



**HAL**  
open science

## IoT-based location and quality decision-making in emerging shared parking facilities with competition

Peng Wu, Feng Chu, Nasreddine Saidani, Haoxun Chen, Wei Zhou

► **To cite this version:**

Peng Wu, Feng Chu, Nasreddine Saidani, Haoxun Chen, Wei Zhou. IoT-based location and quality decision-making in emerging shared parking facilities with competition. *Decision Support Systems*, 2020, 134, pp.113301. 10.1016/j.dss.2020.113301 . hal-02563982

**HAL Id: hal-02563982**

**<https://utt.hal.science/hal-02563982v1>**

Submitted on 22 Aug 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# IoT-based Location and Quality Decision-Marking in Emerging Shared Parking Facilities with Competition

Peng Wu<sup>a</sup>, Feng Chu<sup>a,b,\*</sup>, Nasreddine Saidani<sup>c</sup>, Haoxun Chen<sup>d</sup>, Wei Zhou<sup>e,f</sup>

<sup>a</sup>*School of Economics & Management, Fuzhou University, 350116 Fuzhou, China*

<sup>b</sup>*Laboratory IBISC, Univ Evry, University of Paris-Saclay, 91025 Evry, France*

<sup>c</sup>*Company AZAP, 75008 Paris, France*

<sup>d</sup>*Laboratory LOSI, University of Technology of Troyes, 10300 Troyes, France*

<sup>e</sup>*Information & Operations Management, ESCP Europe, Paris, France*

<sup>f</sup>*Transportation Engineering, School of Naval Architecture, Ocean & Civil Engineering, Shanghai Jiaotong University, Shanghai, China*

---

## Abstract

Shared parking firms offer a double-sided platform for parking space sharing. Many of these firms provide differentiated service levels to both suppliers and buyers. This new phenomenon in the parking industry materialized thanks to recent innovations in IoT-enabled automation and electric vehicle charging technologies. We study shared parking firms. Specifically, we formulate the firm's location and quality decision problem by using a multiplicative interaction model with competition. A non-cooperative game renders the optimized quality levels and location selections at Nash equilibrium in the presence of competition. We illustrate managerial insights with a small-sized problem. For industry practitioners, we propose a tailored branch and bound based exact algorithm and a problem-specific genetic algorithm for large-sized problems. Simulated computational results confirm the effectiveness and efficiency of the proposed shared-parking decision support model.

*Keywords:* Sharing Economy, Shared Parking, Electric Vehicle, Sharing Decision Support

---

## 1. Introduction

New transportation business models have emerged in recent years, thanks in part to increasing popularity of sharing economy and related enabling technologies. Companies that provide shared transportation and parking services such as Uber, Lyft, Onepark, and Pavemint have greatly reshaped the transportation economy and civil infrastructure development in many countries. Moreover, an increasing number of property owners

---

\*Corresponding author: [feng.chu@ibisc.univ-evry.fr](mailto:feng.chu@ibisc.univ-evry.fr) (F. Chu)

have started sharing their parking spaces or membership-exclusive parking facilities to the public via shared parking platforms. The benefits of sharing parking spaces include reduction in the overall parking space requirement and increased revenue for the property owners. With competition from existing public parking providers, emerging shared parking service providers face two strategic decisions that include (1) location selection and (2) service quality choice.

The first decision on location selection is an important classical problem for shared parking firms [1]. Facing competition, companies such as Onepark and Pavemint must consider where to add new shared parking spaces and/or build parking facilities along with an incentive policy and differentiated service levels. The main business model involves incentivizing property owners to share their parking resources on the platform. At the same time, the platform owner also invests in these shared parking properties by providing automated equipment on site (e.g., check-in-out facilities, automated mechanics, video surveillance, sensors and EV charging devices). Unlike classical sharing economy platforms such as Uber or airBnb, there is a substantial (initial) investment requirement for shared parking platform to operate and grow.

The second decision on service quality level has not been considered in published literature on parking marketplace management, to the best of our knowledge. Indeed, it is not until recently that parking service quality has become an important decision because of advances in IoT and electric vehicle technologies. Most shared parking companies not only provide the platform that bridges buyers and suppliers, but also install check-point facilities, EV charging devices, sensors and surveillance equipments upon agreement with the parking space owner. This practice enables the shared parking company to differentiate its service quality. This new phenomenon related to parking service quality choice raises many questions that largely remain unanswered, including its impact on the classical location problem, service quality decision problem, and pricing and competition strategy problems.

We study both the decision problems for a shared parking firm as a competitive location problem (CLP). The set of available parking facility locations is formed from shared private parking spaces. Service quality differentiation is enabled by the installation of onsite equipments/devices that are connected via an IoT-based sensor system infrastructure. This IoT-based system allows for tracking and tracing vehicles as well as provide

information on connected parking facilities such as charging stations and various mechanical devices. It enables the service provider to design and differentiate the quality levels of the parking facilities. On-site electric charging devices offer various levels of charging speed/time. Video and sensor based surveillance of parking facility improves service quality in terms of parking security. Automated checkpoints with RFID and mobile payment facilities improve convenience and help reduce customer wait time.

Existing studies on CLP for parking marketplace management mainly focus on the location decision. However, it is important to simultaneously consider service quality and facility location as both impact a firm's long-term market share. Considering only location may lead to sub-optimality, especially when service quality is measurable and can be controlled. Over the past few years, a number of articles on other industries (e.g., [2–8]) have addressed competitive facility location and design problems in a continuous space by taking into account the service quality decisions of facilities. The quality of a facility is usually determined by the facility size, variety, convenience, maintenance effort, technology level, and other case-specific factors. We consider parking service quality with similar factors. For example, with additional electric charging devices, service can be differentiated by charging speed and quality. In addition, [9–11] considered competitive facility location and design in discrete space. The above studies generalize previous work on competitive facility location by simply assuming that the quality of the existing facilities as given. Although more realistic, response from existing competitors is not considered by any of these studies. As mentioned in [12], considering the response from competitors is a natural extension of CLP. Furthermore, we assume that the probability of a customer choosing a parking site is proportional to the utility of patronizing it. We then adopt a multiplicative interaction (MCI) model [13] to express the probability of a customer patronizing either a new parking site or an existing one.

The contributions of this study are summarized below.

- i) We consider a new shared parking facility location and quality decision problem.
- ii) We present a novel decision model for a shared parking firm that faces competition, competitors' reactions, and limited budget, by adopting the MCI model. The problem is shown to be NP-hard.
- iii) We propose an iterative solution framework composed of two main phases (parking location and quality determination) to solve the proposed model. Given a set of

new open parking facilities, the competitive decision process occurring among new and existing parking facilities is modeled as a non-cooperative game. The service quality of new and existing parking facilities is determined by the Nash equilibrium.

- iv) We develop a tailored B&B algorithm and problem-specific genetic algorithm (GA)-based method to obtain the best set of open parking facilities. Computational results on 320 randomly generated instances under different structures show the effectiveness and efficiency of the proposed approaches.

The remainder of the paper is structured as follows. We introduce and formulate the model in Section 2. In Section 3, we first present an iterative solution framework to solve the proposed problem. Two tailored iterative framework-based approaches: B&B algorithm and GA-based method are developed. Numerical experiments are provided to draw managerial insights and to validate the effectiveness and efficiency of the proposed approaches in Section 4. We conclude and identify future research directions in Section 5.

## 2. The Model

Although advanced technologies and decision support tools have been developed, today it is still considered analytically challenging to optimize location decision in a competitive environment. Competitive location problems may involve not only location decisions, but also other decisions such as quality design and/or pricing. Hotelling's pioneering work on duopoly in a linear market [14] provides an important foundation for today's research on competitive location. The competitive location problems can be categorized as in a continuous space or in a discrete space setting according to whether facilities are located on a plane or on a network (a set of discrete points). For competitive location in a continuous space, [15] studied a single facility location problem that considers competition from another facility. Later, finding the optimal location of a single new facility with the analysis of the market share function was discussed in [16] and [17]. [18] presented a geographic information system (GIS) based competitive location model for a single facility to be located in continuous space. [19] extended this by considering multiple competitive facilities and developed several heuristic algorithms. The earliest contributions to competitive location in discrete space were made by [20, 21]. Following the work of [22, 23], a number of studies have addressed discrete multi-facility location problems in the retail

industry (e.g., [24], [25–27], [28], [29], [30]). Many highlighted competitive location models are available in the literature. We refer to the survey papers [31], [32], [33], and [34] for an overview.

A few published research articles consider competitors' responses in CLP. [35] is the first to consider a firm that chooses to build facilities in a network of either competitive or oligopolistic environment. An economic equilibrium model was developed to describe the competition on the network in terms of equilibrium prices, demands, production levels and shipments. A variational inequality was proposed to specify its equilibrium. [36] considered the location decision and quality design of a new facility in continuous space with the objective of maximizing facility profit. An interval Branch and Bound (B&B) based two-stage algorithm was developed as a solution. Later, [37] developed a game theoretic model to analyze a market situation with two firms entering a new market where customers consider one of the suppliers according to travel distance and service quality. Both firms maximize market share by deciding on the location on a plane and service quality. Recently, [38] established a bi-level programming model to deal with a Huff-like competitive location and design problem where the leader desires to locate a facility such that its profit is maximized after the competitor locates its facility. It is assumed that the follower also considers maximizing its own profit. These studies mainly focus on optimally locating a single or two new facilities and assume that the budget for opening facilities and improving service quality is unlimited. While none of the published CLP research address the parking facility and quality problem, we fill the void by considering CLP for solving the emerging shared parking problem.

In summary, we consider that a shared parking firm is to locate new parking facilities in a network and simultaneously determine the service quality of each parking site to maximize its total profit. In the network, there are existing parking facilities that belong to other parking service firms that may or may not be shared. We assume that these existing parking facilities can improve their quality to compete for market share with new ones but their locations are given and fixed because relocation is expensive. Both the entrant and the existing firm have limited budget. A customer is free to choose any parking to patronize, and two utility factors that include parking quality and travel cost affect her/his location decision. The probability of a customer choosing a parking site is proportional to the patronizing utility. We adopt a multiplicative interaction (MCI)

model [13] to express the probability of a customer patronizing either a new parking site or an existing one. Note that parking owners/members do not pay for using his/her own parking facilities.

