3-15-2023

A comparative performance analysis of intelligence-based algorithms for optimizing competitive facility location problems

Vahid Hajipour

Seyed Taghi Akhavan Niaki

Madjid Tavana

Francisco J. Santos-Arteaga

Sanaz Hosseinzadeh

Machine Learning with Applications 11 (2023) 100443

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

A comparative performance analysis of intelligence-based algorithms for optimizing competitive facility location problems ☆

Vahid Hajipour [a,b], Seyed Taghi Akhavan Niaki [a], Madjid Tavana [c,d,*], Francisco J. Santos-Arteaga [e], Sanaz Hosseinzadeh [b]

[a] Department of Industrial Engineering Sharif University of Technology, Tehran, Iran
[b] Department of Industrial Engineering College of Engineering, West Tehran Branch Islamic Azad University, Tehran, Iran
[c] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, USA
[d] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn, Germany
[e] Departamento de Análisis Económico y Economía Cuantitativa, Universidad Complutense de Madrid, Madrid, Spain

ARTICLE INFO

Keywords:
Competitive facility location
Optimization
Computational intelligence
Meta-heuristics

ABSTRACT

Most companies operate to maximize profits and increase their market shares in competitive environments. Since the proper location of the facilities conditions their market shares and profits, the competitive facility location problem (CFLP) has been extensively applied in the literature. This problem generally falls within the class of NP-hard problems, which are difficult to solve. Therefore, choosing a proper solution method to optimize the problem is a key factor. Even though CFLPs have been consistently solved and investigated, an important question that keeps being neglected is how to choose an appropriate solution technique. Since there are no specific criteria for choosing a solution method, the reasons behind the selection approach are mostly unclear. These models are generally solved using several optimization techniques. As harder-to-solve problems are usually solved using meta-heuristics, we apply different meta-heuristic techniques to optimize a new version of the CFLP that incorporates reliability and congestion. We divide the algorithms into four categories based on the nature of the meta-heuristics: evolution-based, swarm intelligence-based, physics-based, and human-based. GAMS software is also applied to solve smaller-size CFLPs. The genetic algorithm and differential evolution of the first category, particle swarm optimization and artificial bee colony optimization of the second, Tabu search and harmony search of the third, and simulated annealing and vibration damping optimization of the fourth are applied to solve our CFLP model. Statistical analyses are implemented to evaluate and compare their relative performances. The results show the algorithms of the first and third categories perform better than the others.

1. Introduction

Competition occurs in many real-world environments and consequently influences the behavior of nature. In evolution, animals and plants compete for food, nests, sunlight, water, etc. For example, some birds compete to find and occupy nesting sites before others. Indeed, competition exists among the animals that need the same environmental resources (Zeigler, 2014). The competition also exists among service facilities operating in a competitive environment providing similar services. These facilities aim to maximize their market shares and profits influenced significantly by choice of their locations. This brings about the so-called competitive facility location problem (CFLP) to find the best locations for the facilities in a market area. Applications of CFLP involve selecting the locations of restaurants, shopping malls, grocery stores, banks, and the like (Berman et al., 2009).

Competitive location selection deals with the problem of locating new facilities to provide a service (or goods) to the customers of a given geographical area where other facilities (competitors) offering the same service are already present or will be established in the future. The new facilities must compete to capture the maximum market share. There are three types of competition in facility location problems: static competition, dynamic competition, and sequential competition. In the first case, some facilities already exist in the market, the competitors'

☆ Dr. Francisco J. Santos-Arteaga is grateful for the support received from the María Zambrano contract from the Universidad Complutense de Madrid financed by the Ministerio de Universidades with funding from the European Union Next Generation program.

* Correspondence to: Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA 19141, United States.

E-mail addresses: hajipour.v@wtiau.ac.ir (V. Hajipour), niaki@sharif.edu (S.T.A. Niaki), tavana@lasalle.edu (M. Tavana), fransant@ucm.es (F.J. Santos-Arteaga), hosseinzade.sanaz@wtiau.ac.ir (S. Hosseinzadeh).

https://doi.org/10.1016/j.mlwa.2022.100443
Received 26 September 2022; Received in revised form 11 December 2022; Accepted 16 December 2022
Available online 20 December 2022

information is accessible, and the existing facilities do not react to the new facilities. In the second type, competitors react to new facilities and constantly optimize their location. This game continues until a Nash equilibrium is reached, where no moves can improve the conditions of players. Finally, in the third scenario, also called the Stackelberg game, there are two players named leader and follower, and the first player makes their decision based on the decision of the second player.

