 1,2,\dots,J \tag{6.14.4}$$

$$P_{ij} = \frac{exp\,(\theta U_{ij})}{\sum_{j=1}^{J} y_j exp\,(\theta U_{ij}) + \sum_{j=1}^{N^C} exp\,(\theta U_{ij}^{C})}, \quad i = 1,2,\dots,I, \;\; j = 1,2,\dots,J \tag{6.14.5}$$

$$TP \geq TP \cdot Target_5 \tag{6.14.6}$$

$$MS \geq MS \cdot \mathrm{Target}_5 \tag{6.14.7}$$

$$x_{jkl}, y_j \in \{0,1\} \tag{6.14.8}$$

$$\mathrm{Min} \quad Z_2 = \sum_{i=3}^{4}\sum_{k=1}^{4}(w_{ik}^{-}d_{ik}^{-} + w_{ik}^{+}d_{ik}^{+}) \tag{6.14.9}$$

$s.t.$

$$\frac{OC - \max(OC - OC \cdot \mathrm{Target}_{m+1}, 0)}{OC \cdot \mathrm{Target}_m} + d_{3m}^{-} - d_{3m}^{+} = 1 \quad m = 1,2,\dots,4 \tag{6.14.10}$$

$$\frac{LI - \max(LI - LI \cdot \mathrm{Target}_{m+1}, 0)}{LI \cdot \mathrm{Target}_m} + d_{4m}^{-} - d_{4m}^{+} = 1 \quad m = 1,2,\dots,4 \tag{6.14.11}$$

$$\sum_{\pi=1}^{\Pi} z_{k\pi} = 1, \quad k = 1,2,\dots,K \tag{6.14.12}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L_k} t_{kl}x_{jkl}Q_iP_{ij}z_{k\pi} \leq Tw_\pi, \quad \pi = 1, 2, \dots, \Pi \tag{6.14.13}$$

$$x_{k\pi} \leq \sum_{n=1}^{\pi} x_{hn}, \quad k = 1, 2, \dots, K, \qquad \pi = 1, 2, \dots, \Pi, \qquad h \in Pre(k) \tag{6.14.14}$$

$$OC \leq OC \cdot \mathrm{Target}_5 \tag{6.14.15}$$

$$LI \leq LI \cdot \mathrm{Target}_5 \tag{6.14.16}$$

$$z_{k\pi} \in \{0,1\}, \qquad w_\pi \in \mathbb{N} \tag{6.14.17}$$

LPP is adopted in the upper-level and lower-level optimization, respectively, and it allows the PFP and MLB decision-makers represent their preferences on each goal using physically meaningful preference ranges. LPP is proposed by Messac et al. (1996) as a novel approach to multiple objective optimizations. Application of LPP involves the following four steps:

1) Identify each decision criteria as Class 1S (Smaller is Better), Class 2S (Larger is Better), Class 3S (Value is Better), or Class 4S (Range is Better).

2) Define the desirability ranges for each decision criteria: ideal, desirable, tolerable, undesirable, highly undesirable, and unacceptable.

3) Calculate the values of the weights using the algorithm developed by Messac et al. (1996) or Hernandez et al. (2002).

4) Formulate a common deviation function to evaluate the alternatives.

LPP has been applied in industrial engineering and product engineering (Maria et al., 2003, McAllister et al., 2005, Lai et al., 2006, Kongar and Gupta, 2009, Ilgin et al., 2017, Liu et al., 2018). For example, Ilgin et al. (2017) develop an LPP-based disassembly line balancing method to balance a mixed-model disassembly line. A comprehensive review of different variants and applications of physical programming can be found in Ilgin and Gupta (2012).

The upper-level objective function Z_1 in equation (6.14.0) is a common deviation function, which is formulated as a weighted sum of the deviation variables d_{ik}^{-}, d_{ik}^{+} ($i = 1,2$; $k = 1,2,\dots,4$). These deviation variables can be obtained by Constraints (6.14.1) and (6.14.2) where $TP \cdot \mathrm{Target}_k$ and $MS \cdot \mathrm{Target}_k$ are the physically meaningful target values at the desirability level k for the goal TP and MS respectively ($k = 1,2,\dots,5$). These target values are specified by the PFP design decision-maker to quantify the preferences associated with the TP and MS criterions. Different from assigning subjective weights, the weights w_{ik}^{-} and w_{ik}^{+} ($i = 1,2$; $k = 1,2,\dots,4$) can be determined by the algorithm proposed in Hernandez et al. (2002). Similarly, the common deviation function Z_2 in equation (6.14.9) is the lower-level objective function, which is a weighted sum of the deviation variables d_{ik}^{-}, d_{ik}^{+} ($i = 3,4$; $k = 1,2,\dots,4$) derived from Constraints (6.14.10) and (6.14.11). Constraints (6.14.6), (6.14.7), (6.14.15) and (6.14.6) indicate that the fifth level values $TP \cdot \mathrm{Target}_5$, $MS \cdot \mathrm{Target}_5$, $OC \cdot \mathrm{Target}_5$, and $LI \cdot \mathrm{Target}_5$ are unacceptable. The values of decision variables $x_{jkl}, y_j, z_{k\pi}, w_\pi$ are restricted in Constraints (6.14.8) and (6.14.17).

6.5 Nested Bilevel Genetic Algorithms for PFP-MLB Joint Optimization

In this section, a nested bilevel genetic algorithm (NBGA) is developed to find the optimal or near optimal solution of the bilevel optimization model with linear physical programming. The NBGA is a nested sequential approach, in which the PFP and MLB decision-makings are solved by the traditional single-level genetic algorithm (GA) respectively and the lower-level GA is performed for each upper-level feasible solution.

6.5.1 Overview of Nested Bilevel Genetic Algorithms

The flow chart of the NBGA algorithm is shown in Figure 6-4. A step-by-step procedure for the NBGA algorithm can be described as follows:

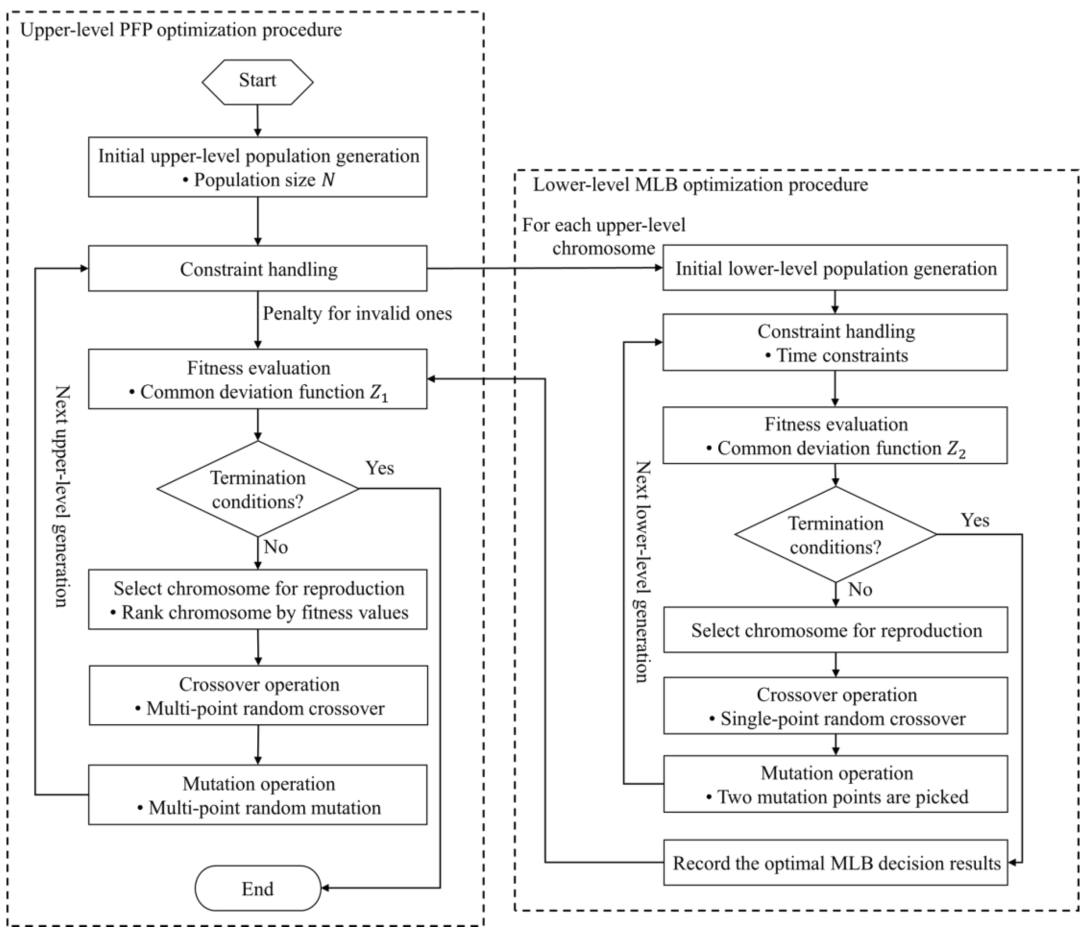


Figure 6-4 Flow chart of NBGA

Step 1: Initialization. An upper-level initial population of size N is randomly generated and constraint handling is carried out to ensure that each generated chromosome satisfy required constraints. Then, for each upper-level chromosome, the corresponding lower-level MLB optimization procedure is executed to obtain the lower-level optimal or near optimal chromosome.

Step 2: Upper-level selection operation. Combine the upper- and lower-chromosomes, and then evaluate each upper-level chromosome by assigning a fitness value based on the common deviation function Z_1 in equation (6.14.0). Choose parent chromosomes for upper-level crossover and mutation operations from the current population by adopting the rank selection method according to fitness values.

Step 3: Upper-level crossover and mutation operations. Offspring chromosomes are created by performing multi-point random crossover and mutation operators, in which single-point random crossover and mutation operators are adopted for each product variant chromosome section.

Step 4: Lower-level optimization. For each upper-level offspring chromosome, the corresponding upper-level decision results are transmitted to the lower-level MLB decision-making problem. After initializing the lower-level population, the fitness value of each chromosome is computed based on the common deviation function Z_2, and a penalizing strategy is adopted for handling those invalid chromosomes that violate the lower-level constraints. Then, the selection, crossover, and mutation operations are carried out successively.

Step 5: Evaluation of offspring chromosomes. Fitness values of upper-level offspring chromosomes are evaluated through combining each upper-level offspring chromosome with its corresponding lower-level chromosome and then computing the upper-level common deviation value.

Step 6: Examination of termination conditions. A maximal number of generations is specified as the criterion for the termination check in both the upper- and lower- genetic algorithm. Proceed to the next generation (Step 2) if the termination check is false.

6.5.2 Encoding and Operators in Upper-Level

To apply GA to the upper-level PFP design decision-making, the integer encoding strategy is adopted for the chromosome structure, as illustrated in Figure 6-5 (a). A chromosome is composed of J product variant sections, and there are K module sub-sections for each product variant section. The value l in the k-th module sub-section in the j-th product variant section represents that the l-th module instance of the k-th selective module is selected for the j-th product variant. Thus, the PFP design decision-making is described by a chromosome with length of KJ. Each chromosome in the upper-level initial population is generated randomly by the approach proposed by Jiao et al. (2007b) to ensure the satisfaction of Constraint (6.14.3). The upper-level fitness function is the common deviation function Z_1 in equation (6.14.0).

A single-point crossover operator is adopted for each product variant substring, and thus the crossover for the upper-level chromosome is carried out with a multi-point crossover operator, as shown in Figure 6-6 (a). Similarly, a single-point mutation operator is employed for each product variant substring, in which one mutation point is picked randomly and then the corresponding module instance is altered at random. Figure 6-6 (b) illustrates the detailed process of multi-point random mutation operators.

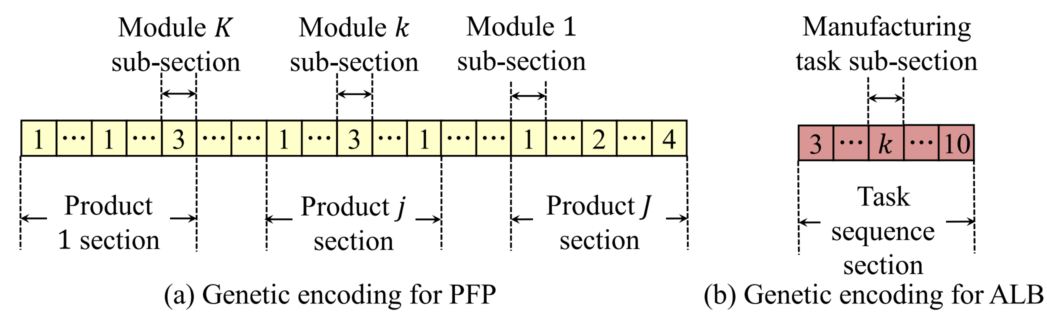


Figure 6-5 Genetic encoding for upper-level PFP and lower-level MLB

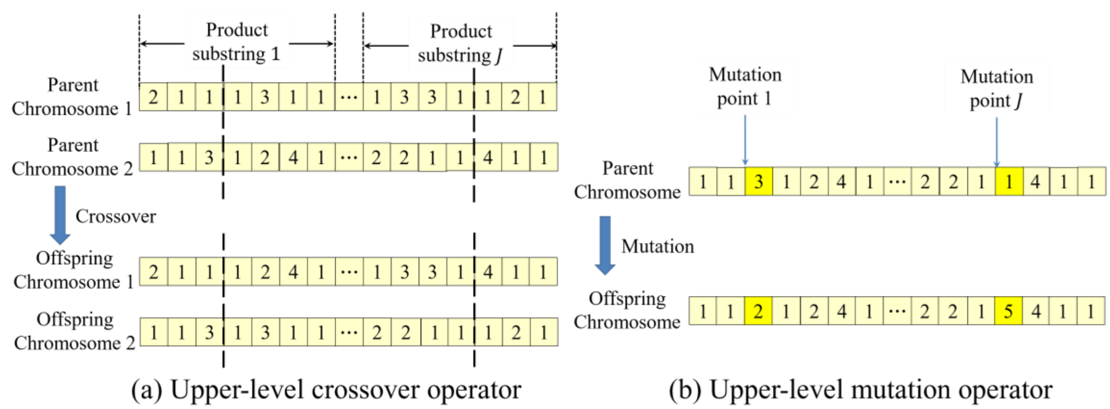


Figure 6-6 Upper-level crossover and mutation operators

6.5.3 Encoding and Operators in Lower-Level

Figure 6-5(b) illustrates the chromosome encoding for the lower-level MLB design decision-making, and there are K assembly task sub-sections for the task sequence section. Each manufacturing task sub-section, i.e., the gene, indicates the manufacturing sequence, and the value in the manufacturing task sub-section, i.e., the allele, represents the corresponding task. In this research, for each lower-level chromosome in the initial population, the manufacturing task substring is generated by the top sort algorithm developed by Hou et al. (2014) to satisfy the precedence constraints, i.e., Constraint

(6.14.14). To ensure the satisfaction of Constraint (6.14.12) and (6.14.13), the algorithm proposed by Leu et al. (1994) is employed for chromosome decoding, i.e., dividing manufacturing tasks into manufacturers. The lower-level fitness function is the common deviation function Z_2 described in equation (6.14.9).

A single-point crossover operator is implemented for the task sequence substring, in which two parent chromosome exchange genetic information after selecting the crossover point randomly. To avoid generating infeasibility offspring chromosomes after crossover, an improved exchange process is adopted for the task sequence substring, as illustrated in Figure 6-7 (a).

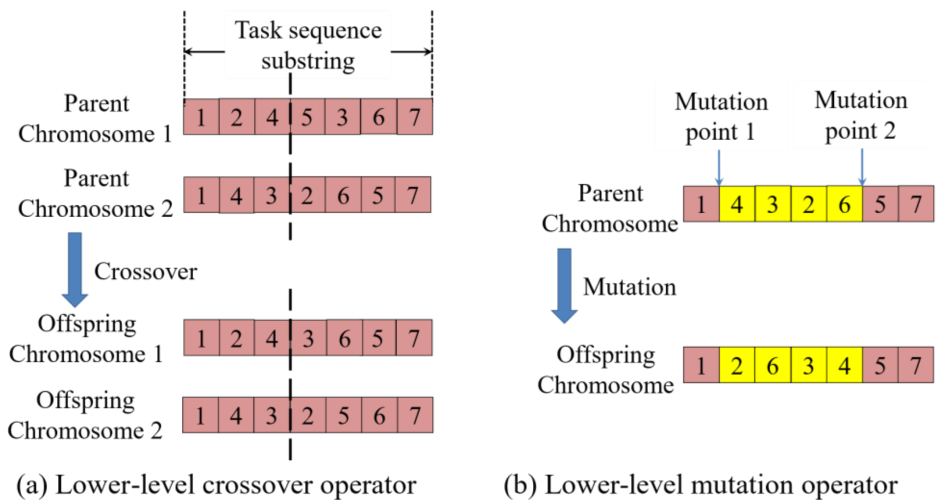


Figure 6-7 Lower-level crossover and mutation operators

It can be observed that the exchanged sequence of the task sequence substring in parent chromosome 1 is 5, 3, 6, and 7, and the ordering of this exchanged sequence in parent chromosome 2 is 3, 6, 5, and 7. Then, generate a new offspring chromosome 1 by replacing the exchanged sequence in parent chromosome 1 with 3, 6, 5, and 7. Finally, the other offspring chromosome 2 can be obtained by using the similar approach, as shown in

Figure 6-7 (a). Figure 6-7 (b) illustrates the lower-level mutation operator. For the task sequence substring, two mutation points are picked randomly, between which the manufacturing tasks are reordered according to the product family precedence graph by the scheme of producing the initial population. Thus, the obtained offspring chromosomes are still feasible after crossover and mutation operations.

6.6 Case Study on Crowdsourcing Task Derivation

6.6.1 Case Description

To illustrate the proposed model and algorithm, the joint PFP and MLB design problem for tank semi-trailer is presented. Suppose that tank trailer company plans to design a family of two trailers, and there are two existing competitive products in the market. The precedence diagram for trailer is shown in Figure 6-8. Suppose that the customized kitchen market has been divided into three segments through market research. The size of each market segment is given in the second row in Table 6-1. Table 6-1 also lists the utility surplus of two existing products for each market segment in the last two rows. The part-worth utilities of module instances for each market segment can be estimated by conjoint analysis or simulation calculation. In this case, the simulated utility data is adopted, and it is generated from a uniform distribution randomly. The estimated manufacturing times are listed in the fifth row in Table 6-2. The estimated purchase costs of module instances are shown in the last row in Table 6-2. The LPP target values for the upper-level PFP goals (TP and MS) and the lower-level MLB goals (OC and LI) are shown in Table 6-3. Applying the algorithm described in Messac et al. (1996), the resulting weights for both the upper- and lower-level common deviation functions are shown in

Table 6-4. The scaling parameter θ in the MNL choice model is set to 0.75. The fixed cost for each manufacturer C^{fix} is set to 500, and the operation cost C^{var} for each MaaS service provider per unit time is set to 0.5. The total planned life T of crowdsourcing operation cost is assumed to be 3000000.

6.6.2 Implementation and Comparison

To solve the bilevel optimization model for joint decision-making of PFP and MLB, the proposed NBGA is conducted. In the upper-level GA, the maximum number of iterations is 500, the crossover probability is 0.8, and the mutation probability is 0.2. In the lower-level GA, the maximum number of iterations is 100, the crossover probability is 0.8, and the mutation probability is 0.2. Based on the above assumptions, the NBGA is run using MATLAB 2017b under the circumstance of Windows 10, Intel i7-7500U 2.90GH and Ram 8G. The running time is 3628.125s.

Figure 6-9 provides the NBGA evolution processes for the upper-level PFP and the lower-level MLB optimizations. It shows the upper-level common deviation function value Z_1 and the lower-level common deviation function value Z_2 for the best individual over generations, which reflects the dynamic interactive decision-making process between the upper- and lower- levels. After 250 iterations, the optimal PFP result and the corresponding MLB result are arrived, which are listed in the third column in Table 6-5.

As shown in Table 6-5, the optimal upper-level chromosome coding scheme is [1 1 2 1 1 1 2 1 2 3 1 1 1 1 1 1 2 1 2 3], the upper-level common deviation function value is 0.5721, the total profit is 1.5600×10^7, and the market share is 92.05%. The corresponding lower-level coding scheme is [1 4 5 6 2 3 7 8 9 10], the lower-level common deviation function value is 0.3820, the operation cost is 1.9507×10^7, the load index is 1.1217. According to the lower-level MLB decision-making result, all the manufacturing tasks are assigned to nine manufacturers, in which task 1 and task 4 are assigned to the same manufacturer. The numbers of MaaS service providers are 2, 2, 2, 2, 1, 2, 3, 1 and 2, respectively.

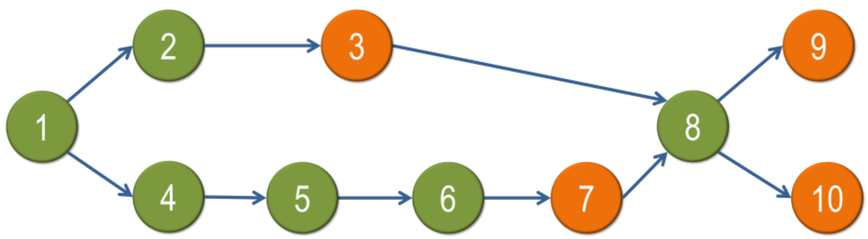


Figure 6-8 Precedence diagram for tank trailer

To verify the validity of the proposed method, two experiments are designed to compare the results of the bilevel approach (TBA) with those of the sequential approach (TSA), i.e., solving the PFP decision-making and the MLB decision-making sequentially in two steps, and the cooperative approach (TCA), i.e., the PFP decision-maker and the MLB decision-maker engage in bargaining and desire a cooperative and binding trade for maximizing their collective interest. For TSA, in the first step, the total operation cost can be estimated based on the historical data, and the PFP problem is solved using the upper-level GA. After obtaining the PFP results in the first step, the MLB problem is solved using the lower-level GA in the second step. In this experiment, the estimated operation cost in the first step is set 2.5×10^7. The optimal PFP and MLB results are listed in the fourth column in Table 6-5.

Table 6-1 Sizes of market segments and utility surplus of existing products

	Segment 1	Segment 2	Segment 3
Estimated number of consumers	250,000	350,000	150,000
Utility surplus of existing product 1	8.7	9.4	8.3
Utility surplus of existing product 2	10.5	8.5	9.6

Table 6-2 Part-worth utilities, manufacturing times and purchase costs for module instances

		M1	M2	M3			M4	M5	M6	M7		M8	M9		M10		
				M31	M32	M33				M71	M72		M91	M92	M101	M102	M103
Part-worth utilities	Segment 1	8.9	9.2	0	6.8	8.3	4.5	6.5	6.5	0	5.9	5.8	0	5.8	0	4.1	6.8
	Segment 2	8.5	9.7	0	6.6	8.9	4.7	6.9	7.2	0	4.8	5.6	0	3.8	0	4.5	6.6
	Segment 3	8.2	9.9	0	6.3	8.5	4.9	6.2	7.9	0	5.2	5.3	0	6.8	0	4.8	6.2
Manufacturing times		0	6	0	5	8	7	5	5	0	8	10	0	4	0	6	9
Module revenues		7.8	8.5	0	5.3	7.6	3.5	5.7	5.2	0	3.9	4.8	0	2.8	0	3.6	4.8

Table 6-3 Sizes of market segments and utility surplus of existing products

	Upper-level PFP decision-maker	
Level k	Total profit $TP \cdot$ Target (10^7)	Market share MSTarget (%)
1	2	90
2	1.5	80
3	1	70
4	0.7	60
5	0.5	50
	Lower-level MLB decision-maker	
Level k	Operation cost $OC \cdot$ Target (10^7)	Load index $LI \cdot$ Target (%)
1	1.5	0.5
2	2	1
3	2.3	1.5
4	2.5	2
5	3	3

Table 6-4 Weights for the goals in the common deviation functions

	Upper-level PFP decision-maker	
Level k	Total profit TP (w_{1k}^{+}/w_{1k}^{-})	Market share MS (w_{2k}^{+}/w_{2k}^{-})
1	0/2.6	0/3
2	0/0.26	0/0.3
3	0/7.15	0/3.63
4	0/21.522	0/7.623
5	0/2.6	0/3
	Lower-level MLB decision-maker	
Level k	Operation cost OC (w_{3k}^{+}/w_{3k}^{-})	Load index LI (w_{4k}^{+}/w_{4k}^{-})
1	1.5	0.5
2	2	1
3	2.3	1.5
4	2.5	2
5	3	3

Table 6-5 Results under the bilevel, sequential, and cooperative approach

		The bilevel approach	The sequential approach	The cooperative approach
Upper-level PFP result configuration	Product variant 1	[1 1 2 1 1 1 2 1 2 3]	[1 1 3 1 1 1 2 1 2 3]	[1 1 2 1 1 1 2 1 1 3]
	Product variant 2	[1 1 1 1 1 1 2 1 2 3]	[1 1 2 1 1 1 2 1 2 3]	[1 1 2 1 1 1 1 1 2 1]
Upper-level objective values	Total profit	1.5600×10^{7}	1.2615×10^{7}	1.3234×10^{7}
	Market share	0.9205	0.9405	0.7530
	Z_1	0.5721	0.6913	1.0316
Lower-level MLB decisions	Task partition solution	1 4 | 5| 6| 2| 3| 7| 8| 9| 10	1 4 | 2| 5| 6| 3| 7| 8| 9| 10	1 2 3| 4| 5| 6| 7| 8| 9| 10
	Manufacturer number per task	2 2 2 2 1 2 3 1 2	2 2 2 2 2 2 3 1 2	2 2 1 1 1 2 1 1
Lower-level objective values	Operation cost	1.9507×10^{7}	2.2508×10^{7}	1.3505×10^{7}
	Load index	1.1217	1.8413	0.5000
	Z_2	0.3820	0.6732	7.4919×10^{-6}

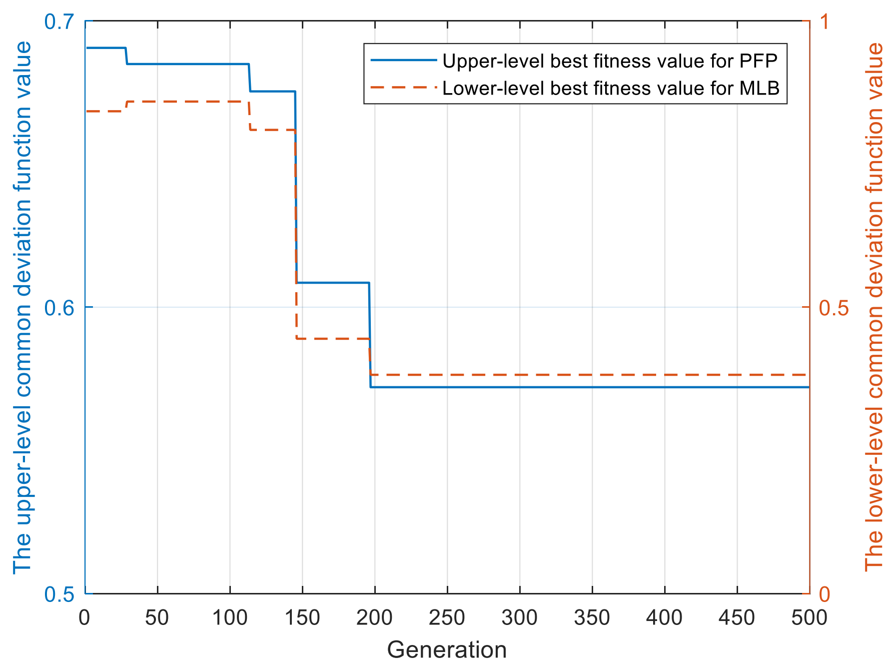


Figure 6-9 The evolution process of NBGA

Figure 6-10 compares the experimental results of the bilevel approach and the sequential approach. It indicates that the total profit increases by 23.66% from 1.2615×10^7 (TSA) in the sequential approach to 1.5600×10^7 (TBA) by adopting the bilevel approach, while the market share decreases by 2.13% from 0.9405 (TSA) to 0.9205 (TBA). Both the operation cost and the load index obtained in the sequential approach are much higher than those using the bilevel approach. For the upper-level common deviation function value Z_1 and the lower-level common deviation function value Z_2, the results obtained by the bilevel approach are 17.24% and 43.26% less than those of the sequential approach, respectively. The reason is that the independent PFP optimization in the sequential approach considers operation costs based on the estimation of existing historical data, and it cannot make full use of the low-cost advantage brought by the interactive design between PFP and MLB.

For TCA, the cooperative decision-making of PFP and MLB can be formulated as a bargaining model which is one single-level optimization formally (Dhingra and Rao, 1995). In this case, the bargaining objective function between PFP and MLB in TCA can be defined as equation (6.15).

$$\text{Max Z} = \left(1 - \frac{Z_1 - Z_1^*}{Z_1^- - Z_1^*}\right)\left(1 - \frac{Z_2 - Z_2^*}{Z_2^- - Z_2^*}\right) \tag{6.15}$$

where Z_1^* and Z_1^- are the upper-level best and worst common deviation function values respectively, and Z_2^* and Z_2^- are the lower-level best and worst common deviation function values, respectively. The traditional GA is employed to solve the proposed single-level optimization model in TCA. Figure 6-11 illustrates the evolution process of the traditional

GA for the cooperative approach. The optimal PFP and MLB results are listed in the last column in Table 6-5.

The target values of TP, MS, OC, and LI under TBA, TSA, and TCA are compared graphically in Figure 6-12. The values of TP and OC in TBA are in the desirable region, and those in TSA lies those in TSA lie in the tolerable region. The value of LI in TBA is in the tolerable region, but that in TSA is in the undesirable region. Compared with TBA, the values of TP and MS in TCA are only in the tolerable region, although the values of MS and LI in TCA are in the ideal region. It can be seen that the solution obtained by adopting the bilevel approach yields a better balance between the four objectives than other approaches.

6.6.3 Sensitivity Analysis

To explore the influence of competitive intensity on the objective values of upper-level PFP and lower-level MLB, the following sensitivity analysis experiment is designed and performed. The competitive intensity can be represented using the utility U_{ij}^{C} of competitive products in the market. Let $\overline{U_{ij}^{C}} = d \cdot U_{ij}^{C}$, where d is fixed as a series of constants from 0.95 to 1.3 in steps of 0.05. The obtained results are shown in Figure 6-13. With the increase of the parameter d, the upper-level objective values TP and MS decrease, and thus the upper-level common deviation function value Z_1 increases gradually. This decrease or increase makes the corresponding lower-level objective values OC and LI, as well as the lower-level common deviation function value Z_2 fluctuate accordingly.

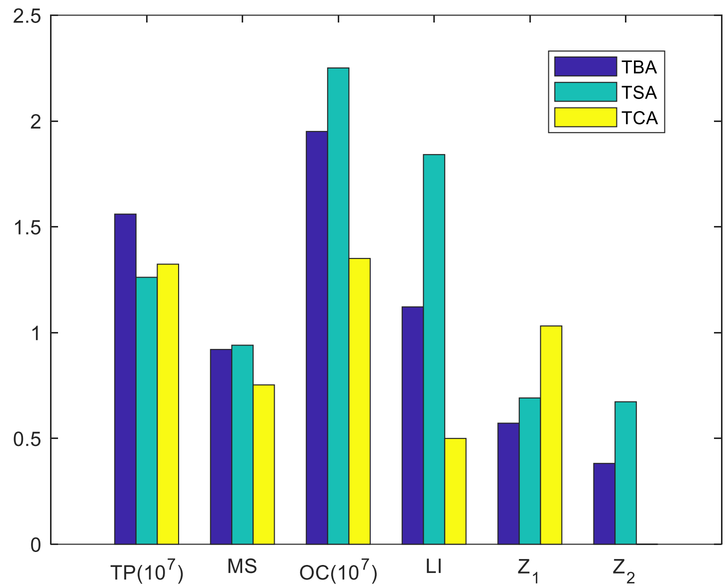


Figure 6-10 Comparison of experimental results

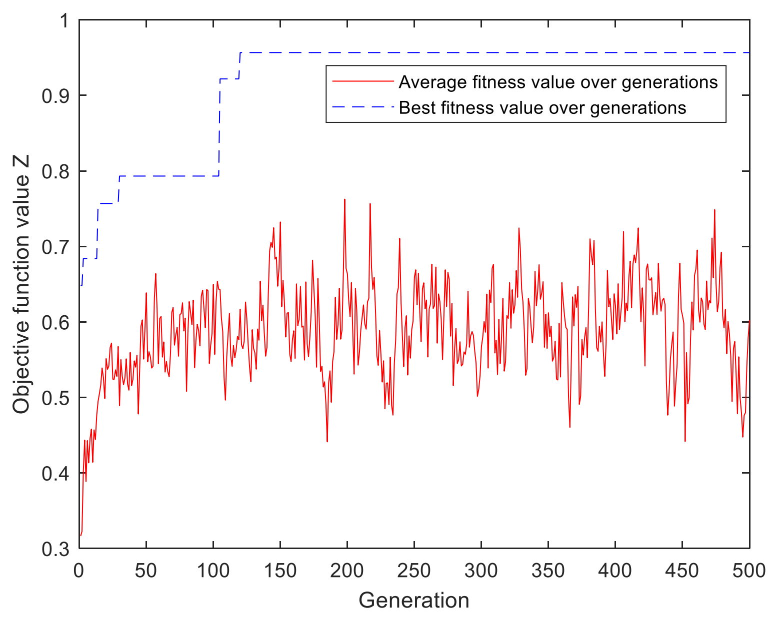
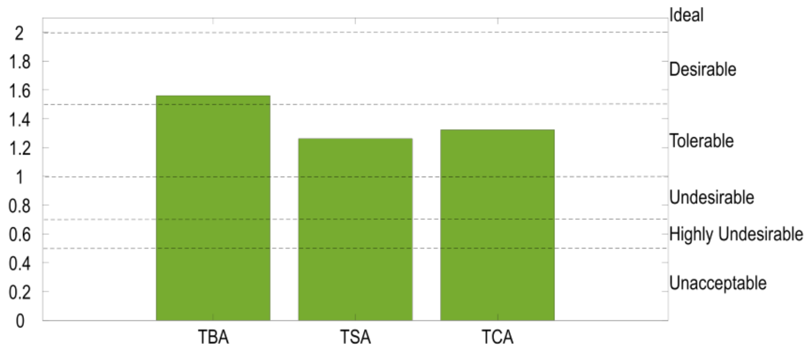
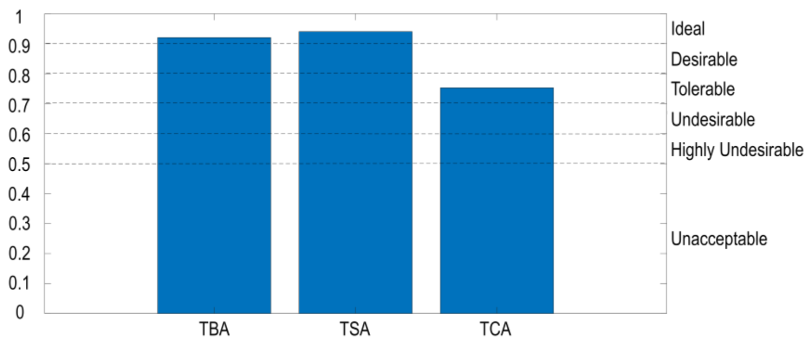


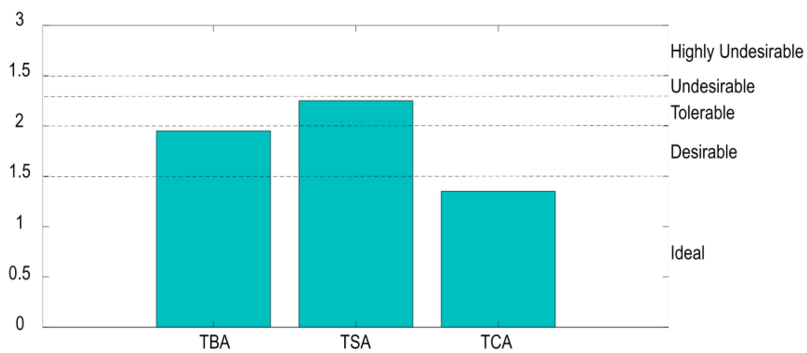
Figure 6-11 The evolution process for the cooperative approach



(a) Comparison of total profit under different approaches



(b) Comparison of market share under different approaches



(c) Comparison of operation cost under different approaches

(d) Comparison of load index under different approaches

Figure 6-12 Comparison of target values under different approaches

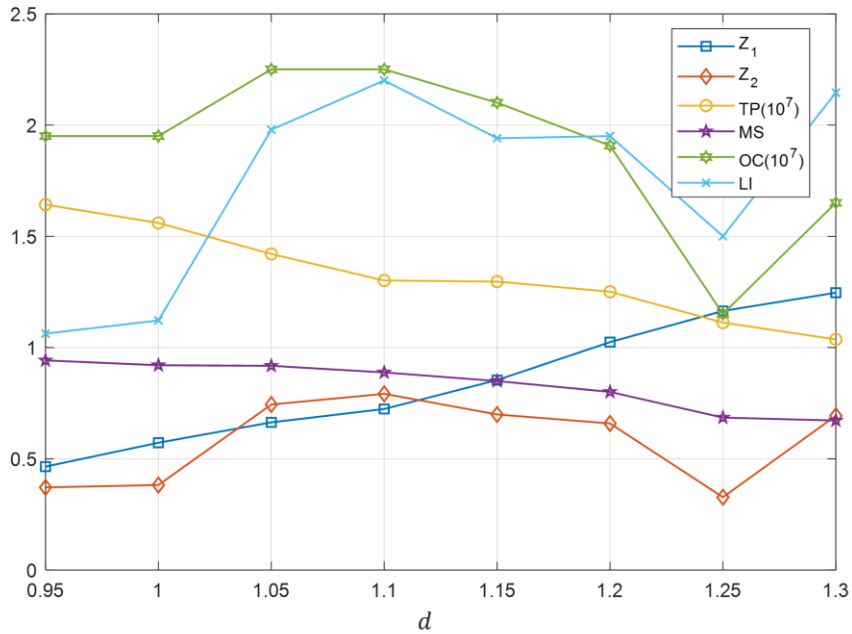


Figure 6-13 The influence of the competitive intensity

6.7 Chapter Summary

Coordinated optimization of PFP and MLB in platform-driven crowdsourced manufacturing adopts an interactive decision-making between various agents. A practical and effective bilevel approach for dynamic interactive design optimization of PFP and MLB is proposed based on Stackelberg game. Consistent with the leader-follower interactive mechanism, a bilevel optimization model with linear physical programming is developed, in which the upper- and lower-level objective functions are the common deviation functions adapted from the corresponding linear physical programs. NBGA with upper-level GA for PFP and lower-level GA for MLB is designed for solving the developed model. The proposed bilevel approach is demonstrated via a joint PFP and MLB design problem for tank trailer product family. Through comparison with other approaches, this bilevel approach is shown to yield satisfactory levels of achievement for PFP and MLB objectives. This approach provides an effective decision-making framework for the multi-agent online interactive product fulfillment faced by enterprises adopting the crowdsourced manufacturing model through service-oriented crowdsourcing platforms.

The managerial insight of this approach is as follows: (1) the PFP optimization decision-maker should take into account the reactions of competitors, since competitive intensity has a significant influence on the objective values; (2) with the progress of technologies and the change of customer preferences, it is important for the PFP design decision-maker to upgrade products and consider the co-evolution of product families and manufacturer crowds.

As a task execution decision support tool for manufacturer in platform-driven crowdsourced manufacturing, this chapter provides an original approach to decompose the product family into a set of executable tasks. It utilizes a bilevel architecture to imitate an interactive decision-making process between cyber platform and manufacturers. The platform can seek a maximum revenue of overall innovation project, while the manufacturer can consider their local constraints and operational revenues. The proposed work shows a crowdsourcing task derivation can be achieved through a bilevel game-theoretic programming. A crowdsourcing manufacturing process can provide the decision-support service for manufacturers and offers a more profitable task derivation result than conventional approach.

CHAPTER 7. NETWORKED MATERIAL FLOW PLANNING FOR CROSS-DOCKING LOGISTIC SERVICES: A BRANCH-AND-PRICE METHOD

Crowdsourced manufacturing through a platform-driven manner has been observed as an emerging trend towards Industry 4.0 by paving the way of delivering MaaS. It utilizes a cyber platform and crowdsourcing to reach external partners' manufacturing knowledge and resources while allowing companies to focus on their core competencies. It addresses an underlying logic that maximizes the reuse of resources by searching similarities among prolific product, process, and manufacturing resources varieties. It also challenges traditional logistic service for manufacturing industries by expanding a simple material flow to a complex, networked, fluctuating one. Cross-docking has been widely recognized as a logistic solution to complex material flow by splitting service routes to pickups and deliveries for maximizing vehicle reuse. It adopts a platform-driven strategy by exchanging loads at the cross-docking. This study formulates the logistic service problem in platform-driven crowdsourced manufacturing as a Crowdsourcing Vehicle Routing Problem with Cross-Docking (C-VRPCD), which integrates logistic solution provider crowds into the manufacturing service process. This study considers the logistic provider as a capacitated homogeneous vehicle started at various pickup points and times in a logistic service. The vehicles are scheduled in a route to visit service requesters synchronously and arrive at the cross-dock center simultaneously for load exchanging. All service requests must be fulfilled in predetermined time duration. Thus, this study formulates a mixed-integer linear programming (MILP) model for C-VRPCD to minimize the total cost of crowdsourcing fleet, which considers logistic solution provider hiring cost and vehicle operating cost. A branch-and-price (B&P) algorithm is proposed to solve this problem by using column generation techniques. This B&P algorithm first proves its effectiveness by solving benchmarking instances and compared with results from the default MILP solver in CPLEX. Computational analysis shows the proposed algorithm excels default MILP solver and can provide exact solutions.

In this regard, the rest of this study proceeds as follows. Section 1 discussed how cross-docking solves challenges raised by platform-driven crowdsourced manufacturing challenges and algorithm solutions of it. Section 2 presents the problem context and mathematical formulation of C-VRPCD. Section 3 establishes a branch-and-price algorithm for C-VRPCD. The computational results and comparison with benchmarks are presented in Section 4. A case study proceeds in Section 5 to verify the applicability of the proposed algorithm in the crowdsourcing environment. Section 6 closes this study.

7.1 Logistic Service in Platform-Driven Crowdsourced Manufacturing

Manufacturing industries are challenged by absorbing more disruptive changes that are impacted by sustainability issues, volatile customer preferences, and macro-environmental fluctuations and effectively delivering manufacturing services by adopting MaaS paradigm (Kusiak, 2019). Platform-driven crowdsourced manufacturing has been proposed to deliver service-oriented manufacturing through an external searching based on crowdsourcing and integrating resources into a manufacturing value chain by a cyber platform (Gong et al., 2021). It provides systematic solutions for manufacturers to peel their peripheral manufacturing activities and thus achieving economies of scale by offering substitutive services. From a perspective from managing material flow across manufacturer network, an optimal decision support on logistic services is critical, which can manage a networked material flow to serve materials and WIP transportation synchronously.

Since a cyber platform in crowdsourced manufacturing operates a two-sided peer-to-peer marketplace to match the open innovators and manufacturers. Thus, multiple value chains will be initiated by the open innovators and executed simultaneously. Since the fulfillment of each value chain requires collaboration among a group of manufacturers, several outstanding manufacturers may be awarded by multiple value chains. Thus, these manufacturers can be viewed as common vertexes in the networked material flow. The logistic service system should offer the logistic services to send the corresponding materials and WIP just-in-time. Because the highly innovative products can be characterized as large variety yet small volume, the logistic service is required to handle the product and production variety. In this regard, a service-oriented scheduling mechanism considering the product and production variety, as well as the manufacturing activities synchronization, can be recognized as the essential function in a logistic service system.

7.1.1 Cross-Docking for Networked Material Flow Management

The material flow management function in logistic services in platform-driven crowdsourced manufacturing aims to send as well as pick up the required material, WIP, subassemblies, or final products on time. Due to the large variety of value chains and the corresponding process variety, a manufacturer can be downstream partners for a set of upstream partners, since it is a vertex in a networked material flow network. Thus, the material and WIP delivery services for this manufacturer has multiple destinations.

Similarly, after the accomplishment of the manufacturing tasks, the picking up services will send the material and WIP to multiple downstream manufacturers. Thus, the process variety will propagate from process domain to the logistics one, therefore challenges the companies with keeping a reasonable cost as well as aligning customers, products, processes, and logistics domain for delivering an increasing product variety (Jiao et al., 2007b). From a platform-based perspective, a cyber platform can collect the information from the manufacturing crowds, formulate the origins and destinations of the service demands, find the common routes in the corresponding transportation service tasks and synchronize the manufacturing activities to achieve just-in-time (Qu et al., 2016).

The participation in platform-driven crowdsourced manufacturing implies that manufacturers open to external partners and allow the integration with partner crowds. Moreover, because recent advancement of information service system enables digitization of the manufacturing activities and the streaming of process data, the logistic service system can retrieve real-time data on the shop floor and making optimal decisions. The new synergy of the IoT and cloud computing architecture enables the visualization of the logistic on the shop floor and applying big data analysis of the material flow inner the manufacturers, thus, paves the way towards a holistic optimal logistic plan balancing the inner- and inter-manufacturers material flows (Zhong et al., 2015). The holistic logistic plan can synchronize transportation tasks and manufacturing activities. Therefore, the time gaps between the manufacturing task accomplishment and picking up as well as the materials or WIP deliveries and the start of order execution can be minimized. The smaller the time gaps are, the potential inventory level on the shop floor can achieve.

7.1.2 Crowdsourcing Vehicle Routing Problem with Cross-Docking

There is a stream of operation solutions for a logistic network with a high material variety with tight time constraints. The service-oriented logistic system installs an agile control mechanism, which entails a user-friendly, flexible, scalable, and widely connected engineering system architecture to link internal and external transportation (Evers et al., 2000). Traditional logistic solutions like direct shipping and milk-run send shipment from origin to destination directly and to multiple destinations in a tour, respectively, has been observed a limited capability of serving small shipment size and geographically dispersed customers (Buijs et al., 2014). To response these shortcomings, warehousing and cross-docking are developed by using a centralized depot.

Cross-dock can allow the inbound trucks to unload the freight and transported directly to the outbound trucks with no or simple storage infrastructure (Wen et al., 2009). Compared to the warehousing that holds an inventory of products to act as a shortage buffer, cross-docking groups similar shipping requirements and fulfilled by immediately recombination to a delivery tour in centralized freight terminal, which as known as cross-dock (Bozer and Carlo, 2008). The cross-docking addresses a platform-driven approach by operating in a similarity exploration and consolidating freight with the same downstream manufacturers utilizing fewer handling efforts to serve product varieties (Ladier and Alpan, 2016). It has been widely accepted as a solution to serve the complex logistic network with a short delivery lead time and the reduction of the storage space (Van Belle et al., 2012). It also requires a tightly synchronization of pickup and delivery routes to achieve a just-in-time paradigm by having no or less storage buffer (Vogt, 2010). Therefore, a successful cross-docking operation creates demands for holistic approach for modeling, quick response to uncertainty, and precise decision-making for resource planning.

Vehicle routing planning with time window (VRPTW) modeling and solutions for pickup and delivery truck fleet management builds up the foundation of cross-docking operation excellence (Shakeri et al., 2012). A large variety of algorithms have been developed to solve the cross-docking planning problem based on VRPTW formulation with time constraints and other management concerns (Buijs et al., 2014). The heuristic algorithm solution for vehicle routing problem with cross-docking (VRPCD) includes tabu search (Lee et al., 2006), multi-objective population-based heuristics (Arabani et al., 2011), large neighborhood metaheuristics (Grangier et al., 2017), to name but a few. Analytical solutions for VRPCD can be derived from MILP formulation by Lee et al. (2006), and it can be accelerated by adopting branch-and-price approach, which utilizes column generation techniques to model the transportation planning problem into a pair of master problem and subproblem and update the possible column pool (Santos et al., 2011).

Other fleet management issues rise along with the instantiation of cross-docking have also been explored, which includes include arrival uncertainty (Konur and Golias, 2013), pickup and deliveries with cross-docking (Santos et al., 2013), split deliveries (Moghadam et al., 2014), resource constraints (Grangier et al., 2021), and queue model-based multi-door facilities (Goodarzi et al., 2021). Platform-driven crowdsourced manufacturing searches a large amount of logistic service provider for materials and WIP deliveries, which rise a challenge for opening conventional vehicle routes and allowing the participation of logistic service provider crowds. Open vehicle routing problem has been proposed to accommodate third party logistic provider (Schopka and Kopfer, 2016).

Vincent et al. (2016) open traditional VRPCD and solve it by simulated annealing. According to our knowledge, an analytical solution for C-VRPCD is not existed.

7.1.3 Chapter Organization

In this regard, the rest of this study proceeds as follows. Section 2 presents problem context and mathematical formulation of C-VRPCD. Section 3 establishes a branch-and-price algorithm for C-VRPCD. The computational results and comparison with benchmark are presented in Section 4. A case study is proceeded in Section 5 to verify the applicability of the proposed algorithm in crowdsourcing environment.