### 2.1. Problem Description

Let  $G = (N, A)$  be a network that includes a set of nodes  $N = P \cup C \cup D$  and a set of arcs  $A$ . The nodes in sets  $P$ ,  $C$  and  $D$  respectively represent the potential sites for opening new shared **P**arking service facilities, **C**ompeting parking service facilities, and **D**emand points (Figure 1). The shortest distance between the two nodes,  $i, j \in N$ , represented by an arc in  $A$  is denoted by  $d_{ij}$ .

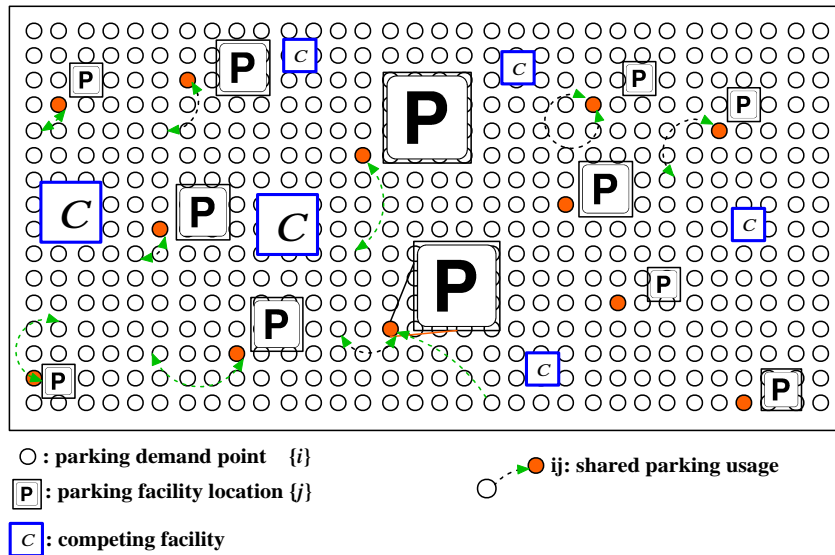


Figure 1: Shared parking location selection with discrete demand points

We consider a general framework that models a set of available parking candidates and decides which to include in the service network and at what quality level. Typical candidates can be a level or a block of parking spaces from a multiple level parking complex. Figure 1 illustrates the mechanism of shared parking services. In the figure, the size of **P** and **C** show that the density of available shared parking space may vary from area to area. The grey circle linking square **P** implies usage of private parking and the circle linking any two different nodes implies the usage of shared parking from a demand spot. The availability of parking spaces for sharing could be an issue because it directly affects the supply.

Suppose that a shared parking service firm has a limited budget  $B$  and is attempting to open one or more new parking facilities from a finite set of potential candidate locations  $P$  that are previously privately owned. The fixed cost for constructing/renovating the parking facility at site  $j$  is denoted by  $f_j, j \in P$ . There is a set of existing parking facilities  $C \subset N(|C| = m)$  that belongs to other parking service firms with service quality  $\alpha_j$  at parking  $j \in C$ . For simplicity, we assume that  $m$  existing parking facilities belong to  $m$  different competitors respectively. However, the proposed model and algorithms given later can be easily extended to the case where some competitors have multiple facilities. The potential parking service demand at demand zone  $i \in D$  is denoted by  $w_i$ . When entering the market, the entrant firm also needs to decide the service quality at each site, denoted by  $y_j, j \in P$ , and carefully take into account possible responses from existing competitors.

We assume that the service quality level of each new parking or the quality improvement level of each existing parking is taken from a discrete set. This assumption does not lead to a loss of generality but simplifies the solution to the studied problem. The entrant firm needs to decide the service quality at each site  $j \in P$ , denoted by  $y_j$ , and carefully take into account the responses from existing competitors. Each existing parking facility  $j \in C$  can improve its service quality to compete for market share subject to a limited budget  $B_j$ . Let  $z_j$  be the increased service quality,  $j \in C$ . Note that reallocating existing facilities is not allowed because it is expensive. We also assume that the service quality of both new and existing parking facilities are bounded and the maximum service quality is denoted as  $q_m, m \in P \cup D$ .

We assume that a potential customer whose destination is zone  $i, \forall i \in D$  is free to decide where to purchase parking service, and her/his decision is mainly affected by two factors: the service quality at facility  $j, \forall j \in P \cup C$  denoting its service level and the travel distance from node  $i$  to facility  $j$ , i.e.,  $d_{ij}$ . The market share of each parking is determined by the utility its customers gain from purchasing the service provided by the parking facility. The utility of a customer whose destination is demand zone  $i$  patronizing parking facility  $j$ , denoted by  $u_{ij}$ , increases in the service quality and decreases in the travel distance. Nevertheless, for facility  $j, \forall j \in P \cup C$ , the higher the service quality provided, the more the cost required. For simplicity, it is assumed that customers with the destination zone  $i \in D$  perceive the same utility [12]. According to [9], the following



utility expressions (1) and (2) for new and existing parking facilities are defined:

$$u_{ij} = y_j^{\beta_1} d_{ij}^{-\beta_2}, i \in D, j \in P \quad (1)$$

$$u_{ij} = (\alpha_j + z_j)^{\beta_1} d_{ij}^{-\beta_2}, i \in D, j \in C \quad (2)$$

where  $\beta_1$  and  $\beta_2$  are non-negative values, respectively representing the sensitivity of customers to quality and travel distance. For practical applications, these are estimated based on available real data. As defined above,  $y_j$  and  $z_j$  are decision variables for the service quality of new and existing parking, respectively. We note that  $d_{ij}$  in (1) is a variable for the new parking's location, while that in (2) is a parameter since the location of an existing parking is known.

Let  $S \subseteq P$  denote the set of selected sites for locating new parking facilities. Then, the probability of a customer  $i \in D$  to patronize facility  $j \in S \cup C$ , denoted by  $p_{ij}$ , is proposed according to an MCI-type model [13] as follows:

$$p_{ij} = \frac{u_{ij}}{\sum_{l \in C} u_{il} + \sum_{k \in S} u_{ik}}, i \in D, j \in S \quad (3)$$

$$p_{ij} = \frac{u_{ij}}{\sum_{l \in C} u_{il} + \sum_{k \in S} u_{ik}}, i \in D, j \in C \quad (4)$$

The market share or customer flow whose destination is demand zone  $i, i \in D$ , captured by a parking  $j, j \in S \cup C$ , is thus written as follows:

$$M_{ij} = w_i p_{ij}, i \in D, j \in S \cup C \quad (5)$$

Then, the profit obtained by parking  $j, j \in S \cup C$  can be defined as follows:

$$\pi_j = F\left(\sum_{i \in D} M_{ij}\right) - (f_j + \phi(y_j)), j \in S \quad (6)$$

$$\pi_j = F\left(\sum_{i \in D} M_{ij}\right) - \phi(z_j), j \in C \quad (7)$$

where  $F$  is a strictly increasing function which transforms the market share into expected sales. Following [2],  $F$  is taken as a linear function, i.e.,  $F(\sum_{i \in D} M_{ij}) = \rho \sum_{i \in D} M_{ij}$ , where  $\rho$  is the unit profit obtained.  $\phi(y_j), j \in S$  and  $\phi(z_j), j \in C$  are functions computing the cost of operating a new parking with quality level  $y_j$  and that of increasing an existing parking in quality level  $z_j$ , respectively. In this paper, we consider a linear service quality

cost function, i.e.,  $\phi(y_j) = q^c y_j$  and  $\phi(z_j) = q^c z_j$ , where  $q^c$  is the cost of increasing one level of service quality.

The problem consists of optimally deciding the location of new entrant parking facilities  $S \subseteq P$  and determining their service quality by taking into account the reactions of competitors' parking facilities and limited budget, so as to maximize the total profit of the entrant firm. Specifically,  $m$  competitors' parking service facilities that are already in the market can maximize their own profit by optimally improving their service quality.

## 2.2. Notations

The notations to be used for the formulation are summarized as follows.

### Indices:

$i$ : index of demand zone.

$j$ : index of existing parking facilities and potential sites.

### Parameters:

$N$ : set of nodes in the network,  $i, j \in N$ .

$A$ : set of arcs in the network.

$P$ : set of potential locations for opening new parking facilities,  $P \subset N$ .

$C$ : set of existing parking facilities,  $C \subset N$ .

$D$ : set of demand zones,  $D \subset N$ .

$w_i$ : demand per unit time at zone  $i \in D$ .

$d_{ij}$ : shortest distance from node  $i$  to  $j$ ,  $(i, j) \in A$ .

$\alpha_j$ : current quality level of existing facility  $j$ ,  $j \in C$ .

$\beta_1$ : customer service quality sensitivity parameter,  $\beta_1 \geq 0$ .

$\beta_2$ : customer travel distance sensitivity parameter,  $\beta_2 \geq 0$ .

$f_j$ : fixed cost associated with the opening of a parking facility at site  $j \in P$ .

$\phi(y_j)$ : cost of operating a new parking  $j$  with quality level  $y_j$ ,  $j \in P$ .

$\phi(z_j)$ : cost of increasing an existing parking  $j$  at quality level  $z_j$ ,  $j \in C$ .

$B$ : budget of the entrant firm available for opening new parking facilities.

$\rho$ : income from a single customer from sharing.

$B_j$ : budget of existing parking  $j \in C$  available for improving its quality level.

$q_m$ : maximum quality level for each parking facility. Note that  $q_m \geq \alpha_j$  for  $j \in C$ .

$q^c$ : cost of increasing one level of service quality.

### Decision variables:

$x_j$ : equal to 1 if a parking is open at site  $j, \forall j \in P$ , and 0 otherwise.

$y_j$ : quality level of new parking  $j, \forall j \in P$ , and  $y_j \in Q_j = \{q_1^j, q_2^j, \dots, q_{m_j}^j\}$ , where  $q_k^j, k = 1, 2, \dots, m_j$  is the  $k$ -th quality level of new parking  $j, 0 \in Q_j$ . Note that  $y_j = 0$  means that site  $j$  is not open.