The selection of an optimization methodology is one of the most important features of CFLPs (Fletcher, 1972). Optimization methodologies can be categorized into the classes of classical (exact) methodologies such as branch and bound (Beresnev & Mel'nikov, 2016; Drezner & Drezner, 2004; Fernández et al., 2014) that give optimal solutions, or heuristic algorithms (Drezner et al., 2007; Fernández et al., 2014; Rohaninejad et al., 2017), and meta-heuristic algorithms (Shan et al., 2019; Zarrinpoor & Seifbarghy, 2011; Zhang et al., 2016) that can generally find near-optimal solutions of more complex CFLPs.

CFLPs belong to the class of NP-hard problems (Fernández et al., 2014), for each of which some of the optimization mentioned above methodologies perform better than others. This chooses an appropriate methodology, a very important quality. While in previous studies, the problem was solved using one or more optimization methodologies, less attention was given to selecting an appropriate optimization methodology. There are no specific criteria in the literature for selecting the right solution approach, and the reason for choosing one is unclear. Therefore, in this study, several optimization methodologies that have been applied to solve CFLPs are analyzed to distinguish them in terms of their performances.

Our objective is to shed light on selecting an appropriate solution approach. We propose a new CFLP that will be solved using different methodologies to fulfill this aim. The new CFLP involves a static competition setting where facility congestion and reliability are considered. The objective function of this problem is to minimize the total cost of the new facilities, including the installation cost, the customers' traveling, waiting costs, and service costs. That is, we consider the main set of factors affecting the choices of customers, defining a real competitive environment where facilities may not always be available.

The remainder of the paper is organized as follows. Section 2 surveys and classifies related CFLP works, reviewing the different solution methods applied. The problem, notations, parameters, decision variables, and the mathematical formulation of the problem are introduced in Section 3. Nine solution algorithms are applied in Section 4 to solve the problem. Even though the parameters of the algorithms are calibrated using the Taguchi approach, several test problems are solved in Section 5 to compare their performances statistically. Finally, the conclusion and recommendations for future research are presented in Section 6.

2. Related works

The related works reviewed in this section are categorized into two classes, including (1) CFLPs and (2) the optimization methods applied to solve CFLPs.

2.1. The literature on CFLPs

Hotelling (1990) was the first to introduce competition in a market with two competing firms in the facility location problem. Later, the CFLP model was employed in different real-world problems, including location determination of clothing stores (Huff, 1964), supermarkets (Bell et al., 1998), and shopping malls (Drezner & Drezner, 2002), among others. Due to the importance of market competition, CFLPs attracted increasing attention, and different conditions, such as congestion and reliability, were considered. For instance, ReVelle (1986) introduced a competitive location–allocation problem. Aboolian et al. (2009) worked on a competitive location–allocation problem of web services and applied a queuing system to model facility congestion. Zarrinpoor and Seifbarghy (2011) developed a CFLP that considers capacity constraints and congestion in facilities. Shiode et al. (2012) followed a Nash equilibrium approach to propose a CFLP with three competitors. Snyder and Daskin (2005) introduced a location problem that considers the reliability of the different facilities. Hajipour et al. (2014) proposed a location–allocation problem where the failure probability of the facilities was considered. Similarly, Zhang et al. (2016) defined a CFLP that considers the failure probability of the facilities. Zarrinpoor et al. (2016) presented a location–allocation problem addressing the reliability of congested facilities.

The introduction of constraints in CFLPs is another important topic tackled by many researchers. For instance, Beresnev and Mel'nikov (2016) proposed a CFLP with bounded capacities of the facilities. Qi et al. (2017) developed a CFLP model with a limited distance for services. They assumed that people only patronize facilities within a range they feel convenient. Wang and Chen (2017) addressed a CFLP by assuming that attractiveness is a function of the distance coverage of a facility.

Table 1 summarizes some of the CFLP literature, where the articles are classified in terms of competition, location space, congestion, queuing system, allocation, and reliability. As seen in Table 1, few papers consider congestion in CFLPs. Moreover, no paper addresses the reliability of facilities in CFLPs. In real-world problems, nonetheless, there is congestion in some facilities, and the facilities sometimes are not available for service. As a result, a CFLP with facility congestion and reliability is considered in this paper to analyze the solution performance of several intelligence-based algorithms.