7.2 Problem Definition and Mathematical Model

7.2.1 Problem Context for C-VRPCD

This study focuses on the logistic service system for platform-driven crowdsourced manufacturing, which is driven by the advantages of cross-docking following a platform-driven strategy and a crowdsourcing fulfillment strategy to utilize external logistic provider. The challenges of optimal scheduling in a crowdsourcing environment, which instantiates platform-driven strategy through cross-docking is solved by the model of C-VRPCD. Figure 1 conceptually illustrates C-VRPCD by incorporating three logistic providers to fulfill logistic service for crowdsourced manufacturing. A crowdsourcing logistic service process starts from a combination of different pickup routes, which has various locations and service time window. Logistic provider hired from a logistic crowd are viewed as homogeneous and are indexed as k, $k \in [1, K]$, $\gamma \in \mathbb{Z}^{+}$. The vehicles are scheduled to visit every manufacturer in a crowdsourcing network synchronously to exchange the WIP and materials they picked and are loaded for delivery routes. The WIP and material for manufacturers in pickup and delivery routes can be different to achieve a fulfillment of various innovative product projects. From a perspective of service quality control, all manufacturers should be visited exactly once per pickup and delivery routes.

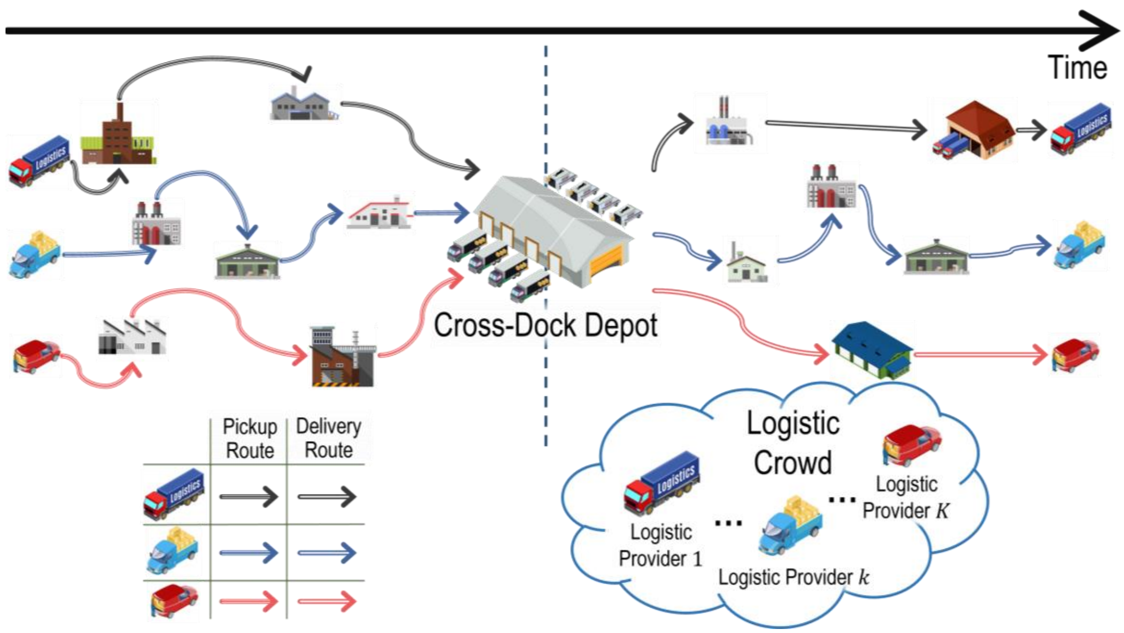


Figure 7-1 Crowdsourcing vehicle routing problem with cross-docking

The overall operational objective for this problem is seeking a minimized transportation cost following scheduled routes and an optimal number of hired logistic providers. It is a modification of the VRPCD which requires all vehicle starts from cross-docking depot. The C-VRPCD integrates logistic provider crowd via crowdsourcing, which implies that services of vehicles are started and ended by various time and locations. From a management perspective, this optimal decision-making process can be further decomposed as a series of VRP problems and an optimal combinatorial problem of combining possible routes to a fleet plan. An architecture of master problem and subproblems entails a negotiation process of a pricing and bidding process to plan logistic providers. As an analytical method of solving large scale MILP, B&P algorithm updates potential solution space through a column generation technique (Barnhart et al., 1998). B&P algorithm can solve m-VRPTW for cross-docking problem by model the route selection problem as a master one and individual routing problem as a subproblem (Santos et al., 2011).

7.2.2 Mathematical Formulation of C-VRPCD

C-VRPCD assumes logistic providers are heterogenous vehicles which park in a dummy cross-dock, which can access to all possible manufacturers with no distance, collect all vehicle without number limit, and be viewed as a logistic crowd. All of hired vehicles arrive at cross-dock simultaneously to enable a division of routes to pickup and delivery process. The nomenclature and MILP model for C-VRPCD is provided as following.

This model uses a graph-based presentation of potential routes. Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ collects all possible locations as vertices and $\mathcal{A}$ collects all possible trips as arcs. The vertices set $\mathcal{V}$ van be further decomposed as $\mathcal{V} = \{0\} \cup \{|0|\} \cup M^U \cup M^D$, where $\{0\}$ represents cross-dock depot, $\{|0|\}$ represents dummy cross-dock, M^U collects all upstream manufacturers who provide WIP and material, and M^D collects all downstream manufacturers who receive WIP and material. The total number of upstream manufacturers M^U is n^U and total number of downstream manufacturers M^D is n^D. The arcs can be further decomposed into $\mathcal{A} = \mathcal{A}^U \cup \mathcal{A}^D$, while $\mathcal{A}^U \cap \mathcal{A}^D = \emptyset$, where $\mathcal{A}^U = \{(i^U, j^U) : i^U, j^U \in \{|0|, 0, 1, \dots, n^U\}\}$ denotes all possible arcs connecting upstream manufacturers, cross-dock depot, and dummy cross-dock. Similarly, $\mathcal{A}^D =$

$\left\{(i^D, j^D): i^D, j^D \in \{|0|, 0, 1, \dots, n^D\}\right\}$ denotes all possible arcs connecting downstream manufacturers, cross-dock depot, and dummy cross-dock. Transportation cost c_{ij} are attached to these arcs, $\left\{c_{ij} \geq 0: (i, j) \in \mathcal{A}\right\}$. Logistic service requests are modeled as triples $\mathcal{S} = \left\{s_{i^U} \coloneqq \left(i^U, i^D, q_{i^U}\right): i^U \in \{1, \dots, n^U\}\right\}$, where each triple regulating the start and end positions and undividable load $q_{i^U} \geq 0$. These loads are shipped from requester upstream manufacturer i^U to receiver downstream i^D. A fleet size of K homogeneous vehicles are responsible for fulfillment all service request. In our model, every vehicle is required to stop at cross-dock depot $\{0\}$ before delivering to downstream manufacturer M^D for possible loads exchange. If the loads q_{i^U} from pickup routes R^P and delivery routes R^D use different vehicle k, an exchanging cost $c_{i^U}^k$ is generated.

Several decision variables and binary parameters are used for problem modeling. Decision variable β_r^k assumes 1 to indicate a pickup tour r utilize vehicle k, 0 for otherwise, and a transportation cost c_r are attached to it. Correspondingly, a decision variable $\gamma_{r'}^k$ assumes 1 for a delivery route use vehicle k, 0 otherwise, and cost is $c_{r'}$. Exchanging decision variable $\tau_{i^U}^k$ is introduced to indicate the load i^U in vehicle k is exchanged in cross-dock depot or not, and a cost $c_{i^U}^k$ is attached. Two binary parameters $a_r^{i^U}$ and $b_{r'}^{i^D}$ describes the pickup route r and delivery route r' in the form of whether it visit upstream manufacturer i^U and downstream manufacturer i^D or not, respectively.

The MILP mathematical formulation of C-VRPCD is equation (7.1.0) – (7.1.7).

$$\text{Min} \sum_{r \in R^P} c_r \sum_{k \in K} \beta_r^k + \sum_{r' \in R^D} c_{r'} \sum_{k \in K} \gamma_{r'}^k + \sum_{k \in K} \sum_{i^U \in S} c_{i^U}^k \tau_{i^U}^k \qquad (7.1.0)$$

$s.t.$

$$\sum_{r\in R^P} \beta_r^k = 1 \;\; \forall k \in K \tag{7.1.1}$$

$$\sum_{r'\in R^D} \gamma_{r'}^k = 1 \;\; \forall k \in K \tag{7.1.2}$$

$$\sum_{r\in R^P} a_r^{i^U} \sum_{k\in K} \beta_r^k = 1 \;\; \forall i^U \in M^U \tag{7.1.3}$$

$$\sum_{r'\in R^D} b_{r'}^{i^D} \sum_{k\in K} \gamma_{r'}^k = 1 \;\; \forall i^D \in M^D \tag{7.1.4}$$

$$\sum_{r\in R^P} \beta_r^k a_r^{i^U} - \sum_{r'\in R^D} \gamma_{r'}^k b_{r'}^{i^D} + \tau_{i^U}^k \geq 0, \forall i^U \in S, \forall k \in K \tag{7.1.5}$$

$$-\sum_{r\in R^P} \beta_r^k a_r^{i^U} + \sum_{r'\in R^D} \gamma_{r'}^k b_{r'}^{i^D} + \tau_{i^U}^k \geq 0, \forall i^U \in S, \forall k \in K \tag{7.1.6}$$

$$\beta_r^k, \gamma_{r'}^k, \tau_{i^U}^k, a_r^{i^U}, b_{r'}^{i^D} \in \{0,1\} \tag{7.1.7}$$

Equation (7.1.0) formulates the objective function of C-VRPCD, which minimizes total cost incurred in pickup, delivery, and exchanging operations through crowdsourcing logistic service process. Equation (7.1.1) and (7.1.2) are convexity constraints to enforce all vehicle are used in pickup and delivery routes. Equation (7.1.3) and (7.1.4) ensure that every request is covered without overlap by pickup and delivery routes. Equation (7.1.5) and (7.1.6) guarantee $\tau_{i^U}^k$ is 1 if a service load i^U use different vehicle. Equation (7.1.7) require all decision variables and binary parameters are either 0 or 1.

Following a branch-and-price modeling approach, the master problem modeled in equation (7.1.0) – (7.1.7) are changed to Restricted Master Problem (RMP) by replacing pickup R^P and delivery routes R^D by a restricted routes pool. B&P utilize linear programming relaxation of RMP by relaxing the integrality constraints. A series of dual variables $\{\upsilon^k : k \in K\}$, $\{\varphi^k : k \in K\}$, $\{\nu_{i^U} : i^U \in M^U\}$, $\{\mu_{i^D} : i^D \in M^D\}$, $\{\pi^k_{i^U} : i^U \in M^U\}$, $\{\chi^k_{i^U} : i^U \in M^U\}$ are assigned to constraints (7.1.1) – (7.1.6). The routes for RMP are updated by seeking routes in two minimization pricing problems modeled in (7.2) and (7.3).

$$\arg\min_{r \in R^P} c_r - \sum_{i^U \in M^U} a_r^{i^U} \nu_{i^U} - \sum_{i^U \in M^U} a_r^{i^U} \pi^k_{i^U} + \sum_{i^U \in M^U} a_r^{i^U} \chi^k_{i^U} - \upsilon^k, \forall k \in K \tag{7.2}$$

$$\arg\min_{r' \in R^D} c'_r - \sum_{i^D \in M^D} b_{r'}^{i^D} \mu_{i^D} + \sum_{i^U \in M^U} b_{r'}^{i^D} \pi^k_{i^U} - \sum_{i^U \in M^U} b_{r'}^{i^D} \chi^k_{i^U} - \varphi^k, \forall k \in K \tag{7.3}$$

7.3 Branch-and-Price algorithm for C-VRPCD

B&P method utilizes column generation technique in an iterative way and combines with branch-and-bond techniques to solve the linear relaxation of large-scale optimization models that involve a large volume of variables and associated columns (Choi and Tcha, 2007). It adopts divide-and-conquer philosophy, which solve the problem that the large-scale optimization has to consider massive volume of columns. B&P utilize the observation that the optimal solution of a combinatorial problem only includes a small subset of columns. Thus, an iteratively updating scheme can significantly limit the scale of master problem, and a pricing problem can be solved to seek a fast cost reduction.

7.3.1 Algorithm Architecture of B&P for C-VRPCD

The flowchart of B&P algorithm for C-VRPCD is shown in Figure 7-2. It starts with a subset of pickup route set R^P and delivery route set R^D as initial column set, which is generated through a heuristic to ensure the feasibility of initial solution. A full B&P algorithm has two cycle, column generation and branching operation. In each iteration of column generation, it solves linear programming relaxation of RMP considering a subset of the columns and yields dual variables, which includes υ^k, φ^k, ν_{i^U}, μ_{i^D}, π_i^k, χ_i^k. These dual variables are used to find negative reduced cost by solving pricing problems modeled in (7.1). If both pricing problems explored negative reduced cost columns, they can be added to the RMP. A new set of dual value can be explored by re-solve the RMP with linear programming relaxation. Otherwise, the current RMP solution founds its optimal and terminates column generation cycle.

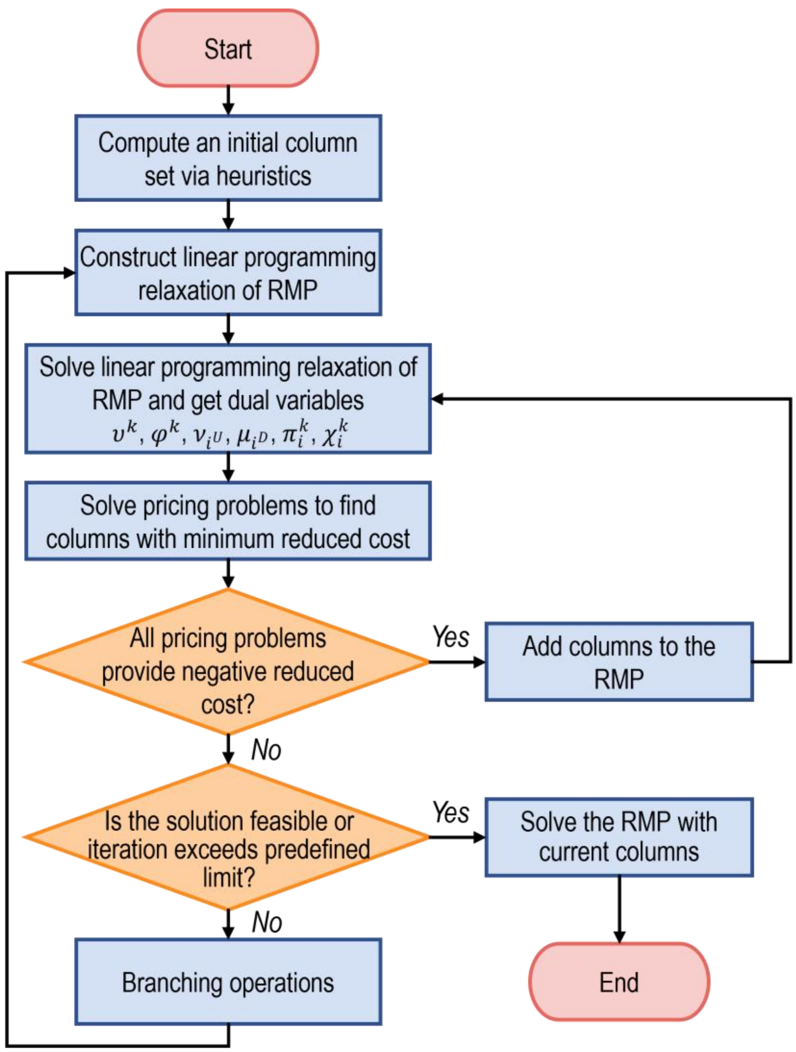


Figure 7-2 Flowchart of branch-and-price algorithm for C-VRPCD

The second cycle is branching operations to add bounding constraints following a branching-based tree search. It branches variables from master problem and finds feasible integer solution of current RMP. After all variables find integer solution after column generation process, the solution of RMP with current columns can be viewed as an optimal solution of the master problem. A predefined limit can be added to restrain the iteration of branching operations.

7.3.2 Pulse Algorithm for Pricing Problem

The objective of the pricing problem is to generate high-quality columns attached to variables that have the potential to improve the solution performance based on the current variables until no such variables can be found. This is done by exploring the variables with minimum negative reduced costs using the dual solution of the current linear programming relaxation of restricted master problem. Pricing problems are modelled in equation (7.2) and (7.3), which can be formulated as an elementary shortest path problem with resource constraints (ESPPRC). It checks the feasibilities of visiting nodes in a certain precedence and calculates objective value based on resource extension functions along a route.

Among the exact solutions for ESPPRC, labeling algorithms is the most widely used solution for pricing problems in B&P problems, which iteratively calculates the label (a tuple to represent a route) following dynamic programming approach and utilizes dominance rule to reduce searching space (Costa et al., 2019). Recently, pulse algorithm has been proposed for VRPTW as a pricing problem to serve column generation method (Lozano et al., 2016). Pulse algorithm firstly finds lower bounds on the cost given an amount of resource consumed, and recursively explores paths connecting vertices based on inexplicit enumeration in a graph through pulse propagation, which addresses a depth-first search of a directed graph. Pulse algorithm incorporates three pruning strategies: 1) feasibility pruning, which prunes infeasible paths by using structural constraints; 2) bound pruning, which utilize primal and lower bounds to discard suboptimal solutions; 3) rollback pruning, which compares pulses differing by the latest vertex visited to discard suboptimal partial paths.

A general pulse algorithm is shown in Table 7-1. Lines 1-4 in Table 7-1 initialize value for the partial path $\mathcal{P}$, the cumulative reduced cost $r(\mathcal{P})$, cumulative path load $q(\mathcal{P})$, cumulative path time consumption $t(\mathcal{P})$. Line 5 calls bounds function to find lower bound for every node of the question, which is shown in Table 7-2. Line 6 triggers recursive pulse which propagates from the start nodes v_s (dummy cross-dock $\{|0|\}$ in pickup routes and cross-dock depot $\{0\}$ in delivery routes). This function will explore all of the information of feasible path from v_s to end node v_e (cross-dock depot $\{0\}$ in pickup routes and dummy cross-dock $\{|0|\}$ in delivery routes).

The pulse algorithm starts with three prune functions, namely Feasible, Bounds, RollBack, which is called in line 7, 8, and 9, respectively. These prune functions can ensure that the pulse algorithm can find an optimal elementary path in a limited space efficiently. If the path is not pruned, the current path is added to partial paths of current node v_i in line 10. Line 11 updates the vehicle loads. Line 12 to line 16 forms a for-loop to propagate the pulse by invoking the pulse procedure to every possible node $v_j \in \mathcal{A}_i^+$, where $\mathcal{A}_i^+$ is the set of accessible nodes set of current one v_i.

Table 7-1 Pseudocode of general pulse algorithm for ESPPRC

Algorithm 7-1: general pulse algorithm for ESPPRC

Input: $\mathcal{G}$ *directed graph;* $\mathcal{v}_s$ *start node;* $\mathcal{v}_e$ *end node;* $\mathfrak{d}$ *bound step size;* $[\underline{t}, \overline{t}]$ *time bound for planning;* $r(\mathcal{P})$ *path reduced cost;* $q(\mathcal{P})$ *path load;* $t(\mathcal{P})$ *path time;* $\mathcal{v}_i$ *current node;* $\mathcal{A}_i^+$ *set of accessible nodes set of current one* $\mathcal{v}_i$.

Output: $\mathcal{P}^*$ *optimal path*

$\mathcal{P} \leftarrow \{0\}$;
$r(\mathcal{P}) \leftarrow 0$;
$q(\mathcal{P}) \leftarrow 0$;
$t(\mathcal{P}) \leftarrow 0$;
$\mathrm{bound}(\mathcal{G}, \mathfrak{d}, [\underline{t}, \overline{t}])$;
$\mathrm{pulse}(\mathcal{v}_s, r(\mathcal{P}), q(\mathcal{P}), t(\mathcal{P}), \mathcal{P})$;
    if $\mathrm{Feasible}(\mathcal{v}_i, q(\mathcal{P}), t(\mathcal{P})) = \mathrm{true}$
        if $\mathrm{Bounds}(\mathcal{v}_i, t(\mathcal{P}), r(\mathcal{P})) = \mathrm{false}$
            if $\mathrm{RollBack}(\mathcal{v}_i, t(\mathcal{P}), r(\mathcal{P}), \mathcal{P}) = \mathrm{false}$
                $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathcal{v}_i\}$;
                $q(\mathcal{P}) \leftarrow q(\mathcal{P}) + q_i$;
                for $\mathcal{v}_j \in \mathcal{A}_i^+$ **do**
                    $r(\mathcal{P}') \leftarrow r(\mathcal{P}) + r_{ij}$;
                    $t(\mathcal{P}') \leftarrow \max\{a_j, t(\mathcal{P}) + t_{ij}\}$;
                    $\mathrm{pulse}(\mathcal{v}_s, r(\mathcal{P}), q(\mathcal{P}), t(\mathcal{P}), \mathcal{P})$;
                end for
            end if
        end if
    end if
end pulse
return $\mathcal{P}^*$

Once the pulse algorithm reached the end node $\mathcal{v}_e$, the best-performed path $\mathcal{P}^*$ will be updated. The algorithm will be terminated until the last recursive propagate reach the end node $\mathcal{v}_e$.

The feasibility pruning is proceeded through function Feasible. The paths that violate structural constraints can be identified and discarded. The constraints covered by this study includes time window, vehicle load capacity, and cycle constraints.

The bound pruning limited the search space by providing the lower bounds $\underline{r}(v_i, t(\mathcal{P}))$ for every node, which is shown in Table 7-2. The bound contains minimum reduced cost of a path $\mathcal{P}$ that reaches v_i.The time bound of planning horizon $[\underline{t}, \overline{t}]$, bound step size ƌ, and directed graph are essential input for this function. The time windows for paths are gradually reduced by the give step size ƌ. The output of this function is denoted as lower bound matrix $\mathbf{B} = [\underline{r}(v_i, \tau)]$, which stores all lower bonds for every node and time step. Lines 4 to 9 solves ESPPRC for every node in that consumption using pulse procedure. Lines 10 – 14 stores optimal reduced cost value found for every node at given time $\underline{r}(v_i, \tau)$. If the optimal path is an empty set, the reduced cost will be set to infinity.

Table 7-2 Pseudocode of bounds function for pulse algorithm for ESPPRC

Algorithm 7-2: Bounds function for pulse algorithm for ESPPRC

Input: $\mathcal{G}$ *directed graph;* ƌ *bound step size;* $[\underline{t}, \overline{t}]$ *time bound for planning.*

Output: $\mathbf{B} = [\underline{r}(v_i, \tau)]$, *lower bound matrix.*

1: $\tau \leftarrow \overline{t}$;
2: **while** $\tau > \underline{t}$ **do**;
3: $\quad \tau \leftarrow \tau -$ ƌ;
4: $\quad$ **for** $v_i \in \mathcal{A}$ **do**
5: $\quad\quad \mathcal{P} \leftarrow \{\}$;
6: $\quad\quad r(\mathcal{P}) \leftarrow 0$;
7: $\quad\quad q(\mathcal{P}) \leftarrow 0$;
8: $\quad\quad t(\mathcal{P}) \leftarrow \tau$;
9: $\quad\quad \text{pulse}(v_s, r(\mathcal{P}), q(\mathcal{P}), t(\mathcal{P}), \mathcal{P})$;
10: $\quad\quad$ **if** $\mathcal{P}^* = \{\}$ **then**;
11: $\quad\quad\quad \underline{r}(v_i, \tau) \leftarrow \infty$;
12: $\quad\quad$ **else**
13: $\quad\quad\quad \underline{r}(v_i, \tau) \leftarrow r(\mathcal{P}^*)$;
14: $\quad\quad$ **end** if
15: $\quad$ **end** for
16: **return B**

Rollback pruning aims to avoid exploration of unpromising regions by making better decisions in the early searching stage by backtrack to a better initial choice in a depth-first graph search. Once a path $\mathcal{P}_{sj}$ reaches node v_j, a reevaluation of potential bypath is possible or not, and correspondingly, the reduced cost of these bypaths will be benefited from or not are proceeded. If the paths are subset of the other one and all of the values are higher than the other one, it can be evaluated as a dominated path candidate (Feillet et al., 2004). Compared to conventional labelling algorithm, this pulse algorithm excels in getting rid-off the storage of saving labels.

7.3.3 Branching Heuristics

B&P synergizes branch-and-bound and column generation by iteratively searching of branching tree, as shown in Figure 7-2. The variable chosen to be branched in this study is $\tau_{i^U}^k$, which links the balance of pickup and delivery routes by Equation (7.1.5) and (7.1.6). The branching of $\tau_{i^U}^k$ can formulate a pair of k and i^U can be used for uploading a father node of the branching tree. A branching uncertainty index $\kappa_{i^U}^k$ is introduced for determining the branching priority, which can be defined in Equation (7.4).

$$\kappa_{i^U}^k \coloneqq \sum_{r \in R^P} \min\left\{a_r^{i^U} \beta_r^k, 1 - a_r^{i^U} \beta_r^k\right\} + \sum_{r' \in R^D} \min\left\{b_{r'}^{i^D} \gamma_{r'}^k, 1 - b_{r'}^{i^D} \gamma_{r'}^k\right\}, \forall k \in K \qquad (7.4)$$

It measures the uncertainty that a vehicle k serve a request i^U or not. If a vehicle k is assigned or unassigned to a request certainly, $\kappa_{i^U}^k$ will approach to zero. Otherwise, $\kappa_{i^U}^k$ will increase to show a high uncertainty of assigned or unassigned to a request. Thus, branching the maximum $\kappa_{i^U}^k$ in all K can be perceived as a high searching efficiency to deviate uncertainty.

If the branch variable $\tau_{i^U}^k$ are branched to zero, a service load i^U will not load to or unload from the vehicle k at cross-dock. $\tau_{i^U}^k = 0$ implies two situations, which are $\sum_{r \in R^P} a_r^{i^U} \beta_r^k = 0$ and $\sum_{r' \in R^D} b_{r'}^{i^D} \gamma_{r'}^k = 0$, which indicates vehicle k never touch service load i^U, and $\sum_{r \in R^P} a_r^{i^U} \beta_r^k = 1$ and $\sum_{r' \in R^D} b_{r'}^{i^D} \gamma_{r'}^k = 1$. Otherwise, $\tau_{i^U}^k = 1$ implies that $\sum_{r \in R^P} a_r^{i^U} \beta_r^k$ and $\sum_{r' \in R^D} b_{r'}^{i^D} \gamma_{r'}^k$ have different value, which is shown in Table 7-3.

Table 7-3 Four nodes of branching of a parent node

		Vehicle k serves service load i^U or not in pickup routes	
		$\sum_{r \in R^P} a_r^{i^U} \beta_r^k = 0$	$\sum_{r \in R^P} a_r^{i^U} \beta_r^k = 1$
Vehicle k serves service load i^U or not in delivery routes	$\sum_{r' \in R^D} b_{r'}^{i^D} \gamma_{r'}^k = 0$	$\tau_{i^U}^k = 0$	$\tau_{i^U}^k = 1$
	$\sum_{r' \in R^D} b_{r'}^{i^D} \gamma_{r'}^k = 1$	$\tau_{i^U}^k = 1$	$\tau_{i^U}^k = 0$

This branching rule makes C-VRPCD a quadtree, and a best-first strategy with maximum $\kappa_{i^U}^k$ can search the branch tree in a depth-first manner.

7.4 Computational Results of C-VRPCD

This experiment is from a case of tank trailer crowdsourced manufacturing. Table 7-4 shows the details of tasks within the planning horizon. The tasks are modified from a real-world logistic routes planning problem. Six types of data are given: task ID, pickup time, delivery time, the manufacturer of arrival, and the manufacturer of departure. The travel time among each manufacturer and cross-dock depot is given in Table 7-5. These data are inputs of the networked material flow algorithm. The planning result is shown in Table 7-6, including the task ID, and logistic service provider's ID.

Table 7-4 Crowdsourcing logistic service tasks

Task ID	Pickup Time	Delivery Time	Arrival	Departure
1169624838194331684	2019-01-16 05:25:00	2019-01-16 09:45:00	'5'	'3'
1169624838190137854	2019-01-16 06:30:00	2019-01-16 18:50:00	'N1'	'93'
1169624838190137684	2019-01-16 05:40:00	2019-01-16 18:04:00	'4'	'95'
1169624838164971555	2019-01-16 07:45:00	2019-01-16 16:05:00	'3'	'4'
1169624838194331933	2019-01-16 12:45:00	2019-01-16 18:05:00	'95'	'95'
1169624838164971640	2019-01-16 06:50:00	2019-01-16 19:10:00	'3'	'93'
1169624838164971620	2019-01-16 11:00:00	2019-01-16 14:20:00	'N1'	'N1'
1169624838164971621	2019-01-16 07:05:00	2019-01-16 15:25:00	'93'	'3'
1169624838190137954	2019-01-16 12:05:00	2019-01-16 17:25:00	'95'	'N1'
1169624838190137681	2019-01-16 05:10:00	2019-01-16 15:30:00	'4'	'5'
1169624838190137969	2019-01-16 06:15:00	2019-01-16 19:39:00	'5'	'3'
1169624838164971543	2019-01-16 09:25:00	2019-01-16 15:45:00	'3'	'93'
1169624838164971545	2019-01-16 12:25:00	2019-01-16 16:41:00	'5'	'5'
1169624838185943261	2019-01-16 08:30:00	2019-01-16 16:46:00	'N1'	'4'
1169624838164971572	2019-01-16 12:35:00	2019-01-16 13:55:00	'4'	'93'
1169624838164971573	2019-01-16 08:40:00	2019-01-16 16:00:00	'95'	'5'
1169624838164971523	2019-01-16 09:45:00	2019-01-16 17:09:00	'4'	'N1'
1169624838164971524	2019-01-16 05:50:00	2019-01-16 19:14:00	'N1'	'3'
1169624838164971614	2019-01-16 09:50:00	2019-01-16 13:06:00	'5'	'95'
1169624838164971533	2019-01-16 05:10:00	2019-01-16 17:34:00	'4'	'93'
1169624838164971575	2019-01-16 06:10:00	2019-01-16 18:30:00	'93'	'4'
1169624838185943255	2019-01-16 05:10:00	2019-01-16 13:26:00	'93'	'5'
1169624838164971534	2019-01-16 10:15:00	2019-01-16 17:39:00	'95'	'4'
1169624838164971584	2019-01-16 12:15:00	2019-01-16 18:31:00	'95'	'N1'
1169624838164971585	2019-01-16 08:20:00	2019-01-16 13:36:00	'5'	'93'
1169624838190137859	2019-01-16 05:20:00	2019-01-16 17:36:00	'3'	'95'
1169624838185943247	2019-01-16 08:30:00	2019-01-16 15:54:00	'N1'	'95'
1169624838164971638	2019-01-16 07:35:00	2019-01-16 13:51:00	'4'	'N1'
1169624838185943248	2019-01-16 12:35:00	2019-01-16 16:59:00	'3'	'95'
1169624838164971566	2019-01-16 11:45:00	2019-01-16 16:05:00	'4'	'N1'
1169624838185943263	2019-01-16 06:45:00	2019-01-16 16:01:00	'4'	'4'
1169624838194332048	2019-01-16 11:45:00	2019-01-16 14:09:00	'5'	'3'
1169624838164971567	2019-01-16 10:50:00	2019-01-16 19:10:00	'5'	'3'

Table 7-5 Travel time among manufacturers and cross-dock

	'3'	'4'	'5'	'93'	'95'	'N1'	'{0}'
'3'		15	4	8	46	39	42
'4'	15		23	18	38	30	35
'5'	4	23		37	18	19	36
'93'	8	18	37		20	16	25
'95'	46	38	18	20		25	18
'N1'	39	30	19	16	25		14
'{0}'	42	35	36	25	18	14	

Table 7-6 Planning result of B&P for C-VRPCD

Task ID	Provider ID	Task ID	Provider ID
1169624838194331684	1000423	1169624838164971573	1000422
1169624838190137854	1000458	1169624838164971523	1000458
1169624838190137684	1000425	1169624838164971524	1000428
1169624838164971555	1000504	1169624838164971614	1000506
1169624838194331933	1000422	1169624838164971533	1000502
1169624838164971640	1000506	1169624838164971575	1000422
1169624838164971620	1000505	1169624838185943255	1000428
1169624838164971621	1000455	1169624838164971534	1000458
1169624838190137954	1000428	1169624838164971584	1000506
1169624838190137681	1000456	1169624838164971585	1000422
1169624838190137969	1000457	1169624838190137859	1000458
1169624838164971543	1000450	1169624838185943247	1000502
1169624838164971545	1000458	1169624838164971638	1000428
1169624838185943261	1000451	1169624838185943248	1000506
1169624838164971572	1000452	1169624838164971566	1000423
1169624838185943263	1000423	1169624838164971567	1000422

7.5 Chapter Summary

Logistic services for manufacturer entail a networked material flow planning, which plans a set of optimal vehicle service routes to link manufacturers as a material network. This chapter propose a cross-docking service method to use a platform-based strategy by splitting service routes into pickup and delivery ones and exploring maximum similarities among them. The vehicle can be used maximally by synchronizing pickup and delivery activities to achieve no or few inventory in cross-dock depot. This service method shows the potential of handling a large number of manufacturers and volatile service requirements in platform-driven crowdsourced manufacturing.

This chapter formulates an optimal decision-making of logistics services for platform-driven crowdsourced manufacturing through cross-docking as C-VRPCD. A B&P algorithm for C-VRPCD is proposed in this chapter. It utilizes a divide-and-conquer philosophy to decompose C-VRPCD into master problem and subproblem, which are connected by dual value and pricing problems to update column pool for size-controlled RMP. Pulse algorithm has been applied as the fundamentals of solving subproblem, which utilizes a recursively depth-first search of the directed graph. The proposed branching rules search integral solutions in a quadtree manner.

This chapter provides logistic service solutions for manufacturers, which optimally plans logistic routes for crowdsourcing network. It enables manufacturers peeling off their logistic department and focusing on manufacturing activities. Also, a B&P algorithm solution is also proposed to solve the emerging crossdocking solution.

CHAPTER 8. TASK DISPATCHING AND SCHEDULING THROUGH REAL-TIME CROWDSOURCING TASK ACCEPTANCE AND ACCOMMODATION: AN INTERACTIVE BILEVEL OPTIMIZATION MODEL

With the fourth-generation industrial revolution, manufacturing industries are focusing on dynamic, fully autonomous, and more customer-oriented production systems. This customer-oriented change converts classically static customer demand into that which is dynamic and real-time, as no prior information regarding customer demand is known in advance. This paper focuses on real-time order acceptance and scheduling (r-OAS) for a data-enabled permutation flow shop. To compensate for the shortage in prevailing approaches that make bottleneck-based decisions or assume that the intermediate buffers among workstations are infinite, an r-OAS scheme is generated based on a data-driven representation, which can concisely predict the dynamic production status of flow shops and the corresponding makespan of a job with finite intermediate buffer constraints. Using this representation, real-time job release planning (r-JRP) can be coupled with r-OAS to minimize various operational costs of flow shops (i.e., the costs of the work-in-process, earliness, and tardiness). In terms of the inherent interactive mechanism between r-OAS and r-JRP, in which r-OAS generates a decision space for r-JRP and r-JRP then feeds the lowest operational costs back for use in r-OAS decision-making, a bilevel interactive optimization (BIO) is formulated to simultaneously address the two subproblems based on the Stackelberg game. The r-OAS acts as the leader, while r-JRP acts as the follower. The BIO is a type of nonlinear integer programming, and a bilevel tabu-enumeration heuristic algorithm is developed to solve it. The efficiency of the BIO is verified through a practical case study.

8.1 Crowdsourcing Task Acceptance and Scheduling

Order acceptance and scheduling (OAS) is an important decision in make-to-order production environments, as it may not be beneficial for firms to accept and produce all potential orders due to limited production capacity and tight delivery due dates. Over the past few decades, OAS has attracted considerable attention from researchers and practitioners (Slotnick, 2011), in which order acceptance is considered when determining which orders to accept, while order scheduling is addressed when identifying a production sequence for the accepted orders (Lin and Ying, 2015). This stream of research focuses on OAS problems with different objectives in various production environments: for example, deterministic and stochastic single- or multiple-machine problems with the objectives of revenue maximization, lead-time and due-date setting, and cost minimization (Slotnick, 2011). In many manufacturing industries, the OAS for permutation flow shops presents a challenging problem for sequence-determined setup times and dynamic job transitions among workstations, which in turn problematize attempts to calculate a given job's makespan. Xiao et al. (2012) and Lin and Ying (2015) study the static OAS problem for permutation flow shops by considering the known arrival time, due date, and composition of each order.

The recent advancement of ICT brings the real-time data stream to the shop-floor manufacturing scheduling, which enables manufacturers to focus on constructing dynamic, fully autonomous, and more customer-oriented production systems. Therefore, OAS is becoming dynamic and customer oriented, whereby customers can place orders with desired order compositions and due dates in the system in real time. This means that decision makers must accept or reject orders in real time after considering the current production capacity of flow shops and customer-specified due dates. The newly accepted orders must be scheduled alongside existing orders, which are either already accepted and in-process by machine(s) or waiting in a processing queue. We define this problem as r-OAS, a term that is used to reference an optimization model that can utilize real-time data to update the pre-determined OAS decision without disrupting production system operations. Rahman et al. (2015) and Rahman et al. (2019) studied the r-OAS problem for permutation flow shops by developing heuristic and meta-heuristic algorithms; however, this research stream oversimplified this problem and may cause challenges when implemented in the real world:

1) Real-time production status identification is the foundation for r-OAS decision-making. In r-OAS problems, the OAS decision is dynamically updated based on the current production status. This status is difficult to identify, as it is affected by various resource- and job-related disruptions. Previous studies assume that production systems are deterministic, such that their real-time status can be derived simply by analyzing production planning and scheduling schemes (Wang et al., 2013, Lin and Ying, 2015).

2) Reliable prediction for job makespan is the core of generating an optimum r-OAS scheme (De Jong et al., 2019). Currently, most permutation flow shop scheduling assumes that the capacity of the intermediate buffers is sufficient, such that the starvation/blockage of workstations will not occur. This assumption allows dynamic job transitions among workstations to be modeled by linear functions, and thus the makespan of a job can be

easily estimated. While these linear functions are beneficial for developing linear programming models and solving methods related to r-OAS problems, they are unrealistic, as the intermediate buffers must be finite for real-time WIP inventory control.

3) The objective of existing studies examining r-OAS focuses on maximizing the number of accepted orders while minimizing the order completion times (Rahman et al., 2015). This objective simplifies the coordination among multiple operation costs in flow shops (i.e., WIP inventory cost, final product inventory cost caused by earliness, and tardiness penalty cost), as many production systems today are required to operate within just-in-time paradigms.

Until now, these simplifications have been considered reasonable, as the precise status of production systems is difficult to perceive in real time. The job transition behaviors among workstations, which are constrained by finite intermediate buffers and affected by various disruptions, are difficult to derive; thus, the makespan of a job cannot be concisely predicted. Due to this shortage, it is impossible to construct an approach for r-JRP, which is vital in coordinating the various operation costs after receiving an r-OAS decision (Chen et al., 2020). Owing to new information technologies (e.g., RFID and sensor networks) in smart manufacturing, many transparent and real-time data that reveal the production system status are being collected and analyzed. Using these data, the effects of disruptions can be evaluated and the multi-stage time-varying transition behaviors of flow shops can then be derived (Chen et al., 2020). This derivation enables the efficient identification of the real-time status of production systems. The makespan of a job and the multiple operation costs of a flow shop can be predicted based on this identification, and the r-JRP decision can then be made along with r-OAS.

This study's primary purpose focuses on developing a joint optimization for r-OAS and r-JRP (referenced as r-OASR herein) problems in permutation flow shops with data-driven production status and job makespan prediction. Recently, with the increased interest in the ability to make real-time decisions, several questions and concerns regarding the effects of using real-time data on production planning and scheduling have been discussed and reported (Hozak and Hill, 2009, Ghaleb et al., 2020). These discussions show that continuous replanning and rescheduling may increase the required setups, transportations, and nervousness of production systems. In our r-OASR problem, the continuous updating of OAS can simultaneously increase the net revenue and reputation of manufacturers, whereas it may also sometimes cause system nervousness (Rahman et al., 2015). Fortunately, today's smart manufacturing devotes to formulating a universal adaptive capacity and thus system nervousness caused by r-OASR can be reduced in a certain extent (Shiue et al., 2018). Moreover, the sequence-dependent setup time for different orders is considered in our r-OASR problem such that the increased nervousness can be efficiently dealt with. When designing the searching algorithm, a special solution initiation approach is constructed, such that the OAS scheme inherited from the previous stage can be maintained to the extent possible. Based on these issues, the contributions of this research can be concluded as the following. (1) A data-driven representation approach is proposed to reveal the time-varying transition behaviors of flow shops with finite intermediate buffer constraints and sequence-dependent setup times. In turn, the real-time production status of flow shops can be derived and the makespan of a job can be precisely predicted. (2) An r-JRP is constructed based on the data-driven representation to minimize the various operation costs in flow shops. (3) A BIO is formulated after addressing the inherent interaction and hierarchical characteristics between r-OAS and r-JRP. (4) A bilevel heuristic algorithm comprising a tabu search and an implicit enumeration with a special solution initiation method is designed to manage this joint optimization. The results of the case study demonstrate that the proposed BIO performs better than bottleneck-based approaches and step-by-step optimization methodologies, in terms of maximizing the net revenue of flow shops.

8.2 Real-time Order Acceptance and Scheduling for Data-enabled Permutation Flow Shops

The research stream from static OAS problem towards a synthesis of real-time information and data-driven method is reviewed in this section. Most methods reviewed focus solely on the r-OAS problem with a tardiness-related objective, while multiple operation costs, which can be coordinated by r-JRP, are less involved. Some researchers have undertaken an examination of r-JRP problems at the scheduling stage to simultaneously minimize tardiness, earliness, and flowtime, but no research has extended the r-JRP into the area of r-OAS or that of scheduling methods with finite intermediate buffer constraints. Driven by the application of information and communication technologies, scholars are now exploring data-driven OAS problems after analyzing the real-time data from shop floors. They use these data to successfully estimate uncertain operational parameters in planning and scheduling models but ignore the potential value of these data in predicting the production status and job makespan of flow shops, which are both essential for r-OASR decision-making. Additionally, the data-rich environment enables smart production planning and control. This smart capacity can reduce the nervousness caused by real-time rescheduling and thus promote the development of r-OASR.

8.2.1 *Static OAS problems*

Static OAS problems in single- or multiple-machine production environments are generally formulated at the beginning of production, when a shop floor receives a pool of orders and must determine which ones to accept based on its available capacity and order due dates (Slotnick, 2011). The OAS that uses a single machine is a generalization of an OAS problem with a specialized scarce resource, which constitutes the bottleneck of a production system (Nobibon and Leus, 2011). Many modeling approaches focusing on the characteristics of this OAS have been developed. For example, Nobibon and Leus (2011) propose two linear formulations for static OAS problems and design two exact branch-and-bound procedures to resolve instances including a maximum of 50 jobs. Silva et al. (2018) examine a new arc-time-indexed mathematical formulation after considering the sequence-determined setup time and develop two exact algorithms based on Lagrangian relaxation and column generation. When the size of the problem is particularly large, various efficient approximation heuristics, whose type can be classified as either construction or improvement, are produced to find a near-optimal solution within a reasonable computation time (Chaurasia and Singh, 2017, Lin and Ying, 2015). Construction heuristics are frequently used to generate a fairly good solution for improvement heuristics or other metaheuristics, while improvement heuristics begin with an initial solution and are then repeated to improve the solution within a reasonable period. With the exception of these approximation heuristics, several metaheuristics, including simulated annealing, genetic algorithm, artificial bee colony algorithm, and tuba search, have been proposed to resolve this OAS problem with different versions (Slotnick, 2011); the last two algorithms demonstrate the best performance in solving this problem.

The OAS problems involving parallel machines or in flow shops are typical extensions of OAS with a single machine. Wang et al. (2015) examine an OAS problem with two identical parallel machines by designing exact and heuristic techniques. The proposed exact technique can solve small OAS problems with a maximum of 20 jobs. Wu et al. (2018) extend the model of Wang et al. (2015) to include multiple identical parallel machines and consider the sequence-dependent setup times among the accepted orders. A flow-like metaheuristic is developed for this OAS, the efficiency of which has been compared to classical particle swarm optimization. Flow shop scheduling has been extensively studied due to its diverse industrial and economic applications (Komaki et al., 2019). In examining whether the job sequence is the same for all machines, scheduling can be decomposed into permutation and non-permutation problems (Rossit et al., 2018). Xiao et al. (2012) and (Xiao et al., 2015) extend permutation and non-permutation flow shop scheduling into OAS, respectively, after addressing order tardiness. To formulate an exact algorithm for the problem, Wang et al. (2013) develop a mixed-integer linear programming model and construct a branch-and-bound algorithm to solve problem instances with a maximum of 20 jobs. If the problem size is large, a parallel neighborhood search (Lei and Guo, 2015) and multi-initiator-simulated annealing (Lin and Ying, 2015) are proposed. Although many new mathematical formulations and solution techniques have recently been developed for OAS using different production environments, this area of research is still in its nascent stage and requires further development because of its widespread applications in today’s make-to-order practice.

8.2.2 Real-time OAS problems

Today, with the development of information and communication technologies, the orders in flow shops are becoming increasingly dynamic and customer-oriented, whereby customers can place orders with their desired compositions and due dates in real-time (Rahman et al., 2019). Several studies consider future order arrivals in order acceptance decision-making and define the related OAS as a dynamic problem (Xu et al., 2015). For example, Ebben et al. (2005) propose several workload-based order acceptance strategies with stochastic order arrivals; Arredondo and Martinez (2010) develop a novel approach for the online adaptation of a dynamic order acceptance policy, in which the average-reward reinforcement learning is used. A few studies investigate dynamic OAS problems with either a sequence-dependent setup time or cost. van Foreest et al. (2010) assume that the arriving orders follow an independent stationary Poisson process and use simulation to compare the performance of several heuristic scheduling and acceptance policies. Xu et al. (2015) formulate a stochastic dynamic programming model by considering the stochastic arrival orders and sequence-dependent setup time.

In terms of the real-time characteristic of r-OAS problems, Rahman et al. (2015) and Rahman et al. (2019) establish programming models for this problem in permutation flow shops and propose rule-based heuristic and particle swarm optimization algorithms, respectively. Eriksen and Nielsen (2016) propose another approach for the r-OAS problem by aggregating incoming customer order requests into a stable inflow. These proposed approaches focus on minimizing the job makespan and the weight tardiness penalty cost, while the inventory costs accrued during accepted order fulfillment are ignored (Yenisey and Yagmahan, 2014, Li et al., 2016). Moreover, these approaches to solving r-OAS problems in flow shops all assume that the intermediate buffers are infinite, and no blockage/starvation exists (Rossit et al., 2018). Because the throughput of a flow shop is determined by job release plans, WIP inventory, job processing time, and finite intermediate buffer constraints (Li et al., 2016), this assumption causes inaccuracies in the real-time estimation for job makespan in flow shops and may also prompt additional tardiness penalty costs when producing these accepted orders.

Obviously, changing too often the OAS affects other scheduled tasks like assignment of machine tools and delivery of raw material. These variations may increase nervousness of a shop floor, which should be taken into consideration from a practical point of view. As presented in previous literature, nervousness causes an increase of global cost, a reduction in productivity, and an increase in the bullwhip effect (Hayes and Clark, 1985, Herrera et al., 2016). In this situation, a company determines a trade-off considering production costs, quality of service, and schedule instability (Blackburn et al., 1986). For example, Framinan et al. (2019) address how rescheduling should be performed after analyzing the instantaneous and accurate information on shop-floor status. Azouz et al. (2018) discuss nervousness in the context of adaptive pull control systems. A new approach which relies both on an adaptive freezing interval and a multi-objective simulation optimization technique is proposed such that nervousness can be reduced. Fahmy et al. (2007) emphasize that the job insertion obtains revised schedules featuring significantly lower system nervousness and slightly higher mean flow time than total rescheduling. The increasing use of sensors, FRID, and networked machines exploit the interconnectivity among machines to fulfill the goal of producing intelligent, resilient, and self-adaptable. This intelligence makes real-time scheduling timelier and more crucial as nervousness can be efficiently reduced by the quick and smart response to abnormal events (Shiue et al., 2018, Zhang et al., 2018).

8.2.3 Data-driven OAS problems

Smart manufacturing anticipates a situation in which the shop floor status is instantly available after analyzing the real-time data (Chen, 2016). Many scholars focus on the potential use of such real-time data in developing data-driven approaches for order scheduling problems. In terms of the large amount of data during production, a few papers study data classification problem in scheduling. Framinan et al. (2019) present a classification distinguishing between a class of model-based data and another of instance-based data; Pinedo (2012) classifies scheduling data as static and dynamic. Based on these data classifications, (Rossit et al., 2018) propose a data-driven architecture for order scheduling to enable decisions to be made ahead of time. Zhong et al. (2015) use shop floor data to estimate the arrival of customer orders and standard operation times. Based on this estimation, real-time advanced production planning and scheduling can ultimately be achieved. The data-driven job shop scheduling and its new perspectives under Industry 4.0 are reviewed by Zhang et al. (2018).