$z_j$ : quality improvement level of parking  $j, \forall j \in C$ , already in the market, and  $z_j \in Q_j = \{q_1^j, q_2^j, \dots, q_{m_j}^j\}$ , where  $q_k^j, k = 1, 2, \dots, m_j$  is the  $k$ -th quality level of new parking facility  $j, 0 \in Q_j$ .

**Intermediate variables:**

$u_{ij}$ : utility of a customer whose destination is demand zone  $i, i \in D$  who patronizes parking facility  $j \in P \cup C$ .

$p_{ij}$ : probability of a customer  $i \in D$  to patronize parking facility  $j \in S \cup C$ .

$M_{ij}$ : market share or customer flow at demand zone  $i, i \in D$ , captured by a parking facility  $j, j \in S \cup C$ .

$\pi_j$ : profit obtained by parking facility  $j, j \in S \cup C$ .

*2.3. Formulation*

Based on the above discussion, the considered problem can be formulated as the following integer non-linear program.

Model  $\mathcal{P}_n$  :

$$\begin{aligned} \max \quad & \sum_{j \in P} \pi_j = \sum_{j \in P} \left( \rho \sum_{i \in D} M_{ij} - \phi(y_j) - f_j x_j \right) \\ & = \sum_{j \in P} \left( \sum_{i \in D} \frac{\rho w_i y_j^{\beta_1} d_{ij}^{-\beta_2}}{\sum_{l \in C} (\alpha_l + z_l)^{\beta_1} d_{il}^{-\beta_2} + \sum_{k \in P} y_k^{\beta_1} d_{ik}^{-\beta_2}} - q^c y_j - f_j x_j \right) \end{aligned} \quad (8)$$

$$\text{s.t.} \quad \sum_{j \in P} (q^c y_j + f_j x_j) \leq B, \quad (9)$$

$$y_j \leq q_m x_j, \forall j \in P, \quad (10)$$

$$q^c z_j \leq B_j, \forall j \in C, \quad (11)$$

$$z_j + \alpha_j \leq q_m, \forall j \in C, \quad (12)$$

$$x_j \in \{0, 1\}, \forall j \in P, \quad (13)$$

$$y_j, z_j \in Q_j, \forall j \in P \cup C. \quad (14)$$

Objective (8) is to maximize the profit obtained by all new parking facilities for the entrant firm. Constraint (9) ensures that the available budget of the entrant firm cannot

be exceeded. Constraint (10) restricts that the value of service quality  $y_j$  is equal to 0 if site  $j$  is not selected and the service quality  $y_j$  is bounded by  $q_m$  if site  $j$  is open. Constraint (11) ensures that the cost of each new parking  $j \in P$  for setting its service quality should not exceed its available budget  $B_j$ . Constraint (12) means that the maximum quality level of each existing parking  $j \in C$  should not exceed  $q_m$ . Constraints (13) and (14) enforce the restrictions on decision variables.

As mentioned above, each existing facility  $j, j \in C$  in the market would react to maximize its profit obtained by improving its service quality. Then, for any  $j \in C$ , the corresponding optimization problem can be also formulated as the following integer non-linear program.

Model  $\mathcal{P}_j$  :

$$\max \pi_j = \sum_{i \in D} \frac{\rho w_i (\alpha_j + z_j)^{\beta_1} d_{ij}^{-\beta_2}}{\sum_{l \in C} (\alpha_l + z_l)^{\beta_1} d_{il}^{-\beta_2} + \sum_{k \in P} y_k^{\beta_1} d_{ik}^{-\beta_2}} - q^c z_j, \quad (15)$$

subject to constraints (9)-(14).

Note that the entrant shared parking facilities and the existing ones would compete with each other for market share such that their own profits are maximized. Such a competitive process can be modeled as a non-cooperative game that will be detailed later. Since a discrete facility location problem in a competitive environment without considering quality decisions and competitors' reactions is shown to be NP-hard [39], the considered shared parking service facility location and quality design problem in a competitive environment considering the competitors' reactions is also NP-hard.

To solve the proposed model presents several challenges; in particular: (i) the incorporation of quality design and competitors' reactions makes it harder to solve; and (ii) the objectives of the model is non-convex and discontinuous (i.e., non-smooth and non-linear). Existing methodologies cannot be directly used to solve the proposed model, thus we develop novel solution methodologies next.

### 3. Solution Methodology

In this section, a solution framework is suggested to solve the considered problem. Its core idea is to decompose the studied problem into two main phases: *Quality determination* and *Location*.

*Quality determination:* Given a set of opened new parking facilities  $S$ , determining the service quality at each parking as well as the improved service quality level at each existing parking;

*Location:* Determining an optimal set of locations  $S$ .

In the framework, the former phase serves as sub-routine for the latter.

### 3.1. Quality determination

Given an optimal set of locations  $S \subseteq P$ , the optimization problem  $\mathcal{P}_n$  reduces to the following non-linear integer program:

Model  $\mathcal{P}'_n$  :

$$\max \sum_{j \in S} \sum_{i \in D} \frac{\rho w_i y_j^{\beta_1} d_{ij}^{-\beta_2}}{\sum_{l \in C} (\alpha_l + z_l)^{\beta_1} d_{il}^{-\beta_2} + \sum_{k \in S} y_k^{\beta_1} d_{ik}^{-\beta_2}} - \sum_{j \in S} q^c y_j \quad (16)$$

$$\text{s.t.} \quad \sum_{j \in S} q^c y_j \leq B - \sum_{j \in S} f_j \quad (17)$$

$$q^c z_j \leq B_j, \forall j \in C \quad (18)$$

$$y_j \leq q_m, \forall j \in S \quad (19)$$

$$z_j \leq q_m - \alpha_j, \forall j \in C \quad (20)$$

$$y_j, z_j \in Q_j, \forall j \in S \cup C, \quad (21)$$

and for any  $j \in C$ , its corresponding problem also reduces to the following non-linear program:

Model  $\mathcal{P}'_j$  :

$$\max \sum_{i \in D} \frac{\rho w_i (\alpha_j + z_j)^{\beta_1} d_{ij}^{-\beta_2}}{\sum_{l \in C} (\alpha_l + z_l)^{\beta_1} d_{il}^{-\beta_2} + \sum_{k \in S} y_l^{\beta_1} d_{il}^{-\beta_2}} - q^c z_j, \quad (22)$$

subject to constraints (17)-(21).

In the quality level determination phase, each parking  $j, j \in S \cup C$  seeks for its best service quality as the best response to the other parking facilities' quality decisions. The determination of the best service quality of both new and existing parking facilities lies in finding an optimal quality combination  $(\mathbf{y}, \mathbf{z})$  such that objectives (16) and (22) are maximized while satisfying constraints (17)-(21). To derive the best quality, we model the competitive quality decision process occurring among all parking facilities as a non-cooperative multi-player game in which a parking and a possible set of quality (or a

possible set of quality improvements) are considered as a player  $j$  and a set of strategies  $Q_j$ , respectively,  $j \in S \cup C$ . Moreover, any feasible strategy of each parking must satisfy its budget constraint. The game's non-cooperative nature motivates us to choose Nash equilibrium as its solution. In Nash equilibrium, the set of quality choices made by the parking facilities are their best responses to the choices of the competitors' parking facilities.

As previously assumed,  $Q_j, \forall j \in S \cup C$  is a finite discrete set, thus the game's solutions are finite. Since in reality either the entrant firm or the competitors usually cannot know the complete market information, a mixed-strategy is adopted to ensure the existence of Nash equilibrium for the game. [40] proved that a finite game has a Nash equilibrium as stated by the following theorem.

**Theorem 1.** *If a game  $G$  is finite, then there is a mixed-strategy Nash Equilibrium for  $G$  [40].*

The considered non-cooperative  $n$ -player game can be represented by the following tuple:

$$\Gamma = (S \cup C, \{Q_j\}_{j \in S \cup C}, \{\pi_j\}_{j \in S \cup C}) \quad (23)$$

According to Theorem 1 in [41], the problem of computing the Nash equilibrium of game  $\Gamma$  can be formulated as an equivalent non-linear minimization program, shown as follows:

Model  $\mathcal{P}_\Gamma$  :

$$\min \sum_{r \in S \cup C} (\pi_r^* - \pi_r(\sigma)) \quad (24)$$

$$\text{s.t.} \quad \pi_r(\sigma_{-r}, s_r^k) - \pi_r^* \leq 0, \forall r \in S \cup C, \forall k = 1, 2, \dots, l_r \quad (25)$$

$$\sum_{k=1}^{l_r} \sigma_r^k = 1, \forall r \in S \cup C \quad (26)$$

$$\sigma_r^k \geq 0, \forall r \in S \cup C \quad (27)$$

where  $\sigma$  is the decision vector denoting the mixed strategy combination;  $\sigma_{-r}$  and  $s_r^k$  denote the mixed strategy vector formed by all players except player  $r$  and  $k$ -th pure strategy of player  $r$ , respectively; and  $\sigma_r^k$  represents the probability assigned to pure strategy  $s_r^k$ . Note that the optimal value of  $\mathcal{P}_\Gamma$  is 0. The value of  $\pi_r^*$  at the optimal point gives the expected payoff of player  $r$ . For more details on the above formulation, we refer to [41].

In this study, we adopt the sequential quadratic programming based quasi-Newton algorithm to solve  $\mathcal{P}_\Gamma$  such that the Nash equilibrium is identified. For more details on this method, interested readers can refer to [41, 42].

### 3.2. Location

In this section, we develop tailored B&B and GA for locating parking facilities. In general, although a B&B algorithm can find the optimal location, it is often computationally hard for large-sized NP-hard problems. Thus, we also develop a meta-heuristic algorithm to solve large-sized instances more efficiently.

#### 3.2.1. Branch and bound method

In this section, we focus on designing an exact method: branch and bound algorithm to find an optimal solution. The B&B method is an iterative method that exploits a tree structure (generally called tree search). The main part of the algorithm is composed of two basic operations: branching and bounding. They can be simply explained as follows.