2.2. Optimization approaches

In this section, several optimization methodologies applied in different CFLPs are reviewed. Generally, the optimization approaches are classified into three categories: exact, approximation, and heuristic. While exact methods cannot find optimal solutions for all types of problems of all sizes, heuristic methods that involve special heuristics and meta-heuristics can find near-optimum solutions to all problems. Approximation approaches are also able to find near-optimal solutions (Chong & Zak, 2013). It should be noted that Hotelling (1990), Huff (1964), Bell et al. (1998), and Drezner and Drezner (2002), who were the first to work on CFLPs, did not apply any optimization approach to find a solution. They just analyzed the CFLP problem and introduced methods and concepts such as facility attraction and market share. Nevertheless, there are many works available in the literature that apply various optimization methodologies to solve CFLPs. The next four sub-sections review CFLP papers based on the methods used to solve the related problems.

2.2.1. Exact and approximation approaches

Among the many research works in which exact methodologies were used to solve CFL problems, Drezner and Drezner (2004) proposed an efficient branch & bound (B&B) algorithm to solve the Huff gravity-based model. Aboolian et al. (2007) used tangent-line approximation (TLA), greedy algorithm (GRA), and steepest ascent heuristic (SAH). Later, Aboolian et al. (2009) decomposed the CFLP into two subproblems and applied CPLEX to solve them. Shiode et al. (2012) employed TLA to solve the CFLP with three competitors. Fernández et al. (2014) utilized a B&B method to solve various CFLPs in medium sizes. Another application of the B&B method to solve the CFLP is given by Beresnev and Mel'nikov (2014). More recently, Gentile et al. (2018) employed the branch & cut (B&C) approach to optimize a CFLP in which competition was modeled using the Stackelberg game.

Since exact methods cannot find the optimal solution to some problems, especially large-size ones, many researchers have used heuristic approaches. To name just a few researchers who have applied approximate algorithms, Beresnev (2009) proposed a method to find upper

Table 1
Some related works on CFLPS.

References	Competition			Location space			Congestion	Queuing system				Allocation	Reliability
	Static	Nash	Stackelberg	Continuous	Discrete	Network		M/M/1	M/G/1	M/M/m	M/M/m/k		
Drezner and Drezner (2004)	✓					✓							
Aboolian et al. (2007)			✓		✓								
Drezner et al. (2007)	✓			✓									
Aboolian et al. (2009)	✓				✓		✓			✓		✓	
Zarrinpoor and Seifbarghy (2011)	✓				✓		✓				✓		
Shiode et al. (2012)			✓			✓							
Ashtiani et al. (2013)			✓		✓								
Fernández et al. (2014)		✓		✓									
Hajipour et al. (2014)					✓		✓			✓		✓	✓
Panin et al. (2014)		✓	✓		✓								
Beresnev and Mel'nikov (2016)			✓		✓								
Hajipour, Farahani, and Fattahi (2016)					✓		✓				✓	✓	
Hajipour, Fattahi, et al. (2016)					✓		✓	✓				✓	
Ivanov and Morozova (2016)			✓		✓								
Zarrinpoor et al. (2016)					✓		✓		✓			✓	✓
Zhang et al. (2016)			✓		✓							✓	✓
Lančinskas et al. (2017)	✓				✓								
Qi et al. (2017)			✓		✓								
Rohaninejad et al. (2017)		✓			✓								
Wang and Chen (2017)	✓				✓								
Shan et al. (2019)		✓			✓								
This Research	✓				✓		✓			✓		✓	✓

bounds for a two-level CFLP model. He then applied a polynomial-time algorithm to find approximate solutions. Beresnev and Mel'nikov (2011) utilized an approximate algorithm in which a local search was applied to improve the solutions. Panin et al. (2014) presented a two-level CFLP model and proposed two approximate algorithms based on alternating heuristics and local search to optimize the model. Rohaninejad et al. (2017) developed an approximate algorithm to optimize a bi-objective CFLP. More recently, Kung and Liao (2018) introduced an approximate algorithm based on demand function approximation, linear relaxation, decomposition of the problem into two sub-problems, and sorting. To illustrate the average performance of the proposed algorithm, they solved the problem using both a genetic algorithm and CPLEX. Papers that have applied heuristic approaches are also discussed in the next section.

2.2.2. Heuristic algorithms

Drezner et al. (2007) proposed a novel CFLP and applied a greedy algorithm (GRA) to solve the problem. Konur and Geunes (2012) developed two heuristic algorithms based on random search (RSM) and self-adaptive projection (SAPM) methods to define a Stackelberg equilibrium location decision. Ashtiani et al. (2013) used a penalty function algorithm (PFA) to solve a competitive facility location model. Sasaki et al. (2014) presented a Stackelberg hub-arc location algorithm (SHALA). Ivanov and Morozova (2016) employed a local search algorithm (LS) to optimize a competitive facility location model. Fernández, Pelegrín, et al. (2017) used both the B&B algorithm and the Weiszfeld-like algorithm (WA) to optimize a CFLP with the static competition.