In OAS decision-making, a reliable estimation of job makespan is critical. De Jong et al. (2017) develop a multilayer perceptron type neural network machine learning algorithm for quick and accurate job makespan prediction. They apply makespan prediction to a wide variety of shop scheduling problems. Because the current estimation approaches are limited to merely generalizing shop layout configurations and non-visual data input, a convolutional neural network algorithm for makespan regression is proposed by De Jong et al. (2019). To address the problem with order acceptance, Duan et al. (2015) develop a completion status prediction approach for new orders and then construct an automated learning-based order admission framework. In addition, considering the disruptions that may significantly affect the performance of production systems, Zou et al. (2017) develop an event-driven method for dynamic production system diagnosis and prognosis. Chen et al. (2020) formulate an event-driven estimation method for the job makespan under make-to-order production environments after addressing the system input and finite intermediate buffer constraints. This work is vital to developing an optimization model for our r-OASR problem.

8.3 r-OASR Problem Formulation and Preliminaries

This section defines the r-OASR problem and demonstrates the four hierarchical aspects driving decision-making for r-OASR: order acceptance, order scheduling, production status prediction, and job release. In terms of this interactive framework, three key technical challenges in modeling and solving the r-OASR problem are addressed.

8.3.1 Replanning policy for r-OASR

In traditional manufacturing supply chains, real-time orders are directly created by downstream customers (e.g., downstream manufacturers). In this situation, the quantity of real-time orders at any one moment in time is low. To decrease the nervousness of the production system, we collect real-time orders placed during the past time horizon t_0 and then update the OAS scheme accordingly. This rolling horizon-driven r-OASR is presented in Figure 8-1 (a). The length of each period is equal to t_0. Figure 8-1 (a) contains the new arrival orders collected during period $Q-1$, the unprocessed orders at the end of period

$Q-1$, and the real-time status of production systems that are inherited by period Q for r-OASR decision-making. However, in platform-driven manufacturing networks, the open platform collects real-time orders from many customers and recommends a list of orders for a production system. This recommendation is regarded as an event that drives the replanning of r-OASR. Because the quantity of recommended orders is sufficiently large in this case, the production system must immediately update its OAS scheme after receiving these new orders. Figure 8-1 (b) illustrates a platform that recommends a list of real-time orders to a production system at time t_{O-1}, t_O, and t_{O+1}, respectively. These events all immediately trigger r-OASR decision-making.

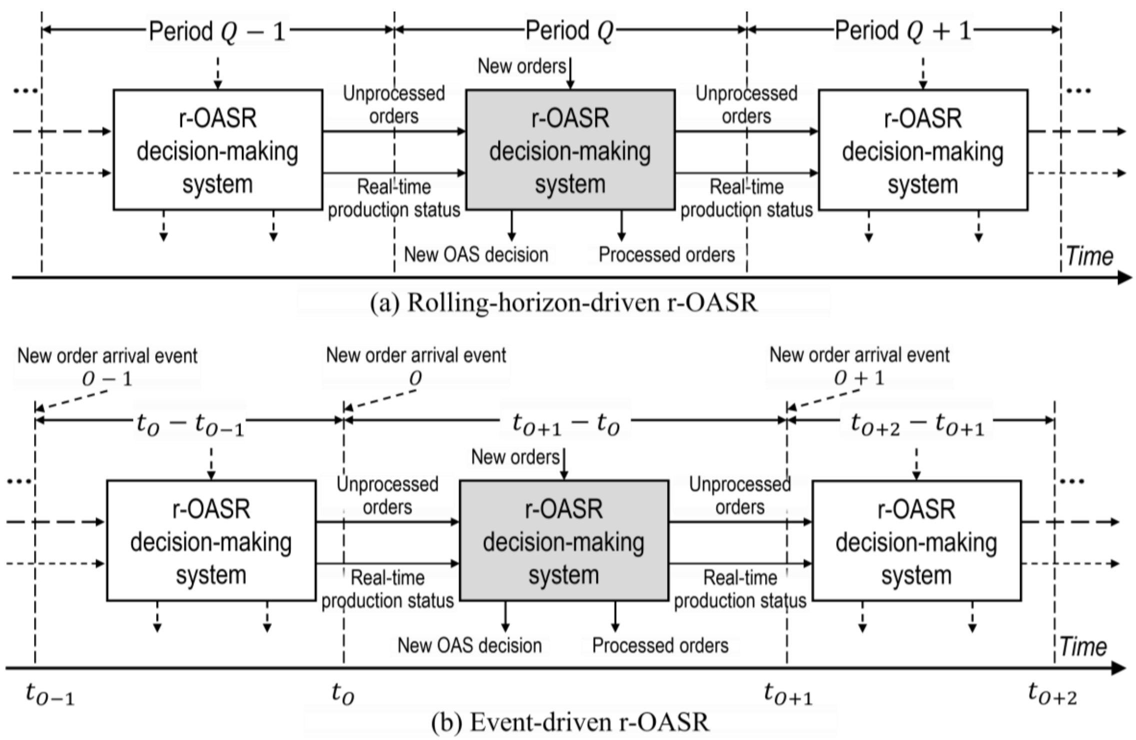


Figure 8-1 Replanning policies of r-OASR problem

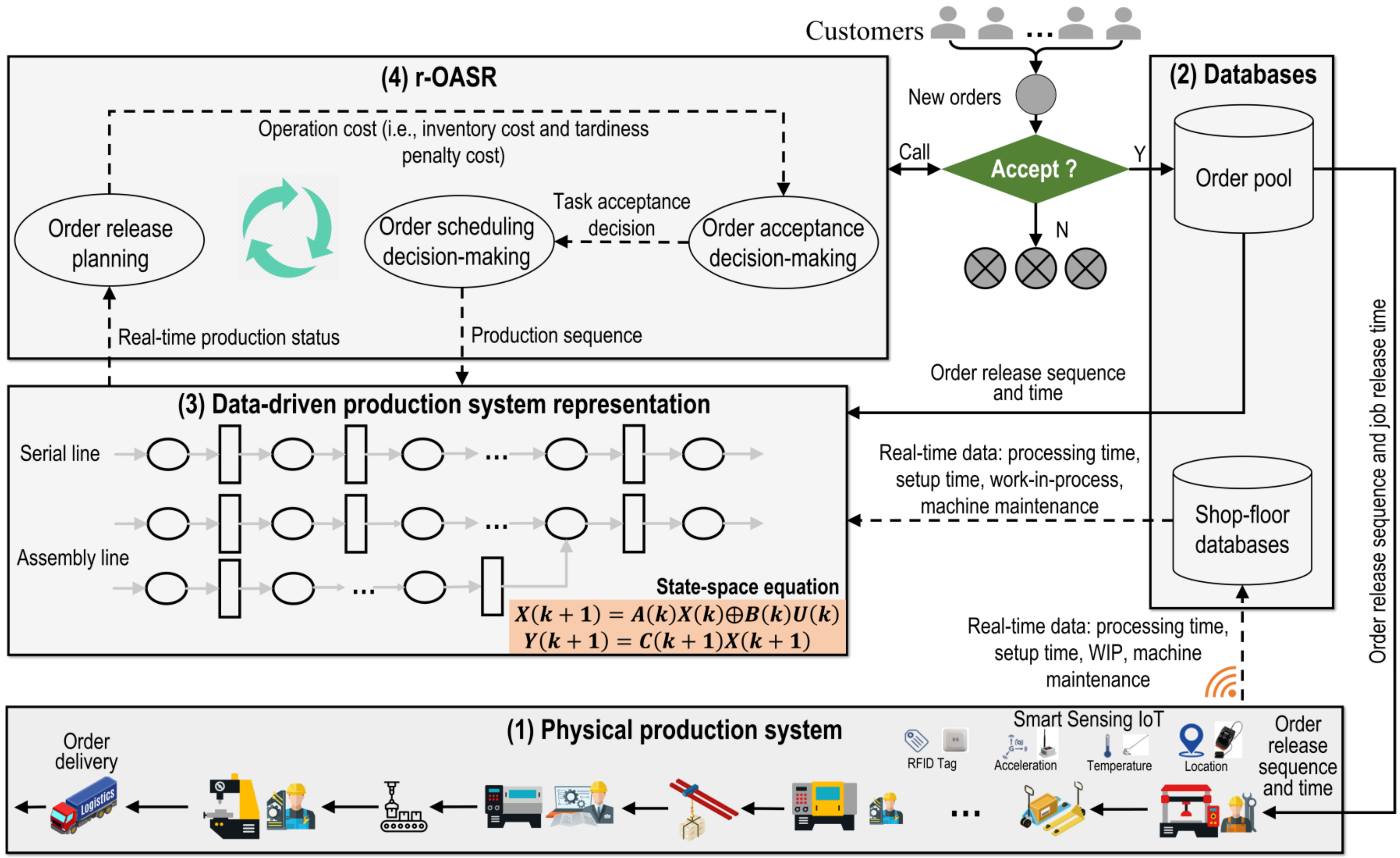


Figure 8-2 Interactive framework of r-OASR problems

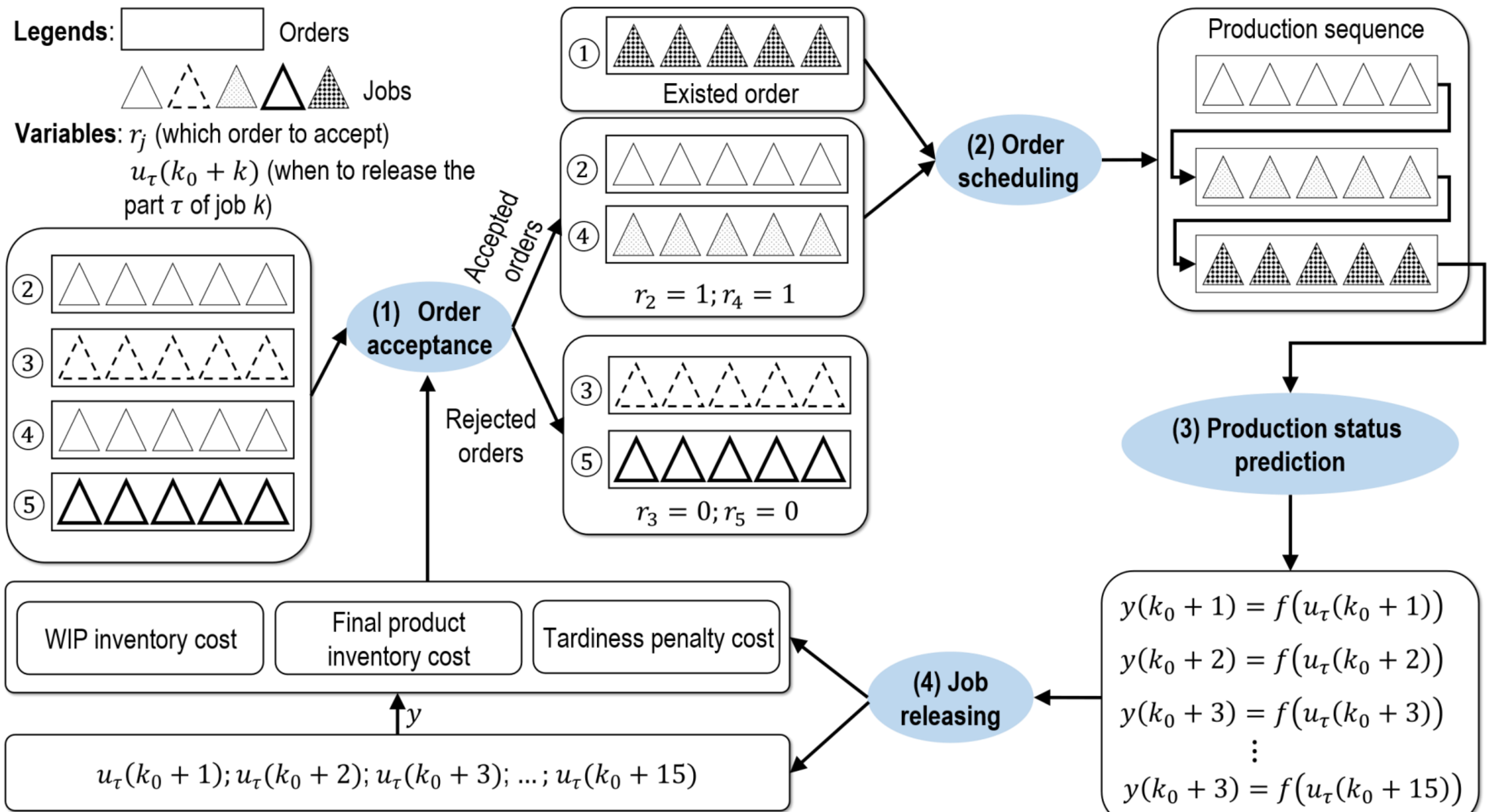


Figure 8-3 r-OASR instantiation with four new orders and one existing order

8.3.2 r-OASR Problem Definition

In terms of the rolling horizon-driven and event-driven mechanisms, the r-OASR problem can be described as follows: A flow line with $\mathcal{M}$ machines ($\mathcal{M} \geq 2$) and finite intermediate buffers receives a set of new generated orders Ω_1 ($|\Omega_1| \in N^+$) from customers during the past time horizon t_0 (or from an open platform at time t_O). The manager must decide which orders to accept or reject based on the real-time production status of this flow line. If some or all the new orders are accepted, they must be scheduled with the existing orders Ω_2 ($|\Omega_2| \in N^+$) that are waiting for processing. Each incoming order j ($j \in \Omega_1$) is identified with a job quantity q_j, earliest release date a_j, latest release date b_j, processing time p_{ij} for each job (i is the machine index $i = 1,2, \dots, \mathcal{M}$), due date d_j, revenue e_j, tardiness penalty coefficient $w_{1,j}$, unit inventory cost $w_{2,j}$ for a final product, and unit cost $w_{3,j}$ for WIP inventory. A sequence-dependent setup time $t_i(k, k')$ occurs if job k immediately precedes job k' in a production sequence. $t_i(k, k') = 0$ if no setup time exists in the machine i. Each machine can only process one job at a time, and any job can only be processed on a downstream machine after completing its processing on the current machine. If we denote the order tardiness penalty cost by $\pi_{1,j}$, the final product inventory cost by $\pi_{2,j}$, and the WIP inventory cost by $\pi_{3,j}$, the net revenue of an order can then be represented by $\pi_j \coloneqq e_j - \pi_{1,j} - \pi_{2,j} - \pi_{3,j}$. If order $j \in \Omega_1$ is accepted, $\pi_j > 0$. The r-OAS targets the generation of the best OAS plan to maximize the total net revenue. After receiving a production sequence from the r-OAS, the r-JRP begins to calculate the lowest cost $\pi_{1,j}$, $\pi_{2,j}$, and $\pi_{3,j}$ by addressing the real-time production status and constraints pertaining to job release. The lowest cost is sent back to the r-OAS for consideration in the decision-making process.

The serial line is a typical flow line production in industries such as automotive, electronics, appliance, and aerospace systems. The merge line comprises several parallel serial lines, in which each sub-serial line is responsible for processing/assembling a part or a component. This paper focuses on serial and merge lines to examine r-OASR problems. The r-OASR decision-making for other flow lines can be extended from our research results.

8.3.3 Hierarchical decision-making of r-OASR

Generally, real-time order acceptance must be accompanied by a scheduling problem, in which the order acceptance determines which order to accept and the scheduling generates a corresponding production sequence with the newly accepted and current existing orders to minimize the operation cost of flow lines. As stated in the introduction, the lowest operation cost of a determined scheduling plan is coordinated by r-JRP. Owing to its multi-stage dynamic transitions, finite intermediate buffers, and bottleneck shifting caused by multiple-product switching, the operation parameters, that is, the job completion time, WIP, and final product inventory of a job release plan, are difficult to measure (Li et al., 2016). To address this problem, a data-driven representation approach should be developed to reveal the status of flow lines and then predict the operational parameters of a job release plan. A programming model can then be formulated for r-JRP based on this prediction. Figure 8-2 represents the interactive framework of real-time order acceptance, scheduling, production status prediction, and job release. The production status prediction necessitates a sensor network to collect the real-time data of flow shops (e.g., job processing time and machine breakdown events) and a representative approach after analyzing these real-time data. After making the hierarchical decisions entailed in r-OASR, the accepted orders, scheduling plan, and job release plan are stored in a database and then released into the physical flow line and representation module in sequence. The flow line produces jobs according to the plan, and the representation module synchronously updates the production status based on real-time data from the shop floors.

The r-OASR instantiation with four new orders and one existing order is established in Figure 8-3. Here, one may observe that if the order acceptance stage (Stage 1) accepts orders 2 and 4, the order scheduling stage (Stage 2) will generate a production sequence that simultaneously considers existing order 1 and newly accepted orders 2 and 4. Based on this sequence, the function between the job makespan ($y(k_0 + k)$, $1 \le k \le 15$) and the release time of its parts $\left(u_\tau(k_0 + k), \dots, u_{P+1}(k_0 + k)\right)$, $P + 1$ is the quantity of parts in a job; the explanation in Section 8.4 can be derived from the data-driven representation (Stage 3). Parameter k_0 is the job number that is processed by machines when the OAS scheme must be updated. In this regard, the lowest operation cost of orders 1, 2, and 4 can be generated by r-JRP optimization and are sent back to Stage 1 to regenerate the order acceptance decisions. This cycle continues until the best solution with a maximum net revenue is found.

8.3.4 Critical Challenges

Focusing on the four decision-making stages of r-OASR, r-OAS decision can be indicated by several 0–1 variables and are easily formulated as a 0–1 programming model.

The technical challenges are then focused on determining how to develop a data-driven representation for physical flow line production, when to release the jobs such that operation costs can be minimized, and how to construct a programming model to demonstrate the multi-stage decisions of r-OASR.

1) ***Data-driven representation for flow line production***: Flow line production is a typical discrete event dynamic system, in which planned deterministic events (e.g., changeover events and job movement between adjacent machines) and uncertain resource/job-related events are contained (Ouelhadj and Petrovic, 2009, Chen et al., 2020). The modeling and analysis of the time-varying transitions of these discrete events using state-space models are of primary importance to revealing the real-time production status of flow lines and predicting the corresponding job makespan with finite intermediate buffer constraints (Ivanov et al., 2012). The prevailing model-based representation (e.g., continuous flow models and Markov models) treats disruptions as noise or possible states and assume that the modes of operation can never be switched (i.e., only one type of product is processed) (Zou et al., 2017, Farahani et al., 2017, Wang and Ju, 2020). These methods inefficiently respond to real-time events, especially in reaction to the frequent switching of products in make-to-order environments. DES is another method that can efficiently represent mixed-model production with real-time data (Jung et al., 2020). However, the development of a simulation model generally depends on the operational logic of a complex system and a professional simulation platform. It should be difficult and time-consuming to encapsulate a simulation model into r-OAS for decision-making. Thus, it is necessary to explicitly propose a representation formalism that is capable of supporting the event-driven propagation of time-varying state-space models among various dynamic

behaviors. This formalism should be a mathematical model and can be easily encapsulated into upper-level programming models.

2) ***The best job release plan for coordinating multiple operation costs***: The operational costs of an r-OAS scheme can be coordinated by a job release plan developed from the data-driven representation (De Schutter and Van Den Boom, 2001, Chen et al., 2020). Conventional wisdom concentrates on maximizing the number of accepted orders after addressing order due date constraints, while ignoring inherent multiple operation costs during order fulfillment (Slotnick, 2011). Thanks to the real-time data collected from shop floors, the multiple operation costs can be concisely predicted and thus an r-JRP can be developed. Unlike previous research, which assumes that intermediate buffers are infinite, and thus the job competition time between two adjacent machines can be represented as a linear function (Silva et al., 2018), the r-JRP problem must manage the finite buffer constraints in flow lines, which converts the dynamic job transitions among machines into nonlinear types (Chen et al., 2020). These characteristics cause previous programming models and heuristic/meta-heuristic algorithms to be inefficient in solving our r-OASR problem.

3) ***Interactive optimization for order acceptance, scheduling, and job release***: r-OASR can be decomposed into an r-OAS problem and an r-JRP problem, where r-OAS formulates the best OAS scheme and r-JRP results in the corresponding minimal operation costs. The r-OAS and r-JRP are closely interdependent because r-OAS generates a solution space for r-JRP, which generates feedback for an objective function of cost for the r-OAS. Thus far, the r-OAS and r-JRP have been addressed separately in the literature, although they are closely interrelated (Slotnick, 2011). In terms of solving a joint optimization, the

prevailing approaches tend to treat these subproblems as an aggregate and assume that self-interested decision makers can always be coordinated (Du et al., 2019b). However, in r-OASR, r-OAS and r-JRP belong to different decision levels and hold conflicting goals regarding maximizing/minimizing their own benefit. In this sense, such joint optimization exemplifies a leader–follower decision-making process, whereby r-OAS occurs first and r-JRP subsequently responds to r-OAS decisions.

8.4 Data-driven Representation with Max-Plus Algebra Framework

The representation approach is vital in predicting the job makespan after receiving r-OAS decisions. This section elaborates the modeling procedures for flow lines by addressing the finite intermediate buffer constraints and analyzing real-time data from shop floors. Max-plus algebra can effectively represent flow shops owing to its ability to derive time-varying state-space models of discrete event dynamic systems. This approach is based on a dioid $\mathbb{R}_{max}$ represented by the structure $(\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$. $\oplus$ and $\otimes$ signify maximization and addition, respectively: $a \oplus b = \max(a, b)$ and $a \otimes b = ab = a + b$. ε denotes the zero element ($\varepsilon = -\infty$), and e denotes the unit element ($e = 0$). The matrix $\mathcal{E}_{m\times n}$ is a $m \times n$ max-plus algebraic zero matrix $[\mathcal{E}_{m\times n}]_{ij} = \varepsilon$ for all $i,\ j$. The E_n is a $n \times n$ max-plus algebraic identity matrix: $[E_n]_{ii} = e$ for all i and $[E_n]_{ij} = \varepsilon$ for all $i,\ j\ (i \neq j)$. If $A, B \in \mathbb{R}_{max}^{m\times n}$ and $C \in \mathbb{R}_{max}^{n\times p}$, then for all $i,\ j$: $(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$, and $(A \otimes C)_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj}$. Moreover, the max-plus algebraic matrix power of $A \in \mathbb{R}_{max}^{m\times n}$ is defined as follows: $A^0 = E_n$ and $A^k = A \otimes A^{k-1}$ for $k = 1,2, \dots$

The transition network is a timed event graph (TEG) of a production line (Baccelli and Schmidt, 1996). Before developing the max-plus based representation for a flow line, the transition network should first be determined to describe the job dynamic transition behaviors. Compared with some advanced DES that perform evaluation functions after receiving r-OASR decisions, our representation formulates the transition network as a state-space model and can therefore be easily incorporated into optimization models.

8.4.1 *Time-Varying State-Space Model of Serial Production Lines*

The serial production line is a time-varying system for multiple-product processing. The TEG can effectively describe this mixed-model production for its ability to iconify and model time-varying transitions. Figure 8-4 shows the time event graph of a serial line. The $\{P_0, \ldots, P_{\mathcal{M}-1}, P'_1, \ldots, P'_{\mathcal{M}-1}, P''_1, \ldots, P''_{\mathcal{M}}\}$ in Figure 8-4 provides the set of places, and $\{u, x_1, \ldots, x_M\}$ is the set of transitions. Transition x_i represents machine i ($i = 1, \ldots, \mathcal{M}$), place P_i represents buffer B_i ($1 \leq i \leq \mathcal{M} - 1$), place P'_i controls the real-time WIP in buffer B_i, and place P''_i ($1 \leq i \leq \mathcal{M}$) controls the restart of the corresponding machine. The $u(k)$ is the instant at which the k-th job is fed into the system. The tokens in place P_i signify the real-time WIP, and the tokens in place P'_i signify the residual capacity of buffer B_i. Let $N_{max} = \max\{N_1, N_2, \ldots, N_{M-1}\}$, where N_i is the capacity of buffer B_i and $x_i(k)$ ($1 \leq i \leq M$ and $1 \leq k$) is the time instant at which the machine m_i begins working on the k-th job. The buffer B_0 is infinite. The time-varying transition rules of the serial production line can then be represented by equation (8.1):

$$x_i(k) =$$
$$x_i(k-1) \otimes \sigma_i(k-1) \oplus x_{i+1}(k - N_{i+1}) \otimes \sigma_{i+1}(k - N_{i+1}) \oplus x_{i-1}(k) \otimes \sigma_{i-1}(k)$$

$$2 \leq i \leq M - 1, \tag{8.1.1}$$

$$x_1(k) = u(k) \oplus x_1(k-1) \otimes \sigma_1(k-1) \oplus x_2(k-N_2) \otimes \sigma_2(k-N_2), \tag{8.1.2}$$

$$x_M(k) = x_M(k-1) \otimes \sigma_M(k-1) \oplus x_{M-1}(k) \otimes \sigma_{M-1}(k), k \geq N_{max} \tag{8.1.3}$$

where $\sigma_i(k)$ is the job processing time of machine i for the k-th job and consists of both pure processing and setup times. $\sigma_i(k)$ is determined by a given production sequence (see Figure 8-4). According to this time-varying equation, if the first-input-first-output principle is obtained, the discrete-event state-space model of this system can be derived as follows (Chen et al., 2020):

$$\overline{\boldsymbol{X}}(k) = \overline{\boldsymbol{A}}(k-1) \otimes \overline{\boldsymbol{X}}(k-1) \oplus \overline{\boldsymbol{B}}(k) \otimes u(k), \tag{8.2.1}$$

$$y(k) = \overline{\boldsymbol{C}}(k) \otimes \overline{\boldsymbol{X}}(k),\ N_{max} - 1 \leq k, \tag{8.2.2}$$

where $\overline{\boldsymbol{X}}(k-1) = [\boldsymbol{X}^T(k-1), \boldsymbol{X}^T(k-2), \ldots, \boldsymbol{X}^T(k-N_{max})]^T$. The matrices, $\overline{\boldsymbol{A}}(k)$, $\overline{\boldsymbol{B}}(k)$ and $\overline{\boldsymbol{C}}(k)$ are represented by equation (8.3). $\overline{\boldsymbol{A}}(k)$ is a $MN_{max} \times MN_{max}$ matrix, E is a $M \times M$ (max, +)-identity matrix, $\overline{\boldsymbol{B}}(k)$ is a MN_{max} matrix, and $\overline{\boldsymbol{C}}(k)$ is a $1 \times MN_{max}$ matrix. Set $\boldsymbol{A}_0^*(k) = \bigoplus_{s' \geq 0} \boldsymbol{A}_0^{s'}(k)$ ($\boldsymbol{A}_0^*(k)$ is converged because $\boldsymbol{A}_0(k)$ is a strictly lower triangular), the matrix $\overline{\boldsymbol{A}}_1(k), \ldots, \overline{\boldsymbol{A}}_{N_{max}}(k)$, and $\overline{\boldsymbol{B}}_0(k)$ in equation (8.3) can be generated by $\overline{\boldsymbol{A}}_{s''}(k) = \boldsymbol{A}_0^*(k) \otimes \boldsymbol{A}_{s''}(k)$, $(0 < s'' \leq N_{max}$ and $\overline{\boldsymbol{B}}_0(k) = \boldsymbol{A}_0^*(k) \otimes \boldsymbol{B}(k)$. The element $[A_{s''}(k)]_{i'i}$ is the firing time $\sigma_i(k - s'')$ of the machine i if the buffer capacity $N_i = s''$ and there is a transition path from machine i to i', as shown in Figure 8-4; otherwise, $[\boldsymbol{A}_{s''}(k)]_{i'i}$ is equal to ε. The element $[\boldsymbol{B}(k)]_{i1}$ is equal to 0 if there is a system input to machine i and $[C(k)]_{1i}$ is equal to the firing time $\sigma_i(k)$ of machine i if there is a system output to this machine; otherwise, the two elements are equal to ε.

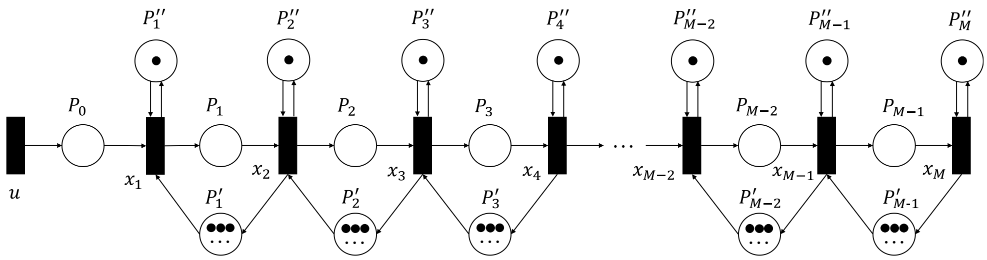


Figure 8-4 Time event graph of serial production lines

$$\overline{\boldsymbol{A}}(k-1) = \begin{bmatrix} \overline{A}_1(k) & \overline{A}_1(k) & \cdots & \cdots & \overline{A}_{N_{max}}(k) \\ \boldsymbol{E} & \mathcal{E} & \cdots & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & \boldsymbol{E} & \ddots & \mathcal{E} & \mathcal{E} \\ \vdots & \vdots & \cdots & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & \mathcal{E} & \mathcal{E} & \boldsymbol{E} & \mathcal{E} \end{bmatrix}, \tag{8.3.1}$$

$$\overline{\boldsymbol{B}}(k) = [\overline{\boldsymbol{B}}_0{}^T(k),\ \varepsilon,\ \ldots,\ \varepsilon]^T, \tag{8.3.2}$$

$$\overline{\boldsymbol{C}}(k) = [\boldsymbol{C}(k),\ \varepsilon,\ \ldots,\ \varepsilon], \tag{8.3.3}$$

8.4.2 Time-Varying State-Space Model of Merge Production Lines

A merge production line can be decomposed into P sub-serial lines and a main-serial line. We denote the machine quantity of the τ-th sub-line by M_τ ($\tau = 1, \ldots, P$) and the machine quantity of the main-line by M_{P+1}.The machines are numbered in each sub-line by $\sum_{\tau'=1}^{\tau-1} M_{\tau'} + h_1$, $h_1 = 1, \ldots, M_\tau$, and the machine in the main-line by $\sum_{\tau=1}^{P} M_\tau + h_2 h_2 = 1, \ldots, M_{P+1}$ ($\sum_{\tau'=1}^{0} M_{\tau'} = 0$). We number the buffers in each sub-line by $\sum_{\tau'=1}^{\tau-1} M_{\tau'} + h_1$, $h_1 = 1, \ldots, M_\tau$, and the buffer in the main line by $\sum_{\tau=1}^{P} M_\tau + h_2$, $h_2 = 1, \ldots, M_{P+1} - 1$. This paper focuses on the assembly merge, which merges stations from all upstream branches simultaneously and assembles them into a single job (Liu and Li, 2010). Figure 8-5 shows the TEG of a merge production line. Except for the merge machine and their predecessor machines, the time-varying transition rule of each machine in Figure

8-5 can be described by equation (8.1). The merge machine in the main line is set as m_τ. For each machine $\sum_{\tau'=1}^{\tau} M_{\tau'}$, the k-th time-varying transition is determined by the k-th transition of the machine $\sum_{\tau'=1}^{\tau} M_{\tau'} - 1$, the $(k-1)$-th transition of machine $\sum_{\tau'=1}^{\tau} M_{\tau'}$, and the $(k - N_{\sum_{\tau'=1}^{\tau} M_{\tau'}})$-th transition of the merge machine m_τ, as shown in equation (8.4.1). For each merge machine m_τ, the k-th time-varying transition is determined by the k-th transition of its upstream machine $m_\tau - 1$ and $\sum_{\tau'=1}^{\tau} M_{\tau'}$, the $(k-1)$-th transition of machine m_τ, and $(k - N_{m_\tau+1})$-th transition of machine $m_\tau + 1$, as shown in equation (8.4.2).

$$x_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k) = \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}-1}(k) x_{\sum_{\tau'=1}^{\tau} M_{\tau'}-1}(k) \oplus \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k-1) x_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k-1) \oplus \sigma_{m_\tau}\left(k - N_{\sum_{\tau'=1}^{\tau} M_{\tau'}}\right) x_{m_\tau}\left(k - N_{\sum_{\tau'=1}^{\tau} M_{\tau'}}\right) \tag{8.4.1}$$

$$x_{m_\tau}(k) = \sigma_{m_\tau}(k-1) x_{m_\tau}(k-1) \oplus \sigma_{m_\tau-1}(k) x_{m_\tau-1}(k) \oplus \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k) x_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k) \oplus \sigma_{m_\tau+1}(k - N_{m_\tau+1}) x_{m_\tau+1}(k - N_{m_\tau+1}) \tag{8.4.2}$$

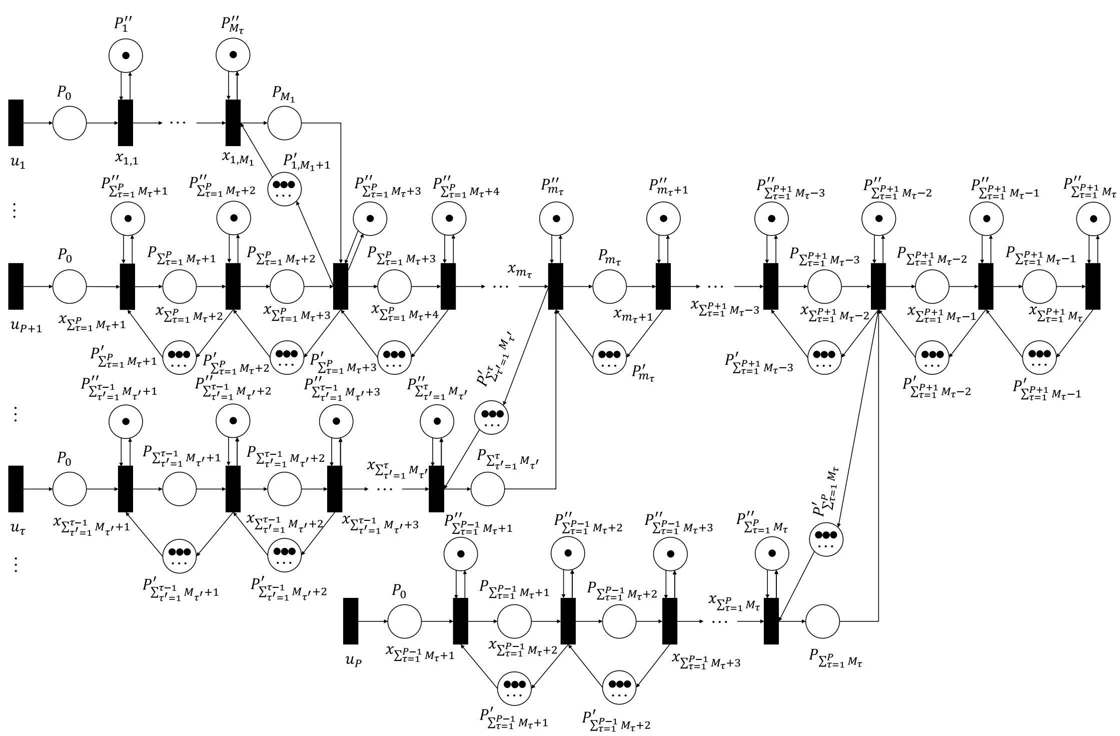


Figure 8-5 Time event graph of merge production lines

If the merge action is not considered, the merge production line can be divided into $P+1$ parallel serial lines. According to transition equation (8.4), the standard state-space equation for the $P+1$ parallel lines can be represented as:

$$\begin{bmatrix} \overline{\boldsymbol{X}}_1(k) \\ \overline{\boldsymbol{X}}_2(k) \\ \vdots \\ \overline{\boldsymbol{X}}_P(k) \\ \overline{\boldsymbol{X}}_{P+1}(k) \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{A}}_1(k-1) & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \overline{\boldsymbol{A}}_2(k-1) & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \ddots & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \overline{\boldsymbol{A}}_P(k-1) & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \overline{\boldsymbol{A}}_{P+1}(k-1) \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{X}}_1(k-1) \\ \overline{\boldsymbol{X}}_2(k-1) \\ \vdots \\ \overline{\boldsymbol{X}}_P(k-1) \\ \overline{\boldsymbol{X}}_{P+1}(k-1) \end{bmatrix}$$

$$\oplus \begin{bmatrix} \overline{\boldsymbol{B}}_1(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \overline{\boldsymbol{B}}_2(k) & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \ddots & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \overline{\boldsymbol{B}}_P(k) & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \overline{\boldsymbol{B}}_{P+1}(k) \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_P(k) \\ u_{P+1}(k) \end{bmatrix}, \tag{8.5}$$

where $\overline{\boldsymbol{X}}_\tau(k) = [\boldsymbol{X}_\tau^T(k-1),\ \boldsymbol{X}_\tau^T(k-2),\ \dots,\ \boldsymbol{X}_\tau^T(k-N_\tau^{max})]^T$ ($\tau = 1, \dots, P+1$), $\boldsymbol{X}_\tau(k-\tau') = \left[x_{\sum_{\tau'=1}^{\tau-1} M_{\tau'}+1}(k-\tau'), x_{\sum_{\tau'=1}^{\tau-1} M_{\tau'}+2}(k-\tau'), \dots, x_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k-\tau')\right]^T$ ($\tau' \in 1, \dots, N_\tau^{max}$), where N_τ^{max} is the maximum buffer capacity of each parallel line and $N_{P+1}^{max} = \max\left\{N_{\sum_{\tau=1}^{P} M_\tau+1}, \dots, N_{\sum_{\tau=1}^{P} M_\tau+M_{P+1}}, \dots, N_{\sum_{\tau=1}^{P} M_\tau}\right\}$. $\bar{A}_\tau(k) \in \mathbb{R}^{(M_\tau N_\tau^{max})\times(M_\tau N_\tau^{max})}$, $\bar{B}_\tau(k) = \mathbb{R}^{(M_\tau N_\tau^{max})\times 1}$. A state-space equation can be simplified from equation (8.6) as follows:

$$\overline{\boldsymbol{X}}(k) = \overline{\boldsymbol{A}}(k-1)\overline{\boldsymbol{X}}(k-1) \oplus \overline{\boldsymbol{B}}(k)\boldsymbol{U}(k) \tag{8.6.1}$$

$$y(k) = \overline{\boldsymbol{C}}(k) \otimes \overline{\boldsymbol{X}}(k), N_{max} - 1 \le k \tag{8.6.2}$$

where $\overline{\boldsymbol{X}}(k) = [\boldsymbol{X}_1^T(k), \boldsymbol{X}_2^T(k), \dots, \boldsymbol{X}_{P+1}^T(k)]^T$ and $\boldsymbol{U}(k) = [u_1(k), \dots, u_{P+1}(k)]^T$. The $(\sum_{\tau=1}^{P} M_\tau N_\tau^{max} + M_{P+1})$-th element of matrix $\overline{\boldsymbol{C}}(k)$ is equal to $\sigma_{\sum_{\tau=1}^{P+1} M_\tau}(k)$ and the other elements are ε. In terms of the merge actions, several extra elements should be added to the state-transition matrix $\overline{\boldsymbol{A}}(k)$. Considering the state transition of the machine $\sum_{\tau'=1}^{\tau} M_{\tau'}, \tau \in \{1, \dots, P\}$ the elements of matrix $\overline{\boldsymbol{A}}(k)$ can be transferred to the following:

$$[\overline{\boldsymbol{A}}(k)]_{\left(M_\tau+\sum_{\tau'=1}^{\tau-1} M_{\tau'} N_{\tau'}^{max}\right)\left(\sum_{\tau'=1}^{\tau-1} M_{\tau'} N_{\tau'}^{max}+M_{P+1}\left(N_{\sum_{\tau'=1}^{\tau} M_{\tau'}}-1\right)+m_\tau-\sum_{\tau'=1}^{P} M_{\tau'}\right)}$$
$$= \sigma_{m_\tau}\left(k - N_{\sum_{\tau'=1}^{\tau} M_{\tau'}}\right). \tag{8.7}$$

Each sub-line τ affects the transition of machines from m_τ to $\sum_{\tau=1}^{P} M_\tau$. This merge action changes the elements of the matrix $\overline{\boldsymbol{A}}(k)$ and $\overline{\boldsymbol{B}}(k)$, and can be represented as equation (8.8):

$$[\overline{\boldsymbol{A}}(k)]_{\left(\sum_{\tau=1}^{P} M_\tau N_\tau^{max}+m_\tau-\sum_{\tau'=1}^{P} M_{\tau'}:\sum_{\tau=1}^{P} M_\tau N_\tau^{max}+M_{P+1}\right)\left(\sum_{\tau'=1}^{\tau-1} M_{\tau'} N_{\tau'}^{max}+1:\sum_{\tau'=1}^{\tau} M_{\tau'} N_{\tau'}^{max}\right)}$$
$$= \begin{bmatrix} \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1) \\ \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1)\sigma_{m_\tau}(k+1) \\ \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1)\sigma_{m_\tau(m_\tau+1)}(k+1) \\ \vdots \\ \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1)\sigma_{m_\tau \sum_{\tau=1}^{P+1} M_\tau}(k+1) \end{bmatrix} \otimes [\bar{A}_\tau(k)]_{(M_\tau)(:)} \tag{8.8.1}$$

$$[\overline{\boldsymbol{B}}(k)]_{\left(\sum_{\tau=1}^{P} M_\tau N_\tau^{max}+m_\tau-\sum_{\tau'=1}^{P} M_{\tau'}:\sum_{\tau=1}^{P} M_\tau N_\tau^{max}+M_{P+1}\right)\tau}$$
$$= \begin{bmatrix} \sigma_{M_j,j}(k+1) \\ \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1)\sigma_{m_\tau}(k+1) \\ \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1)\sigma_{m_\tau(m_\tau+1)}(k+1) \\ \vdots \\ \sigma_{\sum_{\tau'=1}^{\tau} M_{\tau'}}(k+1)\sigma_{m_\tau\left(\sum_{\tau=1}^{P+1} M_\tau\right)}(k+1) \end{bmatrix} \otimes [\bar{B}_\tau(k)]_{(M_\tau)(:)}, \qquad 1 \le \tau \le P \tag{8.8.2}$$

where $[\boldsymbol{A}]_{i(:)}$ is the i-th row elements of matrix $\boldsymbol{A}$ and $\sigma_{vv'}(s+1) = \sigma_v(s+1) \dots \sigma_{v'}(s+1), v < v'$. After updating these elements, the state-space equation (8.6) can be used to derive the real-time status for the merge production lines.

8.4.3 Event-Driven Switch of State-Transition Matrixes

The production status of the flow lines is switched with the disruptions of the shop floors, such as order cancelation, machine breakdown, process quality problems, and material shortage events, among others. These events were collected via RFID techniques and sensor networks. The state-space equation can represent the events by updating the matrices $\overline{\boldsymbol{A}}(k)$, $\overline{\boldsymbol{B}}(k)$, and $\overline{\boldsymbol{C}}(k)$. After modeling the perturbation caused by different disruptions, the accurate status of flow lines can be derived, and an r-OASR scheme with high enforceability can then be determined. We define the n-th disturbing event as $\vec{e}_n = (i, k_n, d_n)$, indicating that the event can last d_n time when the machine i produces the k_n-th job. The processing time of the machine i to the k_n-th job should be transferred to $\sigma_i'(k_n) = \sigma_i(k_n) + d_n$. In this regard, the state-transition matrix, $\overline{\boldsymbol{A}}(k_n)$, $\overline{\boldsymbol{B}}(k_n)$ and $\overline{\boldsymbol{C}}(k_n)$ can be updated with new $\sigma_i'(k_n)$. If a disruption that changes the production sequence occurs, the parameter $\sigma_i(k)$ can be updated with the new production sequence and the matrix $\overline{\boldsymbol{A}}(k)$, $\overline{\boldsymbol{B}}(k)$, and $\overline{\boldsymbol{C}}(k)$ can then be switched.

8.5 Bilevel Interactive Optimization

The single max-plus-based representation shows ineffective in obtaining an optimum r-OASR decision as various constraints and objectives about order acceptance and job releasing should be simultaneously addressed. In this regard, a mathematic programming model must be developed with this representation (van den Boom et al., 2020). This section derives a nonlinear integer programming for the BIO of r-OASR based on the data-driven representation in Section 8.4. In Section 8.5.1, a 0–1 integer programming model is developed for the leader r-OAS after addressing several net revenue and logical constraints. Section 8.5.2 formulates a nonlinear integer programming model for the follower r-JRP by considering the dynamic transition constraints generated from the data-driven representation and the existing job release constraints. Bilevel programming is introduced in Section 8.5.3 to address the interactive optimization of r-OASR.

8.5.1 0-1 Integer Programming for r-OAS Problem

The order acceptance decision is of critical importance for the revenue creation of manufacturers. For each order $j, j \in \Omega_1$, a binary decision variable r_j is used to indicate whether the order is accepted or rejected: $r_j = 1$ if order j is accepted; otherwise, $r_j = 0$. If we denote the completion time of the k'-th job in order j by $C_{j,k'}$,$k' = 1, \dots, q_j$. The order completion time C_j is then equal to C_{j,q_j}. For each order, if $C_j > d_j$, a tardiness penalty cost $\pi_{1,j}$ exists, which is represented by $\pi_{1,j} \coloneqq w_{1,j}(C_j - d_j)$. Because the finished jobs must wait at the stock until all jobs of an order are completed, an inventory cost $\pi_{2,j}$ should be included in this situation. The cost $\pi_{2,j}$ can be represented as $\pi_{2,j}:= \sum_{k'=1}^{q_j} w_{2,j}(C_j - C_{j,k'})$. Apart from $\pi_{1,j}$ and $\pi_{2,j}$, an extra operation cost $\pi_{3,j}$ of a flow line (i.e., the WIP inventory cost in our r-OAS problem) also exists during the order fulfillment procedures. This cost $\pi_{3,j}$ can be represented as $\pi_{3,j}:= \sum_{\tau=1}^{P+1} \sum_{k'=1}^{q_j} w_{3,j}(C_{j,k'} - u_{j,k',\tau})$, where $u_{j,k',\tau}$ is the best release time of the τ-th part of the k'-th job in order j. If $C_j \leq d_j$, no tardiness penalty cost exists. The finished jobs of order j must wait at stock until the delivery time d_j is achieved. In this regard, the inventory cost $\pi_{2,j}$ can be represented as $\pi_{2,j}:= \sum_{k'=1}^{q_j} w_{2,j}(d_j - C_{j,k'})$Based on the above analysis, the total cost π_j to fulfill order j can be determined by equation (8.9) as follows:

$$\pi_j = \pi_{1,j} + \pi_{2,j} + \pi_{3,j}$$

$$= w_{1,j} \max\left\{0, C_{j,q_j} - d_j\right\}$$

$$+ \sum_{k'=1}^{q_j} w_{2,j}\left(\max\{C_j, d_j\} - C_{j,k'}\right) + \sum_{\tau=1}^{P+1} \sum_{k'=1}^{q_j} w_{3,j}\left(C_{j,k'} - u_{j,k',\tau}\right), j \in \Omega_1 \qquad (8.9)$$

If a new order j is accepted, $e_j - \pi_j \geq 0$. Thus, the objective of the r-OAS problem, which aims to maximize the total net revenue, can be formulated by equation (8.10) as follows:

$$F = \sum_{j=1}^{|\Omega_1|} r_j\left(e_j - \pi_{1,j} - \pi_{2,j} - \pi_{3,j}\right) + \sum_{j=1}^{|\Omega_2|}\left(e_j - \pi_{1,j} - \pi_{2,j} - \pi_{3,j}\right), \qquad (8.10)$$

where $\sum_{j=1}^{|\Omega_2|}(e_j - \pi_j)$ is the total net revenue of existing orders. The existing orders cannot be rejected and $e_j - \pi_j \geq 0 \;\; j \in \Omega_2$. To formulate the r-OAS problem, the following assumptions are made:

1) The orders arrive in real-time, and the manufacturer has no prior information regarding these orders;

2) Each order contains only one kind of job;

3) The shop floor has the right to accept or reject any incoming orders;

4) An order in process cannot be interrupted;

5) Once an order is accepted, it cannot be rejected later (This assumption states that our order acceptance scheme cannot be affected by resource-related disruptions, e.g.,

machine breakdown or process quality problems.) (Rahman et al., 2015);

6) The setup time for an order on machines cannot be negligible;

7) Each order should be delivered just-in-time.

The order scheduling problem aims to determine the best production sequence for job release. This scheduling problem can be formulated as a dispatch problem, in which a binary decision variable $z_{j,l}$ is defined as indicating whether order j, $j \in \Omega_1 \cup \Omega_2$, is assigned to position l, $l \in \{1,2,\dots,|\Omega|\}, \Omega = \Omega_1 \cup \Omega_2$, or not: $z_{j,l} = 1$ if order j is assigned to position l; otherwise, $z_{j,l} = 0$. Two types of constraints exist in the scheduling problem: each position should be assigned a maximum of one order and the accepted order j, $j \in \Omega_1 \cup \Omega_2$, must be assigned to a position; see equation (8.11) as follows:

$$\sum_{j=1}^{|\Omega|} z_{j,l} \leq 1\,, j \in \Omega_1 \cup \Omega_2 \text{ and } l \in \{1,\dots,|\Omega|\}, \tag{8.11.1}$$

$$\sum_{l=1}^{|\Omega|} z_{j,l} = r_j\,, j \in \Omega_1, \tag{8.11.2}$$

$$\sum_{l=1}^{|\Omega|} z_{j,l} = 1\,, j \in \Omega_2, \tag{8.11.3}$$

where constraints (8.11.3) ensure that the existing orders cannot be rejected. If the variable $z_{j,l}$ is determined, the real-time production sequence Φ ($|\Phi| = \sum_{j=1}^{|\Omega_1|} r_j q_j + \sum_{j=1}^{|\Omega_2|} q_j$) is generated, as the following example.