At  $n$ -th iteration ( $n \geq 1$ ), a node  $o_n$  at the current tree representing a partial solution with its *associated total fixed cost*  $f_{o_n}$  is selected according to a specified search strategy (for the first iteration, the root node corresponding to the original solution space is considered), and then is branched into two child nodes  $o_n^1$  and  $o_n^2$ , which describes deciding to add a new potential parking site  $j$  or not, respectively. For each node  $o_n^i$ ,  $i = 1, 2$ , we update  $f_{o_n^i} = f_{o_n} + f_j$ . If  $f_{o_n^i} > B$ , which indicates that it does not contain a feasible solution, then node  $o_n^i$  is pruned. If node  $o_n^i$  is located at the  $|P|$ -th level detailed later, which means that all potential parking sites are considered, then we compute a lower bound  $LB_{o_n^i}$  by determining the quality of both new and existing parking facilities, and update the best lower bound found so far  $LB_b$  if  $LB_{o_n^i} > LB_b$ . Otherwise, we define problem  $UB(o_n^i)$  representing a relaxation problem of the corresponding subproblem that is then solved to obtain a corresponding upper bound (if it exists), denoted by  $UB_{o_n^i}$ . If  $UB_{o_n^i}$  is less than or equal to  $LB_b$ , which implies that it does not contain a global optimal solution, then node  $o_n^i$  is closed and the search for  $o_n^i$  is terminated. If there exist unexplored nodes, a new iteration repeats; otherwise, the current lower bound is considered as the optimal solution of the problem and the search is terminated. In what follows, the main components of the proposed branch and bound algorithm including branching rule, lower and upper bounds computation, search strategy and pruning schemes are detailed,

followed by a summarization of the overall algorithm.

*A. Branching rule:* In a traditional B&B method, the branching procedure usually considers all decision variables of an optimization problem. However, for complex optimization problems with multiple sets of variables like the considered problem containing three sets of variables, i.e.,  $x_j$ ,  $y_j$ , and  $z_j$ , a branching procedure taking into account all sets of variables usually requires excessive memory, which may result in slow convergence of a B&B method. To accelerate the speed of the B&B method, we propose a branching procedure based on partial variables, which is given in detail next.

For the studied problem, we can find that the location decision of new parking  $x_j$  would greatly affect the values of  $y_j$  and  $z_j$  as once locations of new parking facilities are determined, the optimal quality of both new and existing parking facilities can be subsequently determined by computing the Nash equilibrium. Such an observation leads us to design a branching procedure depending on variables  $x_j$  instead of all sets of variables.

In our B&B method, the potential parking sites in set  $P$  are sequentially considered through the branching procedure. Two branches are generated for each node. Each node at level  $k$  represents a partial location scheme  $S'$ , in which  $k$  potential parking sites are already determined, and the remaining  $|P| - k$  potential sites are in the un-considered set and sequentially considered next. At each node, two child nodes are created by considering a new potential site  $j$  selected from the un-considered set. In one child node, the new potential site is considered to be opened, i.e.,  $x_j = 1$ , while in the other node the new potential site is considered not to be opened, i.e.,  $x_j = 0$ . Note that the selected parking site for each branching node is the one with the highest fixed cost among the remaining unexplored ones, which aims to eliminate the leaf nodes violating the budget constraint as soon as possible in the B&B tree, thereby accelerating the convergence speed.

*B. Lower bound computation:* To achieve rapid pruning for our B&B algorithm, thereby accelerating its solution speed, we design a constructive heuristic to compute a lower bound of solutions (i.e., a feasible solution). The basic idea of the heuristic is to open as many new parking facilities as possible while respecting limited budget  $B$  to compete for more market share. The proposed heuristic first finds a good feasible set of open locations  $S_0$ , then determines the quality of all parking facilities by computing the Nash equilib-



rium for new opened and existing parking facilities, and finally calculating the objective function value  $\pi_{S_0}$ . Note that the heuristic can also be used as a standalone method. The procedure of the heuristic for computing a lower bound is outlined in Algorithm 1.

---

**Algorithm 1** Heuristic procedure for computing a lower bound

---

- 1: Initialize  $S_0 \leftarrow \emptyset$ , and  $TC \leftarrow 0$ ;
  - 2: Sort all the potential sites in set  $P$  in the increasing order of their fixed cost, and let  $j = 1$ ;
  - 3: **while**  $TC \leq B$  **do**
  - 4: Open the  $j$ -th sorted potential site, i.e.,  $x_j = 1$ , and set  $S_0 = S_0 \cup j$ ;
  - 5: Let  $TC = TC + f_j$ ;
  - 6: **end while**
  - 7: Determine the quality of both new and existing parking facilities  $(\mathbf{y}, \mathbf{z})$  based on  $S_0$  by computing the Nash equilibrium of game  $\Gamma$ ;
  - 8: Let  $TC = TC + \sum_{j \in S_0} q^c y_j$ ;
  - 9: **if**  $TC \leq B$  **then**
  - 10: Output the objective function value  $\pi_{S_0}$  as the lower bound  $LB$  and stop.
  - 11: **else**
  - 12:  $TC = TC - \sum_{j \in S_0} q^c y_j$ ;
  - 13: Close the latest-opened parking  $j'$  and let  $S_0 = S_0 \setminus \{j'\}$ ;
  - 14:  $TC = TC - f_{j'}$  and go to Step 7;
  - 15: **end if**
- 

*C. Upper bound computation:* To accelerate the convergence of the B&B method, an upper bounding scheme is proposed to prune part of branches in the search tree that are not necessary to be explored further. Each node  $o$  in the B&B tree corresponds to a partial solution where the values of some variables  $x_j$ 's are determined. Let  $\Omega(o)$  be these  $j$ s. To calculate an upper bound of the total profit of all new parking facilities corresponding to node  $o$ , for each existing parking  $j \in C$ , we first fix the quality level to its initial one  $\alpha_j$ , i.e.,  $z_j = 0$ ; and for each new parking  $j \in P$ , we set its service level as the maximum level  $q_m$  if it is considered to be opened. Thus, the market share captured by the new parking facilities would be maximal.

In addition, we also relax the budget constraints for existing and new entrant parking facilities and only ensure that the total fixed cost of new entrant parking facilities is equal to or less than the available budget. However, the obtained relaxed model is still non-linear. To make the problem more tractable, we consider the attractions of the existing parking facilities and the sites at which we decide to open a parking only, which is the

denominator part of the objective function. Thus, we obtain a linear objective function. Based on the above analysis, an upper bound of the total profit of all new parking facilities corresponding to node  $o$  can be obtained by solving the following ILP:

Model  $UB(o)$ :

$$UB_o = \max \quad \rho \sum_{i \in D} w_i \frac{\sum_{j \in P} q_m^{\beta_1} d_{ij}^{-\beta_2} x_j}{\sum_{j \in C} \alpha_j^{\beta_1} d_{ij}^{-\beta_2}} - \sum_{j \in P} f_j x_j \quad (28)$$

$$\text{s.t.} \quad \sum_{j \in P \setminus \Omega} f_j x_j \leq B - \sum_{j \in \Omega} f_j x_j^o \quad (29)$$

$$x_j = x_j^o, \forall j \in \Omega \quad (30)$$

$$x_j \in \{0, 1\}, \forall j \in P \setminus \Omega \quad (31)$$

where  $x_j^o$  are those values determined in node  $o$ . Note that the above ILP can be efficiently solved by any optimization solver such as CPLEX.

If there is no solution to  $UB(o)$ , then node  $o$  cannot lead to a feasible solution, and thus can be eliminated from the search tree. Otherwise, the optimal objective function value  $UB_o$  provides an upper bound corresponding to node  $o$ .

*D. Search strategy and pruning schemes:* The selection of a good search strategy is important for a B&B algorithm in terms of its efficiency as measured by computational time and memory requirement. Different search strategies exist in the literature, such as depth-first search, best-bound-search, and breadth-first search [43]. In our B&B algorithm, we use the depth-first-search strategy plus backtracking rule to select a node for the next branching in the search tree because it performs better than the others.

The search procedure in a branch is terminated when each of the following conditions is satisfied:

- i) The current selected node is a leaf or belongs to the  $|P|$ -th level; in other words, all potential parking sites are added.
- ii) The corresponding total fixed cost is greater than the budget  $B$ .
- iii) The corresponding upper bound of the current node is less than or equal to the best lower bound found so far.

*E. Overall algorithm:* The B&B algorithm to solve the addressed competitive shared parking facility location and quality design problem is outlined in Algorithm 2.

---

**Algorithm 2** Branch and bound algorithm

---

- 1: Construct an initial feasible location configuration  $S_0$  of the new parking facilities and compute the corresponding objective function value  $\pi_{S_0}$  with **Algorithm 1**, and set the current best lower bound  $LB_b$  as  $\pi_{S_0}$ ;
  - 2: Sort all the potential sites in set  $P$  in increasing order of their fixed cost;
  - 3: Select an unexplored node in the B&B search tree according to the depth-first-search strategy plus backtracking rule;
  - 4: Split the selected node into two children branches:  $x_j = 1$  and  $x_j = 0$ , where  $j$  denotes the potential site with the largest fixed cost among the un-considered sites;
  - 5: **for**  $k = 1; k \leq 2; k++$  **do**
  - 6:     **if**  $\sum_{i=1}^j f_i x_i > B$  **then**
  - 7:         Node  $o$  is eliminated;
  - 8:     **else if**  $j = |P|$  **then**
  - 9:         Determine the quality of both new and existing parking facilities  $\mathbf{y}, \mathbf{z}$ , which ensures the Nash equilibrium, and update the best lower bound  $\pi_b$  as  $\pi(\mathbf{x}, \mathbf{y}, \mathbf{z})$  if  $\pi(\mathbf{x}, \mathbf{y}, \mathbf{z}) > LB_b$ ;
  - 10:     **else**
  - 11:         Compute an upper bound  $UB_o$  for each branch by solving  $UB(o)$  defined in (28)-(31) with CPLEX and obtain its corresponding location solution  $\mathbf{x}$ ;
  - 12:         **if** there is no solution or  $UB_o \leq LB_b$  **then**
  - 13:             Node  $o$  is eliminated;
  - 14:         **end if**
  - 15:     **end if**
  - 16: **end for**
  - 17: **if** there is an unexplored node **then**
  - 18:     Go to Step 3;
  - 19: **else**
  - 20:     Output the best solution  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  and the corresponding objective function value  $LB_b$  and stop.
  - 21: **end if**
- 

### 3.2.2. Genetic algorithm based method

As the problem is NP-hard, determining optimal solutions is computationally difficult, especially for large-scale problems. Thus, in addition to the B&B based method developed in the previous section, we propose a GA-based method to tackle large-scale problems. GA is a nature-inspired meta-heuristic method introduced by [44] for solving complex optimization problems. Due to its practical applicability and extensive generality, GA has been successfully applied to many different optimization problems including workforce staffing and assignment [45], credit scoring [46], and order clustering [47].