2.2.3. Meta-heuristic algorithms

As previously stated, meta-heuristic algorithms can be classified into four general classes, including (1) evolution-based, (2) human-based, (3) swarm-intelligence-based, and (4) physics-based approaches (Du & Swamy, 2016). Algorithms based on the principle of Darwin's evolutionary theory (such as evolutionary algorithms) are included within the evolution-based class. Human-based algorithms (such as Tabu search) include algorithms based on human behavior or characteristics. Algorithms based on swarm intelligence (such as particle swarm optimization), namely, the social behavior of animals, also belong to a class of intelligence-based algorithms. Finally, algorithms inspired by physical laws (such as simulated annealing) are grouped in the physics-based class. We review the literature on the use of meta-heuristic algorithms to solve CFLPs based on the four categories above.

Redondo et al. (2015) presented a bi-objective model for the CFLP and proposed a new evolutionary multi-objective optimization algorithm to solve it. They compared the results obtained by their solution algorithm to the ones derived based on a B&B algorithm and a non-dominated sorting genetic algorithm (NSGA). Konak et al. (2017) developed a two-objective model for the CFLP and applied a multi-objective genetic algorithm to optimize it. Wang and Chen (2017) presented a two-objective competitive facility location model in which the attractiveness of each facility was determined through the coverage radius. They used a NSGA to solve it. Fernández et al. (2019) employed an evolutionary and B&B algorithm to optimize a CFLP.

Zarrinpoor and Seifbarghy (2011) optimized a competitive facility location model using a genetic algorithm (GA) and a Tabu search algorithm (TSA). They showed the overall better performance of TSA compared to GA. Drezner et al. (2011) introduced a CFLP and optimized it using a TSA. They compared the result with the ones obtained using B&B and GRA. Küçükaydın et al. (2012) utilized a TSA to optimize a bi-objective CFLP and applied an exact method to evaluate the performance of their proposed meta-heuristic. Drezner et al. (2012) employed a TSA to optimize a competitive location model and compared its results with the ones of GRA and B&B. Drezner et al. (2015) proposed a TSA and a B&B to solve a two-level competitive facility location model. Finally, Biesinger et al. (2016), Shan et al. (2019), and Qi et al. (2017) used TSA to optimize their competitive facility location models.

MirHassani et al. (2015) developed several versions of the particle swarm optimization algorithm (PSO) to optimize a competitive model. They applied the Taguchi method to adjust the parameters of the algorithms and used a B&B algorithm to compare and analyze the results obtained. Nasiri et al. (2018) applied PSO and GA to optimize a competitive location model. They also used the Taguchi approach to tune the parameters of their algorithm and GAMS software to derive their results.

Redondo et al. (2009) applied the simulated annealing (SA) algorithm, an evolutionary algorithm, and a Weiszfeld-like algorithm (WA) to optimize a competitive facility location model. Ghaffarinasab et al. (2018) recently developed several versions of SA with different operators to optimize single-level and two-level CFLP models. They solved the problem using CPLEX software to validate and compare the results.

Table 2 summarizes several CFLP works alongside the optimization approaches used to solve them. The solution approaches are explicitly described in Table 3.

As shown in Table 2, meta-heuristics have been applied more frequently than other optimization approaches to solve CFLPs in recent years. Fig. 1 provides a graphical illustration of this claim. Moreover,

Table 2
Research on CFLPs and classification of the optimization methods applied.