Assume that five real-time orders (i.e., $\Omega_1 = \{4,5,6,7,8\}$) arrive at the current production period and each order has five jobs. There are three (i.e., $i = 1,2,3$) existing orders with five jobs each. If Orders 1, 2, 3, 4, 5, 7, and 8 are accepted and they are dispatched to Positions 2, 1, 4, 3, 6, 8, and 7, respectively (i.e., $z_{1,2} = 1, z_{2,1} = 1, z_{3,4} = 1$, $z_{4,3} = 1$, $z_{5,6} = 1$, $z_{7,8} = 1$, and $z_{8,7} = 1$), the new production sequence should be $\overbrace{1\ 2\ 3\ 4\ 5}^{\text{Order 2}}\ \overbrace{1\ 2\ 3\ 4\ 5}^{\text{Order 1}}\ \overbrace{1\ 2\ 3\ 4\ 5}^{\text{Order 4}}\ \overbrace{1,2,3,4,5}^{\text{Order 3}}\ \overbrace{[\ ]}^{\text{Order 6}}\ \overbrace{1,2,3,4,5}^{\text{Order 5}}\ \overbrace{1,2,3,4,5}^{\text{Order 8}}\ \overbrace{1,2,3,4,5}^{\text{Order 7}}$. This means that Job 1 in Order 2 is produced first, and then followed by Jobs 2, 3, and so on. Order 6 is rejected and Position 5 in this sequence is therefore empty.

8.5.2 A Nonlinear Integer Programming for r-JRP Problem

The r-JRP searches for the best job release plan to coordinate the three kinds of operation costs after predicting the system dynamic status (Chen et al., 2020). Assume the real-time production status of a flow line be $\overline{\boldsymbol{X}}(k_0)$, where k_0 means the k_0-th job that is processed by machines when new orders arrive. This r-JRP can then be defined as follows: we find the best release plan $\{u_\tau(k + k_0)\}$ for jobs, where $u_\tau(k + k_0)$ is an integer and $1 \leq k \leq |\Phi|$, based on the real-time status of flow lines at step k_0 to minimize the WIP inventory cost $\sum_{j=1}^{|\Omega|} \pi_{3,j}$, the final product inventory cost $\sum_{j=1}^{|\Omega|} \pi_{2,j}$, and the tardiness penalty cost $\sum_{j=1}^{|\Omega|} \pi_{1,j}$, thus subjecting them to job release time and state-space equation constraints. Parameter k denotes the k-th job in the newly generated production sequence, which it receives from the leader r-OAS problem. According to equation (8.10) and state-space equations (8.2) and (8.6), these costs can be calculated using equation (8.12). WIP minimization can be achieved by minimizing the gap between the system output and its corresponding release time, as shown in equation (8.12.1). The modeling framework of this r-JRP can be represented by an IDEF0 model: the input of this r-JRP is the order due date d_j, $j = 1, \ldots, |\Omega|$, order acceptance decision r_j, $j \in \Omega_1$, and a job release sequence Φ, $|\Phi| = \sum_{j=1}^{|\Omega_1|} r_j q_j + \sum_{j=1}^{|\Omega_2|} q_j$; the output is the best job release plan $u_\tau(k + k_0) 1 \leq k \leq |\Phi|$; the control includes the goals of cost minimization (i.e., $\sum_{j=1}^{|\Omega|} \pi_{1,j}$ (Π_1), $\sum_{j=1}^{|\Omega|} \pi_{2,j}$ (Π_2), and $\sum_{j=1}^{|\Omega|} \pi_{3,j}$ (Π_3)), the constraints pertaining to the job release time $u_\tau(k + k_0)$, and the constraints regarding state-space equation (8.2) and (8.6); the mechanism comprises the real-time processing status $\overline{\boldsymbol{X}}(k_0)$ and $y(k_0)$, the predictive processing time $\sigma_i(k + k_0)$, and an optimization approach.

$$\Pi_3 = \sum_{l=1}^{|\Omega|} \sum_{k=1}^{\sum_{j=1}^{|\Omega|} z_{j,l} q_j} \sum_{j=1}^{|\Omega|} z_{j,l}\, w_{3,j} \sum_{\tau=1}^{P+1} \left[\begin{array}{l} y\left(k_0 + \sum_{l'=1}^{l-1} \sum_{j=1}^{|\Omega|} z_{j,l'} q_j + k\right) \\ -u_\tau\left(k_0 + \sum_{l'=1}^{l-1} \sum_{j=1}^{|\Omega|} z_{j,l'} q_j + k\right) \end{array} \right] \tag{8.12.1}$$

$$\Pi_2 = \sum_{l=1}^{|\Omega_1|+|\Omega_2|} \sum_{k=1}^{\sum_{j=1}^{|\Omega|} z_{j,l} q_i} \left[\begin{array}{r} \sum_{j=1}^{|\Omega|} w_{2,j} z_{j,l} \max\left\{ y\left(k_0 + \sum_{l'=1}^{l} \sum_{j=1}^{|\Omega|} z_{j,l'} q_j\right), d_j \right\} \\ -y\left(k_0 + \sum_{l'=1}^{l-1} \sum_{j=1}^{|\Omega|} z_{j,l'} q_j + k\right) \end{array} \right] \tag{8.12.2}$$

$$\Pi_1 = \sum_{l=1}^{|\Omega|} \max\left\{ \sum_{j=1}^{|\Omega|} w_{1,j} z_{j,l} \left[y\left(k_0 + \sum_{l'=1}^{l} \sum_{j=1}^{|\Omega|} z_{j,l'} q_j\right) - d_j \right], 0 \right\} \tag{8.12.3}$$

The parameters $\sigma_i(k + k_0)$, $a_\tau(k)$, and $b_\tau(k)$ are determined by variables r_j and $z_{j,l}$. We denote the nearest predecessor of position L, $L \in \{2, \ldots, |\Omega_1| + |\Omega_2|\}$, that has been assigned an order of $L-1$. The $\sigma_i(k+k_0)$, $a_\tau(k)$, and $b_\tau(k)$ can then be represented by equation (8.13). If position L has no predecessor position (i.e., $\sum_{j=1}^{|\Omega_1|+|\Omega_2|}\sum_{l=1}^{L-1} z_{j,l}q_j = 0$), $\sigma_i(k+k_0) = \sum_{j=1}^{|\Omega_1|+|\Omega_2|} y_{j,L}p_{ij}$ if $k=1$.

$$\sigma_i(k+k_0) = \begin{cases} p_{ij} + t_i(k+k_0, k+k_0-1), if \sum_{j=1}^{|\Omega|} z_{j,L} = 1 \; and \; k = \sum_{j=1}^{|\Omega|}\sum_{l=1}^{L-1} z_{j,l}q_j + 1 \\ p_{ij}, if \sum_{j=1}^{|\Omega|} z_{j,L} = 1 \; and \sum_{j=1}^{|\Omega|}\sum_{l=1}^{L-1} z_{j,l}q_j + 1 < k \le \sum_{j=1}^{|\Omega|}\sum_{l=1}^{L} z_{j,l}q_j \end{cases} \quad (8.13.1)$$

$$a_\tau(k) = a_j, \sum_{j=1}^{|\Omega|} z_{j,l} = 1 \; and \sum_{j=1}^{|\Omega|}\sum_{l'=1}^{l-1} z_{j,l'}q_j + 1 \le k \le \sum_{j=1}^{|\Omega|}\sum_{l'=1}^{l} z_{j,l'}q_j \,, 1 \leqslant l \leqslant |\Omega| \quad (8.13.2)$$

$$b_\tau(k) = b_j, \sum_{j=1}^{|\Omega|} z_{j,l} = 1 \; and \sum_{j=1}^{|\Omega|}\sum_{l'=1}^{l-1} z_{j,l'}\, q_j + 1 \le k \le \sum_{j=1}^{|\Omega|}\sum_{l'=1}^{l} z_{j,l'}\, q_j \,, 1 \leqslant l \leqslant |\Omega| \quad (8.13.3)$$

8.5.3 0-1 Integer Programming for r-OAS Problem

Based on equations (8.1) – (8.13), the optimization for the r-OASR problem can be described as a BIO model. Consistent with a Stackelberg game, the leader r-OAS problem performs as an upper-level optimization, while the follower r-JRP acts as a lower-level optimization. Therefore, the general form of the joint order acceptance, scheduling, and job-release decisions can be represented in the following BIO model.

$$\max F\left(r_j, z_{j,l}, u_\tau(k+k_0)\right) = \sum_{j=1}^{|\Omega_1|} r_j e_j + \sum_{j=1}^{|\Omega_2|} e_j - \sum_{j=1}^{|\Omega|} \pi_{1,j} - \sum_{j=1}^{|\Omega|} \pi_{2,j} - \sum_{j=1}^{|\Omega|} \pi_{3,j} \quad (8.14.0)$$

$s.t.$

$$\sum_{j=1}^{|\Omega|} z_{j,l} \leq 1, l = 1, \dots, |\Omega| \tag{8.14.1}$$

$$\sum_{l=1}^{|\Omega|} z_{j,l} = r_j, j \in \Omega_1 \tag{8.14.2}$$

$$\sum_{l=1}^{|\Omega|} z_{j,l} = 1, j \in \Omega_2 \tag{8.14.3}$$

$$r_j\left(e_j - \pi_{1,j} - \pi_{2,j} - \pi_{3,j}\right) > 0, j \in \Omega_1 \tag{8.14.4}$$

$$e_j - \pi_{1,j} - \pi_{2,j} - \pi_{3,j} > 0, j \in \Omega_2 \tag{8.14.5}$$

$$r_j \in \{0, 1\}, j \in \Omega_1 \tag{8.14.6}$$

$$z_{j,l} \in \{0, 1\}, j \in \Omega_1 \cup \Omega_2, l = 1, \dots, |\Omega| \tag{8.14.7}$$

$$\min f\left(u_\tau(k + k_0)\right) = \sum_{j=1}^{|\Omega|} \pi_{1,j} + \sum_{j=1}^{|\Omega|} \pi_{2,j} + \sum_{j=1}^{|\Omega|} \pi_{3,j} \tag{8.14.8}$$

$s.t.$

$$\sigma_i(k + k_0) = \Theta_1\left(z_{j,l}\right), j, l\text{=1,…,}|\Omega| \tag{8.14.9}$$

$$a_\tau(k) = \Theta_2\left(z_{j,l}\right), j, l\text{=1,…,}|\Omega| \tag{8.14.10}$$

$$b_\tau(k) = \Theta_3\left(z_{j,l}\right), j, l\text{=1,…,}|\Omega| \tag{8.14.11}$$

$$\overline{\boldsymbol{X}}(k + k_0) = \overline{\boldsymbol{A}}(k + k_0 - 1) \otimes \overline{\boldsymbol{X}}(k + k_0 - 1) \oplus \overline{\boldsymbol{B}}(k + k_0)$$

$$\otimes \boldsymbol{U}(k+k_0), 1 \leq k \leq |\Phi| \tag{8.14.12}$$

$$y(k+k_0) = \overline{\boldsymbol{C}}(k+k_0) \otimes \overline{\boldsymbol{X}}(k+k_0), 1 \leq k \leq |\Phi| \tag{8.14.13}$$

$$b_\tau(k) \geq u_\tau(k+k_0) \geq a_\tau(k), 0 < k \leq |\Phi|, 1 \leq \tau \leq P+1 \tag{8.14.14}$$

$$u_\tau(k+k_0) \in \mathbb{Z}^+, 0 < k \leq |\Phi|, |\Phi| = \sum_{j=1}^{|\Omega_1|} r_j q_j + \sum_{j=1}^{|\Omega_2|} q_j\,, 1 \leq \tau \leq P+1 \tag{8.14.15}$$

Functions Θ_1, Θ_2, and Θ_3 are formulated according to equation (8.13). Some constraints exist in both the upper- and lower-level optimizations. equation (8.14.0) indicates the objective of the r-OAS problem by summarizing the net revenue of all orders. Constraints (8.14.1) require that each production position is assigned a maximum of one order. Constraints (8.14.2) ensure that the newly accepted orders must be assigned to a position. The existing order cannot be rejected by Constraints (8.14.3). The net revenue of each order is promised by Constraints (8.14.4) and (8.14.5). Constraints (8.14.6) and (8.14.7) enforce the binary integrality of the respective decision variables. The objective for lower-level optimization is revealed by equation (8.14.8). Constraints (8.14.9) – (8.14.10) describe the relationship between $\sigma_i(k+k_0)$, $a_\tau(k)$, $b_\tau(k)$ and variable $z_{j,l}$, respectively. The state-space equation constraints are represented by equations (8.14.11) and (8.14.12). The job release time is restricted by constraints (8.14.13). Constraints (8.14.14) enforce the non-negativity and integrity of the decision variables at the lower level.

8.6 Solution Approach

The solving processes for the BIO model are complex because the optimization models for r-OAS and r-JRP are NP-hard. Traditional solution approaches in bilevel programming can be classified into direct and indirect categories. The indirect methods convert a bilevel model into a single-level model like the Karush-Kuhn-Tucker condition and penalty function (Ji et al., 2013). The extreme-point search and some intelligence algorithms are direct approaches for bilevel programming problems (Xiong et al., 2018). Considering the NP-hard characteristic of BIO, various heuristic algorithms have been developed, including genetic algorithms, particle swarm optimization, and tabu search (Xiong et al., 2018). This paper establishes a direct solution approach comprising a tabu search and an implicit enumeration algorithm. This algorithm's role is to generate a feasible solution for the BIO model and derive several managerial conclusions accordingly. The pseudocode of the bilevel tabu-enumeration algorithm is presented in Table 8-1. To decrease the nervousness of the production systems, a new insertion procedure is designed to generate the initial solution (See Section 8.6.1), just as (Fahmy et al., 2007) shows. This procedure inserts real-time orders into the initial scheduling plan, such that the maximum net revenue can be achieved. With this insertion, we do not change the scheduling plan that is inherited from the previous stage until it can generate more benefit. Moreover, to avoid the exponential increase in search time, only the η most promising neighborhood solutions after a tabu search are transferred to the enumeration algorithm to search for the best job release plan (Gromicho et al., 2012). Note that the parameter η setting simultaneously affects the problem solutions and computation time. For example, setting $\eta = 1$ results in the nearest neighbor heuristic and $H = \infty$ results in a whole neighborhood search.

Table 8-1 The pseudo-code of the tabu-enumeration algorithm

Algorithm 8-1: *General interactive procedures of the tabu-enumeration algorithm*

Input: *A serial of real-time orders Ω_1, a serial of orders that are waiting at job pools Ω_2, the real-time production status $X(k_0)$ of flow line, a tabu tenure K, an enumeration searching step δ, parameter η, and the I_{max}.*

Output: *The best order acceptance decision r_j, $j \in \Omega_1$, scheduling decision $z_{j,l}$, $j \in \Omega_1 \cup \Omega_2$, $l = 1, \ldots, |\Omega_1| + |\Omega_2|$, and job release plan $u_\tau(k + k_0)$, $1 \le k \le |\Phi|$, $1 \le \tau \le P + 1$.*

begin
  $I_{current} \leftarrow 0$;
  Initial solution generation;
  repeat
    $I_{current} \leftarrow I_{current} + 1$;
    foreach *a pair of elements in a solution* **do**
      Swap the two elements;
      Reject the orders with a net revenue equal to or smaller than zero and generate the most feasible solution;
      Insert the rejected orders and generate the corresponding feasible solutions;
      Calculate the fitness of each feasible solution with the assumption that $u_\tau(k + k_0) = u_\tau^{lower}(k + k_0)$, $1 \le k \le |\Phi|$;
    end
    Sort the feasible solutions in descending order based on their fitness;
    Select the first η feasible solutions and set the fitness of other solutions to zero;
    foreach *selected feasible solution* **do**
      Call the enumeration search algorithm for best release plan generation, i.e., $u_\tau(k + k_0)$, $1 \le k \le |\Phi|$;
      Calculate the operation cost and feed the results back into the r-OAS problem;
      Revise the fitness of each selected feasible solution;
    end
    Let current solution be the feasible solution with the greatest fitness;
    if *the fitness of current solution is smaller than the previous* **then**
      Replace the old solution with the new solution and the new fitness;
    else
      Keep the old solution and its fitness;
    endif
    Update tabu list
    if $rand < e^{(-I_{current}/50)}$ **do**
      Local search based on the drop-insert operator;
      Call the enumeration search algorithm for best release plan generation;
      Calculate the operation cost and feed the results back into the r-OAS problem;
      Revise the fitness of the current solution;
    end
    if *the fitness of the current solution is smaller than the previous* **then**
      Replace the old solution with the new solution and new fitness;
    else
      Keep the old solution and its fitness;
    endif
    Until $I_{current} > I_{\max}$ || **stop criteria**
  Return r_j^*, $z_{j,l}^*$, and $u_\tau^*(k + k_0)$
end

8.6.1 Tabu Search to Solve r-OAS Problem

Tabu search is a well-known heuristic for handling combinatorial optimization problems and is particularly successful in its ability to resolve job scheduling problems (Cesaret et al., 2012). Motivated by the successful applications of tabu search, this section extends this algorithm to make OAS decisions. The implementation procedures of the tabu search are described as follows.

1) ***Solution representation***: A solution for the r-OAS problem can be represented by a vector with size $|\Omega|$, in which the value of the l-th element indicates the assigned position of order j in a production sequence. If an order is not accepted, the value of the corresponding element is zero. For example, a solution with 15 orders ($|\Omega_1| = 12$ and $|\Omega_2| = 3$) can be represented by $\{1,3,5,6,2,0,0,4,0,0,7,0,0,8,0\}$. This representation means that Orders 6, 7, 9, 10, 12, 13, and 15 are rejected, while Orders 1, 2, 3, 4, 5, 8, 11, and 14 are processed in the first, third, fifth, sixth, second, fourth, seventh, and eighth positions. Because orders 1, 2, and 3 are existing orders, their position value in a solution must be greater than zero.

2) ***Solution initiation***: An initial feasible solution S_0 for r-OAS can be generated by a greedy rule. This rule first calculates the revenue-load ratio for each order by considering the order revenue and its processing time. This ratio is represented as $RLR_j = e_j/\left(q_j \max_{i=1,..,M} p_{ij} + \sum_{i=1}^{M} p_j\right)$, $j \in \Omega_1 \cup \Omega_2$, which signifies the unit revenue created per production time. The orders are sorted by this ratio, and the order with the highest value is first inserted into the initial production sequence. During this insertion, the selected order is inserted into its possible position in sequence, and the corresponding net revenue is

calculated. For example, the production sequence from solution $\{1,3,5,6,2,0,0,4,0,0,7,0,0,8,0\}$ is $\{1,5,2,8,3,4,11,14\}$ (the element in the set is the order number) and the possible insert position precedes Orders 1, 5, 2, and so on. Finally, the order is inserted into the position that can generate the highest net revenue. At this stage, the costs $\pi_{1,j}$, $\pi_{2,j}$, and $\pi_{3,j}$ of Order j are determined according to equation (8.12) and based on the assumption that $u_\tau(k_0+k) = u_\tau^{lower}(k+k_0)$, $k = 1, \dots, |\Phi|$, $\tau = 1, \dots, P + 1$. If this order insertion causes the revenue of some orders in the new production sequence to be equal to or less than zero, these orders are rejected, and the net revenue of the remaining orders is recalculated. This insertion procedure continues until all newly arrived orders are traversed.

3) ***Neighborhood search with swap operators***: Swap and insertion are common operators in tabu search for similar scheduling problems (Bilge et al., 2007). In swap operators, the neighborhood of the current solution S is searched by swapping two elements of S. This pairwise exchange can change both the set of accepted Orders and the corresponding production sequence. For example, two neighborhood solutions $\{1,3,5,7,2,0,0,4,0,0,6,0,0,8,0\}$ and $\{1,3,5,6,0,0,2,4,0,0,7,0,0,8,0\}$ can be generated by applying swap operators to the solution $\{1,3,5,6,2,0,0,4,0,0,7,0,0,8,0\}$. The first is obtained by swapping the fourth and eleventh elements, while Order 4 is rejected, and Order 7 is accepted in the second after swapping the fourth and seventh elements. The two elements with a value of zero do not need to be swapped, as they cannot generate a new solution. Because the existing orders cannot be rejected in this decision period, their position element can be swapped with the position that has been assigned a nonzero element in the current solution. For example, in the solution $\{1,3,5,6,2,0,0,4,0,0,7,0,0,8,0\}$, Element 1 can

be swapped with Elements 2, 3, 4, 5, 8, 11, and 14. Note that after a swap, the net revenue of each order changes. If any orders belonging to Ω_1 receive a zero or smaller net revenue, this neighborhood swap is forbidden. If any of the newly accepted orders receive a zero or smaller net revenue, they are rejected. Because the number of accepted orders may therefore decrease, an insertion operator is designed to follow swap operators; the local search procedure is shown in the following sequence.

4) ***Tabu list and tabu tenure***: The tabu list of tabu search retains the most recent swap operators to avoid cycling when searching for a new solution. The tabu list in our tabu search algorithm is formed with the K most recently performed swap operators, where K is the tabu tenure. In our tabu list, the retained swap pairs prevent the same elements from being swapped again. For example, if the best neighborhood solution of $\{1,3,5,6,2,0,0,4,0,0,7,0,0,8,0\}$ is $\{1,3,5,6,0,2,0,4,0,0,7,0,0,8,0\}$, swapping the elements corresponding to order pairs $(5,6)$ and $(6,5)$ is tabu during the tabu tenure.

5) ***Local search with a drop-insert procedure***: If the iterations proceed with neighborhood search alone, our tabu search algorithm will likely converge to a poor local optimum. To remedy this, a local search for the current best solution is conducted by applying iterative drop-insert operators. To insert an order with larger net revenue, an order with the lowest net revenue in the current solution is dropped. To select the orders that will be inserted, a roulette wheel selection from genetic algorithms is then used to generate an insertion probability for the rejected orders. The probability of selecting each rejected order is proportional to its RLR_j, such that the orders with higher RLR_j are more likely to be selected. This mechanism can introduce some randomness into our tabu search and create diversified solutions during iterations. This selection is implemented φ times, where φ is

the size of the rejected orders and all the selected but different orders are transferred for insertion. The insertion procedure is shown in part (2). To ensure the convergence of our algorithm, the order drop-insert operator is implemented in a manner similar to that of simulated annealing. That is, the drop-insert operator is implemented if a random number between $(0, 1)$ is smaller than $e^{(\frac{-I_{current}}{50})}$ ($I_{current}$ is the current number of iterations). Otherwise, no order is dropped from the current solution.

6) ***Fitness evaluation and termination criteria***: After the swap operators, the new solutions are transferred to the r-JRP problem for cost calculation (i.e., $\pi_{1,j}$, $\pi_{2,j}$, and $\pi_{3,j}$). The fitness of each solution can then be determined by equation (8.14.0). If the current tabu solution with the highest fitness is better than the fitness of the best-known solution, the current tabu solution is accepted as the best solution. The tabu search is terminated if the number of generations exceeds the upper limit of the iterations.

8.6.2 Implicit Enumeration Algorithm to Solve r-JRP Problem

Implicit enumeration and gradient-based techniques are two typical approaches for production system optimization with analytical methods (Matta et al., 2012). Implicit enumeration is widely applied if the solution space is restricted or if the evaluation procedure for production system performance is carried out rapidly. In terms of the limited searching space, which is restricted by order net revenue and the rapid run speed with the proposed state-space equation, an implicit enumeration algorithm is proposed for r-JRP optimization. Referring to the enumeration procedures of Hashemian et al. (2014), our implicit enumeration can be implemented through the following two steps: enumeration space generation and enumeration strategy with varying search steps.

The size of the enumeration space can obviously affect the searching efficiency in the enumeration space generation. In our programming model for r-JRP, the lower bound of each variable $u_\tau(k+k_0)$, $1 \le k \le |\Phi|$, is determined by the processing time of the first machine in a flow line, such as Machine 1 in serial lines and the Machine $\sum_{\tau'=1}^{\tau-1} M_{\tau'} + 1$, $(\tau = 1, \dots, P)$ in merge lines, refer to equation (8.15):

$$u_\tau^{lower}(k+k_0) = \max\left\{a_\tau(k), \sum_{k''=1}^{k} \sigma_i(k''+k_0), \tau = 1, \dots, P, i = \sum_{\tau'=1}^{\tau-1} M_{\tau'} + 1\right\} \quad (8.15)$$

where $u_\tau^{lower}(k_0) = u_\tau(k_0)$ and $\sum_{\tau'=1}^{0} M_{\tau'} = 0$. This lower bound is reasonable because a smaller value for $u_\tau(k+k_0)$ cannot contribute to improving the system throughput. The upper bound of the variable $u_\tau(k+k_0)$ is derived from the order due date and tardiness penalty cost. For each order, the relationship between the latest delivery time d_j^{latest} and due data d_j can be represented as follows:

$$w_{1,j}\left(d_j^{latest} - d_j\right) + w_{2,j}\frac{q_j\left(q_j - 1\right)}{2}\max_{i=1,\dots,M}\{p_{ij}\} = e_j, j \in \Omega. \quad (8.16)$$

According to equation (8.16), the d_j^{latest} can be derived as follows:

$$d_j^{latest} = \frac{\left(e_j - w_{2,j}\frac{q_j\left(q_j-1\right)}{2}\max_{i=1,\dots,M}\{p_{ij}\}\right)}{w_{1,j}} + d_j, j \in \Omega. \quad (8.17)$$

Referring to equation (8.13), the upper bound of variable $u_\tau(k+k_0)$, $\tau = 1, \dots, P+1$, can be determined by the following:

$$u_{\tau}^{upper}(k+k_0-k'') = \max\left\{b_{\tau}(k), \sum_{j=1}^{\Omega} z_{j,l}\left(d_j^{latest} - k'' \max_{i=1,\dots,M}\{p_{ij}\} - \sum_{i=1}^{\bar{M}} p_{ij}\right)\right\}, \quad (8.18)$$

$$if \sum_{j=1}^{\Omega} z_{j,l} = 1, k = \sum_{j=1}^{\Omega}\sum_{l'=1}^{l} z_{j,l'} q_j, \bar{M} = \{1, \dots, M_{\tau}, m_{\tau}, \dots, M_{P+1}\},, l = 1, \dots, |\Omega|.$$

Furthermore, in enumeration strategy with varying search steps, the cost $\pi_{1,j}$, $\pi_{2,j}$, and $\pi_{3,j}$ are monotonous to variable $u_{\tau}(k+k_0)$. In this case, the best job release plan can be achieved by increasing each variable from the smallest to the greatest. To improve the search efficiency, a varying searching step is designed after analyzing the decision space of variables (see Table 8-1). The six steps of the enumeration are presented as follows.

Step 1: We set $u_{\tau}(k+k_0) = u_{\tau}^{lower}(k+k_0)$, $\tau = 1, \dots, P+1$, $1 \le k \le |\Phi|$, and a search step $\delta = \delta_0$ that increases the value of $u_{\tau}(k+k_0)$ by δ at every step. We determine the initial operation cost f_0 based on equations (8.9) and (9.14.8). We then initiate the enumeration procedures with $k \leftarrow 1$ and $\tau \leftarrow 1$.

Step 2: We set $u_{\tau}(k+k_0) = u_{\tau}(k+k_0) + \delta$, and $u_{\tau}(k''+k_0) = u_{\tau}(k''+k_0) + \delta$, $|\Phi| \ge k'' > k$. The system throughput $y(k''+k_0)$, $1 \le k'' \le |\Phi|$ is then calculated based on state-space equation (8.6) and equation (8.14.8) is used to formulate the operation cost f.

Step 3: If $f \ge f_0$, set $u_{\tau}(k''+k_0) = u_{\tau}(k''+k_0)$, $1 \le k'' \le |\Phi|$, $f_0 = f$, and we return to **Step 2**. If $f < f_0$, we set $u_{\tau}(k''+k_0) = u_{\tau}(k''+k_0) - \delta$, $k \le k'' \le |\Phi|$, and we move on to **Step 4**.

Step 4: We set the search step to $\delta = \lceil \delta/2 \rceil$. If δ is smaller than the unit production time, we move to **Step 5**; otherwise, we return to **Step 2**.

Step 5: We set $k \leftarrow k+1$ and $\delta = \delta_0$. If $k > |\Phi|$, we move on to **Step 6**; Otherwise, we return to **Step 2**.

Step 6: We set $\tau \leftarrow \tau + 1$. If $\tau > P + 1$, we stop the enumeration and feed the function f_0 back to the leader r-OAS problem; otherwise, we return to **Step 2**.

8.6.3 Tabu-Enumeration Interactive Solution Procedure Problem

The nested genetic algorithm is a popular technique for solving the BIO model (Xiong et al., 2018). Because probabilistic techniques are adopted to search for a feasible solution for both leader and follower problems, its convergence is difficult to predict with certainty, especially when the space in which the search is conducted is large. Compared with existing heuristic algorithms, our bilevel tabu-enumeration search empowers a mutual look-out global search strategy by developing a probabilistic search technique for the leader and an exact search technique for the follower. This method can accelerate the speed at which algorithms converge, thus tremendously reducing the computational load, and in turn enhancing computational efficiency. Based on the encoding analysis of tabu search for r-OAS and the implicit enumeration for r-JRP, the bilevel tabu-enumeration is elaborately designed and includes several decision stages, as follows.

Step 1: Initialization. A feasible solution for OAS is randomly generated.

Step 2: Fitness estimation. The swap operator is called to determine all neighborhood solutions. We set $u_\tau(k + k_0) = u_\tau^{lower}(k + k_0)$, $1 \le k \le |\Phi|$, and estimate the fitness of each solution. These solutions are then sorted in descending order based on their estimated fitness and the first η feasible solutions for r-JRP optimization are selected. We set the fitness of these unselected solutions to zero.

Step 3: Implicit enumeration. For each selected solution, we use the implicit enumeration to determine the minimum operation cost f according to equation (8.14.8). We feed cost f back into the tabu search and update the fitness F of each selected solution according to equation (8.14.0).

Step 4: Best solution update. We choose the solution with the highest fitness as the current solution. If the current solution is better than the best solution already identified, we update the best solution and its fitness, and add the corresponding swap operator into the tabu list; otherwise, we take no action.

Step 5: Loach search with a drop-insert operator. We drop an order with minimal net revenue from the current solution and select the unaccepted orders for insertion into the current solution. We update the current and best solutions, as in **Step 4**.

Step 6: We terminate tabu search if the number of generations exceeds the upper limit of the tabu search. We then compare and record the best results and move on to **Step 7**; otherwise, we return to **Step 2**.

Step 7: Stop.

8.7 Case Study of Crowdsourcing Task Acceptance and Accommodation

To demonstrate the performance of the proposed BIO model for the r-OASR problem and test the proposed bilevel tabu-enumeration algorithm, a specific case study from a car seat assembly plant is reported in this section. The bilevel algorithm is performed ten times on a Windows 10 PC with an Intel ® Core ™ i5-8250U 3.4GHz 16GB of RAM. The one yielding the highest fitness is recorded as the best objective value for the BIO model.

8.7.1 Tabu-Enumeration Interactive Solution Procedure Problem

The car seat assembly line contains twelve workstations and eleven buffers, in which the jobs between adjacent workstations are transported by conveyor belts, the data of this case is modified from real-world shop-floor. The TEG of this assembly line is shown in Figure 8-6. Two parallel sub-lines are merged at Station 5: one for cushion assembly and the other for back cushion assembly. The flexibility of each station is high, and at least nine kinds of products can be assembled by this line. Table 8-2 lists the processing time of the stations for each product, and Table 8-3 shows the buffer capability. This assembly line is chosen to verify our BIO approach for a variety of production environments and smart production operations.

1) ***Decision-making for r-OASR within high variety production environments*:** The car seat assembly plant is a first-tier supplier to automakers. Today, with mass customization, many automakers are coupled with platform-based strategies to achieve mixed-model production. Because multiple automakers can be serviced by a car seat assembly plant at the same time, this mixed-model production creates a high level of product variety for each seat assembly line. For example, the special assembly line selected in this section must produce nine kinds of car seats and is currently being expanded to produce twelve products in the future. Moreover, due to various resource-related disruptions related to stamping, welding, spraying, and assembly shops and job-related disruptions from customers, the pre-determined car assembly plan is often changed. This frequent variation creates many real-time orders for car seat assembly plants, as they must respond to the production plan of automakers in a timely manner. This special assembly line is designed to accommodate real-time orders in highly varied production

environments. This line adopts the rolling-horizon-driven replanning policy for r-OASR decision-making to enable the fulfillment of real-time replenishments from automakers. By cooperating with this special line, the nervousness of the whole assembly plant can be decreased, as the schedules of other assembly lines will not be disrupted by new incoming orders. Additionally, as it is motivated by the open manufacturing paradigm, this plant is devoted to designing an open platform to collect personalized orders from customers. Customers can place their orders on the platform in real time, similar to the COSMOPlat of Haier. This development trend also positions r-OASR decision-making as important.

2) ***Smart production operations with real-time data*:** To predict the real-time status of assembly lines and derive smart actions in response to disruptions, various information-gathering techniques have been applied to monitor and collect production data from shop floors. For example, an RFID system is constructed to collect the real-time data of WIP and job completion at each station. A sensor-based network is developed to perceive the real-time data of machines and stations (e.g., machine breakdown events), and an image recognition system is designed to identify the productivity data pertaining to workers (e.g., the absence data of workers). A control system is in place that can formulate smart decisions after fuzzing and analyzing real-time data. In view of the smart manufacturing environment, the real-time production status of this assembly line can be easily derived. Moreover, the control system is an open platform. It can provide powerful computing capacities for users and a flexible environment for the development of new functions. Thus, many specified functions for production that are operated with complex optimization models can be incorporated into this control system, such as r-OASR decision-making, which is the focus of this paper.

Table 8-2 Processing time of each station (unit: sec)

Product type	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}
Type A	82	84	84	**92**	89	74	85	78	70	89	92	90
Type B	78	86	82	**97**	93	78	90	80	68	90	89	95
Type C	88	78	83	80	88	70	92	74	78	**95**	74	89
Type D	74	88	78	84	80	76	**95**	80	71	89	80	91
Type E	77	89	82	89	91	83	**97**	77	82	93	78	84
Type F	85	91	88	70	93	71	90	69	88	**93**	83	87
Type G	75	81	68	96	98	75	**99**	85	69	79	94	78
Type H	70	87	95	87	84	89	91	67	93	**99**	89	80
Type I	72	75	90	73	88	69	78	80	92	**95**	76	84

Table 8-3 Buffer capacity of the assembly line (unit: sec)

Buffer No.	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
Buffer capacity	5	8	6	9	10	12	8	9	6	10	11

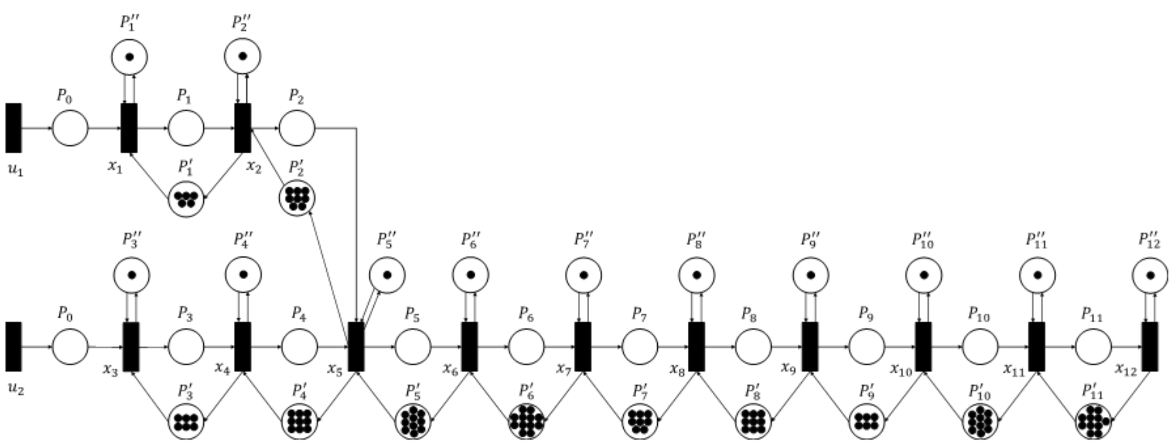


Figure 8-6 Time event graph of the assembly line

Based on these characteristics, the car seat assembly line is extended to make r-OASR decisions. To identify the parameters of our BIO approach, a series of incoming orders during a time horizon $t_0 = 60$min is stochastically selected from the information system. The existing orders are listed in Appendix II. No release date constraint exists in this case study. The tabu tenure is K=5, and the iteration number is $I_{max} = 200$. When new orders arrive, 17 jobs are released to the assembly line and no disruption occurs. Thus, the real-time status $\overline{\boldsymbol{X}}(17)$ of this assembly line can be derived.

8.7.2 Performance Experiments by Benchmarking Using Prevailing r-OASR Approaches

To draw a comparison with the performance of the proposed BIO, two prevailing approaches are presented in this section: (i) the r-OAS model with bottleneck machines (r-OAS-B) (Nobibon and Leus, 2011, Lin and Ying, 2015), and (ii) the r-OAS model for permutation flow lines (r-OAS-F) (Rahman et al., 2015, Rahman et al., 2019). To verify the performance of the r-JRP, we extend these approaches by considering cooperation with r-JRP as a factor. The approaches are as follows: a single r-OAS-B (r-OAS-B I), r-OAS-B with r-JRP (r-JRP-B II), a single r-OAS-F (r-OAS-F I), and r-OAS-F with r-JRP (r-OAS-F II). In r-OAS-B II and r-OAS-F II, the r-OAS and r-JRP problems are operated separately. The r-OAS-B and r-OAS-F are solved by our tabu-enumeration algorithm. For the comparison, the results of our BIO model are also decomposed into two categories: one does not cooperate with r-JRP (BIO I) while the other does (BIO II).

8.7.3 Results and Analysis

Table 8-4 presents the best decision for OAS with $|\Omega_1| = 34$ and $|\Omega_2| = 10$. The r-OAS-B, r-OAS-F, and BIO all accept eighteen new orders. In comparing the results with Appendix II, one can observe that the nervousness of this production system is low, as the scheduling plan for existing orders remains unchanged. In this situation, the total net revenues of r-OAS-B I, r-OAS-B II, r-OAS-F I, r-OAS-F II, BIO I, and BIO II are 338,840, 340,560, 340,470, 345,730, 340,540, and 348,900, respectively. Compared to other approaches, BIO II demonstrates better performance, regardless of cost Π_1, Π_2, or Π_3. This performance is achieved by identifying a better production sequence, as the accepted orders are the same; see the results of r-TAS-B, r-TAS-F, and the BIO. For the flow lines, the state-space equation is important in reducing the tardiness penalty cost because it can accurately predict production capacity (see the results of r-OAS-B and r-OAS-F). In view of the cost Π_1, Π_2, or Π_3, the proposed r-JRP can significantly reduce Π_2 and Π_3, whereas Π_1 can be increased slightly.

In conclusion, when compared to the bottleneck-based and flow-shop approaches, our BIO model can increase the profit of this assembly line by 2.79% and 2.48%, respectively. This improvement is significant, as the average net revenue of this assembly plant has been only approximately 10% over the last ten years. Except for profit improvement, the BIO model can provide accurate information for production process control and can in turn improve the capacity of production systems to make precise decisions. To further test the performance of our BIO, several experiments regarding the quantity of real-time orders are designed. We observe the quantity of real-time orders every 60 min over the preceding five days and find that 80% of the samples are located between 6 and 34 orders. Figure 8-7 shows the total net revenue of the proposed six approaches for r-OASR by changing $|\Omega_1|$ from 6 to 34. Figure 8-7 shows that the total net revenue of the BIO approach decreases with the quantity of incoming orders. This result is consistent with reality and shows that the BIO and its tabu-enumeration algorithm can respond to different production parameters. Upon observing the net revenue of r-OAS-B II, r-OAS-F II, and

BIO II, it can be observed that BIO II consistently demonstrates better performance and yields a more significant benefit for manufacturers. In comparing the results of r-OAS-F and BIO with r-OAS-B, one may observe that considering the job dynamic transition behaviors of flow lines enables production capacity to be predicted more concisely, which can then generate a better OAS scheme. Finally, in examining the results with and without r-JRP, it can be concluded that the r-JRP can decrease operation cost significantly, although it can also occasionally prompt a slight increase in the tardiness cost.

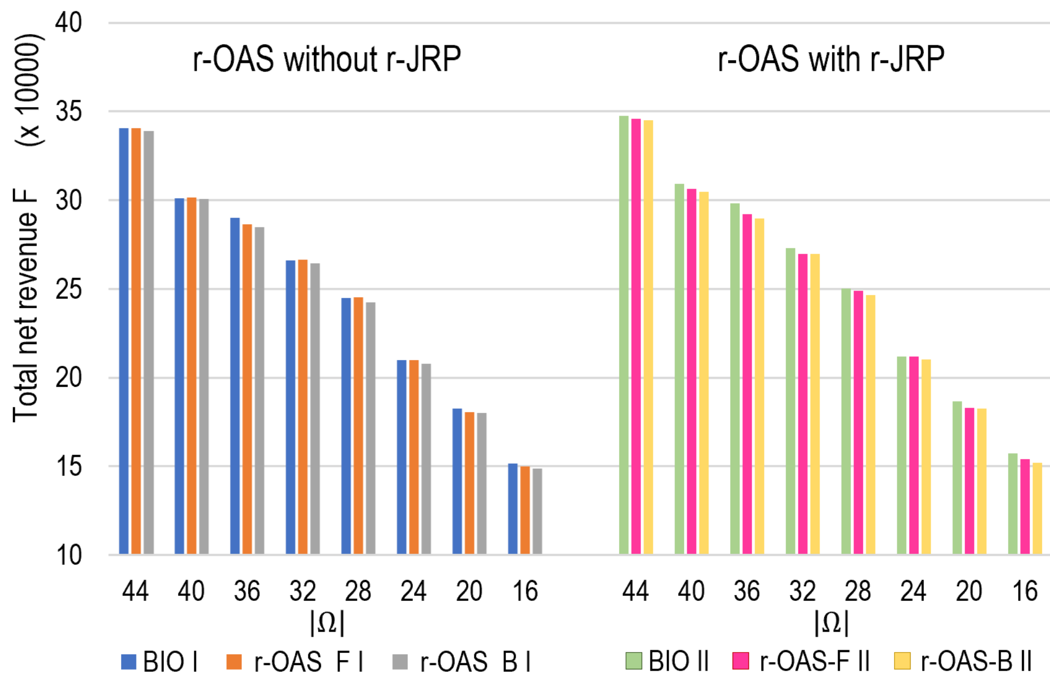


Figure 8-7 The performance of the BIO with different quantities of incoming orders

Table 8-5 and Table 8-6 list the operation costs of the proposed six approaches in these sensitive experiments. Compared to the r-OAS-B, the research results reveal that the BIO and r-OAS-F can sometimes improve the quantity of accepted orders due to a more accurate estimation of the job makespan, such as when $|\Omega_1| = 30$ and $|\Omega_1| = 26$. In other situations, the quantity of orders accepted by the six approaches is the same, but the types of accepted orders and the corresponding schedules are different. Our BIO consistently identifies a better OAS decision, even though the operation costs of production systems are sometimes higher than those of other approaches. For example, the order tardiness penalty cost of BIO II in the case $|\Omega_1| = 26$ is 18,140. When compared to the r-OAS-B II and r-OAS-F II, this cost increases by 34.91% and 47.77%, respectively. However, in this situation, BIO II can increase the net revenue of production systems by 2.09% and 2.03%, respectively. This increase in profit is significant for an assembly plant.

8.7.4 Sensitivity Analysis and Discussions

Sensitivity experiments conducted on parameters $w_{1,j}$, d_j, $w_{2,j}$, and $w_{3,j}$ are designed in this section, and several managerial implications can be concluded based on the results. Figure 8-8 shows the total net revenue F and the accepted order quantity for different $w_{1,j}$ and d_j. The sensitivity experiments decrease $w_{1,j}$ by 2 or increase d_j by 300 at every step, while keeping other parameters at their initial value. As shown in Figure 8-8, the accepted order quantity and total net revenue can be increased by increasing d_j and decreasing $w_{1,j}$, respectively. The results are consistent with reality and demonstrate the efficiency of our algorithm. Moreover, focusing on the effect of parameter $w_{1,j}$, the total net revenue and accepted order quantity can be increased slightly when $\Delta w_{1,j} \leq 4$, while the two figures can be increased linearly when $\Delta w_{1,j} > 4$. Compared to $w_{1,j}$, the effect of d_j is smaller. The total net revenue F and accepted order quantity obviously increase with d_j when $\Delta d_j \leq 900$, while the increasing trends of the two figures slacken when $\Delta d_j > 900$.

Figure 8-9 reveals the total net revenue F and the accepted order quantity with different $w_{2,j}$ and $w_{3,j}$. The sensitivity experiments increase $w_{2,j}$ by 0.2 or increase $w_{3,j}$ by 0.1 at every step, while keeping other parameters at their initial value. As shown in Figure 8-9, the total net revenue F decreases when $w_{2,j}$ and $w_{3,j}$ increase Specifically, the increase in $w_{2,j}$ and $w_{3,j}$ will not reduce (see the rectangle without filling) or even increase (see the rectangle with filling) the accepted order quantity initially. This phenomenon is reasonable, as the BIO model must balance the revenue and cost when searching for a solution. Moreover, comparing the effects of $w_{2,j}$ with those of $w_{3,j}$, it is known that the total net revenue F is more sensitive to $w_{3,j}$. This means that manufacturers should pay more attention to WIP inventory costs when they desire to improve the net revenue of their production systems.

Observed from above research results, several substantive insights about r-OASR development can be obtained. Firstly, the data-rich environment in today's smart manufacturing enables us to make precise r-OASR decisions. As presented in previous literature, the existed bottleneck-based approach and flow-line-based approach with infinite buffers support making r-OASR decisions by estimating system performance (i.e., system throughput, WIP inventory, and final product inventory). This estimation increases the total operational cost of a production system and thus reduces the net revenue, see the results in Table 8-4. In contrast, our data-driven representation for r-OASR can recognize real-time production status and precisely predict system performance by modelling finite buffer constraints and analyzing real-time data collected from shop floors. This precise recognition and prediction can help to make a better r-OASR decision and ultimately increase the net revenue.

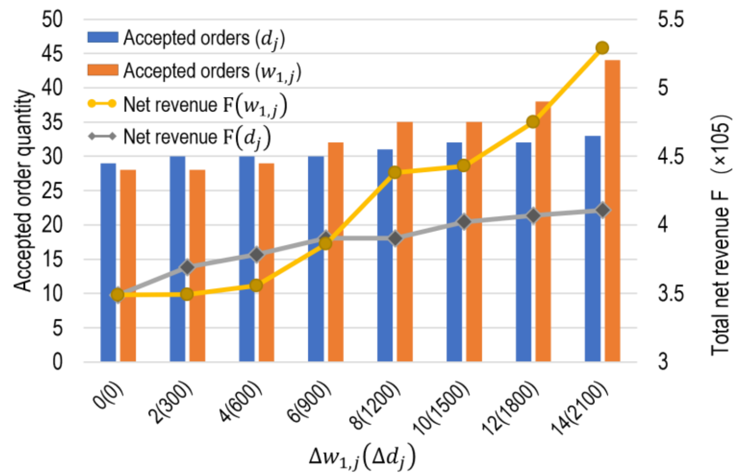


Figure 8-8 The net revenue and accepted order quantity with different tardiness penalty coefficient and due date

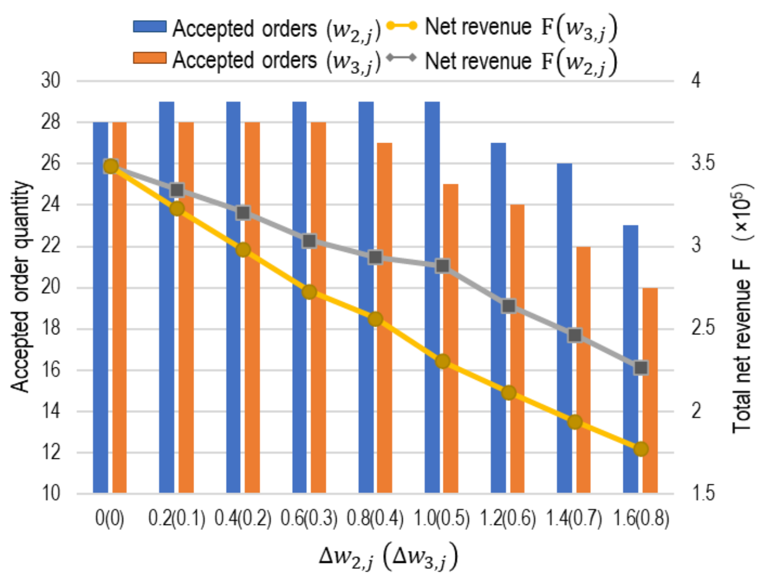


Figure 8-9 The net revenue and accepted order quantity with different unit inventory cost for a final product and inventory WIP cost

Secondly, the data fusion of a shop floor gives birth to more joint optimization problems, in which each subproblem traditionally belongs to different decision-making levels. Conventional wisdom has to separately treat these subproblems because of the impassability of data. The upper-level problem cannot seek cooperation from lower-level and in this regard the lower-level problem also cannot obtain the best input. The data fusion makes joint optimization possible as decisions among different levels can interact with each other freely and quickly, see the example in Figure 8-2 and Figure 8-3. This interaction can create a global optimum solution for a production system. With this solution, the system can be operated more concisely and with lower costs, see the results of r-OASR in Figure 8-7, Table 8-5, and Table 8-6.

