

Received December 30, 2020, accepted March 15, 2021, date of publication March 22, 2021, date of current version April 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068008

Service Family Design Optimization Considering a Multi-Server Queue

ZHUOTONG MIAO¹, XINGGANG LUO^{1,2}, ZHONGLIANG ZHANG², AND QING ZHOU²

¹College of Information Science and Engineering, Northeastern University, Shenyang 110004, China

²Management School, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Xinggang Luo (xgluo@mail.neu.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) Project under Grant 71831006, Grant 71801065, Grant 71771070, and Grant 71932005.

ABSTRACT Service firms not only need to develop differentiated services to meet the requirements of customers with various preferences, but also have to improve service flexibility and the efficiency of the service system. A service family is a strategy by which different modules are configured, based on the service platform, to create a variety of differentiated services. This research considered both the effect of multi-server queues and the heterogeneous service processes in service family design problems to establish a framework of service modularization from three different perspectives—process, activity, and component. To optimize the service family design, a nonlinear integer-programming model was established to determine the optimal configurations of modules and prices for the service family and the optimal number of servers. The model is transformed into a linear form, and thus, can be solved using a commercial optimization software for small-scale problems. An improved genetic algorithm integrated with a neighborhood search was further developed to solve large-scale problems. The correctness of the linearized model and the effectiveness of the meta-heuristic algorithm were demonstrated through case studies and numerical experiments.

INDEX TERMS Linearization, multi-server queue, optimization, service family design.

I. INTRODUCTION

In recent years, the contribution of service industries to the world economy has gradually increased; the service industry in the USA accounts for over 80% of the country's gross domestic product [1]. Faced with diversified customer needs and a competitive market environment, service firms not only need to develop differentiated service products, but also have to improve service flexibility, reduce service response time, and improve the efficiency of their service systems [2]. These goals can be achieved by applying service family, which is a new technology for managing diversified service products that is receiving increasing attention.

Services are special types of intangible products, and to improve flexibility and efficiency, it is necessary to establish a convenient manner of applying relevant theories and methods for product development and design. Fortunately, there are well-developed theories on product families that aim to configure and generate sets of similar products to improve product flexibility and thus meet the requirements of various market segments—through

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchao Jiang¹.

shared product features, components, or substructures—based on their methodology of modular design [2], [3]. Similarly, there are numerous research papers on product families [4].

Considering service characteristics, such as intangibility and perishability, Moon *et al.* [5] introduced the concept of a service family. A service family is established by configuring different functional modules on a service platform, which consists of processes, activities, objects, or features, to create a variety of differentiated services and thus enhance customer value by satisfying diversified requirements [6]. There are various service families in practice. For instance, as shown in Fig. 1, a car-rental company offers a service family consisting of four typical services: low-priced, economic, business, and luxury services. Each service comprises a package of service components (e.g., vehicle types and insurance levels) selected from the service modules. As required elements, vehicle selection and auto insurance are common functional modules, while the global position system, traveling data recorder (TDA), and personal accident insurance (PAI) are variant functional modules. Other service industries, such as tourism, software, logistics, and finance also provide service families that cater to market niches.

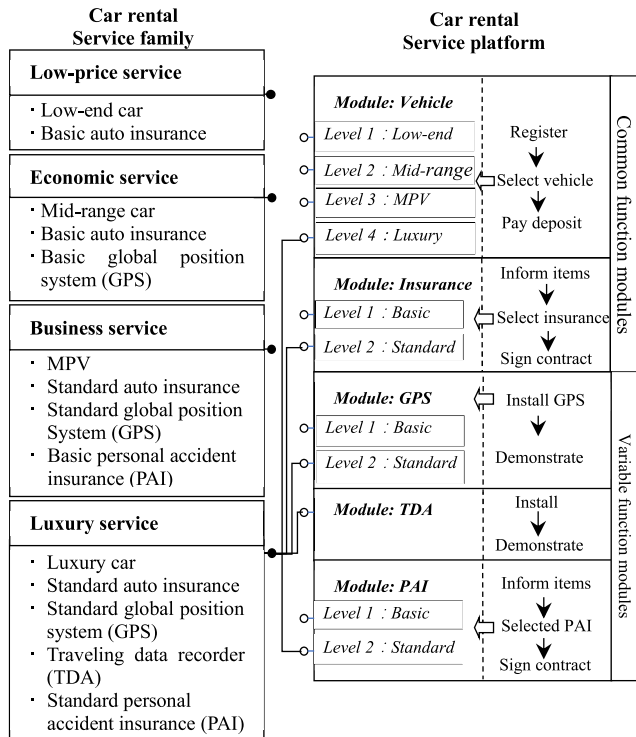


FIGURE 1. Example of a platform-based service family.

Scholars have proposed some theories and methods relating to service family design [7]–[9]; however, they did not consider the effect of the average waiting time in multi-server queues on customer purchase decisions. In fact, the server is one of the most important features of service products and an essential part of the service system. As production and consumption of services are simultaneous, each server can only receive the next customer after completing the service for the current customer. In the service system, queues are formed when the number of customer arrivals exceeds the servers’ capacity. Queuing is common in service industries, and the waiting time has a negative effect on customers’ perceptions and may even lead to loss of customers. The more the servers are in a service system, the less is the average waiting time. However, space or budget constraints limit the number of servers, and firms need to make a reasonable trade-off between reducing waiting time and controlling costs. Therefore, it is necessary to consider the negative impact of queue waiting time on service family design, which can then more accurately express customer preferences, help companies optimize the allocation of service resources, and ensure service quality while maximizing profit.

Moreover, existing methods for service family design do not consider heterogeneous service processes. Service is usually viewed as a process that interacts with customers [10], and its functions are accomplished by performing a series of service activities in a predetermined order [11]. To avoid homogeneous market competition and increase revenue, many service firms have designed differentiated service processes for their service functions. For example, travel

agencies provide tour-service processes for elderly customers who need guidance throughout their visits, while also providing free inter-city transportation and accommodation services that have simplified service processes. Therefore, service functions should not be limited to homogeneous service processes, and, to improve service competitiveness, it is necessary to consider heterogeneous service processes in the service family design.

In this study, we considered both the effect of multi-server queues and heterogeneous service processes in the service family design problem and established a nonlinear integer-programming model to maximize firms’ profits. The model was then transformed into a linear form that can be solved using a commercial optimization software for small-scale problems. An improved genetic algorithm, integrated with a neighborhood search, was further developed to solve large-scale problems. The correctness of the linearized model and the effectiveness of the meta-heuristic algorithm were demonstrated through case studies. Numerical experiments for the sensitivity analysis of various parameters were performed, the characteristics of the mathematical models were analyzed, and we obtained some notable managerial insights. Additionally, numerical experiments with different problem scales were performed to verify the effectiveness of the proposed approach. The main contributions of this study are summarized as follows:

1. To reflect real service scenarios, the average waiting time in a service queue is considered in the service family design optimization model. Linear functions are used in the mathematical model to formulate different sensitivity levels of market segments to waiting times, which describe customers’ purchase behaviors better.
2. Based on the theory of product family design, we proposed some new concepts for the service family, including the development of service modularization from three different perspectives—process, activity, and component. Therefore, the proposed model has better generalizability in representing complex services.
3. Systematic methods for solving the optimization model for the service family design were proposed. The established mathematical model was transformed into an equivalent linear model by introducing intermediate variables and related constraints, thus obtaining the global optimal solution for the small-scale problem. A meta-heuristic algorithm was developed to solve large-scale problems. Some notable management insights were also obtained by analyzing the optimization model through numerical experiments.

The remainder of this paper is organized as follows. In Section 2, we review the related literature on product and service development. In Section 3, we describe the optimization problem and formulation of the mathematical model. In Section 4, we propose an improved genetic algorithm combined with a neighborhood search to solve large-scale problems. In Section 5, we use a car rental service as an industrial case study to explain the proposed method of

service family design and evaluate the model and proposed algorithm. Section 6 concludes the study.

II. LITERATURE REVIEW

A. PRODUCT FAMILY DESIGN

A product family is a group of related products (i.e., product variants) derived from a product platform to satisfy a variety of market niches [2]. It can be regarded as an enabling technology for mass customization and has many advantages, such as increased flexibility, reduced development cost, and improved ability to upgrade products [12]. Shared structures and product technologies form the platform of a product family [3], and a final product is assembled from separate, independent modules [13]; for instance, Ford motor company launches economical, comfortable, and sporty versions of its Focus cars with diversified engine powers, driving assistance systems, and seat technologies and materials. From the viewpoint of marketing operations, a product family design aims to determine the optimal settings of product variant attributes in a product family, with the objective of minimizing performance loss or maximizing expected market profit [2]. Many scholars have contributed to the literature on product family design. A discussion and classification of this research can be found in the survey study by Jiao *et al.* [4].

Due to the similarity between products and services, services are naturally considered in product design problems in creating new customer value. Palsule-Desai *et al.* [14] created product varieties using add-on services while maintaining the identical functionality of the core product, and the service was viewed as an attribute of the product (e.g., delivery and repair services). In addition, the manufacturing and service industries jointly create a system to achieve sustainable growth and profitability. Song *et al.* [15] modularized a product-extension service based on a modified service blueprint and fuzzy graph, and demonstrated the method through an exemplified modular design of a compressor rotor service. Jiao *et al.* [16] outlined an approach connecting customer needs to high-value-added products and services to balance customer satisfaction and cost savings. However, these studies focused more on product design and development while considering services only from the competitiveness perspective.

B. SERVICE MODULARIZATION

Modularization theory originated from Simon's complex system decomposition theory [17] and involves deconstructing an object into its components and recombining them into customizable alternatives. It has been widely used in product design [18]; Sundbo [19] first extended the modularization theory to service production for service customization and personalization. In service modularization, it is important to consider the similarities and differences between a service and a product; a product is assembled using tangible components, while a service, due to its intangibility, is viewed as a "soft" activity. A service is produced and consumed

simultaneously and can also be a process [10]. Hence, customers need to interact with service providers and use the resources in a service process. In addition, customizations in services can be combinational, and a unique service is provided by combining a set of service modules (service contents and processes) [10].

1) SERVICE MODULES

A service module is a system of components that offers well-defined functionality via a precisely described interface and with which a modular service is composed, tailored, customized, and personalized [20]. In the service context, a modular platform can be created using elements or process modules of activities, and service modules can represent one or several service elements offering similar service characteristics [21]; a process module is defined as a standardized and indivisible process step, and it can either refer to information processing or physical activities [22].

In particular, identifying the service modules of a service system provides a basis for the effective composition of customer-specific configurations [20]; hence, by analyzing customer requirements, some scholars have explored how to decompose a service into modules based on service modularity. The existing product modularization methods in the service field are as follows. Simon [17] defined service architecture as the manner in which service system functionalities are decomposed into individual functional elements. Tay and Chen [23] mapped service activities to service modules using a K-means clustering algorithm; the activities of each service module were segregated into common and specific services using an unweighted pair group method with an arithmetic mean. Geum *et al.* [24] employed an interrelationship-based approach to identify the house of quality structure in quality function deployment for service modularization. Dörbecker *et al.* [25] decomposed the requirements of a complex healthcare system into service elements using the multiple domain matrix method for modularization of complex service systems.

2) SERVICE PROCESS MODULARITY

Tuunanen and Cassab [26] defined service process modularization as the systematic combination of the service encounter processes known to both customers and firms to generate new and customizable service packages with increased customer utility. Hence, standardized processes are ordered first; service resources and activities are then divided into sub-processes to achieve mass customization. Feitzinger and Lee [27] proposed a method for decomposing the manufacturing service process into separate sub-processes—based on process postponement—process sequence adjustment and process standardization, to increase service flexibility in mass customizations. Furthermore, some process modeling technologies, such as the Petri net and object-oriented modeling methodology, can help arrange process modularization activities in an orderly manner [28].

In addition, some scholars have explored the impact of service process modularization on service system performance. Tuunanen and Cassab [26] considered service process modularization as a way to leverage existing capabilities and investigated customer responses to modular reuse and modular variation in service encounter processes for new offerings. Carlborg and Kindström [29] explored modularization from a strategic perspective by distinguishing between different service types, presented key issues for service modularization, and contributed insights regarding the balance between efficiency and customization in service development and deployment. Silander *et al.* [30] explored an outpatient care unit in a university hospital to determine which enablers, constraints, and outcomes are related to modularization in advanced healthcare contexts.

3) SERVICE ORGANIZATION MODULARITY

For a service system, organizational modularity is the manner in which resources are used in a flexible manner to enable the use of the core capabilities of a service producer [21]. Meyera and Detoreb [31] proposed a platform-based approach to developing new services based on product family methods and platform designs, organized differently to facilitate the development and deployment of capabilities for an international reinsurer company. Meyer *et al.* [32] improved the platform for patient care services to explore different inpatient and outpatient facilities through examining management departments and other managerial personnel. De Blok *et al.* [33] provided insight into how care organizations can set up their service packages to simultaneously allow for customization and economies of scale in the provision of care services to independently living elderly. Avlonitis and Hsuan [34] put forward new insights for the organizational structure of service firms by empirically exploring, and theoretically advancing, the intersection of modularity and service design for two travel services. Chou *et al.* [35] explored different structures of organizational deployment and different process sequences, according to different service contents.