A GA is usually driven by an initial “population” of feasible solutions, each of which is encoded as a chromosome. A new population of chromosomes is produced following the operations of crossover, mutation, evaluation, and selection. The GA stops once a stopping criterion is reached. Detailed procedure for solving the proposed problem with a GA-based method is described next.

*A. Solution representation:* In a traditional GA, a chromosome is encoded to represent a solution to the original problem. To effectively and efficiently solve the proposed problem by GA, we propose to first use a chromosome to represent a partial solution and then to obtain a complete solution by determining the remaining variables depending on the derived partial solution. For this reason, we consider first encoding new parking facilities' location variables  $\mathbf{x}$  by a chromosome.

Thus, we define a chromosome  $c = \{c_j | j = 1, 2, \dots, |P|\}$ , where  $|P|$  denotes the number of potential locations for opening new parking facilities, to represent the set of decision variables  $\mathbf{x} = \{x_j | j = 1, 2, \dots, |P|\}$ . Each gene  $c_j$  in chromosome  $c$  represents the value of a decision variable  $x_j$ . Then, based on the partial solution specified by a chromosome, we obtain a complete solution by determining the quality of all parking facilities (i.e.,  $(\mathbf{y}, \mathbf{z})$ ) identified by the Nash equilibrium of game  $\Gamma$ .

*B. Initial population generation:* The performance of a GA is affected by the choice of its initial population of chromosomes. Thus, a population initialization procedure tailored for the proposed problem is designed here.

To improve the quality of the initial set of individuals and reduce the search space for an optimal solution, we first propose an upper bound on the number of new parking facilities, denoted by  $\overline{|S|}$ . Let  $q_{min}^j$  denote the lowest quality among set  $Q_j$  of parking,  $j \in P$ . Then,  $\overline{|S|}$  is derived via Algorithm 3 whose core idea is to open the potential parking facilities of the lowest fixed cost with consideration of the lowest service quality while respecting the budget limit. Obviously,  $\overline{|S|}$  is less than or equal to  $|P|$  which helps shrink the solution space.

---

**Algorithm 3** Computing an upper bound on the number of new parking facilities

---

**Input:**  $f_j, q_{min}^j, q^c$ , and  $B$

- 1: Initialize total cost  $TC \leftarrow 0$  and  $\overline{|S|} \leftarrow 0$
  - 2: Sort all potential sites in set  $P$  in the non-decreasing order of their fixed cost  $f_j$
  - 3: Set  $j=1$  and  $TC = TC + f_j + q_{min}^j q^c$ .
  - 4: **while**  $TC \leq B$  **do**
  - 5:      $\overline{|S|} = \overline{|S|} + 1, j = j + 1$
  - 6:      $TC = TC + f_j + q_{min}^j q^c$
  - 7: **end while**
  - 8: Output  $\overline{|S|}$ .
-

Then, we set the size of the population as  $PopSize$ . For a chromosome  $c$ , we first randomly generate an integer  $|S| \in [1, \overline{|S|}]$ , then randomly select  $|S|$  potential sites from set  $P$  and set their corresponding genes  $c_j$  with the value of 1 and the remaining genes  $c_j$  with the value of 0, respectively. Thus, we obtain an initial chromosome  $c = (c_1, c_2, \dots, c_{|P|})$ . We check whether this chromosome satisfies budget constraint (9) by considering a minimum quality level  $q_{min}^j$  for each new opened parking. If so, this chromosome is feasible. Otherwise, it is deleted and a new one is regenerated. In the same way,  $PopSize$  chromosomes are generated as the initial population  $\{c^1, c^2, \dots, c^{PopSize}\}$ . The detailed procedure for the initial population generation is outlined in Algorithm 4.

---

**Algorithm 4** Initial population generation

---

**Input:**  $PopSize$ ,  $\overline{|S|}$ , and  $B$

- 1: Initialize  $l \leftarrow 1$
  - 2: **while**  $l \leq PopSize$  **do**
  - 3:     Set  $c_j^l \leftarrow 0$  for  $\forall j \in \{1, 2, \dots, |P|\}$
  - 4:     Randomly generate an integer number  $|S| \in [1, \overline{|S|}]$
  - 5:     Randomly select  $|S|$  potential sites from set  $P$  and set their corresponding genes  $c_j^l$  with the value of 1
  - 6:     **if**  $\sum_{j \in P} (q^c q_{min}^j + f_j) c_j^l > B$  **then**
  - 7:         Go to Step 3
  - 8:     **else**
  - 9:          $c^l$  is found and set  $l = l + 1$
  - 10:    **end if**
  - 11: **end while**
  - 12: Output  $\{c^1, c^2, \dots, c^{PopSize}\}$ .
- 

*C. Fitness computation:* For a chromosome  $c$  in the population, the objective function of its decoded solution  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  is used to define the fitness function. First, we obtain  $x_j = c_j$  for  $j = 1, 2, \dots, |P|$ . Then, the quality of new and existing parking facilities  $(\mathbf{y}, \mathbf{z})$  are determined by calculating the Nash equilibrium of game  $\Gamma$ .

As the decoded solution may violate the budget constraint (9), we thus transform the budget constraint into a soft constraint based on a budget feasibility test (BFT) function. BFT is defined as  $\max\{0, \sum_{j \in P} (q^c y_j + f_j x_j) - B\}$ . If BFT of a solution is greater than 0, a penalty value will be added to the fitness function. Thus, the fitness value of a

chromosome  $c$  is calculated as follows:

$$Fit(c) = \mathbf{Profit}(\mathbf{x}, \mathbf{y}, \mathbf{z}) - \eta \times \text{BFT} \quad (32)$$

where  $\mathbf{Profit}()$  calculates the total profit of all new entrant parking facilities via (8); and  $\eta$  denotes a penalty factor. The larger the fitness function value, the better the corresponding individual.

*D. Section operation:* The selection of chromosomes is done using roulette wheel section. It is a fitness-proportional-based selection in which chromosomes with higher fitness have larger probability to be selected. For each chromosome  $c^l$ , its probability of being selected is defined as  $p_l = \frac{Fit(c^l)}{\sum_l Fit(c^l)}$ . The selection process is done by spinning the roulette wheel  $PopSize$  times. Each time one chromosome is chosen for a new population with probability  $p_l$ .

*E. Crossover operation:* First, we define a probability  $p_c$  for crossover operation. Then, we randomly generate a real number  $r \in [0, 1]$ . If  $p_c > r$ , two parent chromosomes  $c^l = (c_1^l, c_2^l, \dots, c_{|P|}^l)$  and  $c^k = (c_1^k, c_2^k, \dots, c_{|P|}^k)$  and an integer  $\lambda \in [2, |P| - 1]$  are randomly generated. Thus, two offsprings  $(c_1^l, \dots, c_\lambda^l, c_{\lambda+1}^k, \dots, c_{|P|}^k)$  and  $(c_1^k, \dots, c_k^k, c_{\lambda+1}^l, \dots, c_{|P|}^l)$  are produced.

*F. Mutation operation:* First, we define a probability  $p_m$  for mutation operation. Then, for each gene of the selected parent chromosome, randomly generate a real number  $s \in [0, 1]$  and update the gene's value from 0 to 1 or 1 to 0 if  $p_m > s$ . Thus, a mutated child chromosome is generated.

*G. Termination criterion:* The GA-based algorithm is terminated when a maximum number of generations  $MaxGen$  is reached.

*H. Overall algorithm:* Algorithm 5 presents the GA-based method.

---

**Algorithm 5** GA-based method

---

- 1: Initialize the population size  $PopSize$ , crossover probability  $p_c$ , mutation probability  $p_m$ , and maximum number of generations  $MaxGen$ . Set generation index  $i \leftarrow -1$ ;
  - 2: Initialize  $PopSize$  chromosomes as the initial population calling **Algorithm 4**;
  - 3: Calculate the fitness value of each new chromosome  $c$ ;
  - 4: Set  $x_j = c_j$  for  $j = 1, 2, \dots, |P|$ ;
  - 5: Derive the quality of both new and existing parking facilities  $(\mathbf{y}, \mathbf{z})$  by computing the Nash equilibrium of game  $\Gamma$ ;
  - 6: Calculate the fitness value  $Fit(c)$  via (32);
  - 7: Select chromosomes by spinning the roulette wheel if  $i > 1$ ;
  - 8: Produce new chromosomes via crossover and mutation operations;
  - 9: **if**  $i = MaxGen$  **then**
  - 10:     Output the best found solution  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ , then stop.
  - 11: **else**
  - 12:     Set  $i = i + 1$  and go to Step 3;
  - 13: **end if**
- 

#### 4. Numerical Experiments & Managerial Insights

In this section, we first use a small example for illustration and to draw managerial insights. We then test a large-sized problem with 320 randomly generated instances to verify the performance of the proposed algorithms.