Thirdly, today's smart manufacturing necessitates our r-OASR. The r-OASR can achieve the quick response to abnormal events by minimizing multiple operational costs. This response is significant in constructing smart production and networked manufacturing, see the results in Figure 8-8 and Figure 8-9. Certainly, the nervousness problem of r-OASR is a hot topic in many studies. In smart manufacturing, however, this nervousness can be reduced as the task rescheduling at different decision-making levels can be achieved quickly and automatically along with a r-OASR decision.

Table 8-4 Best decisions for OAS with different approaches

Approach	Accepted orders (r_j)	Production sequence ($z_{j,l}$)	Revenue	Cost Π_1	Π_2	Π_3	Net revenue
r-OAS-B I	17,20,23,25,27,28, 29,30,31,34,35,36, 38,40,41,42,43,44	2,4,1,5,3,8,6,10,7,9,44,17,4 0,25,20,23,36,31,38,41,29, 27,35,28,43,42,30,34	373,100	17,622	6,487	10,154	338,840
r-OAS-B II	17,20,23,25,27,28, 29,30,31,34,35,36, 38,40,41,42,43,44	2,4,1,5,3,8,6,10,7,9,44,17,4 0,25,20,23,36,31,38,41,29, 27,35,28,43,42,30,34	373,100	18,082	4,778	5,247	340,560
r-OAS-F I	17,20,23,25,27,28, 29,30,31,34,35,36, 38,40,41,42,43,44	2,4,1,5,3,6,8,10,9,7,44,17,4 0,20,25,23,36,41,31,35,38, 27,43,28,42,29,34,30	373,100	15,598	6,871	10,166	340,470
r-OAS-F II	17,20,23,25,27,28, 29,30,31,34,35,36, 38,40,41,42,43,44	2,4,1,5,3,6,8,10,9,7,44,17,4 0,20,25,23,36,41,31,35,38, 27,43,28,42,29,34,30	373,100	15,866	6,156	5,359	345,730
BIO I	17,20,23,25,27,28, 29,30,31,34,35,36, 38,40,41,42,43,44	2,4,1,5,3,6,8,10,9,7,44,17,4 0,20,25,23,36,31,38,41,27, 35,28,43,29,30,34,42	373,100	15,598	6,794	10,164	340,540
BIO II	17,20,23,25,27,28, 29,30,31,34,35,36, 38,40,41,42,43,44	2,4,1,5,3,6,8,10,9,7,44,17,4 0,20,25,23,36,31,38,41,27, 35,28,43,29,30,34,42	373,100	16,013	3,755	4,456	348,900

Table 8-5 The operation costs without r-JRP

$\|\Omega_1\|$	r-OAS-B I				r-OAS-F I				r-OAS I			
	$\sum_{j=1}^{\|\Omega_1\|} r_j$	Π_1	Π_2	Π_3	$\sum_{j=1}^{\|\Omega_1\|} r_j$	Π_1	Π_2	Π_3	$\sum_{j=1}^{\|\Omega_1\|} r_j$	Π_1	Π_2	Π_3
34	18	17,622	6,487	10,154	18	15,598	6,871	10,166	18	15,598	6,794	10,164
30	14	12,632	7,027	4,712	**15**	19,930	6,749	8,974	**15**	18,930	7,235	7,838
26	13	13,098	7,898	8,371	13	11,856	7,387	8,618	**14**	17,798	7,883	8,891
22	10	17,184	5,983	8,090	10	15,386	5,951	8,073	10	15,386	5,156	8,028
18	7	17,922	4,801	7,213	7	15,386	4,707	7,248	7	14,386	4,287	6,731
14	5	16,418	2,093	5,744	5	14,562	2,124	5,763	5	13,562	2,124	4,763
10	4	15,432	3,008	5,652	4	14,988	2,108	5,452	4	14,462	2,002	4,052
6	2	14,032	2,320	4,420	2	13,832	3,100	5,020	2	14,002	2,113	4,028

Table 8-6 The operation cost with r-JRP

$\|\Omega_1\|$	r-OAS-B II				r-OAS-F II				r-OAS II			
	$\sum_{j=1}^{\|\Omega_1\|} r_j$	Π_1	Π_2	Π_3	$\sum_{j=1}^{\|\Omega_1\|} r_j$	Π_1	Π_2	Π_3	$\sum_{j=1}^{\|\Omega_1\|} r_j$	Π_1	Π_2	Π_3
34	18	18,082	4,778	5,247	18	15,866	6,156	5,359	18	16,013	3,755	4,456
30	14	12,632	7,027	4,712	**15**	20,090	5,720	4,822	**15**	20,526	2,992	4,725
26	13	13,446	6,441	4,612	13	12,276	5,275	4,592	**14**	18,140	2,793	4,729
22	10	17,668	4,383	4,333	10	15,610	5,836	4,501	10	15,619	2,588	4,420
18	7	18,248	3,737	4,095	7	15,702	3,659	4,229	7	15,740	2,426	4,150
14	5	16,748	2,084	3,444	5	14,880	2,080	3,595	5	14,880	2,080	3,595
10	4	14,082	2,308	4,966	4	13,002	1,984	4,108	4	10,634	1,884	3,524
6	2	12,122	2,086	3,624	2	9,008	1,548	3,208	2	6,562	1,684	3,004

8.8 Chapter Summary

This BIO's efficiency is verified via its practical application to a car seat assembly line. Given the results of various experiments, several important managerial insights for the r-OASR problem can be formulated. (1) An inherent bilevel interactive decision framework exists in r-OASR. The proposed hierarchical joint optimization based on the leader-follower game can generate a more efficient solution for the r-OASR problem. (2) The r-OASR problem manages the limited buffer and blocking constraints by analyzing real-time and transparent data from the shop floor. Various operational costs in terms of inventory and tardiness are considered in this problem. The research results compensate for the insufficiency of Rahman et al. (2015) and can increase the enforceability of an OAS scheme. (3) The parameter design of an order can obviously affects order acceptance, scheduling, and job-releasing decisions, including exogenous parameters $w_{1,j}$ and d_j and endogenous parameters $w_{2,j}$ and $w_{3,j}$. The design of these parameters should be coordinated by outsourcers and manufacturers to maximize the benefit of the entire manufacturing network. (4) Smart manufacturing necessitates r-OASR for its quick response to abnormal events. The nervousness caused by using r-OARS can be reduced by the adaptive capacity of smart manufacturing.

This chapter provides decision-making support for manufacturers to synchronize internal and external material flow, which is achieved by synergizing real-time status of shop-floor, job acceptance planning and sequencing problem together. This original work not only eases the acceptance of crowdsourced manufacturing paradigm, but also provides a new job scheduling architecture in new era.

CHAPTER 9. BLOCKCHAIN-BASED SMART CONTRACTING AND DISTRIBUTED DATA MANAGEMENT FOR INFORMATION SERVICE SYSTEMS IN CROWDSOURCED MANUFACTURING

Following an X-as-a-Service paradigm, MaaS provides information services for platform and manufacturer crowds along a crowdsourced manufacturing process. These services are required to allocate manufacturing resources, track crowdsourcing process, manage product fulfillment data, and serve crowdsourcing contracting. Because manufacturers involved in a value chain are searched through a crowdsourcing process, these manufacturers are widely dispersed, who challenges data management to handle complex process data stream. Moreover, these process data streams are required to accessible by platform and corresponding open innovators for monitoring of execution quality and large-scale cooperation of the manufacturers.

Blockchain technology has been widely applied for highly distributed security database of certain and unerasable records, public ledges, or digitalized events among the participating entities, which is based on smart contracting technology to solidify cooperation relationship among stakeholders (Crosby et al., 2016). It utilizes hash function to encrypt the information to a block and therefore publish the block to every entity in the network. It offers a secure key to protect process data while ensuring the accessibility to every decision agent inner the value chain. Blockchain has been developed to serve a population of companies to connect their engineering software and share the data along with the network (Xu et al., 2016). Based on the model of manufacturing activities, the decision agents can monitor and verify the execution of the untrusted process under a collaborative scenario (Weber et al., 2016). With the synergy of the cloud database technology, it can offer the query and retrieve services for the manufacturing data source from the machine level data, manufacturing resources data, production data, as well as logistic data (Li et al., 2018a). Instantiating blockchain along product fulfillment process acts as a driving force of manufacturing towards a smart and service-oriented one (Chen et al., 2021).

In this regard, this chapter proceeds as follows. Section 9.1reviews essential technical cornerstones of a blockchain-based information service system for platform-driven crowdsourced manufacturing. Section 9.2execute an architectural design of the information service system with a three-layer structure for crowdsourcing environment. Section 9.3implements smart contracts technology to build up the foundation of cooperation relationships among decision agents. Section 9.4develops distributed data streaming architecture to organize crowdsourcing product fulfillment data based on IoT hardware. Section 9.5reports a case study on tank trailer industry to verify the function of proposed information service system and examines the potential of operation excellence in crowdsourcing environment. Section 9.6reflects the proposed methods and information service system by summarizing their contributions and further development threads.

9.1 Blockchain-Based Information Service System

The recent advancement of smart sensing and industrial IoT provides a large variety of data sources to backtrack the life cycle of the product and paves the way for the quality check from a distance (Tian, 2016). The low latency network and ubiquitous connectivity enable the streaming of the real-time status of the material flow in the factory to the cloud database for logistic coordination, which increase utilization rates of equipment and decrease delivery peaks (Qu et al., 2016). The platform-based blockchain structure for IoT can also ease the adoption of crowdsourced manufacturing since it enables interactives among manufacturers via smart contracts in a dispersed and peer-to-peer network without intermediary trust (Bahga and Madisetti, 2016). With the combination of industrial IoT and blockchain, the MaaS can gather the actual process of manufacturers in the real-time and evaluate their contributions to the entire value chain.

The different decision agent group has a distinctive role along the value chain in platform-driven crowdsourced manufacturing. Following a function-behavior-structure analysis in Section 4.1, open innovators, manufacturers, and platform can be categorized as playing a functional, behavior, and structural role, respectively. Open innovators bring design of innovative products and connections to the market to capture product value. Manufacturers act a behavior role who share their manufacturing capabilities for realization of MaaS. Platform plays a structural role who organizes manufacturing network to serve innovative projects. The cooperation of platform and manufacturers implies a reliance on value gathering tole of open innovators. On the other hand, though value gathering capabilities of the innovation products depend on the product design and connections to the market, the quality of the manufacturing and coordination of the manufacturing network will influence the satisfaction of the customer. Therefore, three decision agent clusters are closely linked with each other, and they all succeed and fail together. A product fulfillment block can secure the product fulfillment data and crowdsourcing contracts, which enables the transaction by an exchanging system to serve manufacturing industries in an opened environment (Li et al., 2018b).

MaaS should offer the service of exchanging the product fulfillment block, as well fulfillment data and crowdsourcing contracts. As a point-to-point system, blockchain network lacks a centralized server system to reduce information management and trust establishment cost, thus, employing a consensus mechanism to verify transactions by nodes in the network, which includes Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA) (Du et al., 2017). Traceability is the second essential property carried by blockchain technology, which tracks each cryptocurrency transaction among users, the information storing, and message sending in a blockchain network. Every block and corresponding verified transactions in a blockchain network are associated with unique hash values to locate a block or a transaction, as well as serve as a cryptography method for accessing any block with the hash address (Zheng et al., 2018b). Unchangeability is another important property of the blockchain technology. Once a block is mined, transactions stored in the block can be only read but not edit or delete (Nakamoto, 2008). This ensures that the information in the blockchain cannot be altered without agreement from the majority of nodes in the blockchain network. Alternatively, changes or update to the old information can be sent through new transaction to the blockchain and stored in a different block. Transparency ensures each node or user of the blockchain network can only access the same blockchain network. The synergy of traceability, unchangeability, and transparency ensures that any changes to the blockchain is detectable and traceable to all supply chain participants (Kouhizadeh and Sarkis, 2018). Similarly, for a manufacturing service provided by multiple manufacturers, a blockchain-based information system can provide secure and trust to them.

9.2 Architecture Design of Information Service System

To provide information exchanging services for process along with the platform-driven crowdsourced manufacturing, a conceptual system architecture based on industry IoT and blockchain technology is developed. As shown in Figure 9-1, a three-layer architecture establishes the interactions among decision agents, as well as the decision support and product fulfillment data served by the proposed information service system, which are resource layer, cyber layer, and contracting layer.

The contracting layer is the core of the information service system for platform-driven crowdsourced manufacturing. It provides the way of consolidating cooperation relationships among decision agents, like project manager and manufacturing service provider, by implementing blockchain technology to construct contract database and fulfillment data base. Contract database collects crowdsourcing contracts from cyber layer to record the task assignment result and the essential profile of corresponding manufacturing service provider. Project manager and service provider can request the access to the contract database and then read the contract for collaboration. A blockchain-based fulfillment database collects product fulfillment data from planning and streaming database in lower cyber layer, respectively. It provides trace of a fulfillment process, which includes decision making results and activity logs. The manager and service providers can stream their activities and supervise fulfillment process. Thus, an information exchange can be achieved.

Cyber layer provides an intermediate between the contract layer, which reflects the collaboration relationships among decision agents, and resource layer, which execute the manufacturing service on physical level. It provides a guideline for crowdsourcing task execution and testing, which is achieved by managing product specs and support the derivation of crowdsourcing contracts. A digital twin of physical crowdsourcing network is established in cyber level in the form of a cyber production system model (Chapter 8 proposes a real-time cyber model in detail). It paves way of receiving real-time status of crowdsourcing task execution from smart sensing IoTs in resource level. After the streaming data are collected in streaming database, the decision-making results can be derived and saved into planning database.

Resource layer provides the hardware framework for the platform-driven crowdsourced manufacturing community. The manufacturing resources can be reconfigured according to the crowdsourcing task derivation, as the decomposition result of product precedence diagram. After the task execution initiated, resource layer applies smart sensing IoTs to monitor material flow inter- and inner-manufacturers, which can be visualized as logistic routes and process routes, respectively. The fulfillment data stream can serve the project manager and service provider a way of sense the real-time status of crowdsourcing task execution.

Based on this architecture, the research tasks in information service system for crowdsourced manufacturing can be decomposed into two fields. The first category is using smart contracting technologies to organize crowdsourcing network. The second category is collecting and managing product fulfillment data in a distributive manner.

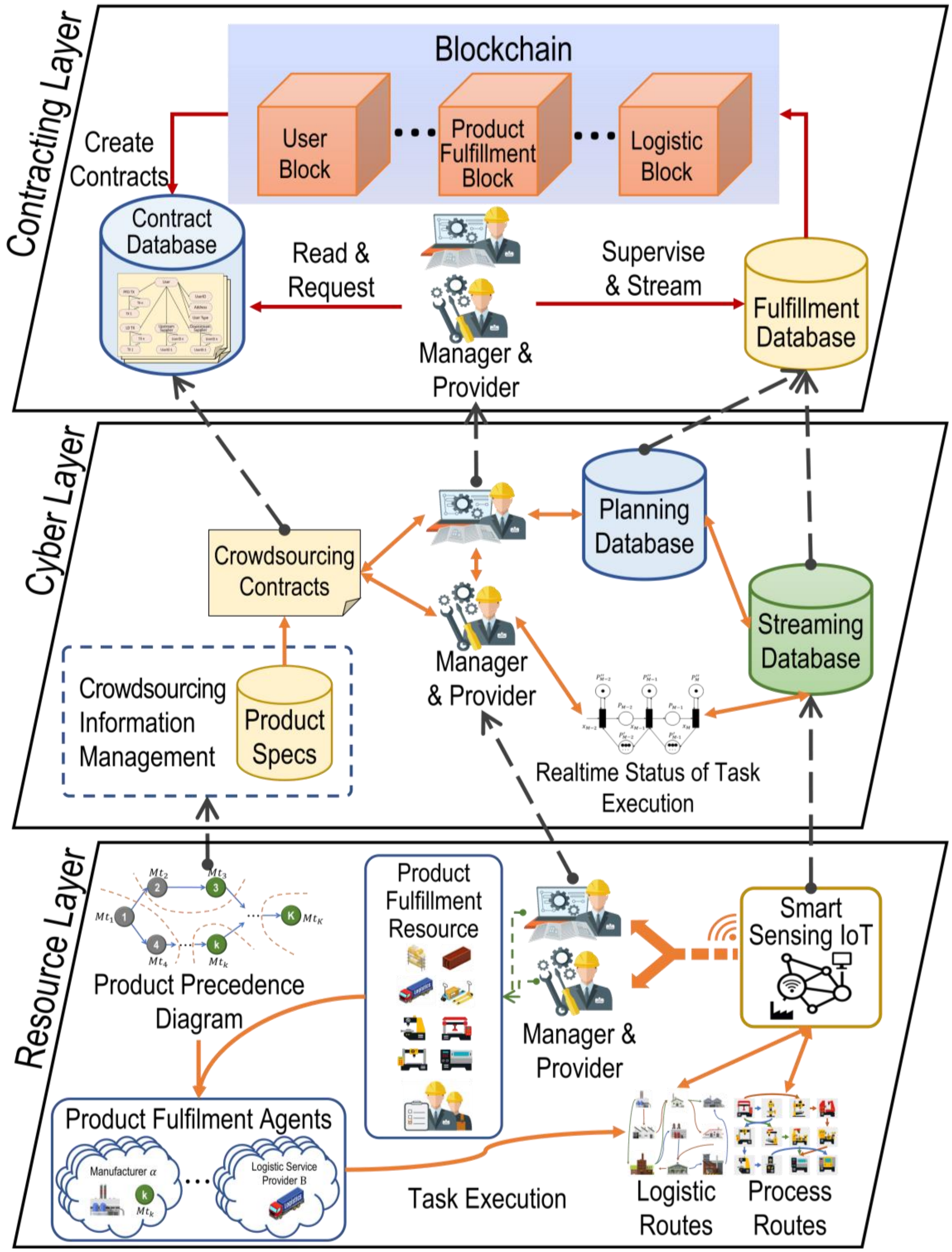


Figure 9-1 System architecture of information service system for platform-driven crowdsourced manufacturing

9.2.1 Contracting Layer

The contracting layer is the top layer of the entire system, which provides services of managing the user information, product fulfillment data, and logistic data based on a blockchain network. Each block in the network contains the unstructured data from cyber layer to the blockchain and the block information which records the transaction process. In the blockchain network, there are three types of blocks which are the user block, product fulfillment block, and logistic block. All three types of blocks contain the sender's account address, the address of the receiver (blockchain), and the transaction hash. Unstructured data in the user block stores the user information of the supply chain participants and the block type. In unstructured data, the user ID is a string data that stores participant's name, and the user type of a supply chain participants is also stored in the user block. Each user block represents one participant in the supply chain. There are three types of participants which are project manager, logistic provider, and manufacturing provider. The first few blocks in the blockchain network are always user blocks in the designed system since user information collection is prior to the supply chain fulfillment data collection. Product fulfillment block stores the extracted manufacturing data from the manufacturing provider, which reflects the manufacturing process across a certain time interval. Similar to the user block, it contains the block information and unstructured data. To record the product fulfillment progress, unstructured data have the part name information, start time, end time, and the number of parts finished. Logistic block stores the data provided by the logistic provider. Unstructured data in the logistic block contains the information of departure location, destination location, current location, and expected arrival time.

To upload the data to the blockchain network, stream product fulfillment data from the logistic and manufacturing provider are uploaded to the distributed streaming database for information extraction. There are two reasons for not uploading raw stream data to the blockchain network directly: (1) To reduce the data size, (2) To reduce the difficulties of reading product fulfillment progress. The coordinate manager supervises the whole uploading process.

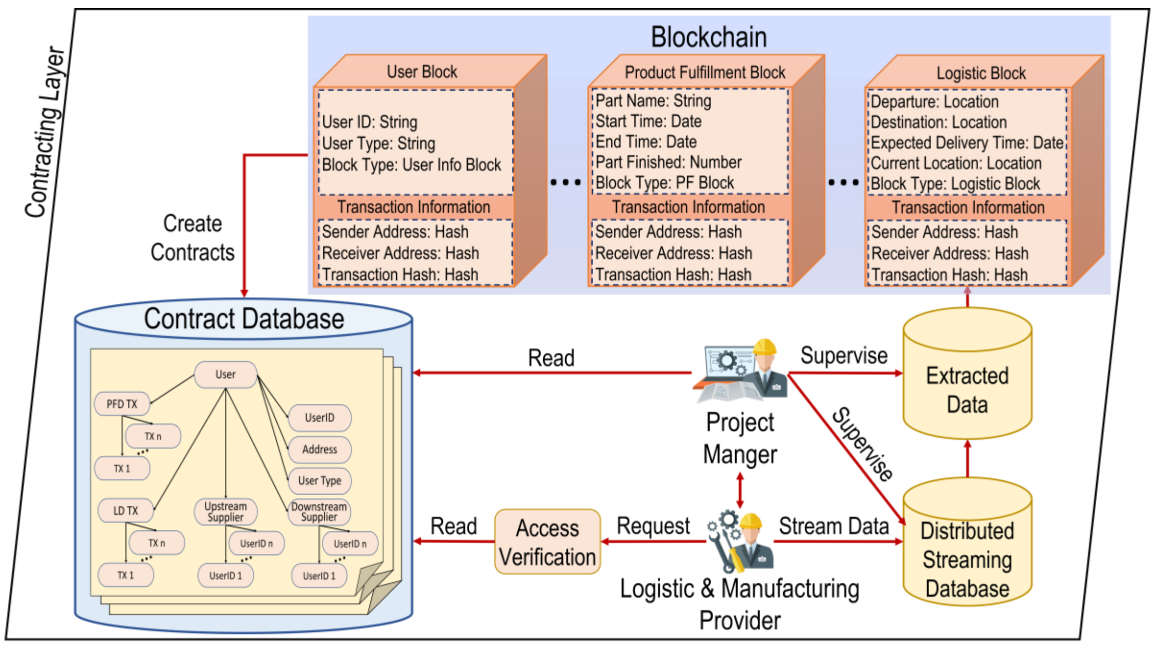


Figure 9-2 Contracting layer of blockchain-based information service system

The mechanism of the blockchain allows the stored information unerasable and traceable. However, unstructured data stored on the chain is difficult to access and manage. In the designed system, raw data is structuralized and stored. The database creation process is periodical since users keep updating product fulfillment progress in the blockchain network. As shown in the figure, the database stored the participants' information, adjacent manufacturing providers and logistic providers, manufacturing tasks fulfillment progress, and logistic tasks fulfillment progress. The project manager has full read access to the database. Data access of the manufacturing provider and logistic provider are limited where they can only access the product fulfillment progress uploaded by the upstream and downstream participants.

9.2.2 Cyber Layer

Cyber layer is the middle layer of the whole system. Product specifications are generated from the product, product family data, and product variant data. The product family specifies the alternative feature choices for similar products, and the product variant data shows a different combination of the product. The product specification clarifies the requirements for each part, required manufacturing process, and required assembly process for each product variant. To construct the supply chain of a certain product variant, its product specification is decomposed into tasks and sent to the project manager. Logistic providers and manufacturing providers are attracted to participate in the crowdsourcing task by broadcasting the specifications. A crowdsourcing contract is established based on the product specification and the negotiation-based contracting mechanism (described in Chapter 5 in detail) among the project manager, logistic provider, and manufacturing provider. Negotiation-based contracting involves the share of profit, expected product delivery date, quantity of orders, and product quality. A planning database is created based on the results uploaded by the logistic and manufacturing provider, respectively. The planning database creation is supervised by the project manager for conformity reason. The planning database act as a handbook for task execution, which includes manufacturing task specifications, logistic task specifications, precedence relationship, and execution guidelines. During task execution, the streaming database stores the fulfillment data stream from smart sensing IoTs in resource layer.

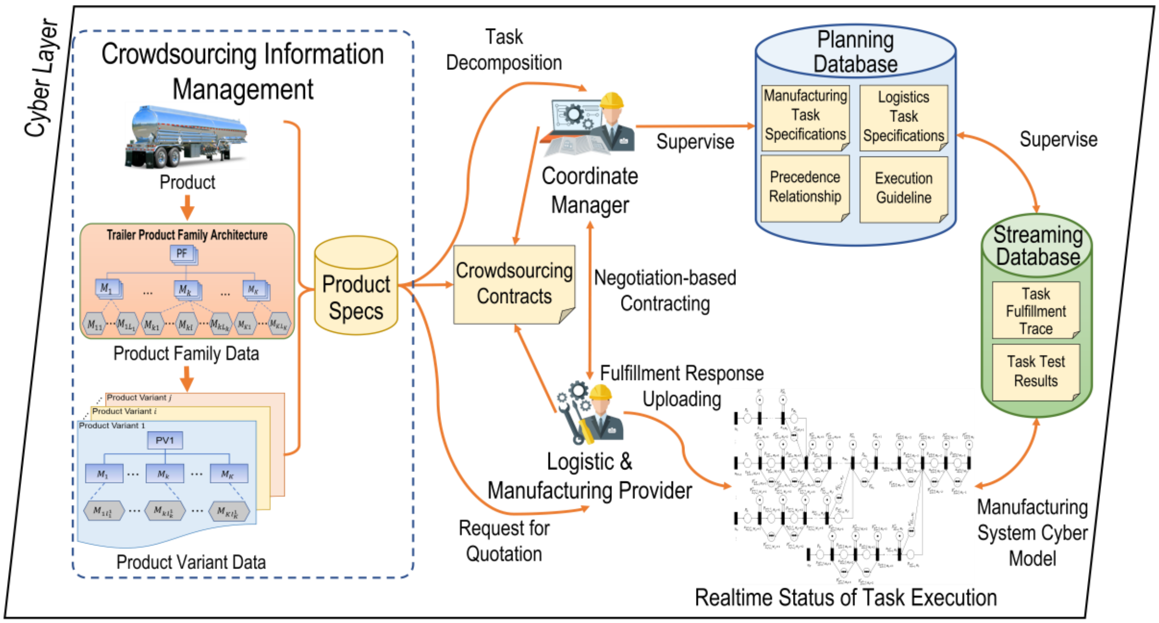


Figure 9-3 Cyber layer of blockchain-based information service system

9.2.3 Resource Layer

Resource layer is the bottom layer of the whole system. In this layer, crowdsourcing tasks are generated and distributed to each participant based on their available resources, and the stream data of each physical entity of the product fulfillment resource are collected and uploaded. The product fulfillment resource of the project manager, logistic provider, and manufacturing provider are reconfigured first. Logistic resources are resources used to receive, deliver a certain amount of material and products to participants in the supply chain. Management resources are used to coordinate and supervise the whole product fulfillment process. Manufacturing resources are the available product manufacturing machines. The product precedence diagram, developed from the contract, contains the manufacturing task M_{t_k} as shown in the figure. The manufacturing tasks are decomposed and regrouped based on the manufacturing specs and requirements. By mapping the tasks decomposed from the product precedence diagram with the product fulfillment resources, the product fulfillment process is established as shown in the figure. In the product fulfillment process, each manufacturing task is distributed to a manufacturing provider. Furthermore, logistic tasks are generated based on the precedence and physical locations of the manufacturing providers. Logistic tasks are distributed to logistic providers for establishing product fulfillment process.

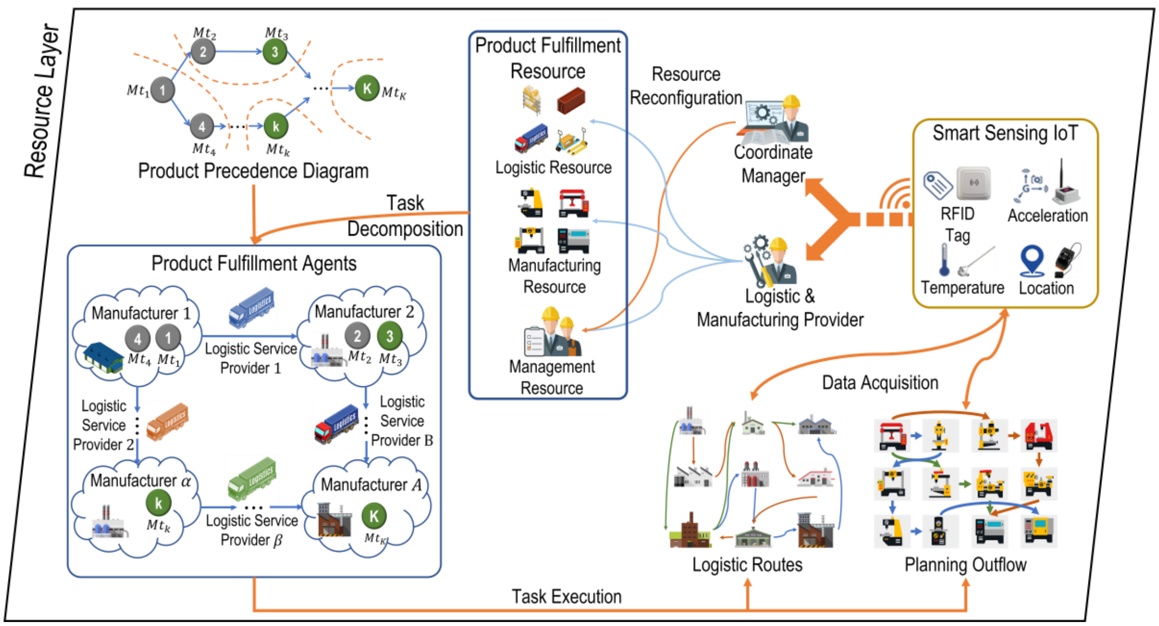


Figure 9-4 Resource layer of blockchain-based information service system

The product fulfillment process is decomposed into logistic routes and planning outflow while executing tasks. Product and materials are transported through logistic routes are from physical locations to physical locations by entities of logistic resources. In the planning outflow, part manufacturing and assembly are from machine to machine among different manufacturing providers. On the same time, stream product fulfillment data are collected while executing the logistic routs and planning outflow through smart sensing

IoT network. Local project manager, logistic provider and manufacturing provider can monitor the working condition through smart sensing IoT network. Meanwhile, stream data are also uploaded to other space for supervising and managing for the distant participants.

9.3 Blockchain-Based Smart Contracting

The architecture of the contracting management is shown in Figure 9-5. Three vital parts in the system are blockchain network, a distributed database based on Interplanetary File System (IPFS) technology, and function portal. IPFS-based Distributed storage system is the intermediate between function portal and blockchain network. On the one hand, IPFS stores large-sized raw data upload by using function portal. On the other hand, it stores the categorized extracted data imported from the blockchain network which enables function portal to provide better access and management service to stakeholders.

9.3.1 IPFS-based Distributed Storage System

IPFS is the protocol for a distributed data storage system. Without a centralized client server, IPFS relies on lots of connected nodes in the P2P network. When a node uploading a file to IPFS, a cryptographical hash address is generated as content identifier (CID). The CID is a unique “fingerprint” to the uploaded file that can permanently record the existence of a file. CID is required for accessing file uploaded by other nodes. When a file is updated in IPFS, a new CID is generated. Therefore, any changes in the uploaded file are detectable in IPFS. It can be concluded that IPFS is a high-performance and secure distributed storage system which enables users to store large-sized data and concurrent data access (Zheng et al., 2018a).

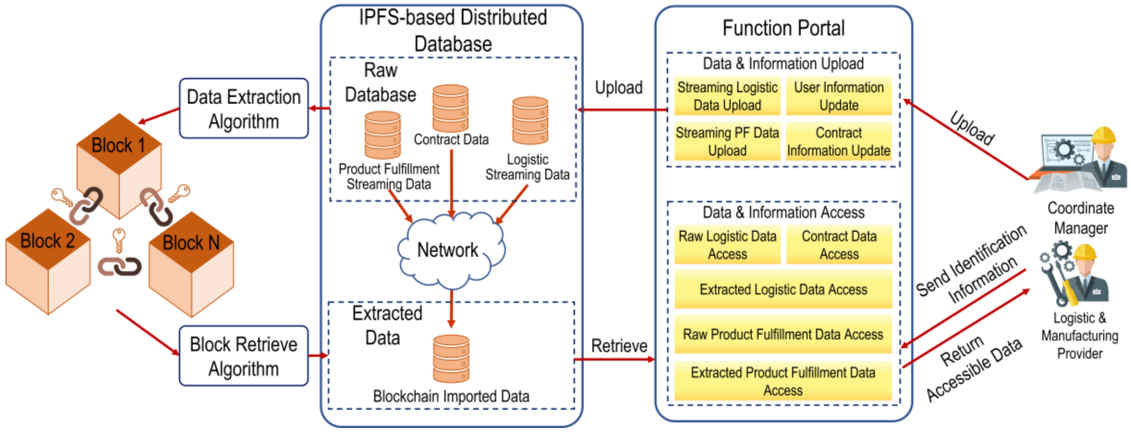


Figure 9-5 Architecture of the Contracting Management

IPFS stores raw data and extracted data of the crowdsourcing manufacturing task in the proposed architecture. The raw data includes product fulfillment streaming data, logistic streaming data, and contract data. In the product fulfillment process, the real-time task execution status is reflected and stored in the streaming product fulfillment data and logistic data collected through IoT based network. Each service provider acts as a node in the IPFS by uploading their local streaming data. Contract data contains manufacturing task specifications, logistic task specifications, precedence relationship and execution guideline, which is created and managed by the coordinate manger. Data extraction algorithm mines critical information in the raw data to reflect task execution status, which can be sent to the blockchain network through transactions. Block data retrieve algorithm can structuralize the extracted data in the blockchain and save it in IPFS. Therefore, the function portal can ease difficulties of operating and managing product fulfillment information in blockchain.

9.3.2 Function Portal

Task execution status and information, tracking, and tracing services are supported by the function portal. In data and information upload field, logistic providers and manufacturing providers update the user information, upload the streaming logistic data, update the contract information, and update the steaming product fulfillment data under the supervision of the coordinate manger.

9.3.2.1 Data and Information Upload Field

There are three service functions in data and information upload field. The first service function is “User Information Update”, which contains names of providers, contact information, and location. It is stored in the contract data in the IPFS which is managed by the coordinate manager. Therefore, the coordinate manager has the full access to the contract database. Logistic and manufacturing providers have the read access to the contract data. All participants can track the changes to the contract data through CID.

“Steaming Logistic Data Upload” is the second service function, which enables logistic provider to upload the streaming logistic through the function portal. The function portal establishes the connection between the local database and IPFS and send the CID to the contract database. File stored in the IPFS is automatically updated periodically based on the streaming data in the local database. The time interval between updating the IPFS file is determined by the negotiation between logistic providers and coordinate manger. The change to the local file is trackable through the corresponding change to the CID.

“Contract Information Update” is the third service function for coordinate manager. It allows coordinate manger to update the contract information based on the feedback and negotiation from service providers. The change to the information in contract database is trackable through IPFS.

"Streaming Product Fulfillment Data Upload" is the fourth service function for establishing the connection between manufacturer's local database that stores the raw streaming product fulfillment data to IPFS. The CID of the file stored in the IPFS is updated due to the change of data stored in the local database for tracking purpose. Manufacturing providers and the coordinate manger can reach the agreement about the time interval between updating.

9.3.2.2 Data and Information Access Field

The Data & Information Access layer provides service for accessing the task execution status of logistic providers and manufacturing providers. In a supply chain network, the downstream participant can prepare for providing the service better with knowing the task execution status from its upstream participant. Based on different types of tasks, different access permission is given to manufacturing provider and logistic provider. The access permission scope defines whose and how much information can be viewed by a participant. Manufacturing providers, logistic providers and the coordinate manager can determine the permission scope for each role to optimize the task execution process. Data & Information Access layer can retrieve the data for a participant based on its permission scope. Five services are provided in this layer.

"Contract Data Access" allows participant to read the accessible information stored in the contract data includes contract information, task execution guidelines, and product specifications. The CID makes it possible for participant to view all history versions and the latest data.

"Raw Logistic Data Access" provides access for participants to view the accessible raw logistic data through the function portal. Since the data is stored in the IPFS, this service returns all CID of the data to the user. Therefore, the participant can track both the latest and history task execution status of other participants.

"Extracted Logistic Data Access" provides access for extracted logistic data. As mentioned in the previous section, extracted logistic data is imported from blockchain and stored in the blockchain imported database. This service retrieves the accessible logistic service execution status to the participant through sending the related blockchain transaction hashes. With using the retrieved transaction hashes, message that carries extracted logistic data is accessible to the participant.

"Raw Product Fulfillment Data Access" plays a similar role as "Raw Logistic Data Access". This service allows the participant to view all versions of the accessible raw product fulfillment data through CIDs.

"Extracted Product Fulfillment Data Access" is a dual of "Raw Product Fulfillment Data Access", which allows the participant to access the latest extracted product fulfillment data through transactions hashes.

9.3.3 Blockchain Network

Transactions that contain simplified user information and extracted task execution status are sent to blockchain network.

"User Information" is the first type of extracted data in blockchain. The complex information of manufacturing providers and logistic providers is simplified to user type and user ID. The user type identifies if the participant is a logistic provider or manufacturing provider. User ID is the name of service provider. A label is also added to the sending transaction for identifying the transaction is a User Data transaction. User Data extraction process is executed when the participant uses the User Service Update service in the function portal. The complex user information is firstly uploaded to IPFS and then extracted to the blockchain network. While sending the transaction, a blockchain network account is created for the user. Sender's address is recorded while sending the transaction and will be mapped with the User ID.

"Product Fulfillment Data" is the second type of data stored in the blockchain network. The raw streaming product fulfillment data is transacted to the blockchain network after users updating the raw product fulfillment data in IPFS. The raw product fulfillment data is extracted to message that can reflect the task execution status. For example, it contains the time that the file is updated in the IPFS, the time of starting to collect local product fulfillment data, the time of ending collecting local product fulfillment data, the physical location of the task execution, the name of the executing task and the number of finished tasks. In the extracted data, status tracked by small time interval is enlarged. Furthermore, multiple manufacturing processes are simplified to one process with yes and no state. Despite the task execution status, the label is added to the extracted data for identifying if it is a product fulfillment data. The extracted data is transacted to the blockchain network by using the same account while sending the extracted user information.

"Logistic Data" is the third type of extracted data. The raw logistic data stored in the IPFS is extracted with the similar mechanism to the product fulfillment data extraction. Data extraction algorithm simplified the status changes in raw data. Extraction enlarges time interval of the status tracking and simplified the complex transportation process. Blockchain network receives the logistic data transaction after the user updates the raw logistic data in IPFS.

9.4 IoT-Based Distributed Data Streaming Management

IoT technology provides an access to each resource and entities in platform-driven crowdsourced manufacturing and serves as a prerequisite for real-time task execution through a ubiquitous network. Sensing real-time task execution status not only significantly gears forward the traditional manufacturing landscape to a smarter and more efficient era, but also envisions a decentralized crowdsourcing manufacturing network through a cyber-physical production system (Lee et al., 2019). Figure 9-6 shows an IoT-based cyber-physical system architecture for platform-driven crowdsourced manufacturing. The integration of physical infrastructure on the lower level and crowdsourcing cyber network on the higher level provides real-time tracing capability by implementing smart sensors. The RFID tags and barcodes can be used for tracing product fulfillment and logistic service resource through a MaaS service process. Synthesizing product fulfillment data streamed by IoT can reflect the status of inner- and inter-manufacturer WIP and material flow.

The raw tracing data is firstly uploaded to the local database, which can be viewed and utilized by local manufacturers for real-time production monitoring and control. Copies of these raw data are transferred as a shared and extracted copy to blockchain-based network. Once entering the blockchain-based network, any updates to the tracing data can be tracked and viewed by corresponding stakeholders, which includes coordinate manager and other service provider.

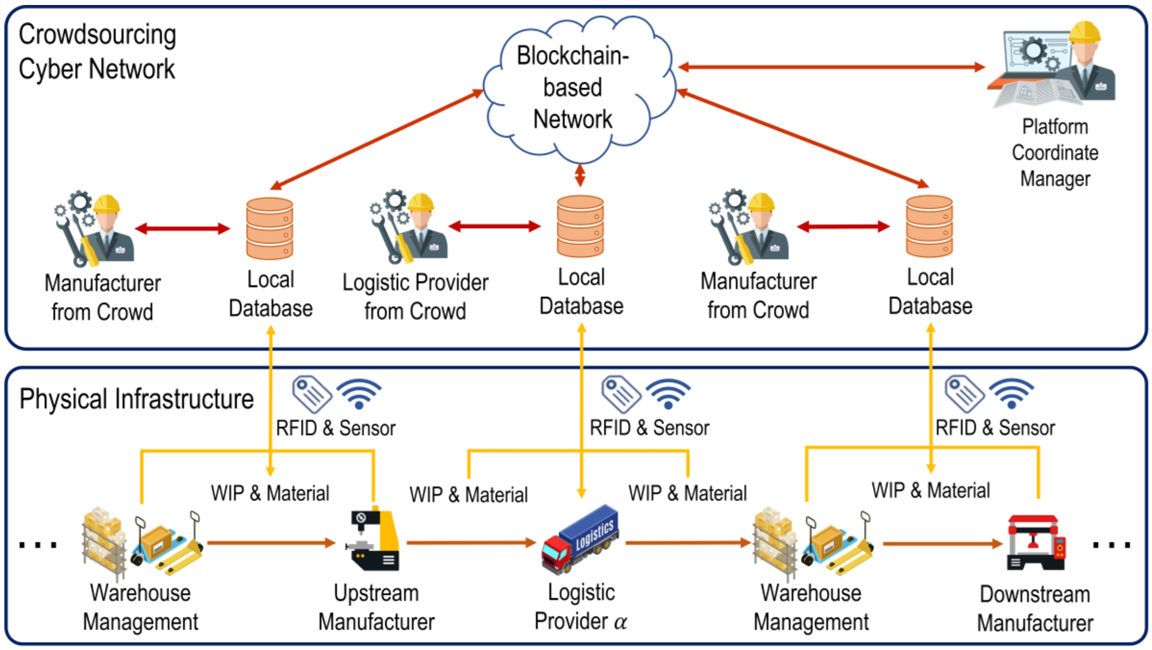


Figure 9-6 IoT-based cyber-physical system for platform-driven crowdsourced manufacturing

Figure 9-7 demonstrates a process of operating the information service system. The raw product fulfillment data is first collected from the local network and updated after each locally determined time interval. The streaming data file in the local storage is copied and transferred to IPFS with a different time interval. As discussed in the Section 9.3, this time interval is determined by both coordinate manger and local service provider based on the requirements and situations of the specific task. The connection between local storage and IPFS is established by manufacturing providers and logistic providers through a connection setup interface. The connection establishment activity in the interface is monitored by the coordinate manager. The connection setup interface also provides basic service for users to check and manage the connection status of local database. For example, the user can check the hash address list of one updating raw data file and the last time when it is updated. In the proposed system, the coordinate manager, manufacturing provider, and logistic provider utilizes connection setup interface for sharing contract, raw product fulfillment data, and raw logistic data, respectively.

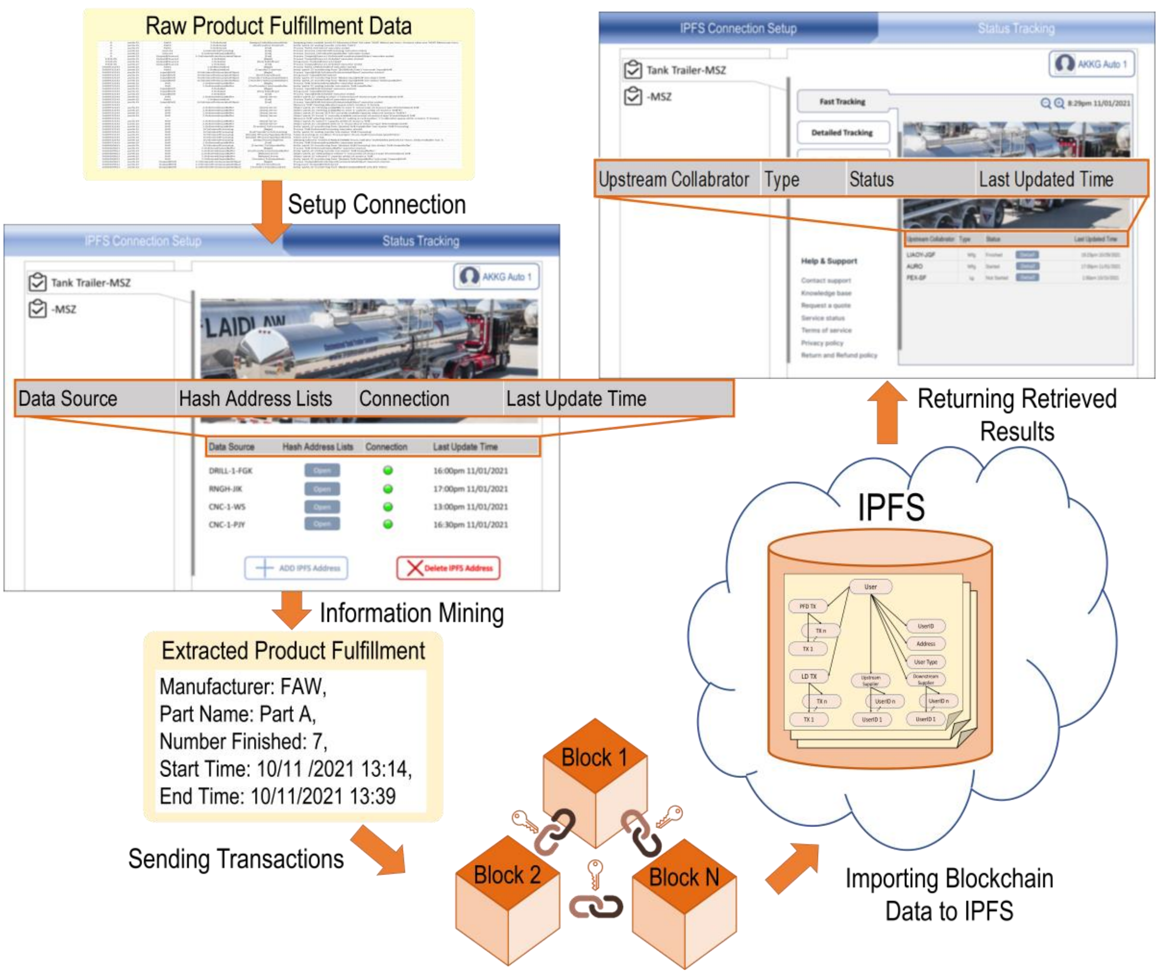


Figure 9-7 Operation flow of the integrated information service system

Information is extracted from the raw data once after update. The extracted data summarizes the task execution status based on the raw data. Among three types of raw data to be extracted, raw product fulfillment data contains the most information, which traces unsynchronous processes in data files. For example, the process can be defined by several elements: the time when process happen or ends, entity ID of the entity that interacts with the process, and the resource grabbed for executing the process. For each entity entering the defined product fulfillment system, it can only leave the system after finishing the last designed step. Therefore, if an algorithm can count the number of entities that have been through the last process for leaving the system, the number of finished tasks can be determined. With the start and end time in the raw data, the name of manufacturer who updated the raw data, the task name, and determined number of finished tasks can be extracted. Table 9-1 shows pseudo-code of product fulfillment data extraction algorithms.

Table 9-1 Example pseudo-code of product fulfillment data extraction

Input: *1. Name of the task.*

2. Table that record when does a process happen in the system. The process is defined as: an entity occupies a resource for executing a certain action and one entity requires several processes for leaving the system.

Output: *Extracted data that summarize the overall action exaction results.*

begin
    Acquire the time when the first process happens in the table as t_{start}*;*
    Acquire the time when the last process ends in the table as t_{end}*;*
    Identify the last process associate with entities for leaving the system p_l*;*
    $L \leftarrow \{\}$;
    Foreach *row in table:*
        Obtain the EntityID in each row and save as I;
          If *I* **not in** *L:*
              Append *I in the list L;*
          end
    end
    $N \leftarrow$ **length** $of\ L$;
    return t_{start};
    return t_{end};
    return *name of task;*
    return *N;*
end

The extracted data is sent to the blockchain network. The extraction data provides better feedback to the distant viewer than raw data. The reduced size of data makes the proposed information service system can adapt most blockchain technologies. For better data management services, data in blockchain network is structuralized and stored in the IPFS. The algorithm of structuralizing the blockchain data is shown in Table 9-2. A struct is firstly defined which contains fields of user ID, account address, user type, list of sent extracted product fulfillment data, list of sent extracted logistic data, list of user ID of adjacent upstream or downstream service providers. Once connected to the blockchain network, the algorithm goes through each transaction of each block. For transactions contain the user information, an instance of the user struct will be created and named by sender's user ID. Sender's address and adjacent service providers are also updated at this time. For transactions contain the product fulfillment data, the algorithm adds that transaction hash to struct instance named by sender's user ID. Similarly, for transactions contain the logistic data, the algorithm adds that transaction hash to struct instance named by sender's user ID. After looping all blocks, a list is created to save all defined struct instances and exported to IPFS.

While accessing extracted data through status tracking interface, structuralized data acts as a catalog which returns the cyber location of accessible information to the user. After sending request of access through status tracking interface, the struct instance named by user ID is obtained and reviewed by the access management algorithm. Then, the algorithm searches for the struct instances named by user ID that appears in the upstream service provider from the list. Once the struct instance of these service provider are obtained, all transaction hashes sent by these service providers will be returned to user. User can view the extracted data in the blockchain by using transaction hashes.