C. SERVICE FAMILY DESIGN

Ramesh *et al.* [36] stated that service families are consistent with product families, in that, “a service family is a set of services that share certain common aspects and have predicted variabilities”; they proposed a framework for a traceability-based knowledge management system to support the design, customization, and delivery of information products and e-service families. However, considering the differences between services and products is vital to developing service modularity. Moon *et al.* [5] stated that “a service family is a set of services based on a service platform that facilitates mass customization by promoting customer value and providing a variety of cost-effective services for different market segments.”. Additionally, “a service platform is a common basis that consists of processes, activities, objects, and/or features that are shared and remain constant from

service to service, within a given service family.” Other scholars explored potential module sharing and determined which service modules used in the platform provided the highest benefit, using a coalitional game [6], [9], Bayesian game [5], or data-mining techniques [7]. Lo and Chiu [8] presented a new approach that simultaneously integrates service design, service quality measurement, and service family planning for home service; concretely, service concepts are generated based on SERVQUAL metrics, and an affinity diagram is prepared to devise service families. Tay and Chen [23] proposed a modeling method for service families that formulates service activities into service function modules using a K-means clustering method and estimates the cost of those service families by combining activity-based cost estimations and functional modularizations.

D. ANALYSIS OF THE OPTIMIZATION METHODS

Gauss *et al.* [37] classified studies on product family design in the past 20 years and reviewed the methods, algorithms, and technologies used. From the optimization method perspective, the literature on product family design can be divided into two groups: the first group applies exact algorithms and the second applies heuristic algorithms [38].

Exact algorithms aim to find global optimal solutions for product family design problems. Some optimization problems of product family design can be formulated as 0-1 integer linear programming models, which have relatively simple mathematical forms. Several exact algorithms, such as branch & bound algorithm or cutting plane algorithm, have been proposed for solving 0-1 integer linear programming models [39]. These classical algorithms have been realized in many commercial software packages, such as ILOG CPLEX, GUROBI, and LINGO. Many scholars are still working on finding the exact solution. For example, Mayer and Steinhardt [40] considered a budget-constrained product line pricing problem and proposed a new branch & bound algorithm, which can solve larger scale problems compared to CPLEX. Scholars have also developed branch-and-price algorithm [41] and branch-and-cut algorithm [42]. Akçay *et al.* [43] established a dynamic pricing model for multiple perishable products and developed an exact algorithm based on dynamic programming. Recently, Bertsimas and Mišič [44] proposed an efficient exact algorithm based on benders decomposition.

Heuristic algorithms aim at obtaining near optimal solutions in product family design, which is regarded as the best available strategy for large-scale cases. Belloni *et al.* [45] classified several heuristic algorithms from three aspects: 1) methods operating in attribute space, such as, genetic algorithm [12] and simulated annealing [46]; 2) methods operating in product space, such as, greedy heuristic [47] and divide-and-conquer heuristic [48], [49]; 3) methods evaluating partially formed products, including beam search heuristic [50] and dynamic programming heuristic [51], [52]. Some scholars developed heuristic algorithms for new optimization problems. Tsafarakis *et al.* [53] developed a fuzzy, self-tuning

differential evolution algorithm to explore the best setting of parameters through statistical analysis. Wu and Chen [54] studied the influence of digital information product piracy on controlling a product version, established a nonlinear programming model through Lagrangean relaxation and subgradient methods, and combined heuristics to obtain the optimal solution for the original problem.

However, none of the existing research studies related to service family design considered multi-server queues and heterogeneous service processes, which are important characteristics of service products and systems.

III. PROBLEM DESCRIPTION AND MODELING

A. DEFINITIONS

Moon et al. [6] put forward the definition of a module-based service family, in which, service modules are categorized into functional and process levels. However, their process module only provides cost information for functional design, and it cannot be used for designing differentiated services. To increase the flexibility of customized services, this study defines a service family and service platform from multiple module dimensions, including process, activity, and component.

A service platform with a modular structure consists of a number of functional service modules, including common service modules—based on basic functions—that are required to be selected in service, as well as variant service modules that are optional and can be used for increasing the variety of services. A functional service module is represented by a replaceable process set (RPS), which is a set of heterogeneous service processes that implement the functional service module; each RPS consists of a number of different service activity modules in a chronological order. A service activity module is an executable and reusable service process step and is represented by a replaceable component set (RCS), which is a set of differentiated service components that meet various customer requirements.

A service family, derived from a service platform, contains a number of differentiated services. A service in a service family is represented by the selected service processes from the RPSs and the selected components from the RCSs of service activities. Fig. 2 shows an example of service family. In a service family design, the RPS and RCS can be used for creating differentiated service process modules and service activities, respectively.

Compared with a product family, one of the most important features of a service family is that customers may wait in line at some of the activity modules of a service process due to limited resources. The key activity module is defined as the most congested activity of a service, and each component of the corresponding RCS is a kind of server (or service desk with a queue).

For instance, Fig. 3 shows one common functional module (rehabilitation) and one variant functional module (physical examination). Rehabilitation is represented by an RPS with

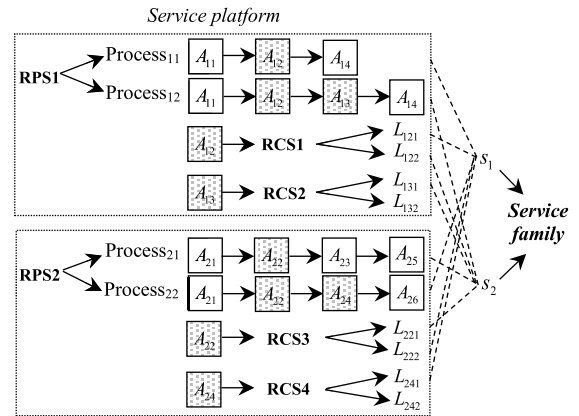


FIGURE 2. Configuration of a service family.

two elements: Process 11 and Process 12, while physical examination is represented by an RPS with three elements: Process 21, Process 22, and Process 23. Medical report is a reusable service activity module that appears in Processes 11, 12, 22, and 23. There are three RCSs—Appointment (Family doctor/Expert), Medical treatment (Therapy A/B), and Examinations for each treatment cycle (Partial/Full). Due to the combination effect of the RCSs, Processes 22 and 23 have two variants, and Processes 11 and 12 have four variants with different configuration of RCSs.

B. DESCRIPTION OF THE OPTIMIZATION PROBLEM

Take for instance, a service firm planning to develop a service family for N market segments, based on an established service platform that contains I functional service modules, including common and variant functional modules. Particularly, a dummy process is added into the RPS associated with the variant functional module, so that logically, all functional service modules can be regarded as common functional modules. The RPS associated with the i -th ($i = 1, 2, \dots, I$) functional service module has J_i activity modules and contains K_i replaceable service processes, in which the k -th ($k = 1, 2, \dots, K_i$) service process is composed of part or all of J_i activity modules. The selection relationships between processes and activity modules are described as a binary matrix h_{ikj} , where $h_{ikj} = 1$ if the k -th service process of the i -th functional service module selects the j -th ($j = 1, 2, \dots, J_i$) activity module. There are L_{ij} components in the RCS associated with the j -th activity module of the i -th functional service module.

It is assumed that the j_0 -th activity module in the i_0 -th RPS is the key activity module and each component in the RCS of the key activity module is a type of server. For instance, in a security check service, there are a number of common security doors for ordinary passengers and a few special security doors for VIP passengers. The service family contains S services, each of which corresponds to one type of server for the key activity module. The quality level of a server limits the module level of selected processes, θ_{ik}^P , and the module level of selected components, θ_{ijl}^L , which are ordered by the costs of

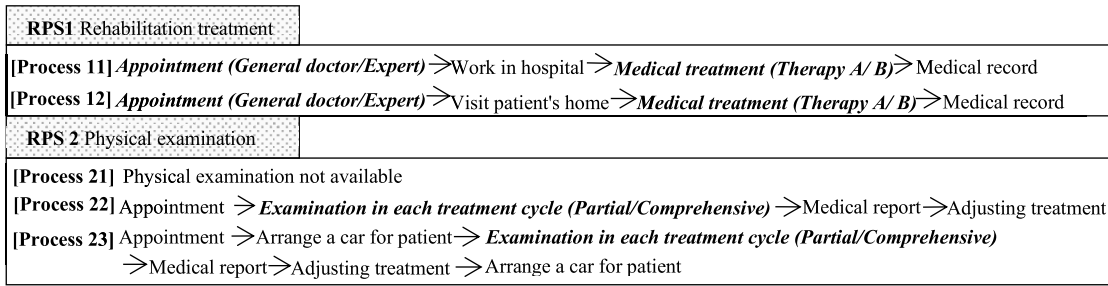


FIGURE 3. Example of the related definitions.

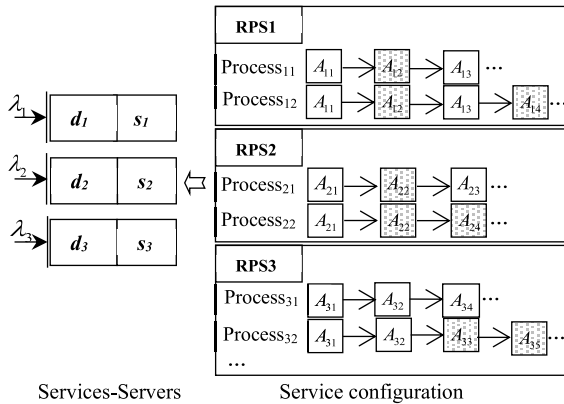


FIGURE 4. Description of the optimization problem.

all processes or components in an RPS and RCS, respectively. In terms of the customers' purchase behavior, a customer only chooses the service with the highest and non-negative utility surplus, and the waiting time at the server has different degrees of negative connotations for each market segment. Suppose that the average number of customers arriving at the d -th type of servers per unit time is λ_d and the average service duration rate is μ_d ; customers are allowed to wait when the service intensity $\rho \in (0, 1)$ and the remaining number of servers is equal to 0. The average waiting time for the q -th level of the d -th kind of servers is $t_{dq}(\lambda_d, \mu_d, Q_{dq}^S)$, which can be computed based on queuing theory.

As shown in Fig. 4, the optimization problem in this study involves achieving the maximum profit by reasonably configuring the services in a service family, while considering the average waiting time in the multi-server queue. We primarily address the following three issues: (1) How to select the optimal processes and components for both the RPSs and RCSs of the service family. (2) How to choose the number of servers required to reach a compromise between reducing costs and satisfying customer requirements. (3) How to set the appropriate price for each service in the service family.

C. MODEL FORMULATION

To facilitate the modeling, the following notations are used in this study:

1) UTILITY

The utility of a product variant can be considered as a linear function of the part-worth utilities of the configured components for the RCSs [12]. Similarly, in the service context, a customer's utility of a modular service is the sum of the part-worth utility, u_{nik}^P , of the configured processes for all RPSs and the part-worth utility, u_{nijl}^C , of the configured components for all RCSs in the processes (as shown in (1)). Particularly, u_{nik}^P represents the customer preference of the process when disregarding internal components.

$$U_{ns} = \sum_{i=1}^I \sum_{k=1}^{K_i} y_{sik} \left(u_{nik}^P + \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} x_{sijl} u_{nijl}^C \right) \quad (1)$$

In a service, queuing has a negative effect on the utility. Jain and Bala [55] found that all customers are averse to long average waiting times and derived a linear disutility function. Similarly, we presumed that the utility variation is proportional to the average waiting time, which is related to the number of servers and average arrival and service rates. Thus, the utility variation of a service caused by queuing is formulated as (2).

$$\Delta U_{ns} = \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq} \quad (2)$$

where Λ_{sd} describes the one-to-one mapping between the s -th service and the d -th kind of servers.

2) COST

In product family design, a linear-additive cost model is used to estimate the product cost, contained variable cost, and fixed cost [12]. Tay and Chen [23] assumed that a service's total cost is equal to the summation of the relevant modular cost and indirect service activity cost for a service family. In this research, the service's variable cost is divided into the variable costs of the process and activity modules; the former denotes the material and staff costs related to all selected activity modules in the configured process, while the latter is the component cost for the configured activity modules. The variable cost of the s -th service, C_s^{var} , is the sum of the configured processes' cost, C_{ik}^P , of all RPSs and the corresponding components' cost, C_{ijl}^C , of all RCSs in the

TABLE 1. Notation.