References	Competition	Optimization methods				
		Exact	Approximate	Heuristic	Meta-heuristic	Commercial solver
Yang and Wong (2000)	Static	✓				
Fischer (2002)	Nash	✓		✓		
Drezner and Drezner (2004)	Static	✓		✓		
Suárez-Vega et al. (2004)	Stackelberg			✓	✓	
McGarvey and Cavalier (2005)	Static	✓		✓		✓
Aboolian et al. (2007)	Stackelberg	✓		✓		✓
Drezner et al. (2007)	Static			✓		
Aboolian et al. (2008)	Static					✓
Marianov et al. (2008)	Static				✓	
Redondo et al. (2008)	Static				✓	
Beresnev (2009)	Stackelberg		✓			
Lee and O'Kelly (2009)	Nash			✓		
Redondo et al. (2009)	Static			✓	✓	
Beresnev and Mel'nikov (2011)	Stackelberg		✓	✓		
Drezner et al. (2011)	Static	✓		✓	✓	
Küçükaydın et al. (2011)	Static	✓		✓		✓
Pelegrín-Pelegrín et al. (2011)	Nash					✓
Zarrinpoor and Seifbarghy (2011)	Static				✓	
Konur and Geunes (2012)	Stackelberg			✓		
Küçükaydın et al. (2012)	Stackelberg	✓			✓	
Shiode et al. (2012)	Stackelberg		✓			
Drezner et al. (2012)	Static	✓			✓	
Saidani et al. (2012)	Nash	✓	✓			
Ashtiani et al. (2013)	Stackelberg			✓		
Lüer-Villagra and Marianov (2013)	Static				✓	
Beresnev and Mel'nikov (2014)	Stackelberg	✓				
Panin et al. (2014)	Stackelberg		✓			
Sasaki et al. (2014)	Stackelberg			✓		
Fernández et al. (2014)	Nash	✓		✓		
Drezner et al. (2015)	Stackelberg	✓			✓	
MirHassani et al. (2015)	Stackelberg	✓			✓	✓
Lančinskas et al. (2015)	Static				✓	
Redondo et al. (2015)	Static	✓			✓	
Beresnev and Mel'nikov (2016)	Stackelberg	✓		✓		
Biesinger et al. (2016)	Stackelberg			✓	✓	
Ivanov and Morozova (2016)	Stackelberg			✓		
Rahmani (2016)	Stackelberg	✓				✓
Zhang et al. (2016)	Stackelberg	✓			✓	
Sadjadi et al. (2016)	Static				✓	
Fernández, Tóth, et al. (2017)	Static	✓		✓	✓	
Fernández, Pelegrín, et al. (2017)	Stackelberg			✓	✓	
Konak et al. (2017)	Stackelberg				✓	
Niknamfar et al. (2017)	Stackelberg				✓	✓
Qi et al. (2017)	Stackelberg				✓	
Bilir et al. (2017)	Static					✓
Lančinskas et al. (2017)	Static				✓	
Wang and Chen (2017)	Static				✓	
Rohaninejad et al. (2017)	Nash		✓			✓
Bagherinejad and Niknam (2018)	Stackelberg	✓			✓	✓
Beresnev and Mel'nikov (2018a)	Stackelberg	✓				
Beresnev and Mel'nikov (2018b)	Stackelberg	✓				
Gentile et al. (2018)	Stackelberg	✓				
Ghaffarinasab et al. (2018)	Stackelberg				✓	✓
Nasiri et al. (2018)	Stackelberg				✓	✓
Kung and Liao (2018)	Static		✓			
Ljubić and Moreno (2018)	Static	✓				✓
Fernández et al. (2019)	Static	✓			✓	
Shan et al. (2019)	Nash				✓	

some meta-heuristic methods display a more satisfactory performance. As there is no specific guideline in the literature to select a proper meta-heuristic, this paper aims to shed light on the selection process of the meta-heuristics used to solve a novel CFLP introduced in the next section. To this end, different meta-heuristic algorithms are applied to solve the problem, and their relative performances are compared.

Table 3
Research works that applied meta-heuristics.

References	Meta-Heuristics			
	Evolution based	Swarm intelligence based	Human-based	Physics-based
Suárez-Vega et al. (2004)			✓	
Marianov et al. (2008)			✓	
Redondo et al. (2008)	✓			
Redondo et al. (2009)	✓			✓
Drezner et al. (2011)			✓	
Küçükaydın et al. (2011)			✓	
Zarrinpoor and Seifbarghy (2011)	✓		✓	
Drezner et al. (2012)			✓	
Lüer-Villagra and Marianov (2013)	✓			
Drezner et al. (2015)			✓	
Lančinskas et al. (2015)	✓			
MirHassani et al. (2015)		✓		
Redondo et al. (2015)	✓			
Biesinger et al. (2016)	✓		✓	
Sadjadi et al. (2016)		✓		
Zhang et al. (2016)	✓		✓	
Fernández, Tóth, et al. (2017)	✓			
Fernández, Pelegrín, et al. (2017)	✓			
Konak et al. (2017)	✓			
Lančinskas et al. (2017)	✓			
Niknamfar et al. (2017)	✓			
Qi et al. (2017)			✓	
Wang and Chen (2017)	✓			
Bagherinejad and Niknam (2018)			✓	
Ghaffarinasab et al. (2018)				✓
Kung and Liao (2018)	✓			
Nasiri et al. (2018)	✓	✓		
Fernández et al. (2019)	✓			
Shan et al. (2019)			✓	