Table 9-2 Pseudo-code of structuralizing blockchain information

```
struct User{
    string userID;
    string address;
    string usertype;
    list PFDtx;
    list LDtx;
    list upstream;
    list downstream;
    function addPFD(PFD) {
        self.PFDtx.append(PFD);
    }
    function addLD(LD) {
        self.PFDtx.append(LD);
    }
    function addupstream(US) {
        self.upstream.append(US);
    }
    function adddownstream(DS) {
        self.downstream.append(DS);
    }
}
Connect to Blockchain network

for i in range(1 to number of block ) {
    list tx_list ← list of transactions in block i;
    for c_tx in tx_list {
        string extracted_data ← string stored in c_tx
        if extracted_data contains user information{
            string c_uid ← userID stored in extracted_data;
            User globals()c_uid;      //create a User struct instance named with string in c_uid
            globals()c_uid.address ← sender's account address;
            globals()c_uid.usertype ← usertype information in extracted_data;
            adddownstream(string that contains downstream userID);
            adduptream(string that contains upstream userID);
        }
        elseif extracted_data contains product fulfillment data{
            addPFD(c_tx.address);   //append the current transaction hash to PFDtx field
        }
        elseif extracted_data contains user logistic data{
            addLD(c_tx.address);   //append the current transaction hash to LDtx field
        }
    end
```

9.5 Case Study of a Tank-Trailer

A case study of tank-trailer crowdsourced manufacturing verifies the performance and feasibility of the proposed blockchain-based information service system. The product is fulfilled through a collaboration of five crowdsourced manufacturers and two logistic providers. This case study demonstrates the functions of smart contracts, distributed information management, and product fulfillment data streaming based on IoT technology. There are four parts in this case study: the description of the crowdsourcing case, smart contracts generation, information management for task execution, and product fulfillment streaming data services.

The information system developed for the case study utilizes several environments at different stages of the development process. To illustrate the development process, environments and tools are categorized by the purpose of usage which is shown in Table 9-3. The case study is based on Ethereum blockchain network due to its advanced developer support Ethereum blockchain stores programs create digital twin of the traditional contract as “Smart Contracts” through executing functions. With using smart contract, secured management to the transactions is achieved. Therefore, building a blockchain test network based on Ethereum is the first usage purpose which employ Truffle and Ganache-cli. Truffle is a development framework for the Ethereum network which provide services like smart contract management and test network management. By utilizing truffle, a personal Ethereum test network can be established, and user have the options to adjust network properties like block size, block mining logic, and the cost of sending transactions. Ganache-cli is auxiliary tool for the Truffle in the case study, it can access the Ethereum test network built by Truffle and show the blockchain information in a GUI. It acts as a demonstration tool for the test Ethereum network.

Table 9-3 Development environments and tools

Usage Purpose	Tools and Environment
Building blockchain test network	Ubuntu Linux 21.10, 24 processors, 32 GB RAM Ganache-cli Truffle
Simulating PF data	Simio
Developing web-based raw PF data upload interface	IPFS JavaScript, CSS React.js Web3.js Meta Mask Solidity
Developing blockchain information importer	Web3.py IPFS

This case study utilizes DES software to generate simulated product fulfillment data to verify real-time streaming function. Simulations are commonly used to verify design performance during the late development stages. To apply DES, Simio is a software broadly applied. Simio offers model-to-model transition, and the simulations can be used to derive and export various outputs. It can export trace of the simulation model as the product fulfillment data.

Developing a web-based raw product fulfillment data upload interface is essential in this case study. The interface will transfer the user uploaded file to the IPFS through web browser. Meanwhile, it can process the raw product fulfillment data for extraction and send the extracted task execution status to the blockchain network. IPFS is the foundation tool here for providing the distributed file storage service. The distributed file storage provides the remote access service to users. React is a JavaScript library for developing web-based use interface which is the main development tool for the web UI design. Web3 is a JavaScript library that used to let user interact with a Ethereum node using HTTP, IPC and WebSocket. MetaMask is crypto currency wallet that can manage Ethereum account though web browser. Smart contract is written in Solidity for executing blockchain command. In summary, when using the web-based interface, Web3 library establishes the connection between IPFS to the test blockchain network, MetaMask provides account management services to user for confirming or rejecting a blockchain transaction, and smart contract executes blockchain commands to interact with the blockchain network.

A blockchain information importer algorithm is also a critical element in this case study. As mentioned in the previous section, it can categorize and structuralize the transaction information, the structuralized data is sent to IPFS. Web3 python library assists this development process which allows user to retrieve information from the blockchain test network.

9.5.1 Description of crowdsourcing case

This chapter studies product is demonstrated in Figure 5-2, which shows the genetic product process structure of the tank-trailer product variant. After crowdsourcing task decomposition and derivation, five tasks are generated. Five product fulfillment providers and two logistic providers are selected to participate crowdsourcing task execution. A crowdsourcing supply network can be constructed.

Figure 9-8 demonstrates the scenario of the tank-trailer crowdsourcing supply network where five manufacturers act as the product fulfillment providers. Logistic providers transport the finished sub-assembly between product fulfillment providers.

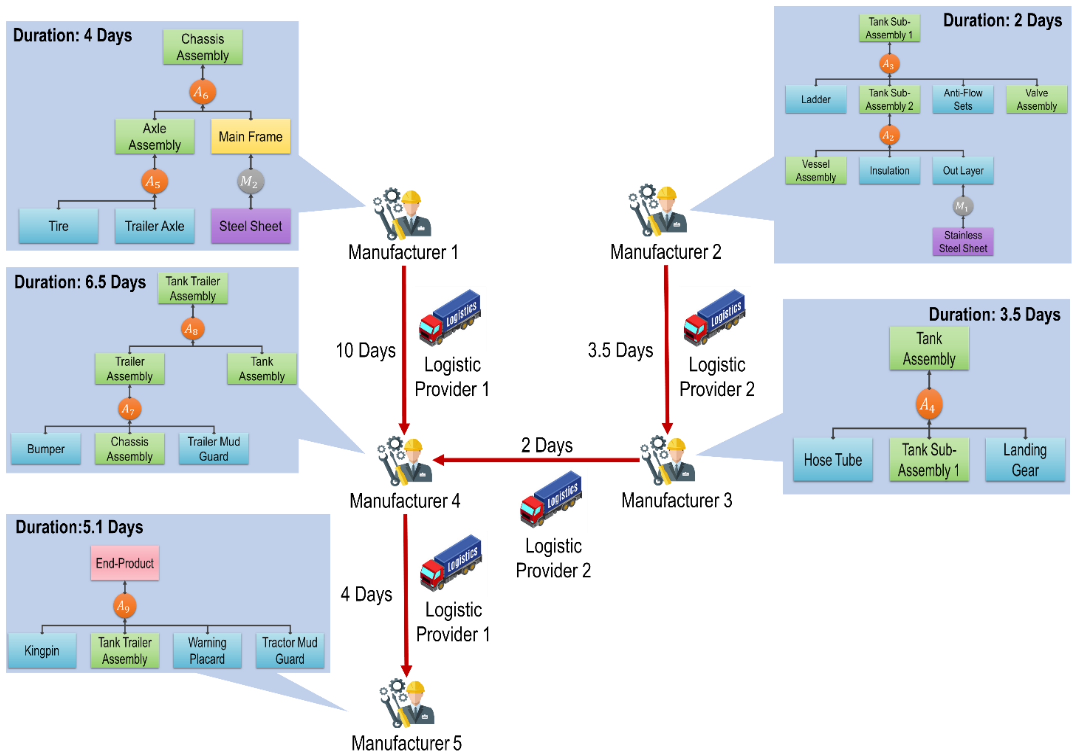


Figure 9-8 Crowdsourcing supply network for tank trailer

9.5.2 Smart contracts generation case

Smart contract “Storage” is deployed to the blockchain network after initializing the test network through truffle as shown in Figure 9-9. The block size in the test network is defined by the “gas limit” which is 6721975. To simulate the proof of authority consensus mechanism in the test network, a new block is mined at a time interval. The block 0 is mined after the deployment of smart contract “Storage”. An Ethereum address “0xF7955da8a54C23547c86f379047c852D57B54886” is assigned to the deployed contract as shown in Figure 9-9. All kinds of information uploading to the blockchain network are in the form of sending transactions to this address. Smart contract “Storage” acts as a portal between users to blockchain network which verifies identities of users and receives the information.

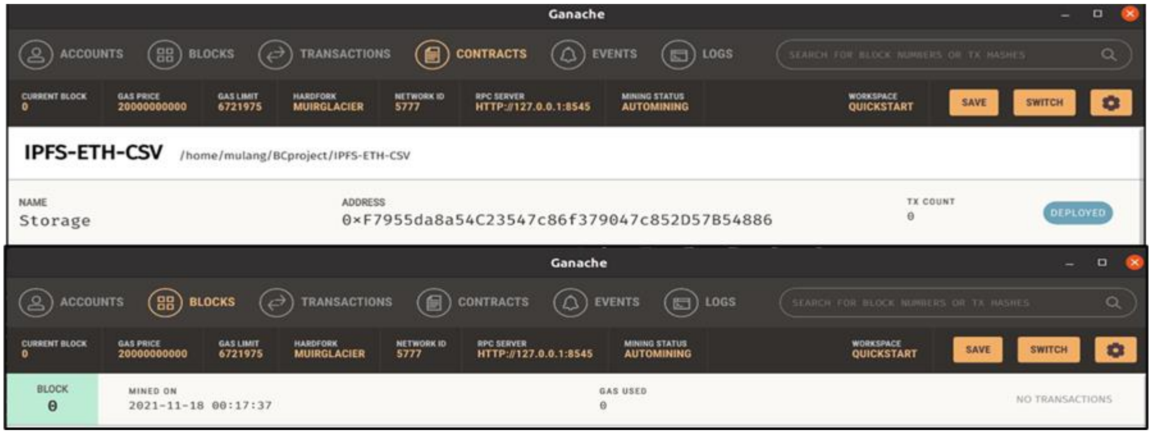


Figure 9-9 Smart contracts and its storage block

"Storage" contains three functions in the case study which are user information upload, product fulfillment data upload, and logistic data upload. "Storage" play as a significant role for only allowing certain types of users to upload certain types of data. These three functions are matched with services introduced in Section 9.3.2. However, function in the smart contract only contain partial described service which require assistance of other components for providing a complete service.

After the initialization, coordinator manager firstly registers information of both product fulfillment providers and logistic providers to the blockchain network through using user information upload service. In the case study, the coordinate manger must enter three types of information for each user which are Ethereum account address, user id, and user type. Since number of coordinate managers are small the case study, their Ethereum account addresses are stored in the smart contract for verification. If the transaction sender's address is not matched with stored addresses, the non-coordinate manger user can't access the user information upload function. Smart contract is not able to finish the service alone, the web-based interface and MetaMask acts as the frontend for direct interaction with the coordinate manager. In the case study, coordinate manger uploads information of five product fulfillment providers and two logistic providers. Figure 9-10 shows an example of uploading information of "Manufacturer 1" to the blockchain network. As shown in the figure, coordinate manager enters account address, user id "M1" and user type "1" in the input box, after clicking the button "Write Information to ETH Blockchain", a transaction is made from coordinate manager's account address to the address of "Storage". MetaMask detect the action and notify the user to confirm the transaction. Once coordinate manager confirm the transaction, the smart contract works at the backend which verifies sender's address. Information is stored in the blockchain through smart contract through the first transaction. The input in the transaction contains a string "User, M1, 1", where "User" is a label that means this transaction contains user information. "M1" is the user ID and "1" is the user type that means "M1" is a product fulfillment provider. Another transaction is sent after the first one where account address of manufacturer 1 and its user ID "M1" are mapped. In the mapping process, address is the key and "M1" is the value. This mapping is sent to the blockchain through the second transaction. The mapping process is significantly important for other two functions.

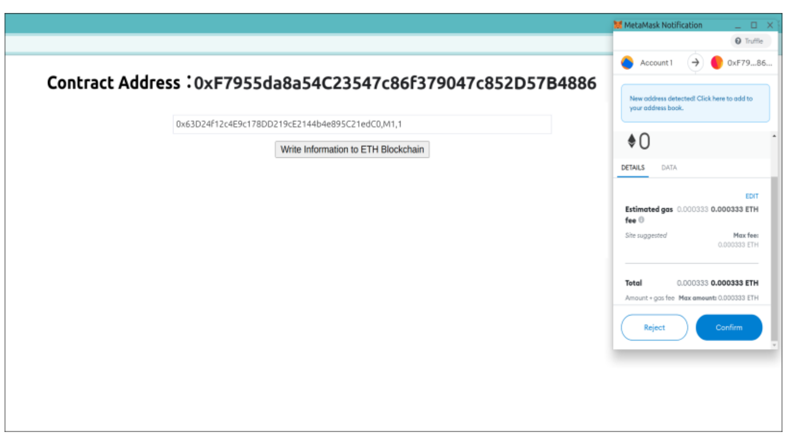


Figure 9-10 Example of uploading information to the blockchain network

Product fulfillment data upload function includes two parts, the first part is to verify sender's information. When the product fulfillment provider uploads its raw streaming data in the product fulfillment web-based interface, product fulfillment data upload function employs product fulfillment provider's account address as key to retrieve the user ID mapped with the address. If the user ID contains letter "M", the function allow user to send product fulfillment transaction to the blockchain. This verification is a simplified case which similar algorithm could be applied to use other elements for identity verification. A detailed use case for product fulfillment providers will be illustrated in Section 9.5.4. Logistic fulfillment data upload function works with a similar algorithm. Instead of letter "M", "L" is used for checking the types of users since user IDs of two logistic providers are "L1" and "L2" in the case study.

9.5.3 Information management for task execution

The information management involves both raw streaming data stored in the IPFS and extracted data stored in the blockchain. In the case study, information of crowdsourced task, task specifications, and supply chain precedence file are uploaded to IFPS through the coordinate manger. The hash address called CID is the pointer to uploaded file which is shared with all participants in the case study. To access the file, participants can add the hash address after "https://ipfs.io/ipfs/" and browse the HTTP address in the web browser. Files stored in IPFS is immutable, participants can always access the shared file through hash address. Coordinate manger can upload an updated file to IPFS if necessary. However, coordinate manger always needs to share the hash address to other participants.

For product fulfillment providers and logistic providers, uploading raw data to IPFS is integrated with the web-based interface. After uploading the file to IPFS, information is extracted from the file and sent to the blockchain network with generated CID. Therefore, the raw data CID and corresponding extracted task execution information are stored together in the blockchain. To increase the versatility, blockchain stored information is categorized as a json file. The json file is stored in IPFS which can be accessed through web-based interface. Figure 9-11 show the json file of imported categorized blockchain information of manufacturer 4. When manufacturer 4 use the web-based interface to access the task execution status of its upstream with using userID “M4” and account address, algorithm integrated in web searches struct contains “M4” in the userID field if it passes identity verification. As discussed in the previous section, coordinate manger, product fulfillment provider and logistic providers determine the access scope for users.

As shown in Figure 9-11, upstream field of manufacturer 4 contains “M1”, “L1”, “M3”, “L2” which represents accessible upstream data. Therefore, struct that contains these userIDs is searched and the material stored in their LDTX and PFDTX field is retrieved back to manufacturer 4.


```
{
"userID"                :M4,
"usertype"              :1,
"Downstream"            :[L1,M5],
"Upsteam"               :[M1,L1,M3,L2],
"LDTX"                  :[],
"PFDTX"                 :[
                         0x46970cc7ccdcfc2a68f39a86cc2dd0b258c8feef8af358df06df274c0db33757,
                         0x165a81164960e95fr5e174f40a4c75fe72e907ef05aed8b01777008cb5be5793,
                         0xc8de03a730816696d3068bf202eec44b4140e8a63fb71a3123bc3e99ed019011
                         ]
}
```


Figure 9-11 Example of upstream field for manufacturer 4

9.5.4 *Product fulfillment streaming data execution services*

In this section, detailed process of raw product fulfillment data upload and extraction process is illustrated. Raw product fulfillment data is generated through Simio model. Figure 9-12 shows the model layout of five manufacturers in Simio where an entity goes through several servers for simulating the machining process.

In each Simio model, an entity is assumed to be finished once it entered the sink object at the end of assembly line. Model trace data in CSV format generated by Simio simulation is employed as raw product fulfillment data. As a DES software, model trace data of the simulation run records every event happening during the simulation chronologically. Figure 9-13 shows an example of model trace data. Events in the simulation is defined by five domain which are time, entity ID, object name, process ID, step name and action. Simulation is paused for several times for each model, and a model trace data is generated at the pause to simulate the raw data acquisition at each time interval.

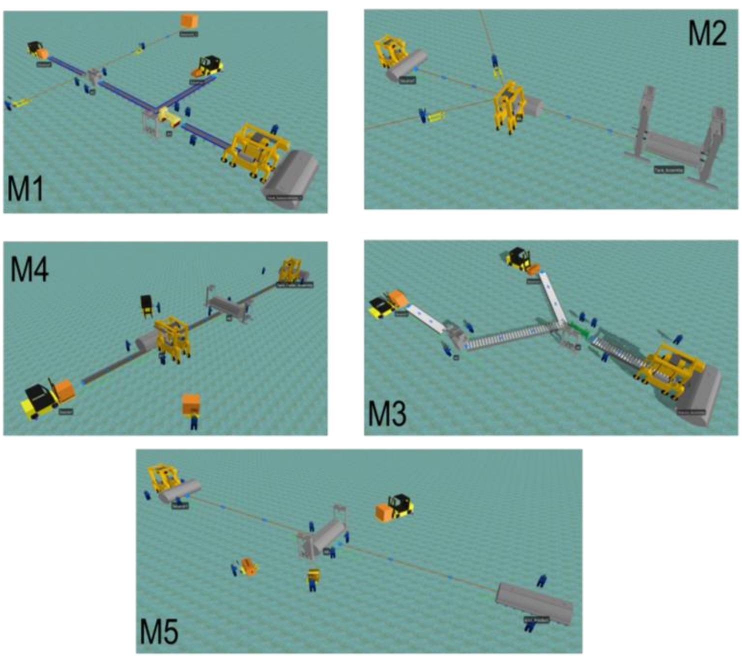


Figure 9-12 Simio simulation model of manufacturer

Time(Hour	EntityID	ObjectName	ProcessID	StepName	Action
0	--	--	--	--	System initialization completed.
0	--	--	--	--	Time until next event of timer 'Source1.EntityArrivals' is '0' Hours.
0	--	--	--	--	Firing event 'Source1.EntityArrivals.Event'.
0	--	--	--	--	Time until next event of timer 'Source1.EntityArrivals' is '0.00655278052637456' Hours.
0	Source1	Source1	0.OnEntityArrival	[Begin]	Process 'Source1.OnEntityArrival' execution started.
0	Source1	Source1	0.OnEntityArrival	[Create] Entities	Creating '1' new object(s) of entity type 'partA'.
0	Source1	Source1	0.OnEntityArrival	[Create] Entities	Created object 'partA.15'.
0	partA.15	Source1	1.OnEntityArrival	[Transfer] ToProcessing	Entity 'partA.15' transferring from '[FreeSpace] Source1' into station 'Source1.Processing'.
0	Source1	Source1	0.OnEntityArrival	[End]	Process 'Source1.OnEntityArrival' execution ended.
0	partA.15	Source1	2.OnEnteredProcessing	[Begin]	Process 'Source1.OnEnteredProcessing' execution started.
0	partA.15	Source1	2.OnEnteredProcessing	[EndTransfer] IntoProcessing	Entity 'partA.15' ending transfer into station 'Source1.Processing'.
0	partA.15	Source1	2.OnEnteredProcessing	[Transfer] ToOutputBuffer	Entity 'partA.15' transferring from '[Station] Source1.Processing' into station 'Source1.OutputBuffer'.
0	partA.15	Source1	1.OnEntityArrival	[End]	Process 'Source1.OnEntityArrival' execution ended.
0	partA.15	Source1	0.OnEnteredOutputBuff	[Begin]	Process 'Source1.OnEnteredOutputBuffer' execution started.
0	partA.15	Source1	0.OnEnteredOutputBuff	[EndTransfer] IntoOutputBuffer	Entity 'partA.15' ending transfer into station 'Source1.OutputBuffer'.
0	partA.15	Source1	0.OnEnteredOutputBuff	[Transfer] ToOutputNode	Entity 'partA.15' transferring from '[Station] Source1.OutputBuffer' into node 'Output@Source1'.
0	partA.15	Output@Source1	1.OnEnteredFromAssoc	[Begin]	Process 'Output@Source1.OnEnteredFromAssociatedObject' execution started.
0	partA.15	Output@Source1	1.OnEnteredFromAssoc	[Fire] EnteredEvent	Firing event 'Output@Source1.Entered'.
0	partA.15	Output@Source1	1.OnEnteredFromAssoc	[Transfer] ToOutboundLink	Entity 'partA.15' transferring from '[Node] Output@Source1' onto link 'Path1'.
0	partA.15	Path1	3.OnEntered	[Begin]	Process 'Path1.OnEntered' execution started.
0	partA.15	Path1	3.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'partA.15.Movement.Rate' the value '5040' Meters per Hour. Previous value was '5040' Meters per Hour.
0	partA.15	Path1	3.OnEntered	[EndTransfer] OntoPath	Entity 'partA.15' ending transfer onto link 'Path1'.
0	partA.15	Path1	3.OnEntered	[End]	Process 'Path1.OnEntered' execution ended.
0	partA.15	Source1	2.OnEnteredProcessing	[End]	Process 'Source1.OnEnteredProcessing' execution ended.
0	partA.15	Source1	0.OnEnteredOutputBuff	[End]	Process 'Source1.OnEnteredOutputBuffer' execution ended.
0	partA.15	Output@Source1	1.OnEnteredFromAssoc	[End]	Process 'Output@Source1.OnEnteredFromAssociatedObject' execution ended.
9.92E-05	partA.15	Output@Source1	1.OnExited	[Begin]	Process 'Output@Source1.OnExited' execution started.
9.92E-05	partA.15	Output@Source1	1.OnExited	[Fire] ExitedEvent	Firing event 'Output@Source1.Exited'.
9.92E-05	partA.15	Output@Source1	1.OnExited	[End]	Process 'Output@Source1.OnExited' execution ended.
0.000732	partA.15	Path1	1.OnReachedEnd	[Begin]	Process 'Path1.OnReachedEnd' execution started.
0.000732	partA.15	Path1	1.OnReachedEnd	[Transfer] FromPath	Entity 'partA.15' transferring from '[EndOfLink] Path1' into node 'Input@Drill'.
0.000732	partA.15	Input@Drill	0.OnEnteredToAssociat	[Begin]	Process 'Input@Drill.OnEnteredToAssociatedObject' execution started.
0.000732	partA.15	Input@Drill	0.OnEnteredToAssociat	[Fire] EnteredEvent	Firing event 'Input@Drill.Entered'.
0.000732	partA.15	Input@Drill	0.OnEnteredToAssociat	[Transfer] ToAssociatedObject	Entity 'partA.15' transferring from '[Node] Input@Drill' into object 'Drill'.
0.000732	partA.15	Input@Drill	0.OnEnteredToAssociat	[Transfer] ToAssociatedObject	Entity 'partA.15' transferring from '[Node] Input@Drill' into station 'Drill.InputBuffer'.
0.000732	partA.15	Drill	2.OnEnteredInputBuffe	[Begin]	Process 'Drill.OnEnteredInputBuffer' execution started.

Figure 9-13 Example model trace data from Simio

Manufacturer 1 to 5 upload their trace model data through the web-based interface. As shown in the Figure 9-14, manufacturer 1 uploads the model trace data through the interface, the file is firstly saved to IPFS, and a hash address is generated and returned on the web page. By clicking on “Write information to ETH blockchain”, an integrated algorithm processes the raw CSV file for information extraction. The algorithm employs the basic mechanisms introduced in table 9-1 which counts number of objects that enters the sink in simulation.

After the information extraction, the web-interface attempts to send transactions to the blockchain from the current Ethereum account logged in the MetaMask, which notifies user about transaction confirmation. Once the transaction is confirmed, the product fulfillment data upload function in the deployed smart contract “Storage” receives the transaction. “Storage” employs the sender’s account address as the key to retrieve the mapped user ID for verification. “Storage” sends the extracted product fulfillment information and the raw data file hash of manufacturer 2 is sent to the blockchain network after passing the identity verification. The actual sent string is “PFD, Part_name: part1, Number_finished: 8, Time_used(days): 0.71382, Qma8eBBR3eKZ1uqGn7Xx83cJakMCuK1qn4jRMowRq7zil". "PFD" indicates that the transaction contains product fulfillment data. The hash address of the file is attached to the end.

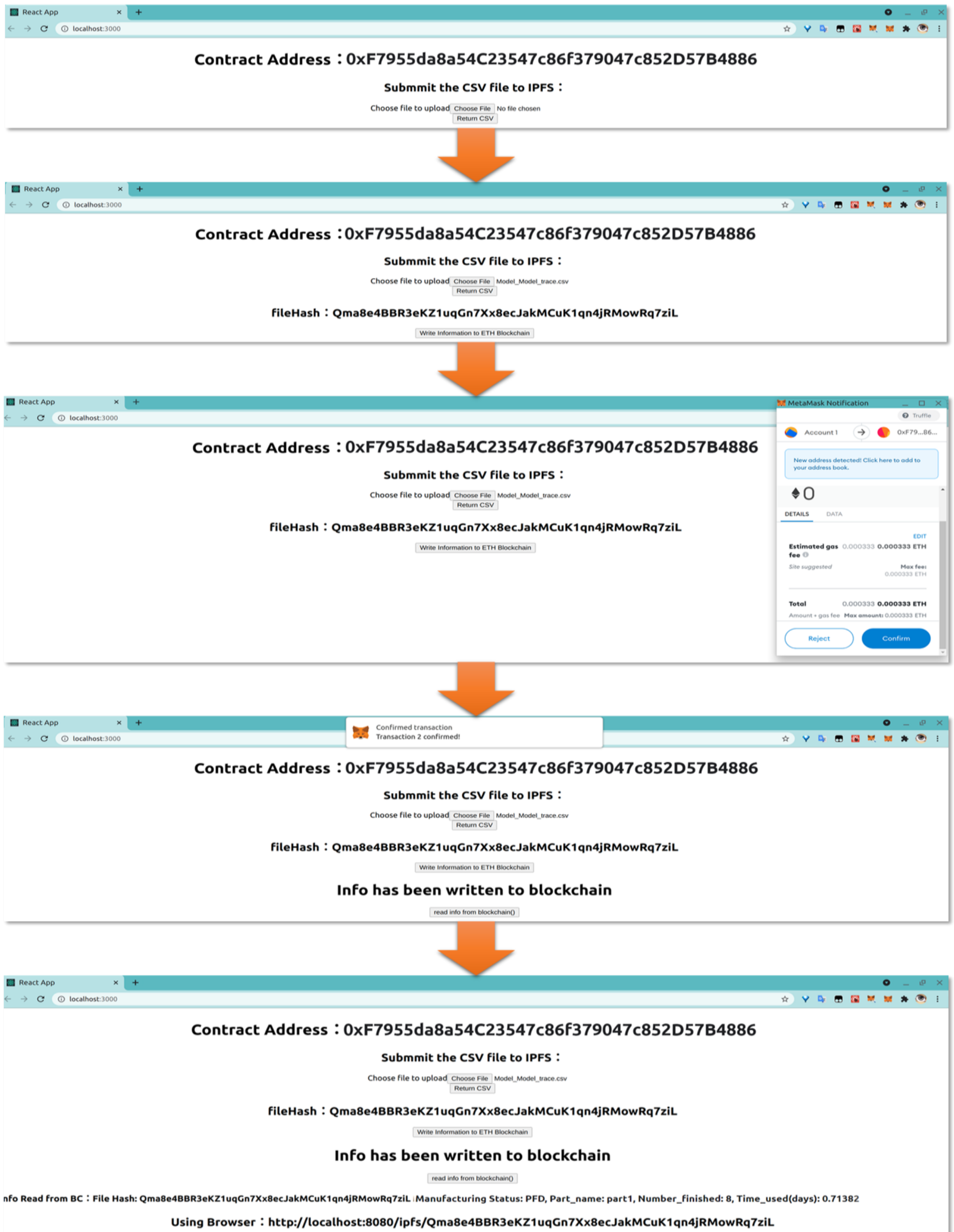


Figure 9-14 Product fulfillment streaming data uploading

In the case study, five manufacturers and two logistic providers utilizes the web-based interface for uploading with the similar process. Transactions sent in the same time interval are includes in the same block. Figure 9-15 demonstrates the block mined in the test network where each block contains several types of transaction from different uses. As mentioned in the Section 9.5.3, the block information is converted to json file and stored in the IPFS for access.

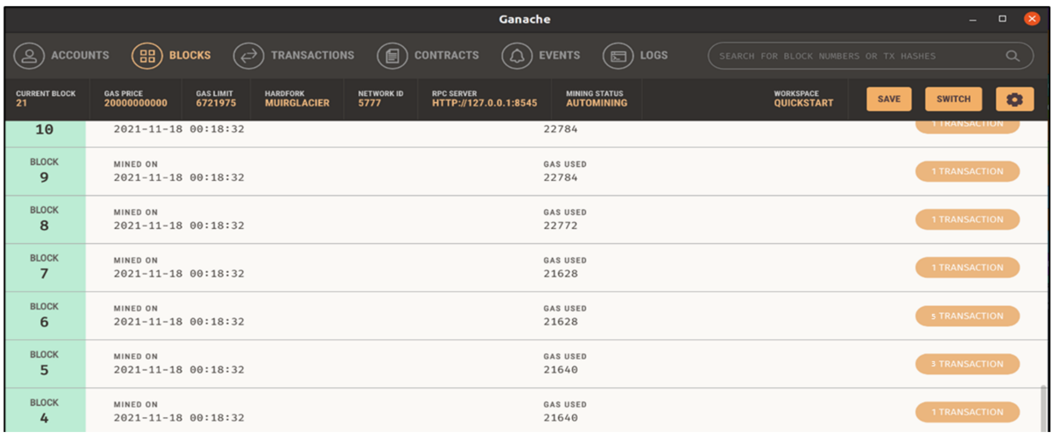


Figure 9-15 Block mining for transactions from different users

9.6 Chapter Summary

Information service is the kernel of platform-driven crowdsourced manufacturing and the cornerstone of delivering MaaS. This chapter proposes the methodologies and proceeds architecture design of a blockchain-based information service system for platform-driven crowdsourced manufacturing. The proposed architecture includes three layers, namely the contracting layer, cyber layer, and resource layer.

The contracting layer enables the deployment of blockchain-based smart contracting, which provides a data management function portal to assign manage access to stakeholders, data extraction algorithm, as well as a data retrieving and updating access for stakeholders. The cyber layer plays the role of an intermediate, which establishes cyber architecture to connect the contracting layer and cyber layer. It not only digitalizes the resource that necessary for the contracting layer but also receives and stores uploaded raw streaming data from the resource layer. The resource layer is the infrastructural layer for resource management. It configures and categorizes the available resource of task participants which provides prerequisites for tasks derivation and allocation. Raw task execution data is managed since it is another type of resource for decision-making.

The architecture of a blockchain-based smart contracting and distributed data information service system is designed in this chapter. It delivers functions for platform-driven crowdsourced manufacturing to deliver MaaS through the three-layer architecture. It can provide information service for geographically distributed stakeholders in a secure way through blockchain technology. A case study evaluates the feasibility and verify the effectiveness of proposed architecture.

CHAPTER 10. REVENUE SHARING IN CROWDSOURCED MANUFACTURING THROUGH POPULATION DYNAMICS MODELING AND ANALYSIS: AN EVOLUTIONARY GAME MODEL

Performed as a platform-driven technology for achieving MaaS, crowdsourced manufacturing involves dynamics of group interactions on structured populations. To reach the full potential of crowdsourcing, it is necessary to study the effectiveness of crowdsourcing in relation to the level of collectivism in facing the problem. There exists an intricate relationship between the number of participants and the difficulty of the problem, indicating the optimal size of the crowdsourced group (Guazzini et al., 2015). This implies the need for a modeling framework in the context of utilization of potential capabilities in crowds with measurement-based analysis of a crowdsourcing platform and the capacity balance among manufacturer crowds (Hoßfeld et al., 2011).

In addition, group interactions are a particularly important and widespread class, representative of the dynamic analysis of decision space in large crowds is formulated as the public goods game (Chen et al., 2015). Population dynamics is inherent in crowdsourcing, for which the study of evolutionary dynamics of group interactions on top of structured populations is necessary, including pattern formation, equilibrium selection, and self-organization in evolutionary games (Perc et al., 2013).

Moreover, the manufacturer crowds form a complex cooperation network of manufacturing service, whereby the inherent openness inevitably leads to dynamics in the governance of the networks towards the effectiveness of open design and manufacturing (Tiwana et al., 2010). The intra-relationship among the crowd population is multifold throughout collaboration among various manufacturing clusters, which are divided by the capabilities they have. Besides, the manufacturing service capacity of a crowdsourcing network relies on the willingness of the manufacturing crowds to bid for an open call while competing with their peers. However, such willingness is dependent on the operational success of the crowdsourcing platform and the individual partner's incentive. Thus, from a long time-span perspective, the proportions of making bidding decision in the population of manufacturer crowds conform to a robust co-evolutionary relationship. If the bidding decisions can bring an excessive profit in a cluster, the proportion of bidding manufacturer in it will increase. Otherwise, a decrease will be observed. It is thus imperative to develop population dynamics models to depict the adoption and reversion of the crowdsourced manufacturing among the crowd population. The analysis of population dynamics provides a critical guideline for individuals' decision making in an open enterprise environment.

In this regard, this chapter examines the population dynamics underlying multiple manufacturer clusters in crowdsourced manufacturing. Section 10.1 formulate the dynamics among manufacturers as an evolutionary competition-cooperation (ECC) game model. Section 10.2 reviews the research threads and proposes research architecture towards multi-verses evolutionary game for crowd behavior analysis. Section 10.3 develops two-player ECC model for platform-driven crowdsourced manufacturing, which provides formulation assumptions, payoff matrix, and replicator equations. Section 10.4 develops multiplayer ECC game model to accommodate multiple manufacturer clusters. Section 10.5 studies stability analysis methods and operational implications. Section 10.6

examines characteristics and potentials of ECC game model for platform-driven crowdsourced manufacturing by demonstrating case study of tank trailer industry. Section 10.7 summarizes this chapter by examining contributions, limitations, and future threads towards population dynamics analysis of manufacturer crowd in platform-driven crowdsourced manufacturing.

10.1 Evolutionary Competition-Cooperation Game Model

The decision support for crowdsourcing platform is aimed for operational excellence for operating the two-folded demand-capacity marketplace. Different from the traditional product manufacturing process which plans the manufacturing processes inner an enterprise or outsources several peripheral activities to designated partners, the decision making in the platform-driven crowdsourced manufacturing shows a collective and distributive characteristic. MaaS requires moving beyond exclusive use of hierarchical decision making, drawing on the power of crowdsourcing and markets wherever possible. Because the crowdsourced manufacturing entails competitive and collaborative workflows that rely on a group decision support system to facilitate the problem-solving process (Thuan et al., 2013b), a successful crowdsourcing platform operation management indicates an understanding of the behavior and evolution of not only manufacturer individual but also the crowd population.

The manufacturer crowd is naturally divided into various manufacturing clusters according to their competitive edges. Thus, the manufacturers who are affiliated to one cluster are confronted with a massive impact of competition. Because of the existence of the awarding process by the manufacturing evaluation broker in the platform, only the best-performed manufacturer in each cluster can be selected and awarded with the contracts S. Besides, the realization of the value chains is essentially a multi-party process, which requires various competitive edges and a large volume of capacity. From this perspective, the relationships among the manufacturers are not only competition but also cooperation.

On the one hand, there is an inter-cluster cooperation relationship shown among the manufacturer crowd, due to the significance of capacity matching. High participation of all clusters will indicate various capabilities and a high manufacturing service capacity. In contrast, low capacity in one cluster will transform this cluster to a bottleneck along the value chain. For example, if the diversity and capacity of the steel sheet factories are limited in the platform's connections, the realization of the tank trailer for the food industry may be hindered, which requires special requirements like edible grade or anti-corrosion material. This phenomenon requires the inter-cluster cooperation to increase the capabilities that the platform connected and the attraction to the open innovators to initiate the value chain.

On the other hand, the inner-cluster cooperation relationship among the manufacturers can be observed from the willingness of participation in one cluster is triggered by the participation status of the upstream and downstream clusters. The higher number of participated manufacturers in the upstream and downstream clusters imply an abundant number of potential value chains, and therefore, a high likelihood of awarding manufacturer in the cluster in the middle. This inner-cluster cooperation indicates the manufacturers need to cooperate with their peers to participate in the bidding to attract more manufacturers in other clusters for future success.

Besides, there is a robust co-evolutionary relationship in the entire manufacturer population. The decision-making process of participating in a value chain or not is based on the revenue of their peers at the current time point. If the average revenue of participating is higher than non-participating, the manufacturers are more likely to participate the crowdsourced manufacturing. Similarly, if the average revenue of participating is lower than non-participating, the manufacturers are more hesitant to participate. This fitness-decreasing behavior can be categorized as an evolutionary puzzle and has been considered as a game-theoretic decision-making scheme (Roca et al., 2009). To find the equilibrium of the evolutionary dynamic supply contracting mechanism, the evolutionary game model is widely applied (Reeves et al., 2005). Since the incentive can be utilized as the factor of sustaining a crowdsourcing ecology, the game model can be used for the incentive design in the crowdsourcing product fulfillment (Tian et al., 2014).

10.2 From Two-Player Game to Multiple Manufacturer Crowds

It has been long time for utilizing game theory to describe interactions among players, but the idea from biology gears forward the development of it towards evolutionary game theory (Smith, 1982). It has been widely applied to model and promote collaboration of a population of rational individuals to achieve a minimization of deviation from the social optimum (Taylor and Nowak, 2007). The basic form of evolutionary game starts explores prey-predator relationship, which is formulated as a two-player population dynamics problem, and a more general form accommodates simultaneous decision-making from multiple players with multiplayer matrix game (Broom et al., 1997). These streams of research build the fundamental of exploring non-linear and chaos system (Nowak and May, 1992). The application of evolutionary games covers dynamic modeling of population genetics in ecology (Hashimoto and Aihara, 2009), management policy derivation for social science (Rosas, 2010), mass collaboration for problem solving (Souza et al., 2009), customer-manufacturer relationship analyzation for social welfare (Ji et al., 2015), and capacity balance dynamics analysis for product fulfillment crowds (Gong, 2018).

The platform-driven crowdsourced manufacturing is aimed to fulfill various customer needs by collaborating with manufacturers from various clusters. This elaborates a collaboration scenario of various industries for fulfilling complex products along the corresponding value chain. A multi-player model is essential to model the dynamics among various industries, which is modeled as manufacturing agent cluster $\{\mu^1, \mu^2, \ldots, \mu^\alpha, \ldots, \mu^{\mathrm{A}}\}$ in Section 3.4. In this regard, this chapter explores relationships among manufacturer clusters under multiplayer evolutionary game and utilizes its dynamics behavior as essential operational excellence approach to promote collaboration among crowds.

10.3 Two-Player ECC Game Model for Capacity Balance

Corresponding to the research agenda proposed in Section 10.2, a two-player ECC game model builds the fundamentals of the population dynamics analysis. The ECC game model is established to imitate the relationship between two cluster of manufacturers, μ^1 and μ^2. This model has considered the cooperation, which is the result of capacity matching and participation abundant, the competition of the agents' peers in their domain, and the co-evolutionary based on the payoff of the states. Based on the evolutionary game theory, several assumptions have been set to formulate the model:

Assumption 1: The potential population of the μ^1 and μ^2 is large enough;

Assumption 2: The variation of the total amounts of the μ^1 and μ^2 is minimal;

Assumption 3: The contracts can be formed with every agent in the population;

Assumption 4: The agents can only select bidding or non-bidding as their strategies;

Assumption 5: The potential service capacity of each manufacturer cluster is matched.

The first assumption ensures an infinite model is adequate for this case, in which a differential equation-based dynamics analysis can be applied to the population. The second assumption implies that the population sizes of manufacturer clusters are stable enough to ignore the fluctuation. The third assumption indicates that manufacturers are homogeneous in every cluster. The fourth assumption regulates the dimension of state space. The fifth assumption implies that the exhaustive search of the manufacturers will establish a balanced manufacturing service capacity.

10.3.1 Model Development Incorporating Revenue Sharing

Based on these assumptions, the model is established as follows. The $X_1(t)$ and $X_2(t)$ are the fraction of the agents who chose bidding strategy in μ^1 and μ^2, respectively. Which $0 < X_1(t) < 1$ and $0 < X_2(t) < 1$. Based on the capacity matching thinking, the capacity unbalance index (CUI) c^u is introduced to measure the unbalance between the capacities of different domains, which can be interpreted as the proportion of the $X_1(t)$ and $X_2(t)$, as defined in equation (10.1). CUI can be interpreted as the ratio of the $X_1(t)$ over $X_2(t)$, it measures the unbalance of the capacity of different clusters.

$$c^u = \frac{X_1(t)}{X_2(t)} \tag{10.1}$$

The cost structure of the agents is modeled in three parts. The first part is the fundamental income of the agents, which can be categorized as manufacturing fundamental income π_1 and π_2, for cluster 1 and 2, respectively. This pair of variables depict the basic operation status of the agents.

The second part is the bidding cost, which can be categorized manufacturing bidding cost b_1 and b_2, for cluster 1 and 2, respectively. This pair of the variables are modeled based on the cost of generating bids. However, this cost is not only related to the fixed cost of making bids, but also the cost resulted from the unbalanced capacity. For instance, in the case the $X_1(t)$ is high and $X_2(t)$ is low, the bidding cost for μ^1 is relatively high, because the probability of awarding in this case is minimal. Meanwhile, the bidding cost of the μ^2 is relatively low, because in such case, the probability of awarding is high. In the worst case, the c^u is approximate to positive infinity, the bidding cost of the μ^1 will approximate to positive infinity, and the bidding cost of the μ^2 will approximate to b_2. Thus, the corrected bidding cost b_1^* and b_2^* are introduced in equation (10.2).

$$\begin{cases} b_1^* = b_1 \cdot (1 + c^u) \\ b_2^* = b_2 \cdot \left(1 + \dfrac{1}{c^u}\right) \end{cases} . \tag{10.2}$$

The third part of the cost structure is the income from the MaaS. The extra income Π is the highest extra income the crowdsourcing platform can reach, and the corrected extra income $\Delta\pi$ is the extra income considering the participation will, which is represented in equation (10.3).

$$\Delta\pi = \Pi \cdot X_1(t) \cdot X_2(t) \tag{10.3}$$

Moreover, the distribution coefficient g is introduced to measure the income distribution between the μ^1 and μ^2 , where $g \cdot \Delta\pi$ will be sent to the μ^1, and $(1-g) \cdot \Delta\pi$ will be sent to μ^2.

10.3.2 Replicator Equations

The agents μ^1 and μ^2 make the decision to choose bidding or non-bidding states based on their own situation. The state space of the μ^1 includes $\{C_1, D_1\}$, where C_1 represents manufacturers in cluster μ^1 cooperatively choose bidding states, while D_1 represents manufacturers in cluster μ^1 defectively choose non-bidding states. Similarly, the state space of μ^2 includes $\{C_2, D_2\}$, which are manufacturers in cluster μ^2 choose bidding and non-bidding states, respectively. Thus, applying the method from Friedman (1991), the state space of the manufacturer coevolution can be represented as $\mathbb{S} = \big\{\{C_1, D_1\}, \{C_2, D_2\}\big\}$. The $\mathbb{S}$ can be quantitatively expressed by $\big(X_1(t), X_2(t)\big)$ in the square of $[0,1] \times [0,1]$. Following a notation that is suggestive of cooperative dilemmas (Nowak, 2012), the payoff of the agents in different states can be established in the Table 10-1. In this context, R denotes the reward of a pair of cooperators, T denotes the temptation to defect, S denotes the “sucker” payoff for a cooperator being exploited by a defector, and P denotes the punishment for a defecting pair.

Table 10-1 The game payoff matrix of the two-player ECC game model

Manufacturer in cluster μ^1	Manufacturer in cluster μ^2	
	$X_2(t)$ Choose Bidding C_2	$1 - X_2(t)$ Choose Non-bidding D_2
$X_1(t)$ Choose bidding C_1	$\pi_1^R = \pi_1 - b_1^* + \Delta\pi \cdot g$ $\pi_2^R = \pi_2 - b_2^* + \Delta\pi \cdot (1 - g)$	$\pi_1^S = \pi_1 - b_1^* + \Delta\pi \cdot g$ $\pi_2^S = \pi_2$
$1 - X_1(t)$ Choose Non-bidding D_1	$\pi_1^T = \pi_1$ $\pi_2^T = \pi_2 - b_2^* + \Delta\pi \cdot (1 - g)$	$\pi_1^P = \pi_1$ $\pi_2^P = \pi_2$

Table 10-1 can determine the fitness of choosing a state by measuring the corresponding income, which is determined by the cooperation proportion of manufacturer in the counterpart cluster. Thus, the fitness functions of μ^1 and μ^2 can be defined as f_1^i and f_2^j, respectively, where $i \in \{C_1, D_1\}$, $j \in \{C_2, D_2\}$. The f_1^i and f_2^j can be defined in equation (10.4) and (10.5).

$$\begin{cases} f_1^{C_1} = X_2(t) \cdot \pi_1^R + (1 - X_2(t)) \cdot \pi_1^S \\ f_1^{D_1} = X_2(t) \cdot \pi_1^T + (1 - X_2(t)) \cdot \pi_1^P \end{cases} \tag{10.4}$$

$$\begin{cases} f_2^{C_2} = X_1(t) \cdot \pi_2^R + \left(1 - X_1(t)\right) \cdot \pi_2^T \\ f_2^{D_2} = X_1(t) \cdot \pi_2^S + \left(1 - X_1(t)\right) \cdot \pi_2^P \end{cases} \tag{10.5}$$

Subsequently, the average fitness function of μ^1 and μ^2 is noted as $\overline{f_1}$ and $\overline{f_2}$, respectively, and is depicted in equation (10.6) and (10.7).

$$\overline{f_1} = X_1(t) \cdot f_1^{C_1} + \left(1 - X_1(t)\right) \cdot f_1^{D_1} \tag{10.6}$$

$$\overline{f_2} = X_2(t) \cdot f_2^{C_2} + (1 - X_2(t)) \cdot f_2^{D_2} \tag{10.7}$$

The replicator dynamics describes the frequencies of states in a population, and the increasing rate of applying a strategy is proportional to its relative fitness (Hofbauer and Sigmund, 1998). The replicator equations can be derived from the differences between the fitness of a state to the average fitness, which is shown in equation (10.8) and (10.9).

$$\dot{X_1}(t) = \frac{dX_1(t)}{dt} = X_1(t) \cdot \left(f_1^{C_1} - \overline{f_1}\right) = X_1(t) \cdot \left(1 - X_1(t)\right) \cdot \left(f_1^{C_1} - f_1^{D_1}\right) \quad (10.8)$$

$$\dot{X_2}(t) = \frac{dX_2(t)}{dt} = X^2(t) \cdot \left(f_2^{C_2} - \overline{f_2}\right) = Y(t) \cdot \left(1 - X_2(t)\right) \cdot \left(f_2^{C_2} - f_2^{D_2}\right) \quad (10.9)$$

Substitute the fitness functions of μ^1 and μ^2 in equation (10.4) and (10.5), the replicator equations (10.8) and (10.9) can be simplified to equation (10.10).

$$\begin{cases} \frac{dX_1(t)}{dt} = X_1(t) \cdot \left(1 - X_1(t)\right) \cdot [\pi_1^R - \pi_1^T] \\ \frac{dX_2(t)}{dt} = X_2(t) \cdot \left(1 - X_2(t)\right) \cdot [\pi_2^R - \pi_2^T] \end{cases} \quad (10.10)$$

Let the equation (10.10) equal to zero, five equilibrium points can be found: $e_1\ (0^+, 0^+)$, $e_2(0^+, 1)$, $e_3(1,0^+)$, $e_4(1,1)$ and the fifth equilibrium point is $e_5({X_1}^*, {X_2}^*)$. Following "R-T-S-P" payoff framework, $e_1\ (0^+, 0^+)$ can be perceived as punishment point, $e_2(0^+, 1)$ is a temptation point, $e_3(1,0^+)$ is a "sucker" point, and $e_4(1,1)$ is a reward point. Considering Where ${X_1}^*$ and ${X_2}^*$ is represented in equation (10.11).

$$\begin{cases} {X_1}^* = \frac{\sqrt{b_2(b_1(1-g) + b_2 \cdot g)}}{\sqrt{b_1(g-1)^2 \Pi}} \\ {X_2}^* = \frac{\sqrt{b_2(b_1(1-g) + b_2 \cdot g)} \cdot \sqrt{b_1(1-g)^2 \Pi}}{b_2 \cdot \Pi \cdot g(1-g)} \end{cases} \quad (10.11)$$

The e_5 is also an equilibrium point when $({X_1}^*, {X_2}^*) \in [0,1] \times [0,1]$, the constraints can be expressed as equation (10.12).

$$\begin{cases} 0 < b_1 < \dfrac{g \cdot \Pi}{2} \\ \dfrac{b_1^2}{b_1 - g \cdot \Pi} \cdot \dfrac{g-1}{g} < b_2 < \dfrac{\sqrt{b_1(g-1)^2(b_1 + 4g \cdot \Pi)} - b_1(1-g)}{2g} \end{cases}. \tag{10.12}$$

10.4 Multi-Player ECC Game Model for Infinite Manufacturer Crowds

Following the research agenda proposed in Section 10.2, a multi-player ECC game model expands the two-player game model in Section 10.3 into a multi-player one to accommodate various manufacturer clusters. The multi-player ECC game model is established to imitate the relationship among manufacturer cluster $\{\mu^1, \mu^2, \dots, \mu^\alpha, \dots, \mu^\mathrm{A}\}$, which considers capacity matching and revenue sharing among clusters. Based on the 2-strategy, n-player games formulated by Broom et al. (1997), assumptions in Section 10.3 have to be modified as follows:

Assumption 1: The potential population of $\mu^1, \dots, \mu^\alpha, \dots, \mu^\mathrm{A}$ is large enough;

Assumption 2: The variation of the total amounts of $\mu^1, \dots, \mu^\alpha, \dots, \mu^\mathrm{A}$ is minimal;

Assumption 3: The contracts can be formed with every agent in the population;

Assumption 4: The agents can only select bidding or non-bidding as their strategies;

Assumption 5: The strategy change is only based on expected payoff;

Assumption 6: The potential service capacities of each manufacturer clusters are matched.