Notation	Description
Parameters	
I	The total number of RPSs.
N	The number of market segments.
N_n^{seg}	The size of the n -th market segment.
K_i	The number of processes for the i -th RPS.
J_i	The number of activity modules in the i -th RPS.
T_s	The service time of the s -th service.
V	The number of service price levels.
Q	The number of servers' quantity levels.
L_{ij}	The number of components of the j -th activity module in the i -th RPS, and if $L_{ij} > 1$, then $j \in \phi_i$.
λ_d	Average arrival rate for the d -th kind of servers.
μ_d	Average service rate is equal to $1/T_s$.
θ_{ik}^P	The module level of the k -th process for the i -th RPS.
θ_{ijl}^L	The module level of the l -th component of the j -th activity module in the i -th RPS.
U_{ns}	The utility of the s -th service for the n -th market segment.
u_{nik}^P	Part-worth utility of the k -th process in the i -th RPS for the n -th market segment.
u_{nijl}^L	Part-worth utility of the l -th component of the j -th activity module of the i -th RPS for n -th market segment.
Δu_n	Unit utility variation per unit time for n -th market segment.
h_{ikj}	If $h_{ikj} = 1$, the j -th activity module belongs to the k -th process for the i -th RPS, otherwise, it does not belong.
r_n	Utility surplus of competitive service in the market segment.
c_{ik}^P	The variable unit cost of the k -th process of the i -th RPS.
c_{ijl}^L	The variable unit cost of the l -th component of the j -th RCS of the i -th RPS.
c_d^S	The cost of the d -th kind of servers.
p_{sv}	The v -th discrete price level of the s -th service.
t_{dq}	Average waiting time of the q -th level of the d -th kind of server.
Q_{dq}^S	The q -th quantity level of the d -th kind of servers.
w_{ik}^P	If $w_{ik}^P = 1$, the cost of the k -th process for the i -th RPS is related to service time, then, $w_{ik}^P = 0$, it is not related.
w_{ijl}^L	If $w_{ijl}^L = 1$, the cost of the l -th component of the j -th activity module in the i -th RPS is related to service time, then, $w_{ijl}^L = 0$, it is not related.
$\sigma_{ss'}$	A binary matrix, if the server's quality level of the s -th service is higher than the s' -th service, $\sigma_{ss'} = 1$; otherwise, $\sigma_{ss'} = 0$.
Decision variables	
x_{sijl}	Binary variables, $x_{sijl} = 1$, if the l -th component of the j -th activity module is selected for the i -th RPS of the s -th service; otherwise, $x_{sijl} = 0$.
y_{sik}	Binary variables, $y_{sik} = 1$, if the k -th process of the i -th RPS is selected for s -th service; otherwise, $y_{sik} = 0$.
ω_{ns}	Binary variables, $\omega_{ns} = 1$, if the n -th market segment will purchase the s -th service; otherwise, $\omega_{ns} = 0$.
γ_{sv}	Binary variables, $\gamma_{sv} = 1$, if the s -th service sets the v -th price level; otherwise, $\gamma_{sv} = 0$.
η_{daq}	Binary variables, $\eta_{daq} = 1$, if the d -th kind of servers chooses the q -th quantity level of server; otherwise, $\eta_{daq} = 0$.

processes (as shown in (3)).

$$C_s^{var} = \sum_{i=1}^I \sum_{k=1}^{K_i} y_{sik} \left(C_{ik}^P + \sum_{j \in \phi_i} h_{ikj} x_{sijl} C_{ijl}^L \right) \quad (3)$$

However, the service cost may change with the service time due to the instantaneity of the service. After introducing two parameters, w_{ik}^P and w_{ijl}^L , the variable costs of the function and activity modules can be obtained by (4) and (5), respectively. Here, if the cost of the k -th process of the i -th RPS is related to the service time, then $w_{ik}^P = 1$; otherwise, the cost is constant. The variable unit costs, c_{ijl}^L and c_{ik}^P , are estimated by human experts.

$$C_{ik}^P = w_{ik}^P T_s c_{ik}^P + (1 - w_{ik}^P) c_{ik}^P \quad (4)$$

$$C_{ijl}^L = w_{ijl}^L T_s c_{ijl}^L + (1 - w_{ijl}^L) c_{ijl}^L \quad (5)$$

Servers are regarded as a part of the service infrastructure; however, the cost of servers is a variable cost, as shown in (6). Although a larger number of servers can decrease the average waiting time, they have higher procurement costs. It is necessary that firms make a compromise to reasonably choose the number of servers and maximize profit.

$$C_d^{var} = c_d \sum_{q=1}^Q \eta_{dq} Q_{dq}^S \quad (6)$$

3) OPTIMIZATION MODEL

Therefore, a firm's total profit is the sum of the net profit for all services minus the variable costs for servers and total fixed cost (7). The optimization model (Model I) for the service family design optimization can be established as [Model I], as shown at the bottom of the next page, where M is a larger position number.

With respect to the service configuration, constraints (8) and (9) ensure that only one element is selected for a service module, and $\phi_i = \{j | L_{ij} > 1\}$. For the conjoint analysis method, when an activity module is not included in a process, it is regarded as a dummy component. If $y_{sik} = 1$ and $h_{ikj} = 0$, then constraints (10–11) can ensure that $x_{sijl} = 1$; if $y_{sik} = 1$ and $h_{ikj} = 1$, then, $x_{sijl} = 0$. ϕ_i contains the indices of RCSs that are not shared in all processes of the i -th RPS. Servers work at a service desk for the components of the key activity module; therefore, constraint (12) confines $x_{si0j0d} = 1$. Constraints (15–17) describe a deterministic customer choice behavior—customers only purchase the service with the highest non-negative utility surplus when $\omega_{ns} = 1$; otherwise, a segment does not choose any services in the service family and prefers competitive services in the market. Each service should have a reasonable price level to maximize revenue (constraint (18)), and service price P_{sv} can be computed by $\sum_{v=1}^V \gamma_{sv} p_{sv}$. To reduce the average waiting time and control costs, a particular kind of servers only chooses the optimal number of servers (constraint (19)). Constraint (20) guarantees that a segment is limited to purchasing a service at most. Finally, constraint (21) confines decision variables as binary.

In addition, it is assumed that the module levels of processes and components are subject to the ordering of servers; in other words, the configuration of all RPSs or RCSs for a service should not be higher than the service configuration with a higher server quality level. For example, JingDong™

launched a gentle logistic service for valuable goods, which has a shorter delivery time and more considerate service than a basic logistic service. If the server quality level of the s -th service is not higher than that of the s' -th service, that is, $\sigma_{ss'} = 0$, then constraint (13) ensures that the module levels of the RPSs of the s -th service are subject to those of the s' -th service. Additionally, Constraint (14) ensures that the levels of RCSs involved in the service processes of the s -th service are smaller than those of the s' -th service. If $\sigma_{ss'} = 1$, then, constraints (13–14) will have no effect.

Further, each function module has K_i kinds of processes, and a process contains $\prod_{j \in R_1} h_{ikj} L_{ij}$ combinations of components; here, $R_1 = \{j \in \phi_i | i = i_0, j \neq j_0\}$. Since a function module only chooses a process, all possible services are combinations of different elements selected from I sets. Due to the upper bound of the module level, the number of possible services is not more than $\prod_{i \in I} \sum_{k \in K_i} \prod_{j \in R_1} h_{ikj} L_{ij}$. If $\sigma_{ss'} = 0, \forall s, s' = 1, \dots, S$, then, the number of possible services is maximum.

By combining the model linearization approach, we transformed the integer non-linear program model into a linear model (see Model II of Appendix A) so that the model can be solved by commercial optimization software packages (e.g., LINGO and ILOG CPLEX), and we proved that Model II has the same optimization result as that of Model I, as shown in Appendix A.

Theorem 1: Model II has the same optimization result as that of Model I.

The proof of Theorem 1 is given in Appendix A. Compared with Model I, the cost of the transformation lies in the intermediate variables and constraints appended to Model II.

Theorem 2: In Model I, if Δu_n is changed to $\Delta u'_n$ and the change of Δu_n does not incur a value change to ω_{ns} : ① if $t'_{dq} \in (\max(0, \beta_1), \beta_2)$, then, P_{sv} is unchanged ② if $t'_{dq} \in [0, \beta_1], \beta_1 > 0$, then P_{sv} increases; ③ if $t'_{dq} \in [\beta_2, \beta_3)$, then P_{sv} decreases; where $\beta_1 = \frac{1}{\Delta u'_n}(U'_{ns} - U_{ns}^ + \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* - \Delta p_s + \chi_n)$, $\beta_2 = \frac{1}{\Delta u'_n}(U'_{ns} - U_{ns}^* + \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* + \chi_n - \psi_n)$,*

[Model I]

max profit

$$= \sum_{s=1}^S \sum_{n=1}^N \omega_{ns} N_n^{seg} (\sum_{v=1}^V \gamma_{sv} p_{sv} - \sum_{i=1}^I \sum_{k=1}^{K_i} y_{sik} (C_{ik}^P + \sum_{j \in \phi_i} h_{ikj} x_{ijl} C_{ijl}^L)) - \sum_{d=1}^S c_d \sum_{q=1}^Q \eta_{dq} Q_{dq}^S - c^{fix} \tag{7}$$

$$\sum_{k=1}^{K_i} y_{sik} = 1, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; \tag{8}$$

$$\sum_{l=1}^{L_{ij}} x_{sijl} = 1, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; j \in \phi_i; \tag{9}$$

$$x_{sijl} \geq y_{sik} (1 - h_{ikj}), \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; \tag{10}$$

$$x_{sijl} \leq 1 - y_{sik} h_{ikj}, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; \tag{11}$$

$$\sum_{k=1}^{K_i} y_{sik} h_{i_0 k j_0} x_{s i_0 j_0 d} = \Lambda_{sd}, \quad s = 1, 2, \dots, S; d = 1, 2, \dots, S; \tag{12}$$

$$\sum_{k=1}^{K_i} \theta_{ik}^P y_{sik} \leq M \sigma_{ss'} + \sum_{k=1}^{K_i} \theta_{ik}^P y_{s'ik}, \quad s, s' = 1, 2, \dots, S, s \neq s'; i = 1, 2, \dots, I; \tag{13}$$

$$\sum_{k=1}^{K_i} \sum_{l=1}^{L_{ij}} y_{sik} h_{ikj} x_{sijl} \theta_{ijl}^L \leq M (1 - \sum_{k=1}^{K_i} \sum_{l=1}^{L_{ij}} y_{s'ik} h_{ikj} x_{s'ijl} + \sigma_{ss'} \cdot \sum_{k=1}^{K_i} \sum_{l=1}^{L_{ij}} y_{s'ik} h_{ikj} x_{s'ijl}) + \sum_{k=1}^{K_i} y_{s'ik} h_{ikj} x_{s'ijl} \theta_{ijl}^L, \quad s, s' = 1, 2, \dots, S, s \neq s'; i = 1, 2, \dots, I; j \in \phi_i; \tag{14}$$

$$\sum_{s=1}^S \omega_{ns} (U_{ns} - \Delta U_{ns} - \sum_{v=1}^V \gamma_{sv} p_{sv} - U_{ns'} + \Delta U_{ns'} + \sum_{v=1}^V \gamma_{s'v} p_{s'v}) \geq 0, \quad n = 1, \dots, N; s' = 1, \dots, S; s' \neq s; \tag{15}$$

$$\sum_{s=1}^S \omega_{ns} (U_{ns} - \Delta U_{ns} - \sum_{v=1}^V \gamma_{sv} p_{sv} - r_n) \geq 0, \quad n = 1, \dots, N; \tag{16}$$

$$\sum_{s=1}^S \omega_{ns'} (U_{ns'} - \Delta U_{ns'} - \sum_{v=1}^V \gamma_{s'v} p_{s'v} - r_n) \geq U_{ns'} - \Delta U_{ns'} - \sum_{v=1}^V \gamma_{s'v} p_{s'v} - r_n, \quad n = 1, \dots, N; s' = 1, \dots, S; s' \neq s; \tag{17}$$

$$\sum_{v=1}^V \gamma_{sv} = 1, \quad s = 1, \dots, S; \tag{18}$$

$$\sum_{q=1}^Q \eta_{dq} = 1, \quad d = 1, \dots, S; \tag{19}$$

$$\sum_{s=1}^S \omega_{ns} \leq 1, \quad n = 1, \dots, N; \tag{20}$$

$$y_{sik}, x_{sijl}, \gamma_{sv}, \eta_{dq}, \omega_{ns} \in \{0, 1\} \tag{21}$$

$\beta_3 = \frac{U'_{ns} - r_n}{\Delta u'_n}$, χ_n denotes the maximum surplus utility, and ψ_n is the maximum value of the surplus utility except χ_n , and $\psi_n \geq 0$.

The proof of Theorem 2 is given in Appendix B. In the real market, if the arrival rate increases within a certain period, then, the average waiting time t'_{dq} increases and $t'_{dq} > t^*_{dq}$; customers' anxious feelings may also cause Δu_n to increase. The service company needs to adjust their selling strategies to avoid losing customers to other competing services. According to Theorem 2, the company has three strategies for timely responses to situations: (1) if U'_{ns} increases, then β_1 and β_2 may increase so that t'_{dq} belongs to (β_1, β_2) , and service prices do not change; the company can provide coupons or extra services to eliminate or reduce customers' negative feelings. (2) If the number of servers increases, then t'_{dq} decreases, and service price may increase ($t'_{dq} \in (0, \beta_1), \beta_1 > 0$) or remain unchanged ($t'_{dq} \in (\beta_1, \beta_2)$). The company can adjust their prices according to measurable t'_{dq} or increase servers as a direct way to manage the situation. (3) If service prices P_{sv} decreases according to Theorem 2, while the number of servers remains unchanged, then the company can provide discount to customers to reduce their negative feelings. Hence, the service company can respond quickly to some practical situations to avoid losing customers.

IV. SOLVING ALGORITHMS

The original optimization model (Model I) is a nonlinear integer programming one, which is difficult to solve. However, the transformed optimization model (Model II) is equivalent to Model I, and Model II is a linear 0-1 programming model. Linear 0-1 programming model can be efficiently solved using a classic branch and bound algorithm for small-sized problems and the global optimality of the solution can be obtained for small-sized problems. However, for large-sized problems, no algorithm guarantees the global optimality for a linear 0-1 programming model. Therefore, we proposed using a commercial optimization software package, supporting the branch and bound algorithm (e.g., ILOG CPLEX), to solve Model II for small-sized problems, and we developed an improved genetic algorithm combined with a neighborhood search (meta-heuristic algorithm) to obtain the near-optimal solutions for large-sized problems. The purpose of adding the neighborhood search is to improve its local search ability. The main characteristics of the proposed algorithm are described as follows.

A. INITIAL POPULATION USING THE NEIGHBORHOOD SEARCH

The initial population contains N_p individuals, and each integer-coded chromosome consists of four sections: RPS and RCS configurations, quantity levels of servers, and price levels. The number of genes is a variable for the RCS configuration, because processes in a RPS include some RCSs satisfied by $h_{ikj} = 1$. To facilitate encoding, a chromosome is divided into two parts, κ_1 and κ_2 ; for instance, in Fig. 5,

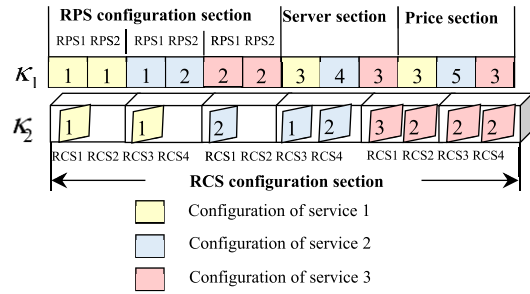


FIGURE 5. An example of the encoded chromosome.

the RPS contains different numbers of RCSs, according to the matrix h_{ikj} . If constraints (13–14) are satisfied, then configuration sections of the RPSs and RCSs for the s -th service are randomly generated based on the valid boundary of the s' -th service genes, and if constraints (13) and (14) are not satisfied for $s' = 1, 2, \dots, S, s' \neq s$, then they are generated in $[1, K_i]$ and $[1, L_{ij}]$, respectively. The last two sections that represent the quantity levels of the server and price levels are randomly generated in $[1, Q]$ and $[1, V]$, respectively. Fig. 5 shows an example of an encoded chromosome in which the genes of each service are represented by a specific color. This example describes a service family, including three services with different module levels of 2 RPSs and 4 RCSs. The process and component levels of Service 1 are not more than those of Services 2 or 3. Similarly, the process and component levels of Service 2 are not more than those of Service 3.