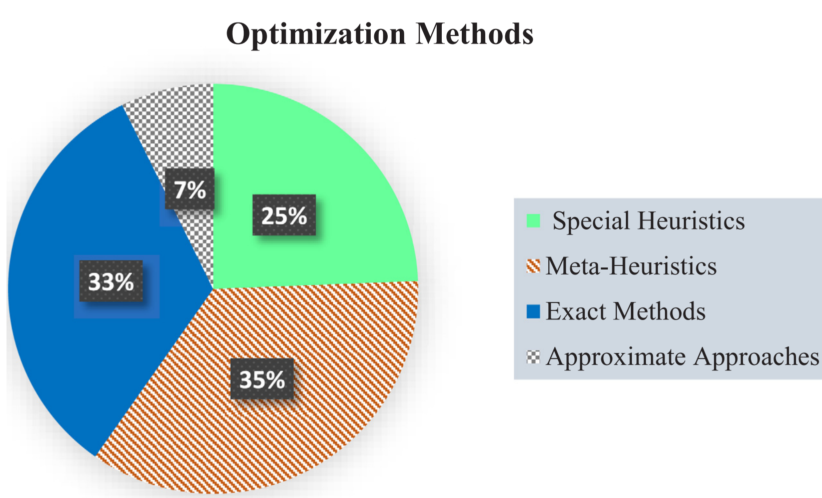

Fig. 1. Usage of different optimization methods to solve FCPLs.

2.3. Current research trends

We summarize below the main lines of research on CFLPs currently being developed in the literature. Most of the models described in this section are solved via heuristic and metaheuristic techniques, highlighting their relevance within the main branches of the literature on CFLP.

2.3.1. Uncertain customer behavior

Beresnev and Melnikov (2020) assumed that customers followed a binary decision rule summarized within a list of possible scenarios. Given this information, the Stackelberg leader decided before a scenario was realized. Both events led to a decision being made by the follower. Santos-Peñate et al. (2020) integrated the linear programs of the leader and the follower into an algorithm considering binary and S-shaped customer choice rules. Lančinskas, Žilinskas, et al. (2020) defined a population-based heuristic algorithm to solve a discrete CFLP with a binary customer choice rule and an asymmetric objective function. Ma et al. (2020) introduced heterogeneity in the choice rules determining customers' behavior and relative proportions. Yu (2020) analyzed a CFLP where customers' behavior was uncertain and categorized into two main types. Yu (2022) further considered uncertain demand types in the definition of a location problem. Lin and Tian (2021a) defined

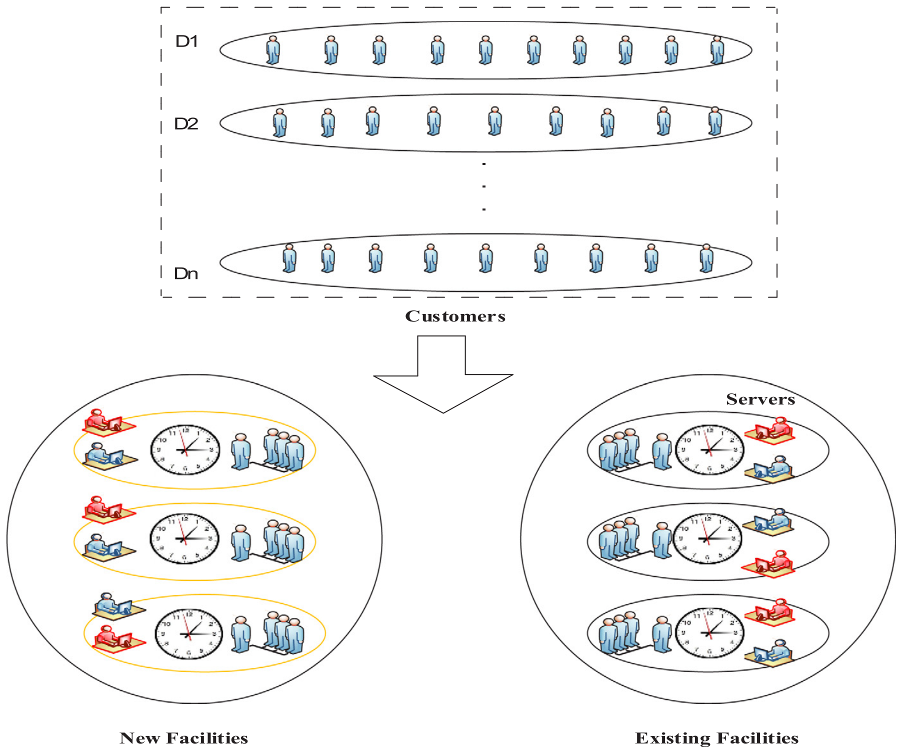

Fig. 2. A congested competitive facility location problem.

a generalized setting where customers followed either the proportional or the partially binary choice rule.