Based on these assumptions, the model is established in a general form as follows. The fraction of the agents chose bidding strategy in manufacturer cluster $\mu^1, \dots, \mu^\alpha, \dots, \mu^\mathrm{A}$ are represented as $X_1(t), \dots, X_\alpha(t), \dots, X_\mathrm{A}(t)$, respectively. All $X_\alpha(t)$ are from 0 to 1. From a capacity-matching perspective, the CUI c^u, which indicates the unbalance of capability. Different from two-player c^u, which is described in equation (10.1), a general form of c^u is essential for multi-player one. The CUI represented in equation (10.13) stands from the manufacturer cluster that explored over the product of rest clusters. It can be interpreted as the measurement of the participation will of one cluster over the participation will of the rest clusters to explore the unbalance of the capacities among different clusters.

$$c_\alpha^u = \frac{X_\alpha(t)}{\prod_{1 \le \beta \le \mathrm{A}, \beta \ne \alpha} X_\beta(t)} \tag{10.13}$$

The cost structure for a manufacturer can be also modelled in three parts. The first part is the fundamental income of manufacturer cluster μ^α, which can be represented as π_α. The second part is the bidding cost, which can be noted as b_α for manufacturer in cluster α to represent their cost of generating bids. Considering the capability unbalance will change the bidding cost, the corrected bidding cost b_α^* are introduced in equation (10.14).

$$b_\alpha^* = b_\alpha \cdot (1 + c_\alpha^u) \tag{10.14}$$

The third part of cost structure is the extra income from the MaaS, which can be formulated as the product of highest extra income and the bidding fraction of all manufacturer clusters, as shown in equation (10.15).

$$\Delta\pi = \Pi \cdot \prod_{1}^{\mathrm{A}} X_{\alpha}(t) \tag{10.15}$$

In addition, the distribution coefficient g is also modified to be a general form g_{α}, which reflects the distribution among all manufacturer clusters. The sum of g_{α} from all clusters are set to 1, which is shown in equation (10.16).

$$\sum_{1}^{\mathrm{A}} g_{\alpha} = 1 \tag{10.16}$$

Similar to the 2-player games in Section 10.3.2, manufacturers in every cluster have two strategies, which construct a state space for μ^{α} as $\{C_{\alpha}, D_{\alpha}\}$. C_{α} represents manufacturers in cluster α choose bidding as a cooperative strategy, while D_{α} represents non-bidding as a defective strategy. Therefore, the coevolutionary manufacturer state space of a 2-strategy multi-player ECC game can be represented as $\mathbb{S} = \{\{C_1, D_1\}, \dots, \{C_{\alpha}, D_{\alpha}\}, \dots, \{C_{\mathrm{A}}, D_{\mathrm{A}}\}\}$. It expands the two-dimensional square of $[0,1] \times [0,1]$ in two-player to a A-dimensional space to allow a quantitative expression. The payoff of strategies in all manufacturer clusters is shown in Table 10-2. It can be used to determine the fitness of choosing a strategy by measuring the income, which is influenced by the participation of the rest of manufacturer clusters.

Thus, the fitness functions of μ^{α} can be represented as f_{α}^{i}, where $i \in \{C_{\alpha}, D_{\alpha}\}$, which can be defined in equation (10.17).

$$\begin{cases} f_{\alpha}^{C_{\alpha}} = \pi_{\alpha}^{C} \\ f_{\alpha}^{D_{\alpha}} = \pi_{\alpha}^{D} \end{cases} \tag{10.17}$$

Subsequently, the average fitness function of μ^{α} is noted as $\overline{f_\alpha}$ and derived in equation (10.18).

$$\overline{f_\alpha} = X_\alpha(t) \cdot f_\alpha^{C_\alpha} + \left(1 - X_\alpha(t)\right) \cdot f_\alpha^{D_\alpha} \tag{10.18}$$

Table 10-2 The game payoff matrix of the multi-player ECC game

Manufacturer in cluster	Strategies of a manufacturer	
	C	D
1	$\pi_1^C = \pi_1 - b_1^* + \Delta\pi \cdot g_1$	$\pi_1^D = \pi_1$
2	$\pi_2^C = \pi_2 - b_2^* + \Delta\pi \cdot g_2$	$\pi_2^D = \pi_2$
...	...	...
α	$\pi_\alpha^C = \pi_\alpha - b_\alpha^* + \Delta\pi \cdot g_\alpha$	$\pi_\alpha^D = \pi_\alpha$
...	...	...
A	$\pi_{\mathrm{A}}^C = \pi_{\mathrm{A}} - b_{\mathrm{A}}^* + \Delta\pi \cdot g_{\mathrm{A}}$	$\pi_{\mathrm{A}}^D = \pi_{\mathrm{A}}$

The replicator equations describe the possibility bidding manufacturer meets other essential bidding collaborators from rest clusters, and shows the increasing rates of adopting a strategy in their cluster (Gokhale and Traulsen, 2010), whose general form can be derived as equation (10.19).

$$\begin{aligned} \dot{X_\alpha}(t) &= \frac{dX_\alpha(t)}{dt} = X_\alpha(t) \cdot \left(f_\alpha^{C_\alpha} - \overline{f_\alpha}\right) \\ &= X_\alpha(t) \cdot \left(1 - X_\alpha(t)\right) \cdot \left(f_\alpha^{C_\alpha} - f_\alpha^{D_\alpha}\right) \end{aligned} \tag{10.19}$$

Substitute equation (10.17), the replicator equations can be simplified to equation (10.20).

$$
\begin{cases}
\dfrac{dX_1(t)}{dt} = X_1(t) \cdot \big(1 - X_1(t)\big) \cdot (\pi_1^C - \pi_1^D) \\
\qquad\qquad\qquad \cdots \\
\dfrac{dX_\alpha(t)}{dt} = X_\alpha(t) \cdot \big(1 - X_\alpha(t)\big) \cdot (\pi_\alpha^C - \pi_\alpha^D) \\
\qquad\qquad\qquad \cdots \\
\dfrac{dX_\mathrm{A}(t)}{dt} = X_\mathrm{A}(t) \cdot \big(1 - X_\mathrm{A}(t)\big) \cdot \big(\pi_\mathrm{A}^C - \pi_\mathrm{A}^D\big)
\end{cases} \tag{10.20}
$$

10.5 Stability Evaluation

The concept of evolutionary stable strategies (ESS) is introduced to observe the dynamic behavior of the population (Hines, 1987). In the evolutionary game theory, the ESS is a refinement of Nash Equilibrium (NE), and all games played have an ESS as an optimal solution (Thomas, 2012). The ESS can be interpreted as a stable condition after a long-time of evolution, and such stability can resist the mutation of small invasion of the population (Friedman, 1991). At an ESS condition, the composition of the states keeps stable and can prevent the turbulence of alternative strategies.

10.5.1 Two-Strategies, Two-Player Games

The population games describe the interactional behaviors among a considerable number of anonymous agents; such behaviors are based on the simple rules called revision protocol (Thomas, 2012). Over a long time-span, the aggregate behavior can be well modeled by the differential equations. Because the proposed evolutionary game is a 2×2 planar system, the stability of the equilibrium points in replicator equations can be analyzed by applying the trace-determinant plane analysis on the Jacobian matrix of the replicator equations. The Jacobian matrix $\mathcal{J}$ of the replicator equations (10.10) is established in equation (10.13). The elements of Jacobian matrix $\mathcal{J}$ are calculated in equation (10.22) – (10.26).

$$\mathcal{J}=\begin{bmatrix}\frac{\partial \dot{X_1}(t)}{\partial X_1(t)} & \frac{\partial \dot{X_1}(t)}{\partial X_2(t)} \\ \frac{\partial \dot{X_2}(t)}{\partial X_1(t)} & \frac{\partial \dot{X_2}(t)}{\partial X_2(t)}\end{bmatrix} \tag{10.22}$$

$$\frac{\partial \dot{X_1}(t)}{\partial X_1(t)}=\frac{b_1\left(3X_1^2\,(t)-2X_1(t)\cdot X_2(t)-X_1(t)-X_2(t)\right)}{X_2(t)}- \\ g(3X_1(t)-2)\cdot X_1(t)\cdot X_2(t)\cdot \Pi \tag{10.23}$$

$$\frac{\partial \dot{X_1}(t)}{\partial X_2(t)}=X_1(t)\cdot\left(1-X_1(t)\right)\cdot\left(\frac{b_1\cdot X_1(t)}{X_2^2(t)}+g\cdot X_1(t)\cdot \Pi\right) \tag{10.24}$$

$$\frac{\partial \dot{X_2}(t)}{\partial X_1(t)}=X_2(t)\cdot\left(1-X_2(t)\right)\cdot\left(\frac{b_2\cdot X_2(t)}{X_1^2\,(t)}+(1-g)\cdot X_2(t)\cdot \Pi\right) \tag{10.25}$$

$$\frac{\partial \dot{X_2}(t)}{\partial X_2(t)}=\frac{b_2\left(3X_2^2(t)+2X_1(t)\cdot X_2(t)-X_1(t)-2X_2(t)\right)}{X_1(t)}- \\ (1-g)\cdot(3X_2(t)-2)\cdot X_1(t)\cdot X_2(t)\cdot \Pi \tag{10.26}$$

The trace and the determinant of $\mathcal{J}$ is noted as $\tau_{\mathcal{J}}$ and $\Delta_{\mathcal{J}}$. The stability of equilibrium points can be evaluated by the following criteria (Hirsch et al., 2012):

1) The equilibrium point is ESS when $\Delta_{\mathcal{J}}>0$ and $\tau_{\mathcal{J}}<0$;

2) The equilibrium is unstable when $\Delta_{\mathcal{J}}>0$ and $\tau_{\mathcal{J}}>0$;

3) The equilibrium is saddle when $\Delta_{\mathcal{J}}<0$.

Based on these criteria, the stability analysis of the $e_1\,(0^+,0^+)$, $e_2(0^+,1)$, $e_3(1,0^+)$, $e_4(1,1)$ is shown in Table 10-3. From Table 10-3, the e_1 is an ESS, which can be interpreted as no manufacturer decide to bid in this population. Although a small fraction of the agents decides to bid, the population will maintain the stable situation at $(0^+,0^+)$ in a long time-span. The equilibrium points e_2 and e_3 are the case that almost all manufacturers in μ^1 or μ^2 decide for bidding, while all the agents in the counterpart cluster choose non-bidding. The stability analysis shows these two are saddle points, which can be interpreted as trajectory has both inner and outer directions, these situations are not evolutionary stable.

Table 10-3 The stability analysis of the first four equilibrium points

Equilibrium Points	$\mathcal{J}$	$\tau_{\mathcal{J}} = \mathrm{tr}(\mathcal{J})$	$\Delta_{\mathcal{J}} = \det(\mathcal{J})$	Stability
$e_1(0^+,0^+)$	$\begin{bmatrix} -3b_1 & b_1 \\ b_2 & -3b_2 \end{bmatrix}$	$-3(b_1+b_2)$	$8b_1 \cdot b_2$	ESS
$e_2(0^+,1)$	$\begin{bmatrix} -b_1 & 0 \\ 0 & \infty \end{bmatrix}$	∞	$-\infty$	Saddle Points
$e_3(1,0^+)$	$\begin{bmatrix} \infty & 0 \\ 0 & -b_2 \end{bmatrix}$	∞	$-\infty$	Saddle Points
$e_4(1,1)$	$\begin{bmatrix} 2b_1 - g \cdot \Pi & 0 \\ 0 & 2b_2 - (1-g) \cdot \Pi \end{bmatrix}$	$2(b_1+b_2) - \Pi$	$(2b_1 - g \cdot \Pi) \cdot (2b_2 - (1-g) \cdot \Pi)$	Undetermined

The fourth equilibrium point can be interpreted as the situation that all the μ^1 and μ^2 apply bidding. Based on the stability analysis criteria, the constraints conditions for each stability situation are summarized in the Table 10-4.

From Table 10-4, it can be observed that the e_4's stability depends on the relationship between the bidding cost, b_1 and b_2, and the extra income distribution, which is determined by g and Π. When the bidding cost is smaller than the half of the received extra income, the e_4 is an ESS. In that case, most of the agents in population will bid, and thus, the prosperous of the open enterprise can be realized. However, in the case that the extra income distribution is unbalanced, the e_4 is a saddle point. It will show different trajectory directions on this point. In the last case, if the bidding cost is too high or the extra income is not enough, e_4 will be unstable. And based on the equation (10.12), the fifth equilibrium point will not exist. This case should be avoided, because in such case, the only ESS is e_1, and the operation of MaaS is hard to sustain.

Table 10-4 The stability constraints conditions of the fourth equilibrium point

Constraints Conditions	Sign $\tau_{\mathcal{J}} = \mathrm{tr}(\mathcal{J})$	Sign $\Delta_{\mathcal{J}} = \det(\mathcal{J})$	Stability Situation
$\begin{cases} 0 < b_1 < \frac{g}{2}\Pi \\ 0 < b_2 < \frac{(1-g)}{2}\Pi \end{cases}$	$-$	$+$	ESS
$\begin{cases} 0 < b_1 < \frac{g}{2}\Pi \\ b_2 > \frac{(1-g)}{2}\Pi \end{cases}$ or $\begin{cases} b_1 > \frac{g}{2}\Pi \\ 0 < b_2 < \frac{(1-g)}{2}\Pi \end{cases}$	$\pm$	$-$	Saddle Points
$\begin{cases} b_1 > \frac{g}{2}\Pi \\ b_2 > \frac{(1-g)}{2}\Pi \end{cases}$	$+$	$+$	Unstable

In the case of the fifth equilibrium point, where $\left(X_1(t), X_2(t)\right) = ({X_1}^*, {X_2}^*)$, the Jacobian matrix and its elements are shown in equation (10.13) – (10.17). To simplify elements in equation (10.22) – (10.26), three variables ρ_1, ρ_2, and ρ_3 is introduced in equation (10.27).

$$\begin{cases} \rho_1 = \sqrt{b_1(1-g)^2 \Pi} \\ \rho_2 = \sqrt{b_2 \cdot g(1-g)\Pi} \\ \rho_3 = \sqrt{b_2(b_1(1-g) + b_2 \cdot g)} \end{cases} \tag{10.27}$$

Thus, the (X^*, Y^*) can be simplified to equation (10.28).

$$\begin{cases} X^* = \dfrac{\rho_3}{\rho_1} \\ Y^* = \dfrac{\rho_3 \cdot \rho_1}{\rho_2^2} \end{cases} \tag{10.28}$$

Substitute equation (10.28) to equation (10.22) – (10.26), the Jacobian matrix can be simplified to equation (10.29), its trace and determinant are calculated in equation (10.30) and (10.31), respectively.

$$\mathcal{J}|_{(X^*,Y^*)} = \begin{bmatrix} \frac{\rho_1 \cdot (1-\rho_3)}{(1-g)^2 \cdot \Pi} & \frac{g \cdot \Pi(\rho_2 + b_2^2 \cdot g) \cdot (\rho_1 - \rho_3)}{\rho_1^3} \\ \frac{b_1[2b_1(1-g) + b_2 \cdot g] \cdot [b_2 g(1-g)\Pi - \rho_1 \cdot \rho_3]}{b_2^2 \cdot g^3 \cdot \Pi} & b_2 - \frac{b_1(1-g)\rho_3}{g\rho_1} \end{bmatrix} \tag{10.29}$$

$$\tau_{\mathcal{J}}|_{(X^*,Y^*)} = b_1 + b_2 - \frac{b_1 \cdot \rho_3}{g \cdot \rho_1} \tag{10.30}$$

$$\Delta_{\mathcal{J}}|_{(X^*,Y^*)} = \frac{2b_2 \cdot \Delta_{\mathcal{J}} \rho_3^4 (\rho_1 - \rho_3)(-\rho_2^2 + \rho_1 \cdot \rho_3)}{b_2 \cdot g^2 \cdot \rho_1^3} \tag{10.31}$$

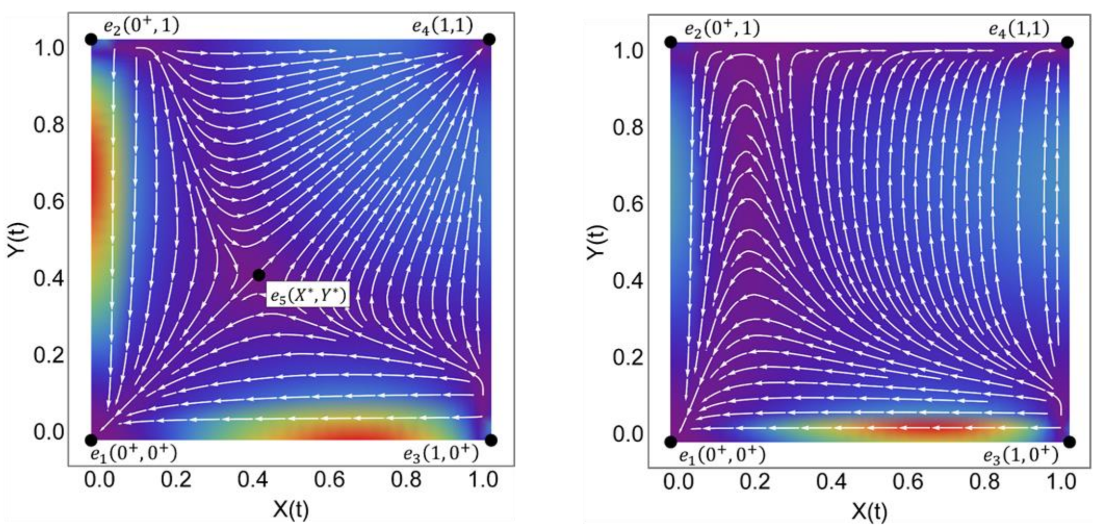
Based on equation (10.29) – (10.31), and the existence conditions for the fifth equilibrium point e_5, the stability of e_5 is processed. The e_5 cannot be an ESS or an unstable point when $(X^*, Y^*) \in [0,1] \times [0,1]$. The e_5 is a saddle point when satisfying equation (10. 32).

$$\begin{cases} 0 < b_1 < \frac{g \cdot \Pi}{2} \\ \frac{b_1^2 \cdot (1-g)}{g \cdot (g \cdot \Pi - b_1)} < b_2 < \frac{(1-g)\sqrt{4g \cdot \Pi}}{2g} \end{cases} \tag{10.32}$$

Therefore, the operation of the open enterprise can consist in two scenarios: 1) the e_4 is an ESS while the e_5 is a saddle point; 2) the e_4 is an ESS while the e_5 is not exist. The first scenario is the case that the only the e_1 and e_4 are the ESS, and the e_5 is existed as a saddle point, the constraint condition is as same as equation (10. 32). To demonstrate this scenario, an illustrative phase diagram is shown in the Figure 10-1 a). In such case, the population can reach the e_4, but a low value of e_5 is the key to enhance the probability of reaching e_4.

The second scenario is the case that the e_1 and e_4 are ESS, the e_5 is not existed due to the unbalance income between μ^1 and μ^2. The constraint condition is shown in equation (10.24). In such case, the MaaS ecology can reach the e_4. The illustrative phase diagram is shown in Figure 10-1 b).

$$\begin{cases} 0 < b_1 < \dfrac{g \cdot \Pi}{2} \\ 0 < b_2 < \dfrac{b_1^2 \cdot (1-g)}{g \cdot (g \cdot \Pi - b_1)} \ or \ \dfrac{(1-g)\sqrt{4g \cdot \Pi}}{2g} < b_2 < \dfrac{(1-g) \cdot \Pi}{2} \end{cases} \tag{10.24}$$



a) Scenario 1

b) Scenario 2

Figure 10-1 Phase diagram of 2-player ECC game

10.5.2 Two-Strategies, Multi-Player Games

It can be observed that the set of replicator equations is a nonlinear system of differential equations in a form as equation (10.21).

$$\begin{cases} X_1'(t) = r_1\left(X_{1(t)}, \dots, X_{\mathrm{A}}(t)\right) \\ \quad\quad\quad\quad \cdots \\ X_{\mathrm{A}}'(t) = r_{\mathrm{A}}\left(X_{1(t)}, \dots, X_{\mathrm{A}}(t)\right) \end{cases} \tag{10.21}$$

Nullcline method has been widely applied to solve the existence of the equilibrium points, by setting $r_\alpha\left(X_{1(t)}, \dots, X_{\mathrm{A}}(t)\right) = 0$, calculate the intersection of nullclines, and observe the direction change in the corresponding direction field (Hirsch et al., 2012). Similar to the derivation of equilibrium points in two-player game, there are two trivial fixed points when the whole population consists of bidding (C^α) or non-bidding (D^α). In an A-player two-strategy game, the maximum number of possible internal equilibrium points is $A - 1$, which requires the quantities of $\pi_\alpha^C - \pi_\alpha^D$ and $\pi_{\alpha+1}^C - \pi_{\alpha+1}^D$ have different signs for all α (Gokhale and Traulsen, 2010).

10.5.3 Operational Implications

From the stability of the equilibrium points, the operational protocol of the ECC game model shows a strong influence on the dynamics of manufacturer population. Firstly, the equality of income distribution can be evaluated by finding the existence of internal equilibrium point. If there is an equilibrium point in real number domain, as shown in scenario 2 of Figure 10-1 b), a severe inequality of income distribution among manufacturer clusters can be perceived. Secondly, manipulating the position of internal equilibrium point is a way of managing manufacturer population, which can be achieved by setting income and bidding cost in this model. From a two-dimensional perspective, equilibrium point e_5 can be viewed as a peak in a landscape, and the link between e_5 to e_1, e_2, and e_3 consist of three ridges, while the area from e_5 to e_4 can be viewed as a valley.

This separation can be mathematically explained by modeling nullclines of the differential equation system constructed. Measuring the current situation of the manufacturer population and reflecting these results on the phase diagram can provide a prediction of the future evolution direction of the crowd. If the state of the manufacturer population falls into the valley area, a long-time prosperity can be predicted. On the other hand, if the state of manufacturer population falls below the internal equilibrium point, a future decay of crowdsourcing ecology can be predicted. If the current state is close to a fully participation and far away from internal equilibrium point, a lower extra income and higher bidding cost will not harm the will of participation in manufacturer crowd. By using the proposed rules and equations, the operational protocol of the manufacturer can be derived, and a judgement of the long-time prosperity can be concluded.

10.6 Case Study of Tank Trailer Multi-Cluster Cooperation

An illustrative case of tank trailer manufacturing service through platform-driven crowdsourced manufacturing is used to illustrate and examine the potential of proposed theory. Through the developing of a crowdsourcing value chain, it demonstrates the application of evolutionary game theoretic dynamic analysis of the partners and an optimal operational protocol manipulating logic. Due to the variety of customer orders, tank trailers are highly customized. By applying platform-driven crowdsourced manufacturing, the open innovator opens its boundary to crowdsource the tasks and utilizes external knowledge and capacities to achieve MaaS. The fulfillment process of a tank trailer requires the collaboration of a trailer frame manufacturer cluster μ^1, fuel tank manufacturer cluster μ^2, and accessory manufacturer cluster μ^3, which is achieved by cyber connections built by crowdsourcing platform. The replicator equations of a 3-palyer ECC game model is shown in equation (10.22).

$$\begin{cases} \dfrac{dX_1(t)}{dt} = X_1(t)\cdot\left(1 - X_1(t)\right)\left(g_1\cdot\Pi\cdot\prod_1^3 X_\alpha(t) - b_1\left(1 + \dfrac{X_1(t)}{X_2(t)\cdot X_3(t)}\right)\right) \\ \dfrac{dX_2(t)}{dt} = X_2(t)\cdot\left(1 - X_2(t)\right)\left(g_2\cdot\Pi\cdot\prod_1^3 X_\alpha(t) - b_2\left(1 + \dfrac{X_2(t)}{X_1(t)\cdot X_3(t)}\right)\right) \\ \dfrac{dX_3(t)}{dt} = X_3(t)\cdot\left(1 - X_3(t)\right)\left(g_3\cdot\Pi\cdot\prod_1^3 X_\alpha(t) - b_3\left(1 + \dfrac{X_3(t)}{X_1(t)\cdot X_2(t)}\right)\right) \end{cases} \quad (10.22)$$

An ECC game model is implemented to explore the population dynamics among manufacturer clusters. Based on the formulation in Section 10.4, a numerical analysis of the dynamics model is executed to demonstrate the proposed theory. The highest extra income Π is set to 1000, which can be interpreted as the profits of the highest profits that a crowdsourced manufacturing service can reach in a value chain. The bidding cost of three manufacturing cluster b_1, b_2 and b_3 are set to 6, 10, and 5, respectively. This cost can be interpreted as the cost of generating the bid. The distribution coefficients of three clusters g_1, g_2, and g_3 are set to 0.3, 0.5, and 0.2. These number entails the distribution among manufacturer clusters. The replicator equations are derived by using these parameters, which are represented in equation (10.23).

$$\begin{cases} \dfrac{dX_1(t)}{dt} = X_1(t)\cdot\left(1 - X_1(t)\right)\left(300\prod_1^3 X_\alpha(t) - 6\left(1 + \dfrac{X_1(t)}{X_2(t)\cdot X_3(t)}\right)\right) \\ \dfrac{dX_2(t)}{dt} = X_2(t)\cdot\left(1 - X_2(t)\right)\left(500\prod_1^3 X_\alpha(t) - 10\left(1 + \dfrac{X_2(t)}{X_1(t)\cdot X_3(t)}\right)\right) \\ \dfrac{dX_3(t)}{dt} = X_3(t)\cdot\left(1 - X_3(t)\right)\left(200\prod_1^3 X_\alpha(t) - 5\left(1 + \dfrac{X_3(t)}{X_1(t)\cdot X_2(t)}\right)\right) \end{cases} \quad (10.23)$$

Set these equations to zero, it can be found that there is only one internal equilibrium point $(0.441, 0.441, 0.376)$. The phase diagram of this case is shown in Figure 10-2. This figure also shows three nullclines of equations (10.23), which are set replicator equations equal to zero separately. Because this case is a three-player ECC game, nullclines are three surfaces. The intersection point of these surfaces is the only internal equilibrium point in this game. The area above three surfaces and equilibrium point will lead to a convergence of ESS, which implies a fully participation of manufacturers. Otherwise, the longtime prosperity cannot be assured. This division shows that manipulating the internal equilibrium point below the current state of manufacturer crowd is essential to a long-time prosperity. In contrast, if the participation state of manufacturer below all these nullclines, it will show a convergence to $(0^+, 0^+, 0^+)$, which is also an ESS. This scenario will lead a platform-driven crowdsourced manufacturing to an end.

Thus, the manipulation of the internal equilibrium point is critical in the management of platform-driven crowdsourced manufacturing. When the participation of the manufacturers is low, a low internal equilibrium point shows its essentiality. It will lead the state to a fully participation state, to achieve the long-time prosperity following the concept of ESS. Such manipulation involves a higher extra income and lower bidding costs. However, when the agents show a higher participation fraction, a low internal equilibrium point is less preferred, because a high extra income to manufacturers implies a lower income to crowdsourcing platform. In this case, a crowdsourcing platform can move the internal equilibrium point in a reasonable distance. The existence and position of internal equilibrium can be utilized as a measurement of a capacity balance. If the equilibrium is an external one or far away from the diagonal of state space, the capacity distribution among manufacturer clusters can be evaluated as a result of an inferior operational protocol.

Lastly, from Figure 10-2, the changing rates of states show low values around the equilibrium points and higher value near nullclines. It implies the participation fraction will change faster near the equilibrium points. From a management perspective, a large distance between the participation state and equilibrium point will result in a quicker reforming, and an approximation between the participation state and equilibrium point will lead to a stable circumstance.

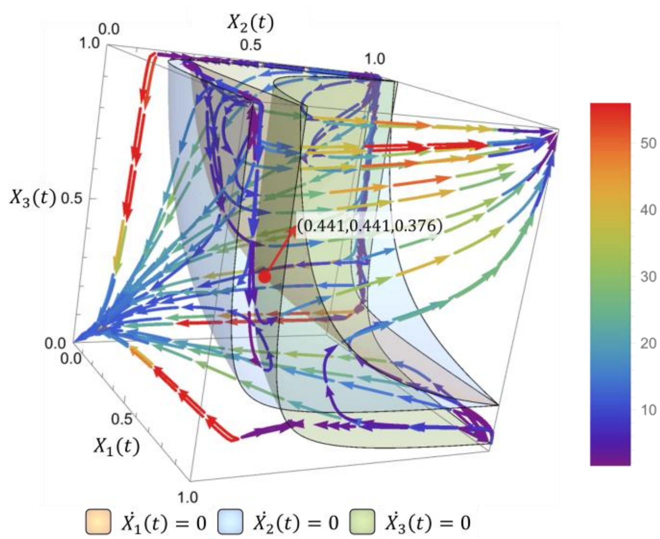


Figure 10-2 Phase Diagram of Tank Trailer Crowdsourced Manufacturing

10.7 Chapter Summary

In this chapter, the ECC game theoretic population dynamic model for platform-driven crowdsourced manufacturing has been established. Different from the traditional manufacturing paradigm, MaaS requires companies to open their boundaries and expand the all-in-one decision making to group behavior via crowdsourcing. The model proposed in this chapter offers a way of describing the adoption and reversion of the open strategy, by formulating competition-cooperation relationships among multiple manufacturer clusters. From an operation excellence perspective, this chapter builds the cornerstone for prediction and management of the behavior of manufacturers.

As a conclusion, a higher income and balanced distribution among the domains will encourage the participation of the agent. Therefore, the long-time prosperity of a crowdsourced manufacturing ecology should be pursued. However, the relationship of the income, distribution balance and the growth rates are changing with the participation fraction. In a high participation and fraction situation, the requirement of the income and distribution balance is loosed. On the other hand, a high income shows a significant influence on the growth rates in the low participation fraction. The open enterprise can manage the agents' behavior by taking the proposed model as a guideline.

This chapter expands the conventional management view, which focuses more on one-time contracting and task execution. A management decision-support should accommodate concept of crowds, which enlarge the number of partner to a population, and adopts a long-time perspective, which considers the evolution of manufacturer clusters.

CHAPTER 11. CONCLUSIONS AND FUTURE WORK

This concluding chapter summarizes the findings and the contributions of this research work. It outlines the conceptual architecture and critical technological methodologies developed for platform-driven crowdsourced manufacturing and how these findings gear forward to MaaS. The limitations and possible improvements are also discussed, along with avenues for future research.

11.1 Conclusions

This dissertation originally proposes platform-driven crowdsourced manufacturing for MaaS through a cyber platform. It bridges the gap between initiatives towards open yet service-oriented manufacturing operational goals and challenges of fulfilling manufacturing tasks in a distributed and collaborative manner through crowdsourcing. This research identifies a series of challenges of extensively searching and utilizing manufacturing resources. This research synergizes qualitative analysis, quantitative decision model, and computational solutions as a MaaS reference model as a transition guidebook for industries to implement platform-driven crowdsourced manufacturing.

In this regard, this work reviews the state-of-the-art research and industry practices to establish a common consensus in a future manufacturing paradigm shift towards platform-driven crowdsourced manufacturing. The stakeholder analysis divides the decision agents along the crowdsourcing value chain into open innovators, platforms, and manufacturers. This trichotomy analysis is based on their roles in the value chain, which is value capturing, structuring, and creating, respectively.

Based on the characteristics of the crowdsourced manufacturing and analysis of applicable industries, the workflow of a crowdsourcing value chain is proposed. Since a successful platform attracts a variety of value chains, the platform is required to offer a stack of crowdsourcing information management fields and crowdsourcing contracting mechanism fields to serve open innovator and manufacturer crowds, respectively. The platform strategy implies maximum reuse of the manufacturers, which leads to the information and material flow sharing manufacturers as a common vertex. The networked information and material flow in the holistic view of crowdsourced manufacturing indicate the necessity of a synergy of information, logistic, as well as production coordination service in crowdsourced manufacturing. In this regard, a MaaS reference model to provide a systematic service for stakeholders in platform-driven crowdsourced manufacturing are proposed, dealing with issues of crowdsourcing contracting mechanism, task execution, operational protocol derivation, and information service system.

First, crowdsourcing contracting requires formulated processes and robust evaluation mechanism to fulfill customer's engineering functional requirements and business operational preference. This study proposes a multi-criteria evaluation method with information content measurement for explicit criteria and decision-tree-based monotone ordinal measurement for inexplicit criteria. A major advantage of proposed method is that it involves quantitative modeling of customer preference and stochastic behavior of candidates to ensure a maximized degree of satisfaction. This process can calculate degree of satisfaction for each criterion and aggregate based on the multi-attribute theory.

Second, coordinated optimization of PFP and MLB in platform-driven crowdsourced manufacturing adopts an interactive decision-making between various agents. A practical and effective bilevel approach for dynamic interactive design optimization of PFP and MLB is proposed based on Stackelberg game. Consistent with the leader-follower interactive mechanism, a bilevel optimization model with linear physical programming is developed, in which the upper- and lower-level objective functions are the common deviation functions adapted from the corresponding linear physical programs. NBGA with upper-level GA for PFP and lower-level GA for MLB is designed for solving the developed model. The proposed bilevel approach is demonstrated via a joint PFP and MLB design problem for tank trailer product family. Through comparison with other approaches, this bilevel approach is shown to yield satisfactory levels of achievement for PFP and MLB objectives. This approach provides an effective decision-making framework for the multi-agent online interactive product fulfillment faced by enterprises adopting the crowdsourced manufacturing model through service-oriented crowdsourcing platforms.

Third, the logistic service through cross-docking acts as an effective solution for platform-driven crowdsourced manufacturing, which organizes logistic service routes in optimally planned pickup and delivery ones. This approach synchronizes pickup and delivery vehicles to let a simultaneously service load exchange to minimize inventory cost. This method addresses a platform-driven solution, which explores the similarities among pickup and delivery demands and integrates them through a centralized cross-dock depot for maximized route reuse. This study formulates a C-VRPCD for optimal decision-making through cross-docking in a crowdsourcing environment. This study proposes a B&P algorithm to solve C-VRPCD, which solves combinations of routes in master problems and finding optimal routes replenishment through a ESPPRC problem. A column generation architecture is proposed to organize branching searching on exchanging operations in master problem and pulse algorithm in subproblem. Through a comparison with other approaches, this B&P approach shown a high solution quality and quick speed. This approach provides efficient platform-driven logistic solutions for crowdsourced manufacturing to achieve MaaS.

Fourth, the r-OAS decision-making is significant in platform-driven crowdsourced manufacturing environments, especially in today's open and collaborative production environment. Given the progress in the use of smart technologies on shop floors, this paper studies the r-OASR problem by simultaneously addressing the subproblems of r-OAS and r-JRP after addressing the objectives of order revenue and various operation costs. Considering the interactive mechanism between r-OAS and r-JRP, a BIO model is established in this study. This interactive approach reveals the inherent tradeoffs of leveraging order revenue and various existing costs. Based on the Stackelberg game theory, the BIO defines r-OAS as the leader and a 0–1 integer programming model is constructed accordingly, while r-JRP is defined as the follower and accompanied by a nonlinear integer programming model. The r-OAS first makes the OAS decision by addressing net revenue and some logical constraints. The r-JRP formulates the best job release plan to minimize operation costs after receiving the r-OAS decision and feeding the cost back to r-OAS. In r-JRP, a data-driven representation approach is proposed to reveal the real-time production status of flow lines and is then used to predict the job makespan. This job-makespan prediction with an r-OAS scheme acts as the key constraint for r-JRP. A bilevel tabu-enumeration algorithm is proposed for the nonlinear hierarchical joint optimization model to determine an efficient solution. The case study reveals that the BIO model can consistently ascertain a better order acceptance, scheduling, and releasing decision than existing approaches that manage the r-OAS and r-JRP separately.

Fifth, a blockchain-based information service system is proposed to serve as a kernel and fundamental of crowdsourced manufacturing through cyber platform. In this study, an information service system architecture design is proceeded for all stakeholders, which allows them to access product fulfillment in a distributed way. Blockchain technology paves a way for smart contracting, which provides an access key for product data uploading, extracting, and supervising in a crowdsourced manufacturing process. This architecture accommodates industrial IoTs, which acts as a cornerstone of other technical elements in this study. The results show a significant improvement by incorporating distributive database for streaming product fulfillment data.

Sixth, in order to manage multiple manufacturer cluster for achieving a long-time prosperous of the crowdsourcing ecology, an operational protocol derivation method based on the multi-verse population dynamics model is proposed. It formulated multiple cluster interactions in platform-driven crowdsourced manufacturing as an ECC game model, which describes not only competition-cooperation but also evolutionary interrelationships among manufacturer clusters in a set of replication equations. By analyzing nullclines of replication equations, the evolution trends of crowdsourcing ecology can be predicted.

11.2 Contributions

The major contributions of the dissertation manifest themselves through the proposal and development of a coherent framework of platform-driven crowdsourced manufacturing for MaaS. The deliverables are entailed in the strategy, fundamentals, methodology, validation, and application aspects, as elaborated below:

(1) At the strategy level, the following consensuses are achieved (Chapters 1 and 2):

- Propose platform-driven crowdsourced manufacturing as a new manufacturing paradigm;
- Build up consensus of MaaS as the next objective of manufacturing industries;
- Highlight the essentiality of adopting platform-driven crowdsourced manufacturing for MaaS.

(2) At the fundamental level, the following findings are obtained (Chapter 3 and 4):

- Proceed dichotomy analysis of industrial applicability;
- Identify essential stakeholders and demonstrate with a running case;
- Propose crowdsourced manufacturing workflow and holistic framework;
- Proceed structural implications of platform-driven crowdsourced manufacturing based on FBS model, which identifies needs for crowdsourcing contracting, task execution services, and management services are essential driving power from open innovators, manufacturers, and platforms, respectively;
- Propose a MaaS reference model of platform-driven crowdsourced manufacturing as a research agenda.

(3) In terms of the methodology and supporting tools, the following deliverables have been promised (Chapters 5, 6, 7, 8, 9, and 10):

- Crowdsourcing contracting evaluation incorporating explicit and inexplicit criteria through information content measurements, decision-tree monotonic ranking, and multi-utility theory;

- Optimal crowdsourcing task derivation and decomposition through bilevel game theoretic decision-making for a joint and equilibrium optimization solution;
- Networked material flow planning through cross-docking logistic services based on a B&P algorithm;
- Task dispatching and scheduling through real-time crowdsourcing task acceptance and accommodation based on an interactive bilevel optimization model;
- Blockchain-based smart contracting and distributed data management for real-time information streaming services in crowdsourced manufacturing;
- Revenue sharing in crowdsourced manufacturing through population dynamics modeling and analysis based on an ECC game model.

(4) As for validation and application, several experimental and case studies have been conducted, including:

- A case of tank trailer cluster to demonstrate fundamental issues of platform-driven crowdsourced manufacturing;
- A case study of evaluating frame welding for a chemical tank trailer;
- A case study of joint PFP and MLB design problem of tank semi-trailer family;
- A case study and experimental verification of crowdsourcing logistic services for a trailer industry cluster through proposed B&P method;
- A case study of car seats assembly plant to demonstrate r-OASR problem based on proposed BIO model;
- A case study of information service system design for a crowdsourcing innovative project throughout a MaaS process;

- A case study of tank trailer multi-cluster cooperation through ECC model and revenue sharing.

11.3 Limitations

As an exploratory study of the proposed platform-driven crowdsourced manufacturing for MaaS, it suffers several limitations.

(1) **The types of crowdsourcing process covered is limited:** This dissertation only covers a crowdsourcing process that the design parameters are defined, and manufacturers bid with their solutions. This process has not considered a concurrent engineering approach, in which the open innovators act as an OEM and find a manufacturer as a collaborator in product research and development. This situation implies a lack of design parameters. Besides, the qualification of manufacturers is not involved in this dissertation.

(2) **Ignore the evolution of manufacturers and product family:** This dissertation holds a stationary view on manufacturers, which ignores capability changing and clusters of manufacturers. However, from a dynamic perspective, the capability and quality of manufacturers will increase along a collaboration relationship. The open innovator can also adjust the design of a product family to achieve a better fit with manufacturers. In service manufacturing, co-evolution of product families and manufacturers over generations caused by market demand changes and technological progress is worthy of in-depth research.

(3) **The robustness of proposed methods:** the robustness of evaluation methods needs to be further verified. It should prove an efficient behavior on a wider spectrum of evaluation criteria and complex manufacturer records. The robustness of methods proposed

in task execution should be verified. The negotiations regarding distinct due dates, revenues, unit tardiness penalty cost, and so on are important in crowdsourcing task execution.

(4) **Experimental validation of proposed methodology:** the methodology proposed in this research is not validated through an experimental to valid the research foundation. The motivation of stakeholders and the efficiency of the proposed platform driven crowdsourced manufacturing mentioned in this research requires a further validation.

(5) **Customer's view:** platform-driven crowdsourced manufacturing aims to fulfill customer's need, while the change of MaaS introduced to the market is not covered in this research.

11.4 Future Threads

Platform-driven crowdsourcing have fundamentally changed the way of organizing product fulfillment resources. Several ideas are elaborated below for potential endeavors in the future.

(1) **Incorporating uncertainty along task execution:** the task execution process should modify the existed optimal decision-making architecture to incorporate uncertainty from real-world. The task derivation and decomposition, which is formulated as PFP and MLB, in an uncertain environment has been separately optimized extensively in the literature, little has been done for the uncertain interactive decision-making of PFP and MLB in MaaS.

(2) **Include a wide scope of real-world management policies:** This dissertation discusses platform-driven crowdsourced manufacturing theoretically. A transition

roadmap towards MaaS through crowdsourced manufacturing is essential. An interview and discussion with professionals from academia and industries will be a milestone, such as challenges identification towards crowdsourcing for engineering design (Shergadwala et al., 2020, Forbes et al., 2020).

(3) **Relax some assumptions for extended applicability:** The methods proposed in this dissertation are based on some assumptions to limit the scope of research. Some of these can be lifted in the future avenue. The ranking algorithm in Chapter 5 assumes that the dataset is clean enough to get rid of any data point that violate monotonic constraints, a better classification and ranking algorithm can perform better for real-world data. Chapter 6 assumes that the manufacturers in each cluster are not able to cross multiple clusters or transfer from cluster to cluster. A modified algorithm to accommodate cluster crossing and transfer cost is promising in future threads. Chapter 8 assume rolling horizon replanning policy and event-driven replanning policies for serial production lines. In future work, the performance assessment of these policies using different production systems can be beneficial for replanning policy selection under different production operation parameters.

(4) **Experiment design for validation:** An experimental design should be covered in the future to validate the related theory in platform-driven crowdsourced manufacturing. Because it is a method to deliver MaaS, the measurement of service level is essential for setting experiment to validate functions of MaaS. Getting along with the increasing instances of platform-driven crowdsourced manufacturing are installed in the real-world, a survey of stakeholders can also validate this research. And the view of customers can be studied based on the real-world instances.

REFERENCES

ABBAS, M. & ELMARAGHY, H. 2018. Co-platforming of products and assembly systems. *Omega,* 78**,** 5-20.

AFUAH, A. & TUCCI, C. L. 2012. Crowdsourcing as a solution to distant search. *Academy of management review,* 37**,** 355-375.

AGOSTINHO, C., DUCQ, Y., ZACHAREWICZ, G., SARRAIPA, J., LAMPATHAKI, F., POLER, R. & JARDIM-GONCALVES, R. 2016. Towards a sustainable interoperability in networked enterprise information systems: Trends of knowledge and model-driven technology. *Computers in industry,* 79**,** 64-76.

AKAY, D., KULAK, O. & HENSON, B. 2011. Conceptual design evaluation using interval type-2 fuzzy information axiom. *Computers in Industry,* 62**,** 138-146.

ALEXOPOULOS, C., SEILA, A. F. & BANKS, J. 1998. Output data analysis. *Handbook of Simulation.* John Wiley & Sons.

ALLON, G. & BABICH, V. 2020. Crowdsourcing and crowdfunding in the manufacturing and services sectors. *Manufacturing & Service Operations Management,* 22**,** 102-112.

ANDREW, K. 2007. The Cult of the amateur: How today's Internet is killing our culture and assaulting our economy. *Nicholas Brealey, London*.

APPLEYARD, M. M. & CHESBROUGH, H. W. 2017. The dynamics of open strategy: from adoption to reversion. *Long Range Planning,* 50**,** 310-321.

ARABANI, A. B., GHOMI, S. F. & ZANDIEH, M. 2011. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert systems with Applications,* 38**,** 1964-1979.

ARREDONDO, F. & MARTINEZ, E. 2010. Learning and adaptation of a policy for dynamic order acceptance in make-to-order manufacturing. *Computers & Industrial Engineering,* 58**,** 70-83.

AZOUZ, N., SILVA BELISÁRIO, L., AMMAR, A. & PIERREVAL, H. 2018. Addressing over-correction in adaptive card-based pull control systems. *International Journal of Computer Integrated Manufacturing,* 31**,** 1189-1204.

BAHGA, A. & MADISETTI, V. K. 2016. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications,* 9**,** 533-546.

BANERJEE, A. G., BECKMANN, B., CARBONE, J., DEROSE, L., GIANI, A., KOUDAL, P., MACKENZIE, P., SALVO, J., YANG, D. & YUND, W. Cloud Computing-

Based Marketplace for Collaborative Design and Manufacturing. International Internet of Things Summit, 2015. Springer, 409-418.

BARNHART, C., JOHNSON, E. L., NEMHAUSER, G. L., SAVELSBERGH, M. W. & VANCE, P. H. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations research,* 46**,** 316-329.

BAYUS, B. L. 2010. Crowdsourcing and individual creativity over time: the detrimental effects of past success. *Available at SSRN 1667101*.

BILGE, Ü., KURTULAN, M. & KıRAÇ, F. 2007. A tabu search algorithm for the single machine total weighted tardiness problem. *European Journal of Operational Research,* 176**,** 1423-1435.

BINNEKAMP, R., VAN GUNSTEREN, L. A., VAN LOON, P.-P. & BARENDSE, P. 2006. *Open Design: A Stakeholder-oriented Approach in Architecture, Urban Planning, and Project Management; Collecting the Following Works: Open Design, a Collaborative Approach to Architecture, Open Design and Construct Management,[and] Open Design, Cases and Exercises*, Ios Press.

BLACKBURN, J. D., KROPP, D. H. & MILLEN, R. A. 1986. A comparison of strategies to dampen nervousness in MRP systems. *Management science,* 32**,** 413-429.

BOGERS, M., ZOBEL, A.-K., AFUAH, A., ALMIRALL, E., BRUNSWICKER, S., DAHLANDER, L., FREDERIKSEN, L., GAWER, A., GRUBER, M. & HAEFLIGER, S. 2017. The open innovation research landscape: Established perspectives and emerging themes across different levels of analysis. *Industry and Innovation,* 24**,** 8-40.

BOISSEAU, É., OMHOVER, J.-F. & BOUCHARD, C. 2018. Open-design: A state of the art review. *Design Science,* 4.

BONABEAU, E. 2009. Decisions 2.0: The power of collective intelligence. *MIT Sloan management review,* 50**,** 45.

BORSHCHEV, A. & FILIPPOV, A. 2004. *From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools*.

BOZER, Y. A. & CARLO, H. J. 2008. Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions,* 40**,** 1007-1018.

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A. & STONE, C. J. 2017. *Classification and regression trees,* Routledge.

BREM, A. & VOIGT, K.-I. 2009. Integration of market pull and technology push in the corporate front end and innovation management—Insights from the German software industry. *Technovation,* 29**,** 351-367.

BRENNER, W., ZARNEKOW, R. & WITTIG, H. 2012. *Intelligent software agents: foundations and applications*, Springer Science & Business Media.

BRETTEL, M., FRIEDERICHSEN, N., KELLER, M. & ROSENBERG, M. 2017. How virtualization, decentralization and network building change the manufacturing landscape: an industry 4.0 perspective. *FormaMente*, 12.

BROOM, M., CANNINGS, C. & VICKERS, G. 1997. Multi-player matrix games. *Bulletin of mathematical biology,* 59**,** 931-952.

BRYAN, A., HU, S. & KOREN, Y. 2007. Concurrent product portfolio planning and mixed product assembly line balancing. *Chin. J. Mech. Eng.,* 20**,** 96-99.