The initial population is composed of individuals with maximum fitness values, in the neighborhood of randomly generated individuals, to improve the performance of the population. The neighborhood of a gene is considered as an incremental or decremental change in the integer value within the gene boundary. The neighborhood of an individual is a set of neighborhoods of all genes. First, the neighborhood of each gene in κ_1 is created by increasing or decreasing one unit within the boundary of the genes; subsequently, it needs to generate or erase genes in κ_2 , according to the relationship between the process and component, and generated genes may be changed to create new neighborhoods within the boundary. In Fig. 6, the second gene in κ_1 only increases one unit because its boundary is equal to 2. Simultaneously, the second part of κ_2 contains RCS4, and it can choose 1 or 2 to create two individuals, as with the third gene. Finally, the best individual in the neighborhood will replace the initial individual.

B. SELF-ADAPTIVE GENETIC OPERATORS

To avoid losing the best individual, the optimal individual of the parent population is incorporated into the child population, and the individuals whose fitness values are higher than the average population fitness value are collected into an elite set. The elite individuals are randomly selected as parents for a crossover operation. Furthermore, self-adaptive crossover and mutation operators are adopted to increase the population diversity at the end of the iteration.

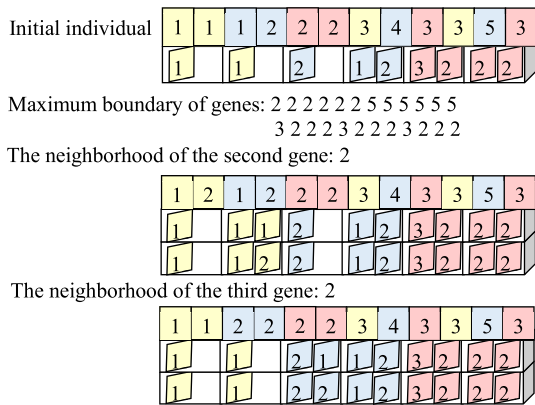


FIGURE 6. An example of the neighborhood of an individual.

The probabilities are computed based on the current population maturity, δ , which is equal to the ratio of the average fitness value of individuals, \bar{f}_{max} —which is greater than or equal to the population’s average fitness value—to the maximum of the population’s fitness value, $f_{max}(22)$ [39]. As the population maturity increases, the crossover probability, p_c , decreases (23) and the mutation probability, p_m , increases to maintain the diversity (24), where α and β are control parameters. The crossover operation is performed based on a 0–1 crossover mask that is generated based on the similarity of parents, π , and the mask of each gene is equal to 1 if a random number is larger than π . Individuals can be completely swapped under the lower similarity condition, while individuals can reduce invalid operations under the condition of higher similarity. In the mutation operation, some genes are randomly modified within the gene boundary.

$$\delta = \bar{f}_{max} / f_{max} \tag{22}$$

$$p_c = 1 / \exp(\alpha \cdot \delta) \tag{23}$$

$$p_m = 1 - \exp(-\beta \cdot \delta) \tag{24}$$

However, individuals easily become infeasible solutions after crossover and mutation operations, which may violate the module level or not satisfy the relationship between processes and components. To modify infeasible solutions, a chromosome-repairing method is applied to satisfy the constraints of the module level, that is, the gene of the infeasible solution is regenerated within the gene bound that is determined by the server quality level. Thereafter, the fitness values of the modified individuals are computed. In addition, the complexity of the chromosome-repairing algorithm is $O(S(IJ_c + N))$, where $J_c = \text{card}(\phi_i)$, $\phi_i = \{j | L_{ij} > 1\}$.

After the genetic operations, the neighborhood search is used to increase the depth of the algorithm. The algorithm searches the neighborhood of individuals with different fitness values, from child population to parent population. It aims at finding better solutions to improve the algorithm’s ability and control the computation time. The complexity of the neighborhood search is $O(S(IJ_c + N)(E_{Rcs} + E_{p\&s} + G_{Rcs}G_{Rps}))$, where E_{Rcs} is the number of the neighborhood individuals of κ_2 , and $E_{p\&s}$ is the number of the neighborhood

individuals of price and server sections of κ_1 . G_{Rps} is the number of individuals of the neighborhood Υ_1 in the RPS configuration section of κ_1 , G_{Rcs} is the number of the neighborhood individuals of modified κ_2 , according to each individual in Υ_1 . In the complexity expression, S , I , J_c , and N are related to the scale of the problem; E_{Rcs} , $E_{p\&s}$, G_{Rcs} , and G_{Rps} are indirectly related to the problem.

C. ELITE SELECTION STRATEGY COMBINED WITH ROULETTE SELECTION

The roulette selection method is commonly used as the selection operator, and it converts the fitness value into a probability. Individuals with larger fitness function values have a greater chance of becoming new individuals in the next generation; however, the method’s randomness may cause the loss of the best individual in the current generation. Therefore, to maintain the diversity of the population, this research adopts the roulette selection method combined with elite selection; the combined method is summarized in the following steps.

Step 1: Save the best individual in the current generation; the number of the new population $pop = 1$;

Step 2: If the average fitness value of the parent population is greater than that of the child population, individuals with different fitness values in the child and parent populations, whose fitness values are superior to the average fitness value of the parent population, are saved to the new population, and the unselected individuals are saved to temporary array; then, go to step 3. Otherwise, individuals with different fitness values in the child and parent populations, whose fitness values are superior to the average fitness value of the child population, are selected into the new population, and the unselected individuals are saved to temporary array; then go to step 3.

Step 3: If $pop < N_p$, use the roulette selection method to select the remaining number of new individuals from the temporary array; otherwise, end;

Due to the characteristics of service products, the combinatorial optimization problem requires several dimensions to decide the problem scale. Based on the above analysis, the complexity of the proposed algorithm depends on the complexity of the neighborhood search, and equals $O(S(IJ_c + N)(E_{Rcs} + E_{p\&s} + G_{Rcs}G_{Rps}))$.

V. CASE STUDY

To facilitate business pick-ups, sightseeing tours, and conference transports, a rental car company plans to launch a series of “half-day car rental” services based on four kinds of cars, classified by prices, and customers are allowed to wait for a certain period. As shown in Fig. 7, the processes or components in italics have different levels for creating customized services. RPS 1–3 belong to common functional modules, and a dummy process is added into RPS 4 so that, logically, all functional service modules can be regarded as common functional modules. For RPS1, customers can rent a car in three ways: taking and returning the car by themselves,

RPS 1 Service pattern module	<i>[Process 11] Customer takes and returns a car to a nearby shop</i> A ₁₁ Information registration→A ₁₂ Choose a kind of car (<i>L</i> _{121 series 1} , <i>L</i> _{122 series 2} , <i>L</i> _{123 series 3} , <i>L</i> _{124 series 4})→A ₁₃ Install Traveling Data Recorder(TDA)→A ₁₄ Inform and sign related papers
	<i>[Process 12] Worker delivers a car to the customer</i> A ₁₁ Information registration→A ₁₂ Choose a kind of car (<i>L</i> _{121 series 1} , <i>L</i> _{122 series 2} , <i>L</i> _{123 series 3} , <i>L</i> _{124 series 4})→A ₁₃ Install Traveling Data Recorder(TDA)→A ₁₄ Inform and sign related papers→A ₁₅ Deliver the car to the appointed place within a certain distance range (<i>L</i> _{152 3km} , <i>L</i> _{153 6km} , <i>L</i> _{154 10km})
	<i>[Process 13] Professional chauffeur provides services</i> A ₁₁ Information registration→A ₁₂ Choose a kind of car (<i>L</i> _{121 series 1} , <i>L</i> _{122 series 2} , <i>L</i> _{123 series 3} , <i>L</i> _{124 series 4})→A ₁₃ Install Traveling Data Recorder (TDA)→A ₁₄ Inform and sign related papers→A ₁₆ Assign a chauffeur to server
RPS 2 Fuel module	<i>[Process 21] Customers need to add fuel to initial reserves</i> A ₂₁ Refueling to the maximum level→A ₂₂ Check dashboard
	<i>[Process 22] Customers need to pay for fuel</i> A ₂₂ Check dashboard→A ₂₄ Settle account→A ₂₅ Workers add fuel
	<i>[Process 23] Complimentary gasoline</i> A ₂₂ Check dashboard→A ₂₃ Gasoline is free of charge within (<i>L</i> _{232 5L} , <i>L</i> _{233 10L} , <i>L</i> _{234 15L})→A ₂₄ Settle account
RPS 3 Insurance module	<i>[Process 31] Insurance A</i> A ₃₁ Information registration→A ₃₃ Collision damage waiver (<i>L</i> _{331 1,000\$} , <i>L</i> _{332 2,000\$} , <i>L</i> _{333 3,000\$})→A ₃₄ Passenger-liability insurance→A ₃₅ Loss damage waiver(<i>L</i> _{351 5,000\$} , <i>L</i> _{352 10,000\$} , <i>L</i> _{353 15,000\$})→A ₃₈ Inform and sign related papers
	<i>[Process 32] Insurance B</i> A ₃₁ Information registration→A ₃₂ Exonerate from partial compensation liability→A ₃₃ Collision damage waiver (<i>L</i> _{331 1000\$} , <i>L</i> _{332 2000\$} , <i>L</i> _{333 3000\$})→A ₃₄ Passenger-liability insurance→A ₃₅ Loss damage waiver (<i>L</i> _{351 5000\$} , <i>L</i> _{352 10000\$} , <i>L</i> _{353 15000\$})→A ₃₆ Third-party liability insurance (<i>L</i> _{362 20,000\$} , <i>L</i> _{363 40,000\$} , <i>L</i> _{364 60,000\$})→A ₃₈ Inform and sign related papers
	<i>[Process 33] Insurance C</i> A ₃₁ Information registration→A ₃₃ Collision damage waiver (<i>L</i> _{321 1,000\$} , <i>L</i> _{322 2,000\$} , <i>L</i> _{323 3,000\$})→A ₃₄ Passenger-liability insurance→A ₃₅ Loss damage waiver(<i>L</i> _{351 5,000\$} , <i>L</i> _{352 10,000\$} , <i>L</i> _{353 15,000\$})→A ₃₆ Third-party liability insurance(<i>L</i> _{362 20,000\$} , <i>L</i> _{363 40,000\$} , <i>L</i> _{364 60,000\$})→A ₃₇ others (<i>L</i> _{372 10,000\$} , <i>L</i> _{373 20,000\$} , <i>L</i> _{374 30,000\$})→A ₃₈ Inform and sign related papers
RPS 4 <i>Tools module</i>	<i>[Process 41] No other ancillary tools</i>
	<i>[Process 42] Provided other ancillary tools</i> A ₄₁ Install car navigation→A ₄₂ debugging

FIGURE 7. Module structure of a service family for car rentals.

TABLE 2. L9 orthogonal array and the obtained cluster centers for RPSs.

	RPS1	RPS2	RPS3	RPS4	<i>n</i> ₁	<i>n</i> ₂	<i>n</i> ₃	<i>n</i> ₄
1	3	1	2	2	0.1	7.5	7.2	56.1
2	2	2	2	1	1.0	23.6	24.2	20.2
3	1	1	1	1	4.3	3.1	2.1	0
4	1	3	2	1	2.1	15.3	17.6	10.4
5	3	3	3	1	0.2	10.4	9.4	65.1
6	3	2	1	1	0.5	8.3	8.9	59.4
7	2	3	1	2	0.7	17.3	18.1	16.8
8	2	1	3	1	1.4	19.3	20.4	17.6
9	1	2	3	2	1.7	13.3	14.8	8.3

having the car delivered by a worker, and using services provided by a professional chauffeur. Further, customers are allowed to pay for fuel fees through three patterns before returning the cars in RPS 2, and RPS 3 offers three insurance options. To satisfy diversified customer segments, different series of cars work as the server for the components of the key activity module A₁₂ and there is a need to decide the number of each kind of car. Finally, four services, involving four series of cars, are composed from the available processes and components to realize the maximum profit.

When combining RPSs with RCSs, modular services have a more flexible structure, so that part-worth utility can be

separately computed using a conjoint analysis. The RPSs of the case study include 3, 3, 3, and 2 processes, and, without choosing a variant functional module, is considered as a dummy process (*process*₄₁). Similarly, when an activity module is not included in a process, it is regarded as a dummy component; therefore, the RCSs of the case include 4, 4, 4, 3, 3, 4, and 4 components. Virtual combinations of RCSs and RPSs enable a separate conjoint analysis application. A number of respondents are invited to participate in a market survey to collect customer preferences for processes and components. The utility is measured as customer’s willingness to pay for services (in dollars). Then, respondents are grouped into four clusters using SPSSTM(IBM, version 22). The clustered, four market segments contain 100, 150, 150, and 50 thousand customers. The utilities of each segment are listed on the right columns in Tables 2 and 3.

Utilities are separately decomposed into part-worth utilities of processes and components by a least-squares linear regression. The results are shown in Tables 4 and 5. In addition, the variable unit cost is estimated by experts. The competitive services of the company are investigated, and the largest utility surpluses of the competitive services in the four market segments are 0.1, 0.3, 0.5, and 0.