##### 4.1. Simple example & managerial insights

A simple simulated example with a five by five customer demand zone (Figure 2) is conducted. We consider 25 demand zones and 6 potential shared parking sites P1-P6. The red zone indicates a high-demand area where an existing competitor C is located. The parameters are set as follows: the current quality level of the competitor is set at 5, the maximum quality level for any parking facility is 15, the cost for increasing one

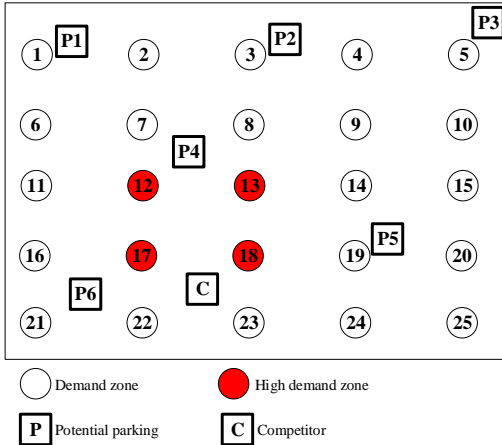


Figure 2: Basic setup

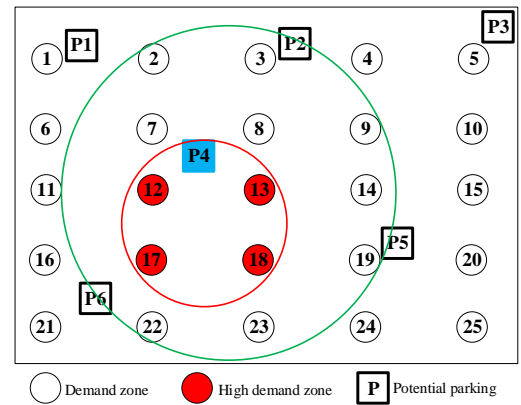


Figure 3: Location decision without competition

level of service quality is 50, the income from a single customer from sharing is 50. The fixed costs of selecting P1, P2, P3, P4, P5, and P6 are 300, 500, 300, 700, 600, and 500, respectively. The customer demand is set to be 10 from zone 1-25, except the red zones 12, 13, 17, and 18, where demand is set to be 100. The distances matrix between potential and existing parking and demand zones are listed in Table 1.

Table 1: The distances between potential and existing parking and demand zones

$d_{ji} \setminus D$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
P1	0.1	5	10	15	20	5	7	11	16	21	10	11	14	18	22	15	16	18	21	25	20	21	23	25	28
P2	10	5	0.1	5	10	11	7	5	7	11	14	11	10	11	14	18	16	15	16	18	23	21	20	21	23
P3	20	15	10	5	0.1	21	16	11	7	5	22	18	14	11	10	25	21	18	16	15	28	25	23	21	20
P4	10	7	8	11	15	7	3	4	8	13	8	4	4	8	13	11	8	9	11	15	15	13	13	15	18
P5	21	18	16	15	16	18	14	11	10	11	16	11	7	5	7	15	10	5	0.1	5	16	11	7	5	7
P6	18	18	20	22	25	13	13	15	18	22	8	9	11	15	20	4	4	9	13	18	3	4	8	13	18
C	20	18	18	19	22	15	13	13	15	18	11	9	8	11	14	9	4	4	8	12	8	4	2	8	12

We consider three scenarios (1 monopoly, 2 duopoly, 3 duopoly with service differentiation) and investigate the optimal location and quality decisions. For the monopoly case, there is no competition so minimizing the weighted travel distance to achieve a high service coverage of demand is simply optimal for the firm. For a fair comparison among three scenarios, the initial quality level of new shared parking facilities for the first two scenarios are set at 5, which is the same as that of the competitor.

Table 2: Computational results for the case study

Scenarios	New entrant firm				The competitor		
	Profit	Cost	Location	Quality	Profit	Cost	Quality improvement
Scenario 1	29650	850	P4	-	-	-	-
Scenario 2	18335.98	1800	P4, P5	-	10364.02	0	-
Scenario 3	14809.73	2000	P3, P4	5, 15	13190.27	500	10

The computational results are summarized in Table 2 and illustrated in Figures 2-6. Figure 2 shows the basic model setup with zones of regular parking demand, high demand zones, potential parking facilities and a competitor. Figure 3 shows that without competition. The best solution for the firm is to select facility P4, which is located closest to the high demand zone. When there is a competitor, as shown in Figure 4, the optimal locations include both P4 and P5. One explanation is that the competitor



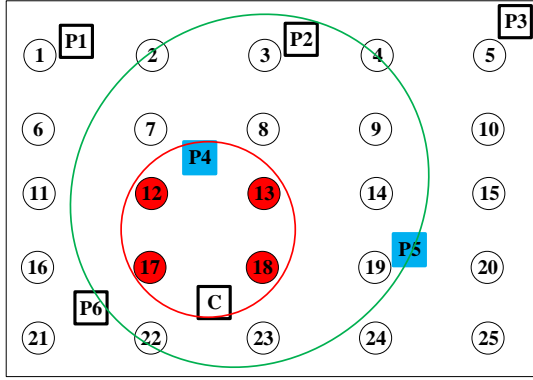


Figure 4: Location decision with competition

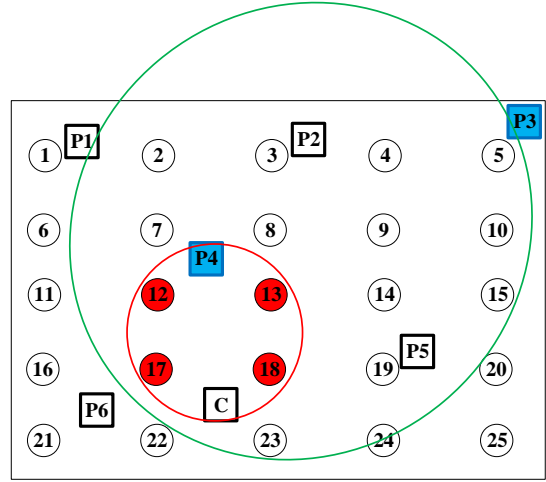


Figure 5: Location + quality decision with competition

changes the demand distribution for the firm, especially in the red zones. To react, the firm includes a new parking facility that is less affected by the competitor. We include differentiated (higher) service offering in Figure 5. The firm's location selection becomes P4 and P3. Higher service offering (such as fast EV charging) gives the customer more incentive to use such parking facilities even if they are farther away. Higher service level expands the maximum travel distance limit between the demand and the facility location. Traditionally if we consider only location, P3 would not be included because it is the most remote site with low coverage rate. With high service level offerings and with competition, P3 now appears in the list of optimal parking locations when the competitor also increases its service level.

From Table 2, we see that the total profits of the new entrant shared-parking firm without competition, with competition, and with quality differentiation are 29650, 18335.98, and 14809.73 respectively. The investment cost increases from 850, 1800, to 2000. For the case without competition, only one facility is established, while two facilities are opened for the cases with competition. Moreover, the area with maximum travel distance is enlarged when the service quality is increased. For the first two cases without quality competition, the center parking points with high demands are considered. With quality differentiation, the firm's profit decreases because of the additional cost to improve quality and because of competitor's reaction by improving its service quality. In this case, the firm's decision includes a very remote parking point P3, which is counter-intuitive if we don't consider parking service competition. Such results are clearly shown in Figures 2-5. The competitor's total profit increases from 10364.02 to 13190.27 when

service quality improvement is considered.

#### 4.2. Larger-sized randomly generated instances

For the large sized problems, different cities/areas have their unique setups in terms of shared parking availability, parking demand pattern, and competition. Each scenario (e.g., map of parking facilities in a city) renders a unique problem setup for the decision maker. The solution algorithms perform differently when the problem size differs. Regarding solvability and solution optimality, we therefore conduct a series of analysis to investigate the suitable solution methods from small to large sized problems.

We evaluate the performance of the proposed algorithms with 32 groups of randomly generated instances under different structures with 10 instances for each group, thus leading to 320 instances in total. Each item for each group in the computational result table is the average value of the 10 instances. The proposed algorithms are coded in C++, with which the sequential quadratic programming based quasi Newton method [41] and IBM ILOG CPLEX 12.2 are combined to compute the Nash equilibrium of game  $\Gamma$  and exactly solve  $UB(o)$ , respectively. CPLEX is run in default options. We set *PopSize* and *MaxGen* to 50 and 200 by rules of thumb. To select a better crossover and mutation probability combination  $(p_c, p_m)$ , various combinations of crossover probabilities  $\{0.7, 0.8, 0.9\}$  and mutation probabilities  $\{0.05, 0.10, 0.15\}$  are compared in the preliminary experiments. The combination  $(0.9, 0.15)$  generally performs better than the others and is then adopted for the GA-based method in the remaining experiments. In addition, for each instance, we test the GA-based method through 10 independent runs and compute the average value for analysis. The computational time for both methods of each instance is limited to 36,000 s. All tests are carried out on a PC equipped with a 2.4 GHz processor and 2 GB of memory.

The data for each test instance is randomly generated in the following way. The customer demand per unit time at each zone  $w_i, i \in D$ , is randomly generated in the interval  $[10, 50]$ . The locations of demand zones, existing and candidate parking facilities are randomly distributed in a  $[0, 50] \times [0, 50]$  Euclidean plane. The shortest distance between each zone and each parking  $d_{ij}, i \in D, j \in P \cup C$ , is calculated accordingly. The service quality of each existing parking  $j$  is randomly generated in the interval  $[1, 10]$ ,  $j \in C$ . Each new parking  $j$  is assumed to have three levels of service quality, i.e.,  $Q_j = \{5, 10, 15\}$ ,  $j \in P$ ; and each existing parking  $j$  is assumed to have two levels of

service quality improvement,  $Q_j = \{5, 10\}, j \in C$ . The cost for increasing one level of service quality  $q^c$  and the fixed cost of opening a potential parking  $f_j, j \in P$  are randomly generated in the interval  $[50, 100]$  and  $[200, 500]$ , respectively. Both sensitivity parameters  $\beta_1$  and  $\beta_2$  are set to 1. The available budget of the entrant firm  $B$  is set to 2000. The available budget of the existing parking is set as:  $B_j = 1000, j \in C$ . The income from a single customer from sharing  $\rho$  is set to 50.

In the experiments, the results obtained by the B&B algorithm are compared with those obtained by the GA-based method in terms of two performance measures : (i) deviation (%) =  $\frac{LB_{GA} - Opt.}{Opt.} * 100\%$  where  $Opt.$  and  $LB_{GA}$  denote the objective function value obtained by the B&B algorithm and GA-based method, respectively; and (ii) computational time ratio =  $\frac{CT_{B\&B}}{CT_{GA}}$  where  $CT_{B\&B}$  and  $CT_{GA}$  denote the computational time (CPU seconds) spent by the B&B algorithm and GA-based method, respectively. The comparison results of both methods are summarized in Tables 3-6.