2.3.2. Stochastic demand

The previous models share intuition with those presented in this subsection regarding the uncertain behavior of consumers. For instance, Ahmadi and Ghezavati (2020) developed a sustainable CFLP where a chance-constrained model was used to formalize the potential dissatisfaction of customers when having to wait too long to receive a given service. Mai and Lodi (2020) studied a CFLP where a random utility function determined the selection of facilities. Rahmani and Hosseini (2021) defined a Stackelberg inventory CFLP where order quantities took place in a stochastic environment. Basciftci et al. (2021) considered a distributionally robust facility location problem where location decisions were strategically determined by the moments of stochastic customer demand. Qi et al. (2022) formalized the sequential opening of facilities when a probabilistic choice model determines demand.

2.3.3. Attractiveness of the facilities

The attractiveness of the facilities to the potential customers located within a reasonable distance constitutes one of the main lines of research within the CFLP literature. Küçükaydın and Aras (2020) defined a model where the preferences of customers were represented as probabilities. Customers choose the closest facility, whose capacity to satisfy the customer is based on two distances determined by the attributes of the customers and the type of facility. Levanova and Gnusarev (2020) studied a model where customers chose facilities to satisfy their demands according to the location and type of facility. Marianov et al. (2020) introduced comparison-shopping in CFLP where consumers visited multiple stores selling substitute products before making a purchase decision.

Lin and Tian (2021b) considered a CFLP where firms maximized profits by selecting a facility–customer attractiveness level. Similarly, Lin and Tian (2021c) applied a mixed-integer quadratic conic approach to find exact solutions to a CFLP whose objective was to define the location and attractiveness of the facilities that maximized profits. Wang and Chen (2021) designed a model where the attractiveness of a facility was determined through a distance-based coverage of the demand points it served. Latifi et al. (2022) analyzed a CFLP where the gains of the leader and the followers' losses were conditioned by the capacity of the former to correctly foresight the latter's response.

2.3.4. Rankings, supply chain interactions, and regret

The final branch of CFLP models currently developed in the literature is more heterogeneous and focuses on scenarios extending the standard analysis framework into various research lines. For instance, Lančinskas, Fernández, et al. (2020) defined random search algorithms based on ranking candidate facility locations. Esmaeili and Hamedani (2022) studied a Stackelberg game environment in two supply chains composed of suppliers, distributors, and customers, where lead-time was considered a competitive factor. Finally, Li et al. (2020) assumed that the leader does not know the follower's response when making a location decision. They categorized the potential responses in terms of the number of new facilities and defined a minimax regret model to minimize the maximum potential loss of the leader.

2.4. Interactions with machine learning techniques

The main features defining CFLPs are also the object of analysis when implementing machine learning techniques to decision models dealing with spatial selection and demand evaluation. For instance, Stepinski and Dmowska (2022) predicted a map of segregated neighborhoods through an empirical model generated by a machine learning algorithm. Adamu et al. (2021) defined a hybrid algorithm based on chaotic crow search and particle swarm optimization to solve a feature selection problem that used k-Nearest Neighbor as a classifier. Di Caprio and Santos-Arteaga (2022) designed different sequential evaluation processes determined by the information retrieval capacity of consumers and analyzed the ability of machine learning techniques to categorize the complexity of these processes correctly. Finally, Asani et al. (2021) considered the opinions of customers and the extrapolation