For all segments, customers are allowed to queue for a certain period when unassigned cars are out of stock.

TABLE 3. L32 orthogonal array and the obtained cluster centers for RCSs.

	A ₁₂	A ₁₅	A ₂₃	A ₃₃	A ₃₅	A ₃₆	A ₃₇	n ₁	n ₂	n ₃	n ₄
1	4	2	3	1	1	4	3	3.5	12.0	14	54.5
2	4	2	2	1	2	3	1	5.7	12.3	13.5	70.4
3	4	4	1	1	2	3	2	6.4	15.3	16.7	47.7
4	1	2	3	3	3	4	2	20.3	18.4	20.3	6.0
5	2	3	4	3	1	3	1	15.6	52.1	39.5	24.4
6	1	4	4	2	3	4	1	27.4	17.3	18.3	7.8
7	2	3	1	2	2	4	3	17.5	49.4	38.5	19.3
8	4	1	4	3	1	2	2	6.7	9.3	10.3	49.5
9	3	4	2	2	1	2	3	11.3	35.4	45.5	40.3
10	1	3	3	1	2	2	4	21.6	18.3	17.2	6.4
11	3	4	3	3	2	1	1	13.2	38.8	47.5	42.4
12	2	2	4	1	1	1	3	18.3	47.6	36.5	22.4
13	4	3	2	3	3	1	3	2.2	11.3	10.1	58.6
14	2	1	2	3	2	4	4	19.3	38.2	27.2	18.2
15	2	4	2	1	3	2	2	16.1	46.2	36.5	20.3
16	3	2	4	2	2	1	2	11.5	39.5	52.4	36.4
17	2	2	1	1	3	2	1	20.4	56.3	51.6	20.1
18	3	1	3	1	3	3	3	14.3	36.2	40.6	35.5
19	4	1	1	2	3	1	4	7.4	8.3	7.6	44.5
20	3	3	4	1	3	3	4	10.3	38.4	50.3	40.4
21	3	2	1	3	1	2	4	12.4	48.4	56.3	37.4
22	1	1	1	1	1	1	1	40.3	18.1	17.4	3.3
23	1	1	4	1	2	2	3	32.2	25.3	23.1	4.5
24	1	2	2	2	1	3	4	23.3	20.1	22.3	5.6
25	4	3	3	2	1	2	1	4.8	11.2	12.4	64.4
26	2	1	3	2	1	3	2	18.9	40.2	32.3	17.3
27	4	4	4	1	1	4	4	3.0	10.3	9.4	50.2
28	2	4	3	1	1	1	4	17.2	44.1	33.6	23.4
29	1	4	1	3	1	3	3	24.6	22.3	23.4	4.0
30	3	3	1	1	1	4	2	13.4	46.5	59.5	38.3
31	1	3	2	1	1	1	2	30.2	24.3	16.3	5.0
32	3	1	2	1	1	4	1	15.6	43.4	41.3	39.2

TABLE 4. Part-worth utilities and variable unit cost (\$) of processes.

Function-Process	Part-worth utility					
	n ₁	n ₂	n ₃	n ₄	c _{ik} ^p	
RPS 1	Process _{S11}	1.67	-9.5	-9.4	-11.97	0.5
	Process _{S12}	0	0	0	0	4.0
	Process _{S13}	-0.77	-11.33	-12.4	42	3.0
	Process _{S21}	0.93	-4.37	-5.13	-6.2	0
RPS 2	Process _{S22}	0.07	0.73	0.93	-1.47	1.0
	Process _{S23}	0	0	0	0	3.0
	Process _{S31}	0.73	-4.77	-5.17	-4.93	0.3
RPS 3	Process _{S32}	-0.03	1.13	1.47	-1.43	0.4
	Process _{S33}	0	0	0	0	0.3
RPS 4	Process _{S41}	0	0	0	0	0
	Process _{S42}	-0.75	-0.63	-0.4	-1.72	0.5
	Based value	0.72	22.7	23.67	23.45	

Based on the parameters of the queue, 10 levels of server numbers are set for four series of cars (see Table 6). The average customer-arrival rates are 2.5, 3.5, 3.0, and 1.0 per hour, for the four series of cars, and a customer has to return the car within five hours. The average service rates of each series of cars are equal to 0.2, and the service intensity

TABLE 5. Part-worth utilities (\$) and variable unit cost (\$) of components.

RCSs-Component	Part-worth utility					
	n ₁	n ₂	n ₃	n ₄	c _{ijl} ^L	
A ₁₂	L ₁₂₁	22.53	9.26	8.04	-49.65	3.5
	L ₁₂₂	12.95	35.51	25.21	-34.3	6.5
	L ₁₂₃	7.79	29.57	37.42	-16.24	6.5
	L ₁₂₄	0	0	0	0	10.0
A ₁₅	L ₁₅₁	4.89	-4.06	-5.5	-5.6	0
	L ₁₅₂	-0.02	0.39	2.89	-0.5	2.0
	L ₁₅₃	0	0	0	0	4.0
	L ₁₅₄	0.45	-2.72	-1.61	-2.59	6.0
A ₂₃	L ₂₃₁	2.17	3.1	3.9	-2.63	0
	L ₂₃₂	-0.16	-1.07	-3.39	2.75	3.0
	L ₂₃₃	-1.4	-2.58	-2.74	1.79	6.0
	L ₂₃₄	0	0	0	0	10.0
A ₃₃	L ₃₃₁	2.49	1.06	0.52	0.04	0.2
	L ₃₃₂	0.97	-2.18	-0.66	-0.61	0.4
	L ₃₃₃	0	0	0	0	0.8
	L ₃₅₁	0.27	0.69	-0.14	-0.71	0
A ₃₅	L ₃₅₂	0	0	0	0	0.4
	L ₃₅₃	-1.12	-0.59	-0.1	-1.51	0.6
A ₃₆	L ₃₆₁	2.54	-0.44	-0.89	0.31	0
	L ₃₆₂	0.69	1.86	3.05	1.17	0.6
	L ₃₆₃	-0.11	0.17	1.26	1.47	0.9
	L ₃₆₄	0	0	0	0	1.2
A ₃₇	L ₃₇₁	2.39	1.25	1.23	4.11	0
	L ₃₇₂	-0.05	0.02	1.58	-2.33	0.4
	L ₃₇₃	0	0	0	0	0.6
	L ₃₇₄	-1.18	-1.68	-0.98	-1.63	0.8
	Based value	1.07	12.5	12.05	56.76	

TABLE 6. Average waiting time of service under different car parameters.

Q	Series 1		Series 2		Series 3		Series 4	
	Q _{1q} ^S	t _{1q} (min)	Q _{2q} ^S	t _{2q} (min)	Q _{3q} ^S	t _{3q} (min)	Q _{4q} ^S	t _{4q} (min)
1	14	118.2	19	128.0	16	219.0	6	176.3
2	15	48.2	20	55.6	17	78.1	7	48.6
3	16	22.6	21	28.1	18	36.1	8	16.7
4	17	11.2	22	15.1	19	18.3	9	6.1
5	18	5.6	23	8.3	20	9.6	10	2.2
6	19	2.8	24	4.6	21	5.1	11	0.8
7	20	1.4	25	2.5	22	2.7	12	0.2
8	21	0.7	26	1.4	23	1.4	13	0.1
9	22	0.3	27	0.8	24	0.7	14	0.0
10	23	0.2	28	0.4	25	0.4	15	0.0

$\rho = \lambda_d / \mu_d Q_{dq}^S$. Table 6 shows $t_{dq}(\lambda_d, \mu_d, Q_{dq}^S)$, the average waiting time for cars under the 10-number levels, based on the $M/M/Q_{dq}^S$ model of queuing theory. Only if the service intensity $0 < \rho < 1$ is satisfied, will an infinite queue not be formed.

Based on the data in this case, Model II can be established and solved with ILOG CPLEX; the results are shown in Table 7. The car rental company sells four services for “poor-rich” segments. Service 1 requires customers to take and return a car for Series 1 from a nearby shop, and they need

TABLE 7. Optimal solutions.

	Traditional approach	Proposed method
Total Profit (\$)	12.92*10 ⁶	13.76*10 ⁶
Optimal Servers	23/28/25/15	16/21/19/7
Service Price (\$)	41.9/79.9/85.9/121.9	41.9/79.9/85.9/120.9
s ₁ : RPS(RCS)	1(1,1)-1(1)-1(1,1,1)-1	1(1,1)-1(1)-1(1,1,1)-1
s ₂ : RPS(RCS)	2(2,2)-2(1)-2(1,1,2,1)-1	2(2,2)-2(1)-2(1,1,2,1)-1
s ₃ : RPS(RCS)	2(3,2)-2(1)-2(1,1,2,1)-1	2(3,2)-2(1)-2(1,1,2,1)-1
s ₄ : RPS(RCS)	3(4,1)-3(2)-2(1,1,2,1)-1	3(4,1)-3(2)-2(1,1,2,1)-1
Choice Behavior	s1/s2/s3/s4	s1/s2/s3/s4

to refuel before returning the car; the driving process is guaranteed by Insurance A, including a \$1,000 collision damage waiver, \$5,000 loss damage waiver, and passenger-liability insurance. Service 2 delivers a car for Series 2 within 3 km and offers Insurance B, including a \$1,000 collision damage waiver, \$5,000 loss damage waiver, \$20,000 third-party liability insurance, and passenger-liability insurance, and customers need to pay for the fuel. In contrast, Service 3 provides a car for Series 3. For Service 4, the company assigns a professional chauffeur to drive a car for Series 4 to the specified place; thus, the customer enjoys the same insurance service as with Services 2 and 3, and the customer only pays for any additional fuel over 5 L. The optimal number of cars, that is, the acceptable compromise between the average waiting time and cost factor, for Series 1, 2, 3, and 4 are 16, 21, 19, and 7, respectively. To maximize the profit, the prices of the four services are \$41.9, \$79.9, \$85.9, and \$120.9, respectively. The four segments purchase Services 1, 2, 3, and 4, respectively. Finally, the car rental company earns \$13.76 million.

Traditionally, a service company first designs the service configuration of a service family, and thereafter, determines the numbers of servers, according to the customer-satisfaction maximization criterion. However, the traditional approach may result in calculation deviation of the customer choice behavior in the real market, since the negative utility caused by waiting time is not considered in the service configuration process. In this research, the negative utility of service queue on customers is integrated into the service family design; thus, a good compromise between cost and customer satisfaction can be achieved by solving the mathematical model. The comparison results in Table 7 show that the optimal service configuration of the traditional approach is consistent with the proposed approach, but service price and numbers of servers are different from those of the proposed method. The total profit of the traditional approach is 12.92*10⁶, while that of the proposed approach is 13.76*10⁶. The results show that the proposed approach gains more profit for service firms.

To explore the relationship between cost and customer satisfaction, an experiment was performed, in which customer sensitivity for the first segment market was set as 1.0, 10.0, 20.0, and 30.0, respectively, and the unit cost of the first kind of servers gradually increased from 20 to 300 thousand in steps of 20 thousand. From Fig. 8 it can be observed that

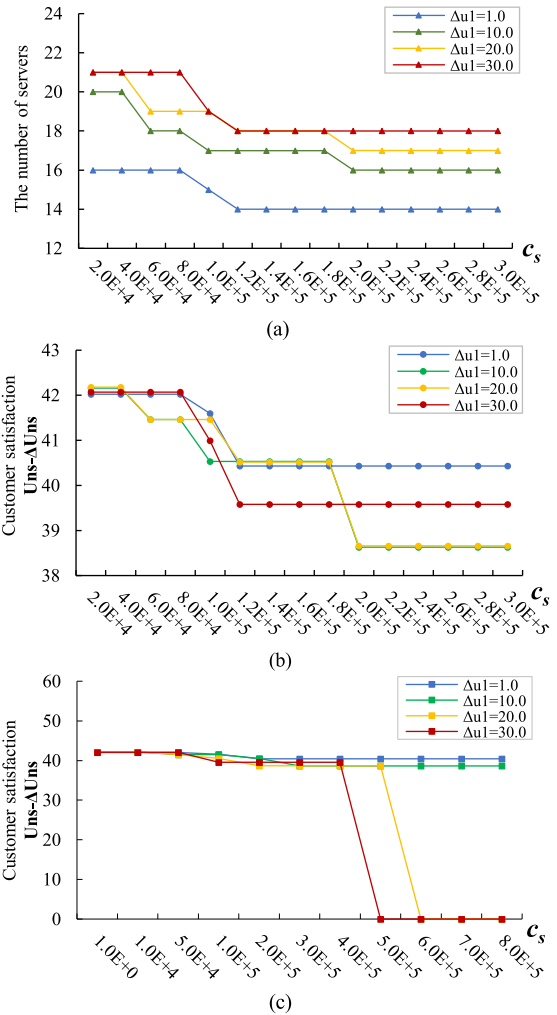


FIGURE 8. The relationship between cost and customer satisfactio.

1) the higher the unit cost of the server, the fewer the number of servers. Additionally, segments with higher sensitivity need more servers to reduce disutility (see Fig. 8.(a)). 2) In Fig. 8.(b), the higher the unit cost of the server, the lower the customer satisfaction. 3) As the unit cost of server gradually increases by larger steps, the market segments with higher sensitivity will forgo choosing a service earlier than the other market segments. In Fig. 8.(c), when $\Delta u_n = 30.0$ and the unit cost is larger than $4e + 5$, customer satisfaction equals to 0. The unit cost c_s is not sensitive to these parameters since c_s varies with relatively larger step, and it helps to control costs for the company while ensuring customer satisfaction.

To explore the influence of customer utility sensitivity on the optimization result, Δu_n of the n -th segment market varies from 0 to 5 in steps of 0.2, and the customer utility sensitivities of other $N - 1$ segments are fixed. As shown in Fig. 9(a), the more sensitive customers are regarding waiting time, the less profit the company will earn. As shown in Fig. 9(b) and (c), when customer utility sensitivity increases, the company can cater to customer needs for waiting time and maintain the stability of price by moderately increasing

TABLE 8. Effect of the average waiting time on the service price.