Table 3: Computational results for instances with  $|P| = 3$

Group	$ C $	$ D $	B&B algorithm	GA based method		Ratio
			CPU time (s)	Deviation (%)	CPU time (s)	
1	3	20	8.50	<b>0.00</b>	35.20	0.24
2		50	8.84	<b>0.00</b>	35.30	0.25
3		100	8.88	<b>0.00</b>	38.30	0.23
4	5	20	13.19	<b>0.00</b>	54.60	0.24
5		50	16.92	<b>0.00</b>	62.50	0.27
6		100	18.01	<b>0.00</b>	87.50	0.21
7	7	20	64.12	<b>0.00</b>	173.20	0.37
8		50	86.29	<b>0.00</b>	339.50	0.25
9		100	120.82	<b>0.00</b>	404.00	0.30
<i>Average</i>			<i>38.40</i>	<i><b>0.00</b></i>	<i>136.68</i>	<i>0.28</i>

Table 3 reports the computational results for instances with  $|P|$  fixed at 3,  $|C| = 3, 5, 7$  and  $|D| = 20, 50, 100$ . From Table 1, we observe that: i) The proposed B&B algorithm can find optimal solution for all the instances within about 2 minutes, which indicates its effectiveness and efficiency in obtaining optimal solutions. The deviations of the solutions found by the GA-based method are 0, implying that an optimal solution is obtained for each instance. This shows that the proposed GA-based method is able to find high-quality solutions; ii) The CPU time for the B&B algorithm (resp. the GA-based method) varies from 8.5 (resp. 35.20) to 120.82 s (resp. 404 s) with its average value 38.4 s (resp. 136.68 s). The B&B algorithm spends less time than the GA-based algorithm over all groups, and the former spends 28% time of the latter on average. This shows that the B&B

algorithm is more efficient than the GA-based method; and iii) The CPU times for both the proposed methods show an increasing trend as the problem size increases. Moreover, for each given  $|C|$  (resp.  $|D|$ ), the CPU time for the B&B algorithm increases with  $|D|$  (resp.  $|C|$ ), which implies that the problem’s complexity increases with  $|C|$  and  $|D|$ .

Table 4: Computational results for instances with  $|P| = 5$

Group	$ C $	$ D $	B&B algorithm	GA based method		Ratio
			CPU time (s)	Deviation (%)	CPU time (s)	
10	3	20	33.47	<b>0.00</b>	150.00	0.22
11		50	36.75	<b>0.00</b>	159.60	0.23
12		100	48.19	<b>0.00</b>	201.60	0.24
13	5	20	92.40	<b>0.00</b>	349.20	0.26
14		50	100.97	<b>0.00</b>	391.20	0.26
15		100	131.30	0.01	465.80	0.28
16	7	20	220.23	<b>0.00</b>	783.10	0.28
17		50	249.01	<b>0.00</b>	1489.37	0.17
18		100	909.29	<b>0.00</b>	3888.42	0.23
<i>Average</i>			<i>202.40</i>	<i>0.00</i>	<i>875.37</i>	<i>0.23</i>

Table 4 presents the computational results for 9 groups of instances with  $|P|$  increased to 5,  $|C| = 3, 5, 7$  and  $|D| = 20, 50, 100$ , from which we can also conclude that: i) the proposed B&B algorithm can find optimal solutions for all the tested instances within a short time (i.e., 910 s), which confirms its efficiency in optimally solving the studied problem; ii) the average deviation varies from 0.00 to 0.01% with its average value being about 0.00, and its value is 0.00 for 8 out of 9 groups, which again shows that the proposed GA-based method is able to find high-quality near-optimal solutions; and iii) the CPU time spent by the B&B algorithm increases from 33.47 to 909.29 s with its average value being 202.40 s, while that of GA-based method varies from 150 to 3888.42 s with its average value being 875.37 s. Moreover, the former is less than the latter over all groups and increases more moderately than the latter with the problem size. These results indicate that the proposed B&B algorithm is superior to the GA-based method in solving the instances with a larger  $|P|=5$ .

Table 5 summarizes the computational results for the instances with  $|P|$  increasing to 10, from which we can further find that: i) the computational time for the B&B algorithm increase as  $|P|$  increases from 3 to 10 for each given  $|C|$  and  $|D|$ . Take groups 9, 18 and 27 as an example, all groups are with  $|C|=7$  and  $|D|=100$ , but  $|P|$  are 3, 5 and 10, respectively, the CPU times are 102.82 s, 909.29 s, and 6900.77 s, respectively. This indicates that the complexity of the problem also increases with  $|P|$ ; ii) the GA-based

Table 5: Computational results for instances with  $|P| = 10$ 

Group	$ C $	$ D $	B&B algorithm	GA based method		Ratio
			CPU time (s)	Deviation (%)	CPU time (s)	
19	3	20	299.17	0.01	390.00	0.77
20		50	321.84	<b>0.00</b>	366.00	0.88
21		100	400.71	<b>0.00</b>	488.00	0.82
22	5	20	645.84	0.01	737.10	0.88
23		50	697.79	0.18	862.00	0.81
24		100	840.97	0.04	814.50	1.03
25	7	20	3268.02	<b>0.00</b>	3356.50	0.97
26		50	4660.38	<b>0.00</b>	5127.20	0.91
27		100	6900.77	<b>0.00</b>	7437.60	0.93
<i>Average</i>			<i>2003.94</i>	<i>0.03</i>	<i>2175.43</i>	<i>0.92</i>

method can obtain close to optimal solution as the deviation varies between 0 and 0.18% with its average value being 0.03%; and iii) the CPU times for both the proposed methods have similar increasing trends as the size of the problem increases.

By comparing all the results in Tables 3-5, we summarize the conclusions below.

- i) The computational time spent by the proposed B&B algorithm increases more rapidly with the number of potential parking facilities (i.e.,  $|P|$ ) compared with that with the number of existing parking facilities  $|C|$  and demand zones  $|D|$ . This may be because the branching procedure of the B&B algorithm is determined by the number of potential location sites and most computational time of the algorithm is spent on solving the integer linear program for computing an upper bound at each explored node; and,
- ii) The computational time spent by the GA-based method is more than that of the B&B algorithm over all groups which may be because the CPU time for the GA-based method depends considerably on the number of existing parking facilities and demand zones, while the B&B algorithm depends relatively slightly on them. However, note that the value of the “Ratio” has an increasing trend as  $|P|$  increases. For example, on average, the ratio is 0.28 for  $|P|=3$  while it is 0.92 for  $|P|=10$ . These results imply that the GA-based method performs more stably than the B&B algorithm as the problem size increases.

In order to further explore the influence of the number of potential parking sites  $|P|$  on the performance of both the proposed algorithms, we also conduct a series of computational experiments that vary the potential number of sites for the location of new parking facilities from 5, 10, 20, 50 to 100, with the number of demand zones fixed at 50 and the number of existing parking facilities fixed at 3. The computational results are summarized

in Table 6.

Table 6: Computational results for large-sized instances with  $|P|=5, 10, 20, 50$  and  $100$

Group	$ P $	B&B algorithm	GA based method		Ratio
		CPU time (s)	Deviation (%)	CPU time (s)	
28	5	35.46	<b>0.00</b>	140.40	0.25
29	10	280.60	<b>0.00</b>	377.20	0.74
30	20	4347.98	0.01	964.00	4.51
31	50	18260.30	0.10	2256.00	8.09
32	100	35223.54	0.05	9822.62	3.59
<i>Average</i>		<i>11699.58</i>	<i>0.03</i>	<i>2712.04</i>	<i>4.29</i>

From Table 6, we can conclude that: i) the proposed B&B algorithm is able to obtain optimal solutions for instances with up to 100 potential parking facilities within the given time. The deviation varies between 0.00 and 0.1% with its average value being 0.03%, which again indicates the high quality of the solutions obtained by the GA-based method; and ii) the CPU time for the B&B algorithm increases from 35.46 to 35223.54 s with its average value being 11629.58 s, while that for the proposed GA-based method varies between 140.40 and 9822.62 s. On average, the former is 4.29 times the latter. Moreover, the CPU time for the B&B algorithm increases sharply with the number of potential parking facilities, while that of the GA-based method increases moderately and remains acceptable even for instances with a large number of potential parking sites  $|P|=100$ . This shows that the GA-based method is superior to the B&B algorithm in terms of computational efficiency for solving large-scale instances with a large number of potential parking sites.

To sum up, the B&B algorithm is quite efficient and suitable for solving small- and medium-sized problems, while the GA-based method is more efficient for solving large-sized problems. The GA-based method is therefore the preferred method (vs. B&B method) to solve practical sized problems.

## 5. Conclusions

This paper investigates an emerging research problem of optimal facility location and associated quality design for a new-entrant shared parking firm when facing competition from existing service providers. The service quality is differentiated by emerging technologies such as IoT, sensor system, charging system, and RFID parking facility surveillance system. These technologies offer vehicle tracking and tracing capability as

well as information on connected parking facilities. On-site electric charging devices can be designed to offer various levels of charging speed/time that is considered as one of the key quality indicators for electric vehicles. Parking facility surveillance based on video and sensor improves parking security service quality. Automated checkpoints with RFID and facilitated mobile payments improve convenience and help reduce customer wait time.

We then introduce a new discrete competitive facility location and design that incorporates the responses of the facilities that are already in the market and limited budget constraints and formulates its decision model. To solve this problem, an iterative solution framework consisting of quality determination and location is developed. In the quality determination phase, the competitive decision process occurring among facilities is modeled as a non-cooperative game and the best qualities of the new facilities and existing ones are identified through Nash equilibrium. A B&B algorithm and a GA are developed to determine the locations of new shared parking facilities. Computational results on 320 randomly generated instances show that the proposed B&B algorithm is able to find optimal solutions in reasonable time for small-sized problems with relatively fewer number of potential parking sites, while the proposed GA-based approach is more efficient and suitable for solving large-sized problems with a large number of potential parking sites.