BRYAN, A., WANG, H. & ABELL, J. 2013. Concurrent design of product families and reconfigurable assembly systems. *Journal of Mechanical Design,* 135**,** 051001.

BÜCHELER, T. & SIEG, J. H. 2011. Understanding science 2.0: Crowdsourcing and open innovation in the scientific method. *Procedia Computer Science,* 7**,** 327-329.

BUHRMESTER, M., KWANG, T. & GOSLING, S. D. 2016. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality data?

BUIJS, P., VIS, I. F. & CARLO, H. J. 2014. Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research,* 239**,** 593-608.

BULL, G. & GROVES, J. 2009. The Democratization of Production. *Learning & Leading with Technology,* 37**,** 36-37.

CAO, Y., LUO, X. G., KWONG, C. K., TANG, J. F. & ZHOU, W. 2012. Joint optimization of product family design and supplier selection under multinomial logit consumer choice rule. *Concurrent Engineering,* 20**,** 335-347.

CHAURASIA, S. N. & SINGH, A. 2017. Hybrid evolutionary approaches for the single machine order acceptance and scheduling problem. *Applied Soft Computing,* 52**,** 725-747.

CHEN, J. 2016. Haier is the sea: CEO zhang ruimin's innovative management. *Management and Organization Review,* 12**,** 799-802.

CHEN, S., CAI, X., WANG, X., LIU, A., LU, Q., XU, X. & TAO, F. 2021. Blockchain applications in PLM towards smart manufacturing. *The International Journal of Advanced Manufacturing Technology***,** 1-15.

CHEN, W., LIU, H. & QI, E. 2020. Discrete event-driven model predictive control for real-time work-in-process optimization in serial production systems. *Journal of Manufacturing Systems,* 55**,** 132-142.

CHEN, X., SZOLNOKI, A. & PERC, M. 2015. Competition and cooperation among different punishing strategies in the spatial public goods game. *Physical Review E,* 92**,** 012819.

CHESBROUGH, H. 2006. *Open business models: How to thrive in the new innovation landscape*, Harvard Business Press.

CHESBROUGH, H. & ROSENBLOOM, R. S. 2002. The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin - off companies. *Industrial and Corporate Change,* 11**,** 529-555.

CHESBROUGH, H. W. 2007. Why companies should have open business models. *MIT Sloan management review,* 48**,** 22.

CHESBROUGH, H. W. & APPLEYARD, M. M. 2007. Open innovation and strategy. *California management review,* 50**,** 57-76.

CHIDA, T., KAIHARA, T., FUJII, N. & KOKURYO, D. 2019. Stability analysis on resource matching in crowdsourced manufacturing. *Procedia CIRP,* 81**,** 405-410.

CHOI, E. & TCHA, D.-W. 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research,* 34**,** 2080-2095.

CLAUDIO, D., KREMER, G. E. O., BRAVO-LLERENA, W. & FREIVALDS, A. 2014. A dynamic multi-attribute utility theory–based decision support system for patient prioritization in the emergency department. *IIE transactions on healthcare systems engineering,* 4**,** 1-15.

COSTA, L., CONTARDO, C. & DESAULNIERS, G. 2019. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science,* 53**,** 946-985.

CRAWFORD, M. 2018. How industry 4.0 impacts engineering design. *The American Society of Mechanical Engineers*.

CROSBY, M., PATTANAYAK, P., VERMA, S. & KALYANARAMAN, V. 2016. Blockchain technology: Beyond bitcoin. *Applied Innovation,* 2**,** 71.

DE JONG, A. W., RUBRICO, J. I., ADACHI, M., NAKAMURA, T. & OTA, J. Big data in automation: Towards generalized makespan estimation in shop scheduling problems. 2017 13th IEEE Conference on Automation Science and Engineering (CASE), 2017. IEEE, 1516-1521.

DE JONG, A. W., RUBRICO, J. I., ADACHI, M., NAKAMURA, T. & OTA, J. 2019. A generalised makespan estimation for shop scheduling problems, using visual data and a convolutional neural network. *International Journal of Computer Integrated Manufacturing,* 32**,** 559-568.

DE SCHUTTER, B. & VAN DEN BOOM, T. 2001. Model predictive control for max-plus-linear discrete event systems. *Automatica,* 37**,** 1049-1056.

DHANARAJ, C. & PARKHE, A. 2006. Orchestrating innovation networks. *Academy of management review,* 31**,** 659-669.

DHINGRA, A. & RAO, S. S. 1995. A cooperative fuzzy game theoretic approach to multiple objective design optimization. *European Journal of Operational Research,* 83**,** 547-567.

DING, K., JIANG, P., LENG, J. & CAO, W. 2016. Modeling and analyzing of an enterprise relationship network in the context of social manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 230**,** 752-769.

DU, G., XIA, Y., JIAO, R. J. & LIU, X. 2019a. Leader-follower joint optimization problems in product family design. *Journal of Intelligent Manufacturing,* 30**,** 1387-1405.

DU, G., ZHANG, Y., LIU, X., JIAO, R. J., XIA, Y. & LI, Y. 2019b. A review of leader-follower joint optimization problems and mathematical models for product design and development. *The International Journal of Advanced Manufacturing Technology,* 103**,** 3405-3424.

DU, M., MA, X., ZHANG, Z., WANG, X. & CHEN, Q. A review on consensus algorithm of blockchain. 2017 IEEE international conference on systems, man, and cybernetics (SMC), 2017. IEEE, 2567-2572.

DU, X., JIAO, J. & TSENG, M. M. 2006. Understanding customer satisfaction in product customization. *The International Journal of Advanced Manufacturing Technology,* 31**,** 396-406.

DUAN, Q., ZENG, J., CHAKRABARTY, K. & DISPOTO, G. 2015. Data-driven optimization of order admission policies in a digital print factory. *ACM Transactions on Design Automation of Electronic Systems (TODAES),* 20**,** 1-25.

DUDEK, G. & STADTLER, H. 2005. Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research,* 163**,** 668-687.

EARTH, G. *Quanpu Zhen*.

EBBEN, M. J., HANS, E. W. & WEGHUIS, F. O. 2005. Workload based order acceptance in job shop environments. *OR spectrum,* 27**,** 107-122.

EISENMANN, T., PARKER, G. & VAN ALSTYNE, M. 2011. Platform envelopment. *Strategic management journal,* 32**,** 1270-1285.

ELMARAGHY, H., AZAB, A., SCHUH, G. & PULZ, C. 2009. Managing variations in products, processes and manufacturing systems. *CIRP annals,* 58**,** 441-446.

ELMARAGHY, H. & ELMARAGHY, W. 2014. Variety, Complexity and Value Creation. Enabling Manufacturing Competitiveness and Economic Sustainability. Springer.

ELMARAGHY, H., SCHUH, G., ELMARAGHY, W., PILLER, F., SCHÖNSLEBEN, P., TSENG, M. & BERNARD, A. 2013. Product variety management. *Cirp Annals,* 62**,** 629-652.

EPPINGER, S. & ULRICH, K. 2015. *Product design and development*, McGraw-Hill Higher Education.

EVERS, J. J., LOEVE, L. & LINDEIJER, D. G. 2000. The service - oriented agile logistic control and engineering system: SERVICES. *Logistics information management*.

FAHMY, S. A., ELMEKKAWY, T. Y. & BALAKRISHNAN, S. 2007. Analysis of reactive deadlock-free scheduling in flexible job shops. *International Journal of Flexible Manufacturing Systems,* 19**,** 264-285.

FARAHANI, S. S., VAN DEN BOOM, T. & DE SCHUTTER, B. 2017. On optimization of stochastic max–min-plus-scaling systems—An approximation approach. *Automatica,* 83**,** 20-27.

FATAHI VALILAI, O. & HOUSHMAND, M. 2014. A platform for optimisation in distributed manufacturing enterprises based on cloud manufacturing paradigm. *International Journal of Computer Integrated Manufacturing,* 27**,** 1031-1054.

FEELDERS, A. & PARDOEL, M. Pruning for monotone classification trees. International Symposium on Intelligent Data Analysis, 2003. Springer, 1-12.

FEILLET, D., DEJAX, P., GENDREAU, M. & GUEGUEN, C. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal,* 44**,** 216-229.

FORBES, H., SHAEFER, D., SHERGADWALA, M. N. & PANCHAL, J. H. Investigating the challenges of crowdsourcing for engineering design: An interview study with organizations of different sizes. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2020. American Society of Mechanical Engineers, V008T08A039.

FRAMINAN, J. M., FERNANDEZ-VIAGAS, V. & PEREZ-GONZALEZ, P. 2019. Using real-time information to reschedule jobs in a flowshop with variable processing times. *Computers & Industrial Engineering,* 129**,** 113-125.

FRANK, A., 2010. UCI machine learning repository. http://archive. ics. uci. edu/ml.

FREITAG, M., BECKER, T. & DUFFIE, N. A. 2015. Dynamics of resource sharing in production networks. *CIRP Annals,* 64**,** 435-438.

FRIEDMAN, D. 1991. Evolutionary games in economics. *Econometrica: Journal of the Econometric Society***,** 637-666.

FUGE, M., TEE, K., AGOGINO, A. & MATON, N. 2014. Analysis of collaborative design networks: A case study of openideo. *Journal of Computing and Information Science in Engineering,* 14.

FÜLLER, J. 2010. Refining virtual co-creation from a consumer perspective. *California management review,* 52**,** 98-122.

GAONKAR, R. & VISWANADHAM, N. 2001. Collaboration and information sharing in global contract manufacturing networks. *IEEE/ASME transactions on mechatronics,* 6**,** 366-376.

GAWER, A. & CUSUMANO, M. A. 2014. Industry platforms and ecosystem innovation. *Journal of product innovation management,* 31**,** 417-433.

GHALEB, M., ZOLFAGHARINIA, H. & TAGHIPOUR, S. 2020. Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns. *Computers & Operations Research,* 123**,** 105031.

GIANNOCCARO, I. & PONTRANDOLFO, P. 2004. Supply chain coordination by revenue sharing contracts. *International journal of production economics,* 89**,** 131-139.

GOKHALE, C. S. & TRAULSEN, A. 2010. Evolutionary games in the multiverse. *Proceedings of the National Academy of Sciences,* 107**,** 5500-5504.

GONG, X. 2018. *Collaborative-crowdsourcing product fulfillment for open design and manufacturing.* Georgia Institute of Technology.

GONG, X., JIAO, R., JARIWALA, A. & MORKOS, B. 2021. Crowdsourced manufacturing cyber platform and intelligent cognitive assistants for delivery of manufacturing as a service: fundamental issues and outlook. *The International Journal of Advanced Manufacturing Technology***,** 1-11.

GONG, X., WANG, S., ZHANG, W., SONG, M., JIAO, R. J. & GEBRAEEL, N. Z. Tournament-based Crowdsourcing Contracting Evaluation by Information Content Measurement. IIE Annual Conference. Proceedings, 2019. Institute of Industrial and Systems Engineers (IISE), 1475-1480.

GOODARZI, A. H., DIABAT, E., JABBARZADEH, A. & PAQUET, M. 2021. An M/M/c queue model for vehicle routing problem in multi-door cross-docking environments. *Computers & Operations Research***,** 105513.

GORE, R. W. 1976. Process for producing porous products. Google Patents.

GOVINDAN, K. & POPIUC, M. N. 2014. Reverse supply chain coordination by revenue sharing contract: A case for the personal computers industry. *European Journal of Operational Research,* 233**,** 326-336.

GRANGIER, P., GENDREAU, M., LEHUÉDÉ, F. & ROUSSEAU, L.-M. 2017. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research,* 84**,** 116-126.

GRANGIER, P., GENDREAU, M., LEHUÉDÉ, F. & ROUSSEAU, L.-M. 2021. The vehicle routing problem with cross-docking and resource constraints. *Journal of Heuristics,* 27**,** 31-61.

GREEN, D. P., FUCHSBERGER, V., KIRK, D., TAYLOR, N., CHATTING, D., MEISSNER, J. L., MURER, M., TSCHELIGI, M., LINDTNER, S. & BJORN, P. Open design at the intersection of making and manufacturing. Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2017. 542-549.

GREWAL-CARR, V. & BATES, C. 2016. The three billion: Enterprise crowdsourcing and the growing fragmentation of work. *Retrieved from Deloitte. com: https://www2. deloitte. com/content/dam/Deloitte/de/Documents/Innovation/us-cons-enterprise-crowdsourcing-and-gro wing-fragmentation-of-work,* 20.

GRINGER, J. 2018. *THE TRAILER CITY* [Online]. motionist.com. Available: https://motionist.com/en/the-trailer-city/ [Accessed 2019].

GROMICHO, J., VAN HOORN, J. J., KOK, A. & SCHUTTEN, J. M. 2012. Restricted dynamic programming: a flexible framework for solving realistic VRPs. *Computers & operations research,* 39**,** 902-909.

GRÖNLUND, J., SJÖDIN, D. R. & FRISHAMMAR, J. 2010. Open innovation and the stage-gate process: A revised model for new product development. *California management review,* 52**,** 106-131.

GUAZZINI, A., VILONE, D., DONATI, C., NARDI, A. & LEVNAJIĆ, Z. 2015. Modeling crowdsourcing as collective problem solving. *Scientific reports,* 5**,** 1-11.

GUO, W. & JIANG, P. 2019. Manufacturing service order allocation in the context of social manufacturing based on Stackelberg game. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 233**,** 1890-1901.

GUPTA, A., AGARWAL, R. K. & WHITMAN, L. Supply chain agent decision aid system (SCADAS). Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304), 2001. IEEE, 553-559.

GUPTA, A., MARANAS, C. D. & MCDONALD, C. M. 2000. Mid-term supply chain planning under demand uncertainty: customer demand satisfaction and inventory management. *Computers & Chemical Engineering,* 24**,** 2613-2621.

HAMALAINEN, M. & KARJALAINEN, J. 2017. Social manufacturing: When the maker movement meets interfirm production networks. *Business Horizons,* 60**,** 795-805.

HANAFY, M. & ELMARAGHY, H. 2017. Modular product platform configuration and co-planning of assembly lines using assembly and disassembly. *Journal of manufacturing systems,* 42**,** 289-305.

HASHEMIAN, N., DIALLO, C. & VIZVÁRI, B. 2014. Makespan minimization for parallel machines scheduling with multiple availability constraints. *Annals of Operations Research,* 213**,** 173-186.

HASHIMOTO, K. & AIHARA, K. 2009. Fixation probabilities in evolutionary game dynamics with a two-strategy game in finite diploid populations. *Journal of theoretical biology,* 258**,** 637-645.

HAYES, R. H. & CLARK, K. B. 1985. Explaining observed productivity differentials between plants: Implications for operations research. *Interfaces,* 15**,** 3-14.

HERNANDEZ, G., SEEPERSAD, C. C. & MISTREE, F. 2002. Designing for maintenance: a game theoretic approach. *Engineering Optimization,* 34**,** 561-577.

HERRERA, C., BELMOKHTAR-BERRAF, S., THOMAS, A. & PARADA, V. 2016. A reactive decision-making approach to reduce instability in a master production schedule. *International Journal of Production Research,* 54**,** 2394-2404.

HINES, W. 1987. Evolutionary stable strategies: a review of basic theory. *Theoretical Population Biology,* 31**,** 195-272.

HIRSCH, M. W., SMALE, S. & DEVANEY, R. L. 2012. *Differential equations, dynamical systems, and an introduction to chaos*, Academic press.

HOFBAUER, J. & SIGMUND, K. 1998. *Evolutionary games and population dynamics*, Cambridge university press.

HOSSEINI, M., PHALP, K., TAYLOR, J. & ALI, R. The four pillars of crowdsourcing: A reference model. 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), 2014. IEEE, 1-12.

HOßFELD, T., HIRTH, M. & TRAN-GIA, P. Modeling of crowdsourcing platforms and granularity of work organization in future internet. 2011 23rd International Teletraffic Congress (ITC), 2011. IEEE, 142-149.

HOU, L., WU, Y.-M., LAI, R.-S. & TSAI, C.-T. 2014. Product family assembly line balancing based on an improved genetic algorithm. *The International Journal of Advanced Manufacturing Technology,* 70**,** 1775-1786.

HOWE, J. 2006. The rise of crowdsourcing. *Wired magazine,* 14**,** 1-4.

HOZAK, K. & HILL, J. A. 2009. Issues and opportunities regarding replanning and rescheduling frequencies. *International Journal of Production Research,* 47**,** 4955-4970.

HU, Q., CHE, X., ZHANG, L., ZHANG, D., GUO, M. & YU, D. 2011. Rank entropy-based decision trees for monotonic classification. *IEEE Transactions on Knowledge and Data Engineering,* 24**,** 2052-2064.

ILGIN, M. A., AKÇAY, H. & ARAZ, C. 2017. Disassembly line balancing using linear physical programming. *International Journal of Production Research,* 55**,** 6108-6119.

ILGIN, M. A. & GUPTA, S. M. 2012. Physical programming: A review of the state of the art. *Studies in Informatics and Control,* 21**,** 349-366.

IVANOV, D., DOLGUI, A. & SOKOLOV, B. 2012. Applicability of optimal control theory to adaptive supply chain planning and scheduling. *Annual Reviews in control,* 36**,** 73-84.

JAHANGIRIAN, M., ELDABI, T., NASEER, A., STERGIOULAS, L. K. & YOUNG, T. 2010. Simulation in manufacturing and business: A review. *European journal of operational research,* 203**,** 1-13.

JI, P., MA, X. & LI, G. 2015. Developing green purchasing relationships for the manufacturing industry: An evolutionary game theory perspective. *International Journal of Production Economics,* 166**,** 155-162.

JI, Y., JIAO, R. J., CHEN, L. & WU, C. 2013. Green modular design for material efficiency: a leader–follower joint optimization model. *Journal of Cleaner Production,* 41**,** 187-201.

JIANG, P. & DING, K. 2018. Analysis of personalized production organizing and operating mechanism in a social manufacturing environment. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 232**,** 2670-2676.

JIANG, P., DING, K. & LENG, J. 2016a. Towards a cyber-physical-social-connected and service-oriented manufacturing paradigm: Social Manufacturing. *Manufacturing Letters,* 7**,** 15-21.

JIANG, P., LENG, J., DING, K., GU, P. & KOREN, Y. 2016b. Social manufacturing as a sustainable paradigm for mass individualization. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 230**,** 1961-1968.

JIAO, J., MA, Q. & TSENG, M. M. 2003. Towards high value-added products and services: mass customization and beyond. *Technovation,* 23**,** 809-821.

JIAO, J. & TSENG, M. M. 1998. Fuzzy ranking for concept evaluation in configuration design for mass customization. *Concurrent Engineering,* 6**,** 189-206.

JIAO, J. & TSENG, M. M. 2004. Customizability analysis in design for mass customization. *Computer-Aided Design,* 36**,** 745-757.

JIAO, J. & ZHANG, Y. 2005. Product portfolio planning with customer-engineering interaction. *Iie Transactions,* 37**,** 801-814.

JIAO, J. R., SIMPSON, T. W. & SIDDIQUE, Z. 2007a. Product family design and platform-based product development: a state-of-the-art review. *Journal of intelligent Manufacturing,* 18**,** 5-29.

JIAO, J. R., XU, Q., WU, Z. & NG, N.-K. 2009. Coordinating product, process, and supply chain decisions: A constraint satisfaction approach. *Engineering Applications of Artificial Intelligence,* 22**,** 992-1004.

JIAO, J. R., YOU, X. & KUMAR, A. 2006. An agent-based framework for collaborative negotiation in the global manufacturing supply chain network. *Robotics and Computer-Integrated Manufacturing,* 22**,** 239-255.

JIAO, J. R., ZHANG, Y. & WANG, Y. 2007b. A heuristic genetic algorithm for product portfolio planning. *Computers & Operations Research,* 34**,** 1777-1799.

JUNG, W.-K., KIM, H., PARK, Y.-C., LEE, J.-W. & SUH, E. S. 2020. Real-time data-driven discrete-event simulation for garment production lines. *Production Planning & Control***,** 1-12.

KADAR, B., EGRI, P., PEDONE, G. & CHIDA, T. 2018. Smart, simulation-based resource sharing in federated production networks. *CIRP Annals,* 67**,** 503-506.

KAIHARA, T. 2001. Supply chain management with market economics. *International Journal of Production Economics,* 73**,** 5-14.

KAIHARA, T. 2003. Multi-agent based supply chain modelling with dynamic environment. *International Journal of Production Economics,* 85**,** 263-269.

KANG, H. S., LEE, J. Y., CHOI, S., KIM, H., PARK, J. H., SON, J. Y., KIM, B. H. & DO NOH, S. 2016. Smart manufacturing: Past research, present findings, and future directions. *International journal of precision engineering and manufacturing-green technology,* 3**,** 111-128.

KEENEY, R. L., RAIFFA, H. & MEYER, R. F. 1993. *Decisions with multiple objectives: preferences and value trade-offs,* Cambridge university press.

KITTUR, A., CHI, E. H. & SUH, B. Crowdsourcing user studies with Mechanical Turk. Proceedings of the SIGCHI conference on human factors in computing systems, 2008. 453-456.

KOCH, M. D. & TUMER, I. Y. Towards Open Design: The Emergent Face of Engineering---A Position Paper. DS 58-3: Proceedings of ICED 09, the 17th International

Conference on Engineering Design, Vol. 3, Design Organization and Management, Palo Alto, CA, USA, 24.-27.08. 2009, 2009. 97-108.

KOHLER, T. 2015. Crowdsourcing-based business models: how to create and capture value. *California management review,* 57**,** 63-84.

KOMAKI, G., SHEIKH, S. & MALAKOOTI, B. 2019. Flow shop scheduling problems with assembly operations: a review and new trends. *International Journal of Production Research,* 57**,** 2926-2955.

KONGAR, E. & GUPTA, S. M. 2009. Solving the disassembly-to-order problem using linear physical programming. *International Journal of Mathematics in Operational Research,* 1**,** 504-531.

KONUR, D. & GOLIAS, M. M. 2013. Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E: Logistics and Transportation Review,* 49**,** 71-91.

KORTMANN, S. & PILLER, F. 2016. Open business models and closed-loop value chains: Redefining the firm-consumer relationship. *California Management Review,* 58**,** 88-108.

KOUHIZADEH, M. & SARKIS, J. 2018. Blockchain practices, potentials, and perspectives in greening supply chains. *Sustainability,* 10**,** 3652.

KRISHNAMURTY, S. 2006. Normative decision analysis in engineering design. *Decision Making in Engineering Design,* 4**,** 21-33.

KUSIAK, A. 2018. Smart manufacturing. *International Journal of Production Research,* 56**,** 508-517.

KUSIAK, A. 2019. Service manufacturing: Basic concepts and technologies. *Journal of Manufacturing Systems,* 52**,** 198-204.

KUSIAK, A. 2020. Service manufacturing= Process-as-a-service+ manufacturing operations-as-a-service. Springer.

KWONG, C. K., LUO, X. & TANG, J. 2010. A multiobjective optimization approach for product line design. *IEEE Transactions on Engineering Management,* 58**,** 97-108.

LADIER, A.-L. & ALPAN, G. 2016. Cross-docking operations: Current research versus industry practice. *Omega,* 62**,** 145-162.

LAI, X., XIE, M. & TAN, K.-C. 2006. QFD optimization using linear physical programming. *Engineering optimization,* 38**,** 593-607.

LAKHANI, K. R. & PANETTA, J. A. 2007. The principles of distributed innovation. *Innovations: technology, governance, globalization,* 2**,** 97-112.

LEE, J., AZAMFAR, M. & SINGH, J. 2019. A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems. *Manufacturing letters,* 20**,** 34-39.

LEE, Y. H., JUNG, J. W. & LEE, K. M. 2006. Vehicle routing scheduling for cross-docking in the supply chain. *Computers & Industrial Engineering,* 51**,** 247-256.

LEI, D. & GUO, X. 2015. A parallel neighborhood search for order acceptance and scheduling in flow shop environment. *International Journal of Production Economics,* 165**,** 12-18.

LEIMEISTER, J. M., HUBER, M., BRETSCHNEIDER, U. & KRCMAR, H. 2009. Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition. *Journal of management information systems,* 26**,** 197-224.

LENG, J., JIANG, P., XU, K., LIU, Q., ZHAO, J. L., BIAN, Y. & SHI, R. 2019. Makerchain: A blockchain with chemical signature for self-organizing process in social manufacturing. *Journal of Cleaner Production,* 234**,** 767-778.

LENG, J., JIANG, P. & ZHENG, M. 2017. Outsourcer–supplier coordination for parts machining outsourcing under social manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 231**,** 1078-1090.

LEWIN, A. Y., VÄLIKANGAS, L. & CHEN, J. 2017. Enabling open innovation: Lessons from Haier. *International Journal of Innovation Studies,* 1**,** 5-19.

LEWIS, K., CHEN, W. & SCHMIDT, L. 2006. Decision making in engineering design.

LI, H., JI, Y., CHEN, L. & JIAO, R. J. 2015. Bi-level coordinated configuration optimization for product-service system modular design. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 47**,** 537-554.

LI, M., YANG, F., UZSOY, R. & XU, J. 2016. A metamodel-based Monte Carlo simulation approach for responsive production planning of manufacturing systems. *Journal of Manufacturing Systems,* 38**,** 114-133.

LI, Z., BARENJI, A. V. & HUANG, G. Q. 2018a. Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform. *Robotics and computer-integrated manufacturing,* 54**,** 133-144.

LI, Z., WANG, W., LIU, G., LIU, L., HE, J. & HUANG, G. 2018b. Toward open manufacturing. *Industrial Management & Data Systems*.

LIN, S.-W. & YING, K.-C. 2015. Order acceptance and scheduling to maximize total net revenue in permutation flowshops with weighted tardiness. *Applied Soft Computing,* 30**,** 462-474.

LIU, X., DU, G., JIAO, R. J. & XIA, Y. 2018. Co-evolution of product family configuration and supplier selection: a game-theoretic bilevel optimisation approach. *Journal of Engineering Design,* 29**,** 201-234.

LIU, Y. 2016. *Game theoretic optimization for high variety assembly system design.* Georgia Institute of Technology.

LIU, Y. & LI, J. 2010. Split and merge production systems: performance analysis and structural properties. *IIE Transactions,* 42**,** 422-434.

LOZANO, L., DUQUE, D. & MEDAGLIA, A. L. 2016. An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science,* 50**,** 348-357.

LUBIK, S., LIM, S., PLATTS, K. & MINSHALL, T. 2013. Market - pull and technology - push in manufacturing start - ups in emerging industries. *Journal of Manufacturing Technology Management*.

LÜTTGENS, D., POLLOK, P., ANTONS, D. & PILLER, F. 2014. Wisdom of the crowd and capabilities of a few: internal success factors of crowdsourcing for innovation. *Journal of Business Economics,* 84**,** 339-374.

MA, S., DU, G., JIAO, J. & ZHANG, R. 2016. Hierarchical game joint optimization for product family-driven modular design. *Journal of the Operational Research Society,* 67**,** 1496-1509.

MANSOURI, S. A., GALLEAR, D. & ASKARIAZAD, M. H. 2012. Decision support for build-to-order supply chain management through multiobjective optimization. *International Journal of Production Economics,* 135**,** 24-36.

MARIA, A., MATTSON, C., ISMAIL-YAHAYA, A. & MESSAC, A. 2003. Linear physical programming for production planning optimization. *Engineering Optimization,* 35**,** 19-37.

MARTINI, A., MASSA, S. & TESTA, S. 2014. Customer co-creation projects and social media: The case of Barilla of Italy. *Business horizons,* 57**,** 425-434.

MATTA, A., PEZZONI, M. & SEMERARO, Q. 2012. A Kriging-based algorithm to optimize production systems approximated by analytical models. *Journal of Intelligent Manufacturing,* 23**,** 587-597.

MCALLISTER, C. D., SIMPSON, T. W., HACKER, K., LEWIS, K. & MESSAC, A. 2005. Integrating linear physical programming within collaborative optimization for multiobjective multidisciplinary design optimization. *Structural and Multidisciplinary Optimization,* 29**,** 178-189.

MEDEIROS, G. H., CAO, Q., ZANNI-MERK, C. & SAMET, A. Manufacturing as a service in industry 4.0: a multi-objective optimization approach. International Conference on Intelligent Decision Technologies, 2020. Springer, 37-47.

MESSAC, A., GUPTA, S. M. & AKBULUT, B. 1996. Linear physical programming: a new approach to multiple objective optimization. *Transactions on operational research,* 8**,** 39-59.

MIAO, C., DU, G., JIAO, R. J. & ZHANG, T. 2017. Coordinated optimisation of platform-driven product line planning by bilevel programming. *International Journal of Production Research,* 55**,** 3808-3831.

MICHALEK, J. J., EBBES, P., ADIGÜZEL, F., FEINBERG, F. M. & PAPALAMBROS, P. Y. 2011. Enhancing marketing with engineering: Optimal product line design for heterogeneous markets. *International Journal of Research in Marketing,* 28**,** 1-12.

MICKLETHWAITE, P. 2012. Open Design Now: Why Design Cannot Remain Exclusive by Bas van Abel, Lucas Evers, Roel Klaassen and Peter Troxler. Taylor & Francis.

MOGHADAM, S. S., GHOMI, S. F. & KARIMI, B. 2014. Vehicle routing scheduling problem with cross docking and split deliveries. *Computers & chemical engineering,* 69**,** 98-107.

MONOSTORI, L., KÁDÁR, B., BAUERNHANSL, T., KONDOH, S., KUMARA, S., REINHART, G., SAUER, O., SCHUH, G., SIHN, W. & UEDA, K. 2016. Cyber-physical systems in manufacturing. *Cirp Annals,* 65**,** 621-641.

MONTES, G. A. & GOERTZEL, B. 2019. Distributed, decentralized, and democratized artificial intelligence. *Technological Forecasting and Social Change,* 141**,** 354-358.

NAKAMOTO, S. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review***,** 21260.

NOBIBON, F. T. & LEUS, R. 2011. Exact algorithms for a generalization of the order acceptance and scheduling problem in a single-machine environment. *Computers & Operations Research,* 38**,** 367-378.

NORTON, M. I. & DANN, J. 2011. Local motors: designed by the crowd, built by the customer. *Harvard Business School Marketing Unit Case*.

NOWAK, M. A. 2012. Evolving cooperation. *Journal of theoretical biology,* 299**,** 1-8.

NOWAK, M. A. & MAY, R. M. 1992. Evolutionary games and spatial chaos. *Nature,* 359**,** 826-829.

OJANPERÄ, S., O’CLERY, N. & GRAHAM, M. 2018. Data science, artificial intelligence and the futures of work. *The Alan Turing Institute*.

OUELHADJ, D. & PETROVIC, S. 2009. A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling,* 12**,** 417-431.

PAKSERESHT, M., MAHDAVI, I., SHIRAZI, B. & MAHDAVI-AMIRI, N. 2020. Co-reconfiguration of product family and supply chain using leader–follower Stackelberg game theory: Bi-level multi-objective optimization. *Applied Soft Computing,* 91**,** 106203.

PARK, J. & SIMPSON, T. W. 2008. Toward an activity-based costing system for product families and product platforms in the early stages of development. *International Journal of Production Research,* 46**,** 99-130.

PERC, M., GÓMEZ-GARDENES, J., SZOLNOKI, A., FLORÍA, L. M. & MORENO, Y. 2013. Evolutionary dynamics of group interactions on structured populations: a review. *Journal of the royal society interface,* 10**,** 20120997.

PINE, B. J. 1993. *Mass customization*, Harvard business school press Boston.

PINEDO, M. 2012. *Scheduling*, Springer.

PIRMORADI, Z., WANG, G. G. & SIMPSON, T. W. 2014. A review of recent literature in product family design and platform-based product development. *Advances in product family and product platform design.* Springer.

POTHARST, R., BEN-DAVID, A. & VAN WEZEL, M. 2009. Two algorithms for generating structured and unstructured monotone ordinal data sets. *Engineering Applications of Artificial Intelligence,* 22**,** 491-496.

PRPIĆ, J., SHUKLA, P. P., KIETZMANN, J. H. & MCCARTHY, I. P. 2015. How to work a crowd: Developing crowd capital through crowdsourcing. *Business Horizons,* 58**,** 77-85.

QIN, S., VAN DER VELDE, D., CHATZAKIS, E., MCSTEA, T. & SMITH, N. 2016. Exploring barriers and opportunities in adopting crowdsourcing based new product development in manufacturing SMEs. *Chinese Journal of Mechanical Engineering,* 29**,** 1052-1066.

QU, T., LEI, S., WANG, Z., NIE, D., CHEN, X. & HUANG, G. Q. 2016. IoT-based real-time production logistics synchronization system under smart cloud manufacturing. *The International Journal of Advanced Manufacturing Technology,* 84**,** 147-164.

QUINLAN, J. R. 2014. *C4. 5: programs for machine learning*, Elsevier.

RAASCH, C., HERSTATT, C. & BALKA, K. 2009. On the open design of tangible goods. *R&d Management,* 39**,** 382-393.

RAGIN, C. C., DRASS, K. A. & DAVEY, S. 2006. Fuzzy-set/qualitative comparative analysis 2.0. *Tucson, Arizona: Department of Sociology, University of Arizona,* 23**,** 1949-1955.

RAHMAN, H. F., JANARDHANAN, M. N. & NIELSEN, I. E. 2019. Real-time order acceptance and scheduling problems in a flow shop environment using hybrid GA-PSO algorithm. *IEEE Access,* 7**,** 112742-112755.

RAHMAN, H. F., SARKER, R. & ESSAM, D. 2015. A real-time order acceptance and scheduling approach for permutation flow shop problems. *European Journal of Operational Research,* 247**,** 488-503.

REDLICH, T. & BRUHNS, F.-L. Open production: a new broker-based approach to interactive value creation and user manufacturing. ASME International Mechanical Engineering Congress and Exposition, 2008. 181-189.

REEVES, D. M., WELLMAN, M. P., MACKIE-MASON, J. K. & OSEPAYSHVILI, A. 2005. Exploring bidding strategies for market-based scheduling. *Decision Support Systems,* 39**,** 67-85.

RICHARDSON, M. 2016. Pre-hacked: Open Design and the democratisation of product development. *New Media & Society,* 18**,** 653-666.

ROCA, C. P., CUESTA, J. A. & SÁNCHEZ, A. 2009. Evolutionary game theory: Temporal and spatial effects beyond replicator dynamics. *Physics of life reviews,* 6**,** 208-249.

ROSAS, A. 2010. Evolutionary game theory meets social science: Is there a unifying rule for human cooperation? *Journal of Theoretical Biology,* 264**,** 450-456.

ROSEN, P. A. 2011. Crowdsourcing lessons for organizations. *Journal of Decision Systems,* 20**,** 309-324.

ROSSIT, D. A., TOHMÉ, F. & FRUTOS, M. 2018. The non-permutation flow-shop scheduling problem: a literature review. *Omega,* 77**,** 143-153.

SAHIN, F. & ROBINSON JR, E. P. 2005. Information sharing and coordination in make-to-order supply chains. *Journal of operations management,* 23**,** 579-598.

SANTOS, F. A., MATEUS, G. R. & DA CUNHA, A. S. 2011. A branch-and-price algorithm for a vehicle routing problem with cross-docking. *Electronic notes in discrete mathematics,* 37**,** 249-254.

SANTOS, F. A., MATEUS, G. R. & DA CUNHA, A. S. 2013. The pickup and delivery problem with cross-docking. *Computers & Operations Research,* 40**,** 1085-1093.

SAXTON, G. D., OH, O. & KISHORE, R. 2013. Rules of crowdsourcing: Models, issues, and systems of control. *Information Systems Management,* 30**,** 2-20.

SCHAEFER, D., THAMES, J. L., WELLMAN, R. D., WU, D., YIM, S. & ROSEN, D. W. Distributed collaborative design and manufacture in the cloud-motivation, infrastructure, and education. 2012 ASEE Annual Conference & Exposition, 2012. 25.472. 1-25.472. 21.

SCHOPKA, K. & KOPFER, H. 2016. An adaptive large neighborhood search for the reverse open vehicle routing problem with time windows. *Logistics management.* Springer.

SCHUH, G., POTENTE, T., WESCH-POTENTE, C., WEBER, A. R. & PROTE, J.-P. 2014. Collaboration Mechanisms to increase Productivity in the Context of Industrie 4.0. *Procedia Cirp,* 19**,** 51-56.

SCHWAB, K. 2017. The fourth industrial revolution. Currency. *New York*.

SHAKERI, M., LOW, M. Y. H., TURNER, S. J. & LEE, E. W. 2012. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers & Operations Research,* 39**,** 2564-2577.

SHEN, B., CHOI, T.-M. & MINNER, S. 2019. A review on supply chain contracting with information considerations: information updating and information asymmetry. *International Journal of Production Research,* 57**,** 4898-4936.

SHERGADWALA, M., FORBES, H., SCHAEFER, D. & PANCHAL, J. H. 2020. Challenges and research directions in crowdsourcing for engineering design: An interview study with industry professionals. *IEEE Transactions on Engineering Management*.

SHIN, M. & JUNG, M. 2004. MANPro: mobile agent-based negotiation process for distributed intelligent manufacturing. *International journal of production research,* 42**,** 303-320.

SHIUE, Y.-R., LEE, K.-C. & SU, C.-T. 2018. Real-time scheduling for a smart factory using a reinforcement learning approach. *Computers & Industrial Engineering,* 125**,** 604-614.

SILVA, Y. L. T., SUBRAMANIAN, A. & PESSOA, A. A. 2018. Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times. *Computers & operations research,* 90**,** 142-160.

SIMARD, C. & WEST, J. 2006. Knowledge networks and the geographic locus of innovation. *Open innovation: researching a new paradigm***,** 220-240.

SIMPSON, T. W., SIDDIQUE, Z. & JIAO, R. J. 2006. *Product platform and product family design: methods and applications,* Springer Science & Business Media.

SISKOS, J., LOCHARD, J. & LOMBARD, J. 1984. A multicriteria decision-making methodology under fuzziness: Application to the evaluation of radiological protection in nuclear power plants. *TIMS Studies in the Management Sciences,* 20**,** 261-283.

SLOTNICK, S. A. 2011. Order acceptance and scheduling: A taxonomy and review. *European Journal of Operational Research,* 212**,** 1-11.

SMITH, J. M. 1982. *Evolution and the Theory of Games*, Cambridge university press.

SOUZA, M. O., PACHECO, J. M. & SANTOS, F. C. 2009. Evolution of cooperation under N-person snowdrift games. *Journal of Theoretical Biology,* 260**,** 581-588.

STEINER, W. J. & HRUSCHKA, H. 2002. A probabilistic one-step approach to the optimal product line design problem using conjoint and cost data. *Review of Marketing Science Working Paper,* 441.

SUH, N. P. 1998. Axiomatic design theory for systems. *Research in engineering design,* 10**,** 189-209.

SUROWIECKI, J. 2005. *The wisdom of crowds,* Anchor.

SWAMINATHAN, J. M., SMITH, S. F. & SADEH, N. M. 1998. Modeling supply chain dynamics: A multiagent approach. *Decision sciences,* 29**,** 607-632.

TAN, C., HU, S. J., CHUNG, H., BARTON, K., PIYA, C., RAMANI, K. & BANU, M. 2017. Product personalization enabled by assembly architecture and cyber physical systems. *CIRP Annals,* 66**,** 33-36.

TAO, F., ZHANG, L., VENKATESH, V., LUO, Y. & CHENG, Y. 2011. Cloud manufacturing: a computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 225**,** 1969-1976.

TAO, F., ZUO, Y., DA XU, L. & ZHANG, L. 2014. IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. *IEEE Transactions on Industrial Informatics,* 10**,** 1547-1557.

TAPSCOTT, D. & WILLIAMS, A. D. 2008. *Wikinomics: How mass collaboration changes everything,* Penguin.

TAYLOR, C. & NOWAK, M. A. 2007. Transforming the dilemma. *Evolution: International Journal of Organic Evolution,* 61**,** 2281-2292.

THOMAS, L. C. 2012. *Games, theory and applications*, Courier Corporation.

THUAN, N. H., ANTUNES, P. & JOHNSTONE, D. Factors Influencing the Decision to Crowdsource. 2013a Berlin, Heidelberg. Springer Berlin Heidelberg, 110-125.

THUAN, N. H., ANTUNES, P. & JOHNSTONE, D. Factors influencing the decision to crowdsource. International Conference on Collaboration and Technology, 2013b. Springer, 110-125.

THURSTON, D. & CRAWFORD, C. 1994. A method for integrating end-user preferences for design evaluation in rule-based systems.

TIAN, F. An agri-food supply chain traceability system for China based on RFID & blockchain technology. 2016 13th international conference on service systems and service management (ICSSSM), 2016. IEEE, 1-6.

TIAN, Y., GOVINDAN, K. & ZHU, Q. 2014. A system dynamics model based on evolutionary game theory for green supply chain management diffusion among Chinese manufacturers. *Journal of Cleaner Production,* 80**,** 96-105.

TIWANA, A., KONSYNSKI, B. & BUSH, A. A. 2010. Research commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information systems research,* 21**,** 675-687.

TRAN, A., HASAN, S. U. & PARK, J.-Y. 2012. Crowd participation pattern in the phases of a product development process that utilizes crowdsourcing. *Industrial Engineering and Management Systems,* 11**,** 266-275.

TRENTIN, A., FORZA, C. & PERIN, E. 2012. Organisation design strategies for mass customisation: an information-processing-view perspective. *International Journal of Production Research,* 50**,** 3860-3877.

TSENG, M., JIAO, R. & WANG, C. 2010. Design for mass personalization. *CIRP annals,* 59**,** 175-178.

VALLANCE, R., KIANI, S. & NAYFEH, S. Open design of manufacturing equipment. Proceedings of the CHIRP 1st International Conference on Agile, Reconfigurable Manufacturing, 2001. 33-43.

VAN BELLE, J., VALCKENAERS, P. & CATTRYSSE, D. 2012. Cross-docking: State of the art. *Omega,* 40**,** 827-846.

VAN DEN BOOM, T. J., VAN DEN MUIJSENBERG, M. & DE SCHUTTER, B. 2020. Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models and dynamic graphs. *Discrete Event Dynamic Systems,* 30**,** 635-669.

VAN FOREEST, N. D., WIJNGAARD, J. & VAN DER VAART, T. 2010. Scheduling and order acceptance for the customised stochastic lot scheduling problem. *International Journal of Production Research,* 48**,** 3561-3578.

VANHAVERBEKE, W. & CHESBROUGH, H. 2014. A classification of open innovation and open business models. *New frontiers in open innovation,* 6**,** 50-68.

VINCENT, F. Y., JEWPANYA, P. & REDI, A. P. 2016. Open vehicle routing problem with cross-docking. *Computers & Industrial Engineering,* 94**,** 6-17.

VOGT, J. J. 2010. The successful cross‐dock based supply chain. *Journal of Business Logistics,* 31**,** 99-119.

WANG, D., DU, G., JIAO, R. J., WU, R., YU, J. & YANG, D. 2016. A Stackelberg game theoretic model for optimizing product family architecting with supply chain consideration. *International Journal of Production Economics,* 172**,** 1-18.

WANG, F. & JU, F. 2020. Transient and steady-state analysis of multistage production lines with residence time limits. *IEEE Transactions on Automation Science and Engineering,* 18**,** 122-134.

WANG, X., HUANG, G., HU, X. & CHENG, T. E. 2015. Order acceptance and scheduling on two identical parallel machines. *Journal of the Operational Research Society,* 66**,** 1755-1767.

WANG, X., XIE, X. & CHENG, T. 2013. Order acceptance and scheduling in a two-machine flowshop. *International Journal of Production Economics,* 141**,** 366-376.

WEBER, I., XU, X., RIVERET, R., GOVERNATORI, G., PONOMAREV, A. & MENDLING, J. Untrusted business process monitoring and execution using blockchain. International Conference on Business Process Management, 2016. Springer, 329-347.

WEBER, S. 2004. *The success of open source*, Harvard University Press.

WEN, M., LARSEN, J., CLAUSEN, J., CORDEAU, J.-F. & LAPORTE, G. 2009. Vehicle routing with cross-docking. *Journal of the Operational Research Society,* 60**,** 1708-1718.

WEYER, S., SCHMITT, M., OHMER, M. & GORECKY, D. 2015. Towards Industry 4.0-Standardization as the crucial challenge for highly modular, multi-vendor production systems. *Ifac-Papersonline,* 48**,** 579-584.

WHEELER, D. A. 2011. Why Free-Libre/Open Source Software (FLOSS)? Look at the Numbers!

WOOLDRIDGE, M. 2009. *An introduction to multiagent systems*, John wiley & sons.

WU, D., GREER, M. J., ROSEN, D. W. & SCHAEFER, D. 2013. Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems,* 32**,** 564-579.

WU, D., ROSEN, D. W., WANG, L. & SCHAEFER, D. 2015. Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Computer-Aided Design,* 59**,** 1-14.

WU, D., THAMES, J. L., ROSEN, D. W. & SCHAEFER, D. Towards a cloud-based design and manufacturing paradigm: looking backward, looking forward. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2012. American Society of Mechanical Engineers, 315-328.

WU, G.-H., CHENG, C.-Y., YANG, H.-I. & CHENA, C.-T. 2018. An improved water flow-like algorithm for order acceptance and scheduling with identical parallel machines. *Applied Soft Computing,* 71**,** 1072-1084.

WU, J., DU, G. & JIAO, R. J. 2021. Optimal postponement contracting decisions in crowdsourced manufacturing: A three-level game-theoretic model for product family architecting considering subcontracting. *European Journal of Operational Research,* 291**,** 722-737.

WU, L. & YANG, C. 2010. A solution of manufacturing resources sharing in cloud computing environment. *Cooperative design, visualization, and engineering.* Springer.

XIAO, T., QI, X. & YU, G. 2007. Coordination of supply chain after demand disruptions when retailers compete. *International Journal of Production Economics,* 109**,** 162-179.

XIAO, W., DU, G., ZHANG, Y. & LIU, X. 2018. Coordinated optimization of low-carbon product family and its manufacturing process design by a bilevel game-theoretic model. *Journal of Cleaner Production,* 184**,** 754-773.

XIAO, Y.-Y., ZHANG, R.-Q., ZHAO, Q.-H. & KAKU, I. 2012. Permutation flow shop scheduling with order acceptance and weighted tardiness. *Applied Mathematics and Computation,* 218**,** 7911-7926.

XIAO, Y., YUAN, Y., ZHANG, R.-Q. & KONAK, A. 2015. Non-permutation flow shop scheduling with order acceptance and weighted tardiness. *Applied Mathematics and Computation,* 270**,** 312-333.

XIONG, Y., DU, G. & JIAO, R. J. 2018. Modular product platforming with supply chain postponement decisions by leader-follower interactive optimization. *International Journal of Production Economics,* 205**,** 272-286.

XU, L., WANG, Q. & HUANG, S. 2015. Dynamic order acceptance and scheduling problem with sequence-dependent setup time. *International Journal of Production Research,* 53**,** 5797-5808.

XU, X., PAUTASSO, C., ZHU, L., GRAMOLI, V., PONOMAREV, A., TRAN, A. B. & CHEN, S. The blockchain as a software connector. 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), 2016. IEEE, 182-191.

XU, Z. & LIANG, M. 2006. Integrated planning for product module selection and assembly line design/reconfiguration. *International journal of production research,* 44**,** 2091-2117.

YANG, D., JIAO, J. R., JI, Y., DU, G., HELO, P. & VALENTE, A. 2015. Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. *European Journal of Operational Research,* 246**,** 263-280.

YENISEY, M. M. & YAGMAHAN, B. 2014. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega,* 45**,** 119-135.

YU, G., YU, Y. & AIJUN, L. 2016. Joint optimization of complex product variant design responding to customer requirement changes. *Journal of Intelligent & Fuzzy Systems,* 30**,** 397-408.

ZHANG, Y., LIU, S., LIU, Y., YANG, H., LI, M., HUISINGH, D. & WANG, L. 2018. The 'Internet of Things' enabled real-time scheduling for remanufacturing of automobile engines. *Journal of cleaner production,* 185**,** 562-575.

ZHAO, X., LIU, Z., DAN, F. & WANG, K. 2016. Plant breeding evaluation with rank entropy-based decision tree. *IFAC-PapersOnLine,* 49**,** 336-340.

ZHAO, Y. & ZHU, Q. 2014. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers,* 16**,** 417-434.

ZHENG, Q., LI, Y., CHEN, P. & DONG, X. An innovative IPFS-based storage model for blockchain. 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018a. IEEE, 704-708.

ZHENG, Z., XIE, S., DAI, H.-N., CHEN, X. & WANG, H. 2018b. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services,* 14**,** 352-375.

ZHONG, R. Y., HUANG, G. Q., LAN, S., DAI, Q., ZHANG, T. & XU, C. 2015. A two-level advanced production planning and scheduling model for RFID-enabled ubiquitous manufacturing. *Advanced Engineering Informatics,* 29**,** 799-812.

ZHOU, F., JIAO, R. J. & LEI, B. 2015. Bilevel game-theoretic optimization for product adoption maximization incorporating social network effects. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 46**,** 1047-1060.

ZOTT, C., AMIT, R. & MASSA, L. 2011. The Business Model: Recent Developments and Future Research. *Journal of Management,* 37**,** 1019-1042.

ZOU, J., CHANG, Q., ARINEZ, J., XIAO, G. & LEI, Y. 2017. Dynamic production system diagnosis and prognosis using model-based data-driven method. *Expert Systems with Applications,* 80**,** 200-209.