Δu	U	Q_{4q}^S	t	$U - \Delta U$	p	Flag	β_1	β_2	β_3
0.0	122.54	6	2.938	122.54	121.9	-	-	-	-
0.2	122.54	6	2.938	121.95	121.9	F	N	3.2	612.700
0.4	122.54	6	2.938	121.36	120.9	↓	N	0.1594	306.350
0.6	122.54	7	0.81	122.05	121.9	↑	1.059	2.725	204.233
0.8	122.54	7	0.81	121.89	120.9	↓	N	0.795	153.175
1.0	122.54	7	0.81	121.73	120.9	F	0.638	1.638	122.540
1.2	122.54	7	0.81	121.57	120.9	F	0.533	1.367	102.117
1.4	122.54	7	0.81	121.41	120.9	F	0.459	1.173	87.529
1.6	122.54	7	0.81	121.24	120.9	F	0.403	1.028	76.587
1.8	122.54	7	0.81	121.08	120.9	F	0.353	0.909	68.078
2.0	122.54	8	0.279	121.98	121.9	↑	0.319	0.819	61.270
2.2	122.54	8	0.279	121.93	121.9	F	N	0.29	55.700
2.4	122.54	7	0.81	120.60	119.9	↓	N	0.268	51.058
2.6	122.54	8	0.279	121.81	120.9	↑	0.632	1.017	47.131
2.8	122.54	9	0.101	122.26	121.9	↑	0.227	0.584	43.764
3.0	122.54	8	0.279	121.70	120.9	↓	0	0.214	40.840
3.2	122.54	7	0.81	119.95	119.9	↓	0.199	0.512	38.294
3.4	122.54	8	0.279	121.59	120.9	↑	0.483	0.777	36.041
3.6	122.54	8	0.279	121.54	120.9	F	N	0.455	34.039
3.8	122.54	9	0.101	122.16	121.9	↑	0.17	0.433	33.247
4.0	122.54	9	0.101	122.14	121.9	F	N	0.161	30.635
4.2	122.54	8	0.279	121.3682	120.9	↓	N	0.153	29.176
4.4	122.54	8	0.279	121.3124	120.9	F	0.146	0.373	27.850
4.6	122.54	8	0.279	121.2566	120.9	F	0.139	0.356	26.639
4.8	122.54	8	0.279	121.2008	120.9	F	0.144	0.342	25.529
5.0	122.54	8	0.279	121.145	120.9	F	0.128	0.328	24.508

* F means that the price is not changed; ‘↑’ denotes that price increases; ‘↓’ denotes that price decreases.

* N denotes $\beta_1 < 0$.

the number of servers. However, if the unit cost of servers is relatively high, it is impractical for the company to increase too many servers, and thus it can improve customer satisfaction by reducing prices. In Fig. 9(c), the price of the 4th service (s4) is adjusted with the fluctuations of the number of servers for increasing customer satisfaction.

It can be observed that the fluctuations of price in the fourth market segment in Fig. 9(b) can be used to explain Theorem 2. The relative parameters of the optimal solutions are listed on the left columns of Table 8. Further, $\beta_1, \beta_2, \beta_3$ are computed according to Theorem 2 and listed on the right columns of table 8. The price of each row is compared with previous rows, and arrows are used to reflect the result of price fluctuations. For example, when Δu_n equals 0.6, $t \in (0, \beta_1)$, where $\beta_1 = 1.059 > 0$, then the price will increase. When $\Delta u_n = 0.8$, $t \in (\beta_2, \beta_3)$ and $\beta_1 < 0$, then the price will decrease. When $\Delta u_n = 1.0$, $t \in (\beta_1, \beta_2)$, then the price remains unchanged. The experiment shows that Δu_n is relatively sensitive to optimization solution, implying that the company should cater to customers to avoid losing them. That is, if the company cannot provide more servers, it may consider providing coupons at peak time or adopting price deduction strategies.

In addition to customer sensitivity and the unit cost of servers, service time may also affect the number of servers. If service time is short, then the average service rate is high. In this experiment, service time is adjusted from 0 to 5 in steps of 0.5; the optimal numbers of servers for the four services are plotted in Fig. 10. Evidently, as service time increases, the total profit gradually decreases because more servers are required for each service. The customer satisfaction for each optimal solution is presented in Table 9. According to the variances of customer satisfactions, it can be observed that customer satisfaction remains relatively stable by sacrificing a small profit, where the price or the number of servers is adjusted.

MatlabTM was used to program the proposed meta-heuristic algorithm. The population size of the genetic algorithm was set as 50, and the results under 100 generations are presented in Fig. 11. The optimal solution was obtained at the 6th generation. Model II is also established based on the case data and the same optimal solution can be obtained by solving Model II with ILOG CPLEX.

To find better solutions with reasonable control parameters of the proposed meta-heuristic algorithm, an experiment was performed, in which, α varied from 0.02 to 2.82 in steps of 0.20, and β varied from 0.20 to 3.00 in steps of 0.20,

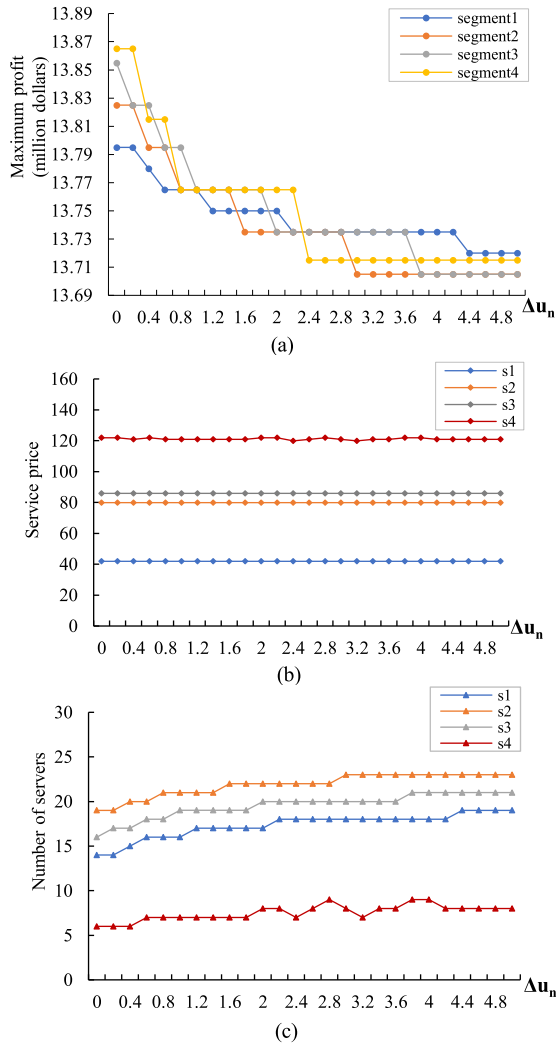


FIGURE 9. Impact of the value of Δu_n on the optimal solution.

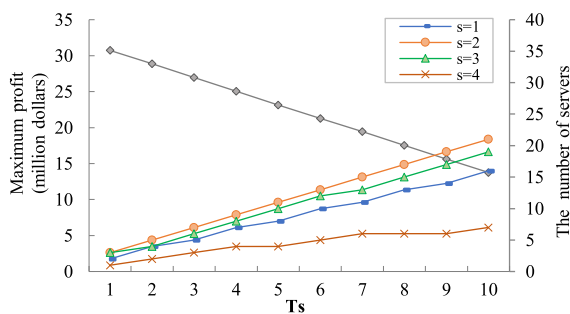


FIGURE 10. Impact of service time T_s on the optimal solution.

while the control parameters of the neighborhood search were set to the minimum value. The best parameter value combination for the optimization is identified as ($\alpha = 0.02$, $\beta = 1.80$). From the experiment, it was found that the control parameter β of the mutation operator is more sensitive in the optimization results than the control parameter α of crossover operator. However, the best parameter value combination may slightly change for cases with different scales; hence, more

TABLE 9. Impact of T_s on customer satisfaction.

T_s	$n1$	$n2$	$n3$	$n4$	Total
0.5	30.250	76.437	81.781	123.430	311.898
1.0	30.304	76.318	81.351	124.440	312.413
1.5	30.133	76.238	81.438	123.600	311.412
2.0	30.248	76.185	81.503	124.430	312.263
2.5	30.111	76.150	81.553	123.070	310.885
3.0	30.226	76.127	81.594	123.390	311.340
3.5	30.116	76.113	81.344	123.580	311.157
4.0	30.220	76.106	81.434	123.000	310.764
4.5	30.130	76.102	81.502	121.750	309.487
5.0	30.221	76.102	81.555	122.570	310.451
Mean	30.1959	76.1878	81.5055	123.326	311.207
Variance	0.005	0.013	0.016	0.649	0.771

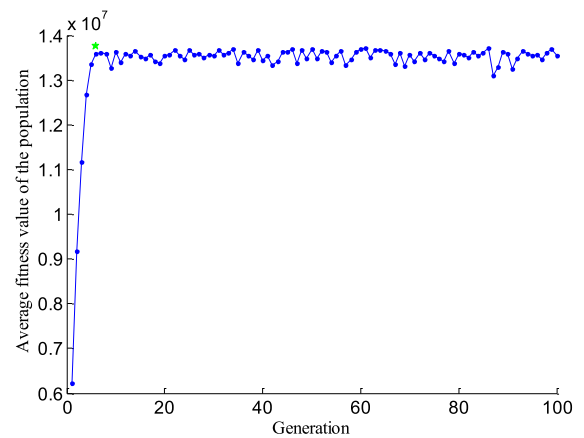


FIGURE 11. Average fitness values of the generation.

experiments of parameter combinations around the suggested one can be performed to further tune the parameters and improve the efficiency of the genetic algorithm.

To evaluate the performance of the proposed approach, a number of cases with different scales were randomly generated. As shown in Table 10, cases 1–8 can be solved by the proposed exact method (linearization). For the smaller cases (1–4), the optimal solutions solved by the proposed meta-heuristic algorithm are the exact solutions. The results indicate the correctness of the linearized model and effectiveness of the meta-heuristic algorithm. However, the solution space of the proposed model rapidly expanded as the numbers of RCSs and RPSs, or segments, increased. For cases with large scales (case 9–10), variables and constraints gradually increased; the cases cannot be globally solved within an acceptable computation time, and the meta-heuristic algorithm can obtain near-optimal solutions for service companies. The computation time of the exact algorithm depends on many factors, including the variables, constraints, and characteristics of the linear model. For cases with small scales (e.g., cases 1–3), the solving time of the linear model is small, while the meta-heuristic algorithm requires more time to prepare the population initialization and perform iterations.

TABLE 10. Computation results for different cases.

Case Number	Case parameter <i>N-S-I-K-J_c-L</i>	The traditional approach		The proposed approach					
		Profit (*10 ⁵)	Time	Exact method (linearization)			Meta-heuristic algorithm		
				Variables	Constraints	Profit (*10 ⁵)	Time	Profit (*10 ⁵)	Time
1	4-4-4-3-9-4	27.675	0.00h	28672	9705	35.250	0.00h	35.250	0.26h
2	4-4-5-3-11-4	45.022	0.02h	42128	14197	54.962	0.02h	54.962	0.20h
3	4-4-6-4-12-4	32.875	0.07h	59168	19837	33.635	0.03h	33.635	0.20h
4	4-4-6-5-15-4	14.728	0.2h	88080	29497	15.078	0.23h	15.078	0.59h
5	4-4-5-5-16-4	18.204	1.32h	91960	30773	19.870	1.90h	19.772	0.42h
6	5-5-5-3-14-5	11.382	1.67h	113610	38101	16.350	1.72h	16.284	0.71h
7	5-5-5-4-12-5	16.490	0.70h	117245	39291	16.780	3.00h	16.770	0.70h
8	5-5-5-4-12-5	14.115	3.36h	129810	43476	14.620	4.80h	14.540	0.79h
9	5-5-5-4-15-5	20.385	0.72h	157865	52851	N/A	N/A	21.537	0.44h
10	5-5-5-5-16-5	13.430	0.91h	176970	59231	N/A	N/A	14.090	0.77h

* *K* represents the maximum value of *K_i*; *L* represents the maximum value of *L_{ij}*.

It can be observed from Table 10 that—for each case in the table—the solution obtained by the proposed approach is better than that obtained by the traditional approach. Therefore, the effectiveness of the proposed approach, including exact method (linearization) and meta-heuristic algorithm, can be verified by the numerical experiments.

VI. CONCLUSION

Since various servers are usually involved in a service family, a service company needs to find a compromise between reducing the average waiting time and controlling the service cost. In this study, formulations for the customer utility and service cost are constructed from the service process and service component dimensions; moreover, service time is introduced to measure the cost from the instantaneity of service. Based on the above considerations, a design optimization model of a service family is established based on a modular service structure. The model maximizes the company’s profit by selecting the optimal process and component configurations and price level, along with the selection of the compromised number of servers to control the average waiting time and service cost. It further determines which services should be provided for each market segment, according to the deterministic customer-purchase choice rule.

To obtain the optimal solution using commercial optimization software packages (e.g., ILOG CPLEX), an equivalent linearized model, based on the original model, is derived. To obtain the near-optimal solutions for the optimization problem with a large solution space, an improved genetic algorithm combined with a neighborhood search is used.

To evaluate effectiveness of the proposed approach, a car-rental service family was introduced to demonstrate the proposed models and algorithms, and sensitivity analysis was performed. A number of cases with different sizes were randomly generated and relevant numeric experiments performed. The results show that the solution obtained by the proposed approach has a higher profit compared with the traditional approach for each of the cases; hence the proposed approach outperforms the traditional approach.