In this paper, we consider a model of available shared parking spaces at the macro level without considering the flow of patrons at the microscopic level where each patron needs to make decisions based on availability (such as choosing the second-best choice). Integrating the consumers' route decision at the micro level is another future research question to extend this work. The pricing of parking services can be designed in the model to further explain the dynamics of competition in the shared parking market. Few articles on parking facility decision models have considered pricing. Our literature review also shows a void in research on parking pricing that is related to parking distance, capacity, and quality. An immediate future research can be directed at investigating shared-parking pricing as an extension of this work. In this paper, we also assume that the new shared parking firm must possess facilities from the market while the competitors may also enter the market. Consequently, a future extension of this research is to take into account the actions of competitor's new facility decision. From another point of

view, shared parking could lead to a reduction in pollution. Thus, future research of green issues in shared parking is practically important [48]. The development of effective methods for the proposed and extended problems [46, 49–55] is another future research topic.

## Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants 71701049 and 71571061, in part by the Natural Science Foundation of Fujian Province, China under Grant 2018J05120.

## References

- [1] L. Chen, J. Olhager, O. Tang, Manufacturing facility location and sustainability: A literature review and research agenda, *International Journal of Production Economics* 149 (2014) 154–163.
- [2] J. Fernández, B. Pelegri, F. Plastria, B. Tóth, et al., Solving a huff-like competitive location and design model for profit maximization in the plane, *European Journal of Operational Research* 179 (3) (2007) 1274–1287.
- [3] B. Tóth, J. Fernández, B. Pelegrín, F. Plastria, Sequential versus simultaneous approach in the location and design of two new facilities using planar huff-like models, *Computers & Operations Research* 36 (5) (2009) 1393–1405.
- [4] J. Redondo, J. Fernández, I. García, P. Ortigosa, Sensitivity analysis of a continuous multifacility competitive location and design problem, *Top* 17 (2) (2009) 347–365.
- [5] J. L. Redondo, A. Arrondo, J. Fernández, I. García, P. M. Ortigosa, A two-level evolutionary algorithm for solving the facility location and design (1|1)-centroid problem on the plane with variable demand, *Journal of global optimization* 56 (3) (2013) 983.
- [6] J. Fernández, S. Salhi, G. Boglárka, et al., Location equilibria for a continuous competitive facility location problem under delivered pricing, *Computers & Operations Research* 41 (2014) 185–195.
- [7] J. L. Redondo, J. Fernández, J. D. Á. Hervás, A. G. Arrondo, P. M. Ortigosa, Approximating the pareto-front of a planar bi-objective competitive facility location and design problem, *Computers & Operations Research* 62 (2015) 337–349.
- [8] J. Fernández, G. Boglárka, J. L. Redondo, P. M. Ortigosa, A. G. Arrondo, et al., A planar single-facility competitive location and design problem under the multi-deterministic choice rule, *Computers & Operations Research* 78 (2017) 305–315.



- [9] R. Aboolian, O. Berman, D. Krass, Competitive facility location and design problem, *European Journal of Operational Research* 182 (1) (2007) 40–62.
- [10] M. G. Ashtiani, A. Makui, R. Ramezani, A robust model for a leader–follower competitive facility location problem in a discrete space, *Applied Mathematical Modelling* 37 (1) (2013) 62–71.
- [11] W. Shan, Q. Yan, C. Chen, M. Zhang, B. Yao, X. Fu, Optimization of competitive facility location for chain stores, *Annals of Operations Research* (2017) 1–19.
- [12] Y. Zhang, Designing a retail store network with strategic pricing in a competitive environment, *International Journal of Production Economics* 159 (2015) 265–273.
- [13] M. Nakanishi, L. G. Cooper, Parameter estimation for a multiplicative competitive interaction model: least squares approach, *Journal of marketing research* (1974) 303–311.
- [14] H. Hotelling, Stability in competition, *The Economic Journal* 39 (153) (1929) 41–57.
- [15] Z. Drezner, Competitive location strategies for two facilities, *Regional Science and Urban Economics* 12 (4) (1982) 485–493.
- [16] F. Plastria, Gbsss: the generalized big square small square method for planar single-facility location, *European Journal of Operational Research* 62 (2) (1992) 163–174.
- [17] T. Drezner, Optimal continuous location of a retail facility, facility attractiveness, and market share: an interactive model, *Journal of retailing* 70 (1) (1994) 49–64.
- [18] R. Suárez-Vega, D. R. Santos-Peñate, P. Dorta-González, Location models and gis tools for retail site location, *Applied Geography* 35 (1) (2012) 12–22.
- [19] T. Drezner, Z. Drezner, S. Salhi, Solving the multiple competitive facilities location problem, *European Journal of Operational Research* 142 (1) (2002) 138–151.
- [20] S. L. Hakimi, On locating new facilities in a competitive environment, *European Journal of Operational Research* 12 (1) (1983) 29–35.
- [21] S. Hakimi,  $p$ -median theorems for competitive locations, *Annals of operations research* 6 (4) (1986) 75–98.
- [22] D. L. Huff, Defining and estimating a trading area, *The Journal of Marketing* (1964) 34–38.
- [23] D. L. Huff, A programmed solution for approximating an optimum retail location, *Land Economics* 42 (3) (1966) 293–303.
- [24] D. Achabal, Multiloc: a multiple store location model, *Journal of Retailing* 58 (1982) 5–25.
- [25] A. Ghosh, C. S. Craig, A location allocation model for facility planning in a competitive environment, *Geographical analysis* 16 (1) (1984) 39–51.
- [26] A. Ghosh, C. S. Craig, Fransys: a franchise distribution system location model, *Journal of*

- Retailing 67 (4) (1991) 466.
- [27] A. Ghosh, S. L. MacLafferty, Location strategies for retail and service firms, Lexington Books, 1987.
- [28] H. Küçükdin, N. Aras, İ. K. Altinel, A discrete competitive facility location model with variable attractiveness, *Journal of the Operational Research Society* 62 (9) (2011) 1726–1741.
- [29] D. Konur, J. Geunes, Competitive multi-facility location games with non-identical firms and convex traffic congestion costs, *Transportation Research Part E: Logistics and Transportation Review* 48 (1) (2012) 373–385.
- [30] A. Lančinskas, P. Fernández, B. Pelegín, J. Žilinskas, Improving solution of discrete competitive facility location problems, *Optimization Letters* 11 (2) (2017) 259–270.
- [31] H. A. Eiselt, G. Laporte, Sequential location problems, *European Journal of Operational Research* 96 (2) (1997) 217–231.
- [32] H. A. Eiselt, G. Laporte, J.-F. Thisse, Competitive location models: A framework and bibliography, *Transportation Science* 27 (1) (1993) 44–54.
- [33] T. Drezner, A review of competitive facility location in the plane, *Logistics Research* 7 (1) (2014) 114.
- [34] M. Ashtiani, Competitive location: a state-of-art review, *International Journal of Industrial Engineering Computations* 7 (1) (2016) 1–18.
- [35] R. L. Tobin, T. Miller, T. L. Friesz, Incorporating competitors’ reactions in facility location decisions: a market equilibrium approach, *Location Science* 3 (4) (1995) 239–253.
- [36] N. Saidani, F. Chu, H. Chen, Competitive facility location and design with reactions of competitors already in the market, *European Journal of Operational Research* 219 (1) (2012) 9–17.
- [37] E. M. Hendrix, On competition in a stackelberg location-design model with deterministic supplier choice, *Annals of Operations Research* 246 (1-2) (2016) 19–30.
- [38] B. G. Tóth, K. Kovács, Solving a huff-like stackelberg location problem on networks, *Journal of Global Optimization* 64 (2) (2016) 233–247.
- [39] S. Benati, P. Hansen, The maximum capture problem with random utilities: Problem formulation and algorithms, *European Journal of Operational Research* 143 (3) (2002) 518–530.
- [40] J. F. Nash, Equilibrium points in n-person games, *Proceedings of the National Academy of Sciences of the United States of America* 36 (1) (1950) 48.
- [41] B. Chatterjee, An optimization formulation to compute nash equilibrium in finite games, in:

- Proceeding of International Conference on Methods and MODELS in Computer Science, 2010, pp. 1–5.
- [42] R. Fletcher, Practical methods of optimization, John Wiley & Sons, 2013.
- [43] T. Ibaraki, Theoretical comparisons of search strategies in branch-and-bound algorithms, *International Journal of Computer & Information Sciences* 5 (4) (1976) 315–344.
- [44] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan, Ann Arbor, MI, 1975.
- [45] N. Ilk, M. Brusco, P. Goes, Workforce management in omnichannel service centers with heterogeneous channel response urgencies, *Decision Support Systems* 105 (2018) 13–23.
- [46] N. Kozodoi, S. Lessmann, K. Papakonstantinou, Y. Gatsoulis, B. Baesens, A multi-objective approach for profit-driven feature selection in credit scoring, *Decision Support Systems* 120 (2019) 106–117.
- [47] R. Kuo, L. Lin, Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering, *Decision Support Systems* 49 (4) (2010) 451–462.
- [48] I. Bose, R. Pal, Do green supply chain management initiatives impact stock prices of firms?, *Decision Support Systems* 52 (3) (2012) 624–634.
- [49] I. Bose, K. Altinkemer, Design of a web site for guaranteed delay and blocking probability bounds, *Decision Support Systems* 38 (1) (2004) 131–140.
- [50] I. Bose, E. Eryarsoy, L. He, Multi-period design of survivable wireless access networks under capacity constraints, *Decision Support Systems* 38 (4) (2005) 529–538.
- [51] P. Ravisankar, V. Ravi, G. R. Rao, I. Bose, Detection of financial statement fraud and feature selection using data mining techniques, *Decision Support Systems* 50 (2) (2011) 491–500.
- [52] F. Wang, X. Lai, N. Shi, A multi-objective optimization for green supply chain network design, *Decision Support Systems* 51 (2) (2011) 262–269.
- [53] X. Chen, I. Bose, A. C. M. Leung, C. Guo, Assessing the severity of phishing attacks: A hybrid data mining approach, *Decision Support Systems* 50 (4) (2011) 662–672.
- [54] L. Zhang, K. Mistry, C. P. Lim, S. C. Neoh, Feature selection using firefly optimization for classification and regression models, *Decision Support Systems* 106 (2018) 64–85.
- [55] L. Meng, Q. Kang, C. Han, M. Zhou, Determining the optimal location of terror response facilities under the risk of disruption, *IEEE Transactions on Intelligent Transportation Systems* 19 (2) (2018) 476–486.