One of the limitations of this research is that multiple series of servers used to perform different functions in a service system are not considered in the model. Furthermore, this study only considers a single-stage queue. Future research may extend the model to fit multi-stage service queues as there may be multiple queues in variant functional modules; in addition, other service characteristics may be introduced into the model.

APPENDIX A

Proof of Theorem 1: Model II is equivalent to Model I.

In Model I, the average waiting time of cars *t_{dq}* is known; all nonlinear parts of the objective function and constraints result from multiplying two or more binary decision variables, and the product of two binary variables is regarded as a binary intermediate variable. Similarly, relative constraints are introduced to realize the replacement—for instance, $\omega_{ns}\gamma_{sv}$ is replaced by Z_{nsv} , and constraints (39–41) are removed in Model II. Because *M* is regarded as a large positive number, constraints (39–41) can ensure that $Z_{nsv} = 1$ if $\omega_{ns} = 1$ and $\gamma_{sv} = 1$; otherwise, $Z_{nsv} = 0$ if $\omega_{ns} = 0$ or $\gamma_{sv} = 0$. In fact, it is equivalent to $Z_{nsv} = \omega_{ns}\gamma_{sv} \cdot Z_{nss'v'}$, G_{nsik} , $G'_{nss'ik}$, O_{sikjl} , and F_{nsdq} are similar replacements. In addition, the product of three or more decision variables can be decomposed into an intermediate variable and a decision variable. Similarly, we introduce the variables $D_{nsikjl} = \omega_{ns}O_{sikjl}$ and $D'_{nss'ikjl} = \omega_{ns}O'_{s'ikjl}$ into the constraints. Note that linear constraints (8–11) and (18–20) need not be processed, and they can move directly into Model II; other parts of Model I are reformulated in the following Model II, as shown at the bottom of the next page.

APPENDIX B

Proof of Theorem 2: Suppose that the sensitivity of the customer to time is Δu , and U_{ns} is determined by the service configuration. Hence, function $Y = U_{ns} - r_n - \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}$ is considered based on constraint (16), and the optimal average waiting time is t^ , as shown in Fig. 12. Because the service price is a discrete*

set with unit step Δp_s , if the service utility with increment α_2 is not greater than $p_s^* + \Delta p_s$, the service price remains unchanged. Then, $\alpha_2 \in [0, \Delta p_s - \chi_n]$, according to (66) and (67); specifically, $\Delta p_s > \chi_n$, otherwise, $\alpha_2 = 0$; and χ_n is equal to the maximum surplus utility. By contrast, a service utility with decrement α_1 still corresponds to the maximum

value $\psi_n (\psi_n \geq 0)$ of the surplus utility in the case of $\omega_{ns} = 0$ (see (64)) based on constraint (15); then, $\alpha_1 \in [0, \chi_n - \psi_n]$.

$$U_{ns}^* - r_n - \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* - \alpha_1 - p_s^* > \psi_n \tag{64}$$

[Model II]

max profit

$$= \sum_{n=1}^N N_n^{seg} \left(\sum_{s=1}^S \sum_{v=1}^V Z_{nsv} p_{sv} - \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} G_{nsik} C_{ik}^P - \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} h_{ikj} D_{nsikjl} C_{ijl}^L \right) - \sum_{d=1}^S c_d \sum_{q=1}^Q \eta_{dq} Q_{dq} - c^{fix} \tag{25}$$

$$\sum_{k=1}^{K_i} y_{sik} = 1, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; \tag{26}$$

$$\sum_{l=1}^{L_{ij}} x_{sijl} = 1, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; j \in \phi_i; \tag{27}$$

$$x_{sijl} \leq 1 - y_{sik} h_{ikj}, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; \tag{28}$$

$$x_{sijl} \geq y_{sik} (1 - h_{ikj}), \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; \tag{29}$$

$$\sum_{v=1}^V \gamma_{sv} = 1, \quad s = 1, 2, \dots, S; \tag{30}$$

$$\sum_{q=1}^Q \eta_{dq} = 1, \quad d = 1, 2, \dots, L_{i0j0}; \tag{31}$$

$$\sum_{s=1}^S \omega_{ns} \leq 1, \quad n = 1, 2, \dots, N; \tag{32}$$

$$o_{sikjl} \leq x_{sijl}, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{33}$$

$$o_{sikjl} \leq y_{sik}, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{34}$$

$$o_{sikjl} \geq 1 + M (y_{sik} + x_{sijl} - 2), \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{35}$$

$$\sum_{k=1}^{K_i} o_{sio_kj0d} h_{io_kj0} = \Lambda_{sd}, \quad s = 1, 2, \dots, S; d = 1, 2, \dots, S; \tag{36}$$

$$\sum_{k=1}^{K_i} \theta_{ik}^P y_{sik} \leq M \sigma_{ss'} + \sum_{k=1}^{K_i} \theta_{ik}^P y_{s'ik}, \quad s, s' = 1, 2, \dots, S, s \neq s'; i = 1, 2, \dots, I; \tag{37}$$

$$\sum_{k=1}^{K_i} \sum_{l=1}^{L_{ij}} o_{sikjl} h_{ikj} \theta_{ijl}^L \leq M \left(1 - \sum_{k=1}^{K_i} \sum_{l=1}^{L_{ij}} o_{s'ikjl} h_{ikj} + \sigma_{ss'} \cdot \sum_{k=1}^{K_i} \sum_{l=1}^{L_{ij}} o_{s'ikjl} h_{ikj} \right) + \sum_{k=1}^{K_i} o_{s'ikjl} h_{ikj} \theta_{ijl}^L, \tag{38}$$

$$Z_{nsv} \leq \omega_{ns}, \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; v = 1, 2, \dots, V; \tag{39}$$

$$Z_{nsv} \leq \gamma_{sv}, \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; v = 1, 2, \dots, V; \tag{40}$$

$$Z_{nsv} \geq 1 + M (\omega_{ns} + \gamma_{sv} - 2), \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; v = 1, 2, \dots, V; \tag{41}$$

$$Z'_{nss'v} \leq \omega_{ns}, \quad n = 1, 2, \dots, N; s, s' = 1, 2, \dots, S; v = 1, 2, \dots, V; \tag{42}$$

$$Z'_{nss'v} \leq \gamma_{s'v}, \quad n = 1, 2, \dots, N; s, s' = 1, 2, \dots, S; v = 1, 2, \dots, V; \tag{43}$$

$$Z'_{nss'v} \geq 1 + M (\omega_{ns} + \gamma_{s'v} - 2), \quad n = 1, 2, \dots, N; s, s' = 1, 2, \dots, S; v = 1, 2, \dots, V; \tag{44}$$

$$G_{nsik} \leq \omega_{ns}, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; \tag{45}$$

$$G_{nsik} \leq y_{sik}, \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; \tag{46}$$

$$G_{nsik} \geq 1 + M (\omega_{ns} + y_{sik} - 2), \quad s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; \tag{47}$$

$$G'_{nss'ik} \leq \omega_{ns}, \quad s, s' = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; \tag{48}$$

$$G'_{nss'ik} \leq y_{s'ik}, \quad s, s' = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; \tag{49}$$

$$G'_{nss'ik} \geq 1 + M (\omega_{ns} + y_{s'ik} - 2), \quad s, s' = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; \tag{50}$$

$$D_{nsikjl} \leq o_{sikjl}, \quad n = 1, 2, \dots, N, s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{51}$$

$$D_{nsikjl} \leq x_{sijl}, \quad n = 1, 2, \dots, N, s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{52}$$

$$D_{nsikjl} \geq 1 + M (\omega_{ns} + o_{sikjl} - 2), \quad n = 1, 2, \dots, N, s = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{53}$$

$$D'_{nss'ikjl} \leq o_{s'ikjl}, \quad n = 1, 2, \dots, N, s, s' = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{54}$$

$$D'_{nss'ikjl} \leq \omega_{ns}, \quad n = 1, 2, \dots, N, s, s' = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{55}$$

$$D'_{nss'ikjl} \geq 1 + M (\omega_{ns} + o_{s'ikjl} - 2), \quad n = 1, 2, \dots, N, s, s' = 1, 2, \dots, S; i = 1, 2, \dots, I; k = 1, 2, \dots, K_i; j \in \phi_i; l = 1, 2, \dots, L_{ij}; \tag{56}$$

$$F_{nsdq} \leq \eta_{dq}, \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; d = 1, 2, \dots, S; q = 1, 2, \dots, Q; \tag{57}$$

$$F_{nsdq} \leq \omega_{ns}, \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; d = 1, 2, \dots, S; q = 1, 2, \dots, Q; \tag{58}$$

$$F_{nsdq} \geq 1 + M (\omega_{ns} + \eta_{dq} - 2), \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; d = 1, 2, \dots, S; q = 1, 2, \dots, Q; \tag{59}$$

$$\sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} G_{nsik} u_{nik}^p + \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} D_{nsikjl} u_{nijl}^L - \Delta u_n \sum_{s=1}^S \sum_{d=1}^S \sum_{q=1}^Q F_{nsdq} \Lambda_{sd} t_{dq} - \sum_{s=1}^S \sum_{v=1}^V Z_{nsv} p_{sv} - \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} u_{nik}^p G'_{nss'ik} - \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} D'_{nss'ijkl} u_{nijl}^L + \Delta u_n \sum_{s=1}^S \sum_{d=1}^S \sum_{q=1}^Q F_{nsdq} \Lambda_{s'd} t_{dq} + \sum_{s=1}^S \sum_{v=1}^V Z'_{nss'v} p_{s'v} \geq 0, \quad n = 1, 2, \dots, N; s = 1, 2, \dots, S; s' \neq s; \quad (60)$$

$$\sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} G_{nsik} u_{nik}^p + \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} D_{nsikjl} u_{nijl}^L - \Delta u_n \sum_{s=1}^S \sum_{d=1}^S \sum_{q=1}^Q F_{nsdq} \Lambda_{sd} t_{dq} - \sum_{s=1}^S \sum_{v=1}^V Z_{nsv} p_{sv} - \sum_{s=1}^S \omega_{ns} r_n \geq 0, \quad n = 1, 2, \dots, N; \quad (61)$$

$$\sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} G'_{nss'ik} u_{nik}^p + \sum_{s=1}^S \sum_{i=1}^I \sum_{k=1}^{K_i} \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} D'_{nss'ijkl} u_{nijl}^L - \Delta u_n \sum_{s=1}^S \sum_{d=1}^S \sum_{q=1}^Q F_{nsdq} \Lambda_{s'd} t_{dq} - \sum_{s=1}^S \sum_{v=1}^V Z_{nss'v} p_{s'v} - \sum_{s=1}^S \omega_{ns} r_n \geq y_{s'ik} u_{nik} + \sum_{i=1}^I \sum_{k=1}^{K_i} \sum_{j \in \phi_i} \sum_{l=1}^{L_{ij}} o_{s'ijkl} u_{nijl} - \Delta u_n \sum_{d=1}^S \sum_{q=1}^Q \eta_{dq} \Lambda_{s'd} t_{dq} - \sum_{v=1}^V \gamma_{s'v} p_{s'v} - r_n, \quad n = 1, 2, \dots, N; s' = 1, 2, \dots, S, s' \neq s; \quad (62)$$

$$y_{sik}, x_{sijl}, \gamma_{sv}, \eta_{dq}, \omega_{ns}, Z_{nsv}, Z_{nss'v}, o_{sijkl}, G_{nsik}, G_{nss'ik}, D_{nsikjl}, D_{nss'ijkl}, F_{nsdq} \in \{0, 1\} \quad (63)$$

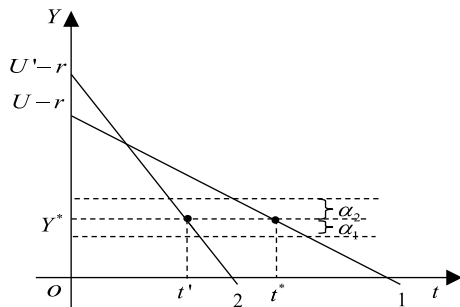


FIGURE 12. Relationship between service utility and average waiting time.

$$\alpha_1 \geq 0 \quad (65)$$

$$U_{ns}^* - r_n - \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* + \alpha_2 < p_s^* + \Delta p_s \quad (66)$$

$$\alpha_2 \geq 0 \quad (67)$$

When the customer sensitivity is adjusted to $\Delta u'$, the service configuration is modified as it changes, and $Y' = U'_{ns} - r_n - \Delta u'_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t'_{dq}$. It is reasonable that $Y' \in (Y^* - \alpha_1, Y^* + \alpha_2)$ does not break constraint (15) and does not affect the service price. According to (68) and (69), if the average waiting time t'_{dq} , related to different quantities of servers belongs to $(\max(0, \beta_1), \beta_2)$, it has no effect on the service price; otherwise, $t'_{dq} \in [0, \beta_1]$, $\beta_1 > 0$, and the price goes up to achieve the maximum profit; if $t'_{dq} \in [\beta_2, \beta_3)$, the price decreases to meet the constraints, and β_3 is computed according to $Y' = 0$.

$$U'_{ns} - r_n - \Delta u'_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t'_{dq} > U_{ns}^* - r_n - \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* - \chi_n + \psi_n \quad (68)$$

$$U'_{ns} - r_n - \Delta u'_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t'_{dq} < U_{ns}^* - r_n - \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* + \Delta p_s - \chi_n \quad (69)$$

where $\beta_1 = \frac{1}{\Delta u'_n} (U'_{ns} - U_{ns}^* + \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* - \Delta p_s + \chi_n)$, $\beta_2 = \frac{1}{\Delta u'_n} (U'_{ns} - U_{ns}^* + \Delta u_n \sum_{d=1}^S \Lambda_{sd} \sum_{q=1}^Q \eta_{dq} t_{dq}^* + \chi_n - \psi_n)$, and $\beta_3 = \frac{U'_{ns} - r_n}{\Delta u'_n}$.

REFERENCES

- [1] Y. Xing, L. Li, Z. Bi, M. Wilamowska-Korsak, and L. Zhang, "Operations research (OR) in service industries: A comprehensive review," *Syst. Res. Behav. Sci.*, vol. 30, no. 3, pp. 300–353, May 2013.
- [2] T. W. Simpson, Z. Siddique, and R. J. Jiao, *Product Platform and Product Family Design: Methods and Applications*. New York, NY, USA: Springer, 2006.
- [3] F. Erens and K. Verhulst, "Architectures for product families," *Comput. Ind.*, vol. 33, nos. 2–3, pp. 165–178, Sep. 1997.
- [4] J. Jiao, T. W. Simpson, and Z. Siddique, "Product family design and platform-based product development: A state-of-the-art review," *J. Intell. Manuf.*, vol. 18, no. 1, pp. 5–29, Jul. 2007.
- [5] S. K. Moon, J. Sim, J. Shu, and T. W. Simpson, "Strategic module sharing for customized service family design using a Bayesian game," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat.*, Aug. 2007, pp. 1–6.
- [6] S. K. Moon, J. Shu, T. W. Simpson, and S. R. T. Kumara, "A module-based service model for mass customization: Service family design," *IIE Trans.*, vol. 43, no. 3, pp. 153–163, Dec. 2010.
- [7] S. K. Moon, T. W. Simpson, J. Shu, and S. R. Kumara, "A method for platform identification to support service family design," *Int. J. Services Oper. Informat.*, vol. 3, pp. 294–317, 2008.
- [8] H.-P. Lo and M.-C. Chiu, "A service family design and business development methodology with a home service case study," *J. Ind. Prod. Eng.*, vol. 31, no. 7, pp. 458–470, Oct. 2014.
- [9] A. Aggarwal, F. T. S. Chan, and M. K. Tiwari, "Development of a module based service family design for mass customization of airline sector using the coalition game," *Comput. Ind. Eng.*, vol. 66, no. 4, pp. 827–833, Dec. 2013.
- [10] C. A. Voss and J. Hsuan, "Service architecture and modularity," *Decis. Sci.*, vol. 40, no. 3, pp. 541–569, Aug. 2009.
- [11] H. Tanriverdi, P. Konana, and L. Ge, "The choice of sourcing mechanisms for business processes," *Inf. Syst. Res.*, vol. 18, no. 3, pp. 280–299, Sep. 2007.

- [12] X. G. Luo, C. K. Kwong, J. F. Tang, S. F. Deng, and J. Gong, "Integrating supplier selection in optimal product family design," *Int. J. Prod. Res.*, vol. 49, no. 14, pp. 4195–4222, Jul. 2011.
- [13] R. Sanchez and J. T. Mahoney, "Modularity, flexibility, and knowledge management in product and organization design," *Strategic Manage. J.*, vol. 17, no. S2, pp. 63–76, Dec. 1996.
- [14] O. D. Palsule-Desai, D. Tirupati, and J. Shah, "Product line design and positioning using add-on services," *Int. J. Prod. Econ.*, vol. 163, pp. 16–33, May 2015.
- [15] W. Song, Z. Wu, X. Li, and Z. Xu, "Modularizing product extension services: An approach based on modified service blueprint and fuzzy graph," *Comput. Ind. Eng.*, vol. 85, pp. 186–195, Jul. 2015.
- [16] J. Jiao, Q. Ma, and M. M. Tseng, "Towards high value-added products and services: Mass customization and beyond," *Technovation*, vol. 23, no. 10, pp. 809–821, Oct. 2003.
- [17] H. A. Simon, "The architecture of complexity," in *Facets of Systems Science*, vol. 7. New York, NY, USA: Springer, 1991, pp. 457–476.
- [18] C. de Blok, B. Meijboom, K. Luijckx, and J. Schols, "The human dimension of modular care provision: Opportunities for personalization and customization," *Int. J. Prod. Econ.*, vol. 142, no. 1, pp. 16–26, Mar. 2013.
- [19] J. Sundbo, "Modulization of service production and a thesis of convergence between service and manufacturing organizations," *Scandin. J. Manage.*, vol. 10, no. 3, pp. 245–266, Sep. 1994.
- [20] T. Tuunanen, A. Bask, and H. Merisalo-Rantanen, "Typology for modular service design: Review of literature," *Int. J. Service Sci., Manage., Eng., Technol.*, vol. 3, no. 3, pp. 99–112, Jul. 2012.
- [21] S. Pekkarinen and P. Ulkuniemi, "Modularity in developing business services by platform approach," *Int. J. Logistics Manage.*, vol. 19, no. 1, pp. 84–103, May 2008.
- [22] M. R. Hoogeweegen, W. J. M. Teunissen, P. H. M. Vervest, and R. W. Wagenaar, "Modular network design: Using information and communication technology to allocate production tasks in a virtual organization," *Decis. Sci.*, vol. 30, no. 4, pp. 1073–1103, Sep. 1999.
- [23] C. K. Tay and S. L. Chen, "Cost estimation of a service family based on modularity," *Int. J. Prod. Res.*, vol. 54, no. 10, pp. 3059–3079, May 2016.
- [24] Y. Geum, R. Kwak, and Y. Park, "Modularizing services: A modified HoQ approach," *Comput. Ind. Eng.*, vol. 62, no. 2, pp. 579–590, Mar. 2012.
- [25] R. Dörbecker, O. Tokar, and T. Böhm, "Deriving design principles for improving service modularization methods-lessons learnt from the complex integrated health care service system," in *Proc. 23rd Eur. Conf. Inf. Syst.*, Münster, Germany, 2015, p. 38.
- [26] T. Tuunanen and H. Cassab, "Service process modularization: Reuse versus variation in service extensions," *J. Service Res.*, vol. 14, no. 3, pp. 340–354, Aug. 2011.
- [27] E. Feitzinger and H. L. Lee, "Mass customization at hewlett-packard: The power of postponement," *Harvard Bus. Rev.*, vol. 75, pp. 116–123, 1997.
- [28] A. Kusiak, "Integrated product and process design: A modularity perspective," *J. Eng. Des.*, vol. 13, no. 3, pp. 223–231, Sep. 2002.
- [29] P. Carlborg and D. Kindström, "Service process modularization and modular strategies," *J. Bus. Ind. Marketing*, vol. 29, no. 4, pp. 313–323, Apr. 2014.
- [30] K. Silander, P. Torkki, P. Lillrank, A. Peltokorpi, S. A. Brax, and M. Kaila, "Modularizing specialized hospital services: Constraining characteristics, enabling activities and outcomes," *Int. J. Oper. Prod. Manage.*, vol. 37, no. 6, pp. 791–818, Jun. 2017.
- [31] M. Meyer, "PERSPECTIVE: Creating a platform-based approach for developing new services," *J. Product Innov. Manage.*, vol. 18, no. 3, pp. 188–204, May 2001.
- [32] M. H. Meyer, E. Jekowsky, and F. G. Crane, "Applying platform design to improve the integration of patient services across the continuum of care," *Manag. Service Qual. Int. J.*, vol. 17, no. 1, pp. 23–40, Jan. 2007.
- [33] C. de Blok, K. Luijckx, B. Meijboom, and J. Schols, "Modular care and service packages for independently living elderly," *Int. J. Oper. Prod. Manage.*, vol. 30, no. 1, pp. 75–97, Jan. 2010.
- [34] V. Avlonitis and J. Hsuan, "Exploring modularity in services: Cases from tourism," *Int. J. Oper. Prod. Manage.*, vol. 37, no. 6, pp. 771–790, Jun. 2017.
- [35] H. Chou, Y. Wang, M. Tseng, and C. Chieng, "Dynamic service product family for mass service customization," *Int. J. Innov. Res. Sci., Eng. Technol.*, vol. 2, pp. 6575–6585, 2013.
- [36] B. Ramesh, A. Tiwana, and K. Mohan, "Supporting information product and service families with traceability," in *Proc. Int. Workshop Softw. Product-Family Eng.* Berlin, Germany, 2002, pp. 353–363.
- [37] L. Gauss, D. P. Lacerda, and P. A. C. Miguel, "Module-based product family design: Systematic literature review and meta-synthesis," *J. Intell. Manuf.*, vol. 32, no. 1, pp. 265–312, 2020.
- [38] G. Bechler, C. Steinhardt, J. Mackert, and R. Klein, "Product line optimization in the presence of preferences for compromise alternatives," *Eur. J. Oper. Res.*, vol. 288, no. 3, pp. 902–917, Feb. 2021.
- [39] J. Jian, "Elite adaptive hybrid genetic algorithm and its realization," *Comput. Eng. Appl.*, vol. 45, pp. 34–35, 2009.
- [40] S. Mayer and C. Steinhardt, "Optimal product line pricing in the presence of budget-constrained consumers," *Eur. J. Oper. Res.*, vol. 248, no. 1, pp. 219–233, Jan. 2016.
- [41] X. Wang, J. D. Camm, and D. J. Curry, "A branch-and-price approach to the share-of-choice product line design problem," *Manage. Sci.*, vol. 55, no. 10, pp. 1718–1728, Oct. 2009.
- [42] I. Méndez-Díaz, J. J. Miranda-Bront, G. Vulcano, and P. Zabala, "A branch-and-cut algorithm for the latent-class logit assortment problem," *Discrete Appl. Math.*, vol. 164, pp. 246–263, Feb. 2014.
- [43] Y. Akçay, H. P. Natarajan, and S. H. Xu, "Joint dynamic pricing of multiple perishable products under consumer choice," *Manage. Sci.*, vol. 56, no. 8, pp. 1345–1361, Aug. 2010.
- [44] D. Bertsimas and V. V. Mišić, "Exact first-choice product line optimization," *Oper. Res.*, vol. 67, no. 3, pp. 651–670, May 2019.
- [45] A. Belloni, R. Freund, M. Selove, and D. Simester, "Optimizing product line designs: Efficient methods and comparisons," *Manage. Sci.*, vol. 54, no. 9, pp. 1544–1552, Sep. 2008.
- [46] S. Tsafarakis, "Redesigning product lines in a period of economic crisis: A hybrid simulated annealing algorithm with crossover," *Ann. Oper. Res.*, vol. 247, no. 2, pp. 617–633, Dec. 2016.
- [47] S. Kumar and A. K. Chatterjee, "A profit maximizing mathematical model for pricing and selecting optimal product line," *Comput. Ind. Eng.*, vol. 64, no. 2, pp. 545–551, Feb. 2013.
- [48] P. E. Green and A. M. Krieger, "Models and heuristics for product line selection," *Marketing Sci.*, vol. 4, no. 1, pp. 1–19, Feb. 1985.
- [49] D. Bertsimas and V. V. Mišić, "Robust product line design," *Oper. Res.*, vol. 65, no. 1, pp. 19–37, Feb. 2017.
- [50] S. K. Nair, L. S. Thakur, and K.-W. Wen, "Near optimal solutions for product line design and selection: Beam search heuristics," *Manage. Sci.*, vol. 41, no. 5, pp. 767–785, May 1995.
- [51] A. Alptekinoglu and C. J. Corbett, "Leadtime-variety tradeoff in product differentiation," *Manuf. Service Operations Manage.*, vol. 12, no. 4, pp. 569–582, Oct. 2010.
- [52] P. B. Tookanlou and H. Wong, "Determining the optimal customization levels, lead times, and inventory positioning in vertical product differentiation," *Int. J. Prod. Econ.*, vol. 221, Mar. 2020, Art. no. 107479.
- [53] S. Tsafarakis, K. Zervoudakis, A. Andronikidis, and E. Altsiadias, "Fuzzy self-tuning differential evolution for optimal product line design," *Eur. J. Oper. Res.*, vol. 287, no. 3, pp. 1161–1169, Dec. 2020.
- [54] S.-Y. Wu and P.-Y. Chen, "Versioning and piracy control for digital information goods," *Operations Res.*, vol. 56, no. 1, pp. 157–172, Feb. 2008.
- [55] A. Jain and R. Bala, "Differentiated or integrated: Capacity and service level choice for differentiated products," *Eur. J. Oper. Res.*, vol. 266, no. 3, pp. 1025–1037, May 2018.



ZHUOTONG MIAO is currently pursuing the Ph.D. degree with the College of Information Science and Engineering, Northeastern University, Shenyang, China. Her current research interests include service family design and new service development.



XINGGANG LUO received the M.Sc. degree in mechanical design and manufacturing from Northeastern University, Shenyang, China, and the Ph.D. degree in system engineering from Northeastern University, Boston, MA, USA.

He is currently a Professor with the College of Information Science and Engineering, Northeastern University. He has authored more than 50 articles on peer-reviewed journals, including *Decision Support Systems*, the *IEEE TRANSACTIONS*

ON SYSTEMS, MAN, AND CYBERNETICS, the *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, the *European Journal of Operational Research*, the *International Journal of Production Research*, and the *International Journal of Production Economics*. His current research interests include intelligent public transportation, service operation management, and optimization algorithms.



QING ZHOU has been holding a postdoctoral position with the Beijing University of Technology, since 2013. He is currently a Professor with the Management School, Hangzhou Dianzi University, China. He is also the Dean of the School of Management and the Head of the Institute of Management Decision and Innovation, HDU. His recent research interests include innovation management and technology innovation alliance.

...



ZHONGLIANG ZHANG received the Ph.D. degree in system engineering from Northeastern University, China.

He is currently an Associate Professor with the Department of Information Management and Information System, Hangzhou Dianzi University, Hangzhou, China. He has authored more than eight articles on peer-reviewed journals, including *Information Sciences*, *Knowledge-Based Systems*, *Applied Soft Computing*, *Neurocomputing*, *Engineering Applications of Artificial Intelligence*, and so on. His current research interests include data mining, imbalanced learning, and evolutionary algorithms.