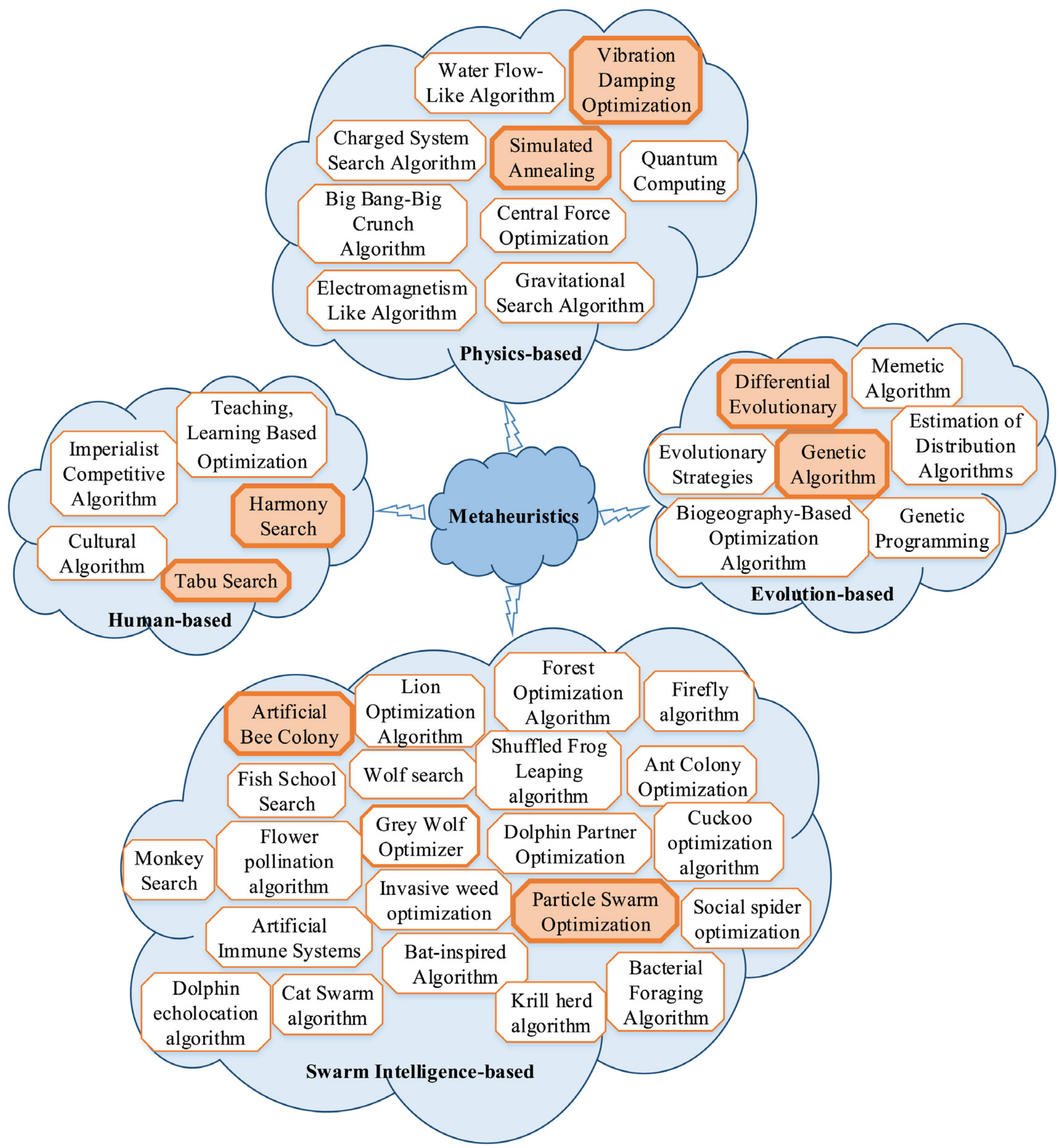

Fig. 3. Classification of meta-heuristic algorithms.

of their food preferences through a semantic approach to enhance the quality of a restaurant recommender system. These models, along with many others applying machine learning techniques, can be easily adapted and implemented within the formal structure of CFLPs to incorporate additional features and extend the framework of analysis.

3. Problem description

To have closer to reality CFLP and better location decisions, the reliability of the chosen facilities and the customer congestion in front of these facilities are investigated in the current research. The competition type is assumed to be static, i.e., an organization intends to establish several new facilities in a competitive market with some existing operating facilities. Facilities are congested and behave as an $M/M/m$ queue system. Each facility has several servers that may not be available with a certain probability when customers visit them. At the same time, the facility will lose customers when all the servers are unavailable. Customers often prefer more available facilities. The distance between a customer and a facility is limited, and customers can only be allocated to facilities closer to the maximum acceptable distance. A schematic of this problem is presented in Fig. 2. In the next two subsections, the indices, parameters, and decision variables used to model the problem at hand are introduced. Then, the mathematical formulation of the problem is derived.

3.1. Indices, parameters, and decision variables

The indices, parameters, and decision variables used to model the problem are defined as follows

j	Index of facilities (existing facilities: $j = 1, 2, \dots, f$; new facilities: $j = f + 1, f + 2, \dots, f + p$)
i	Index of demand points ($i = 1, 2, \dots, N$)
r	Index of assignment levels ($r = 1, 2, \dots, R$)
λ_j	Arrival rate of customers to the facility located at node j
μ_j	Service rate at facility j
h_i	Demand rate at demand point i
m	Number of servers in each facility

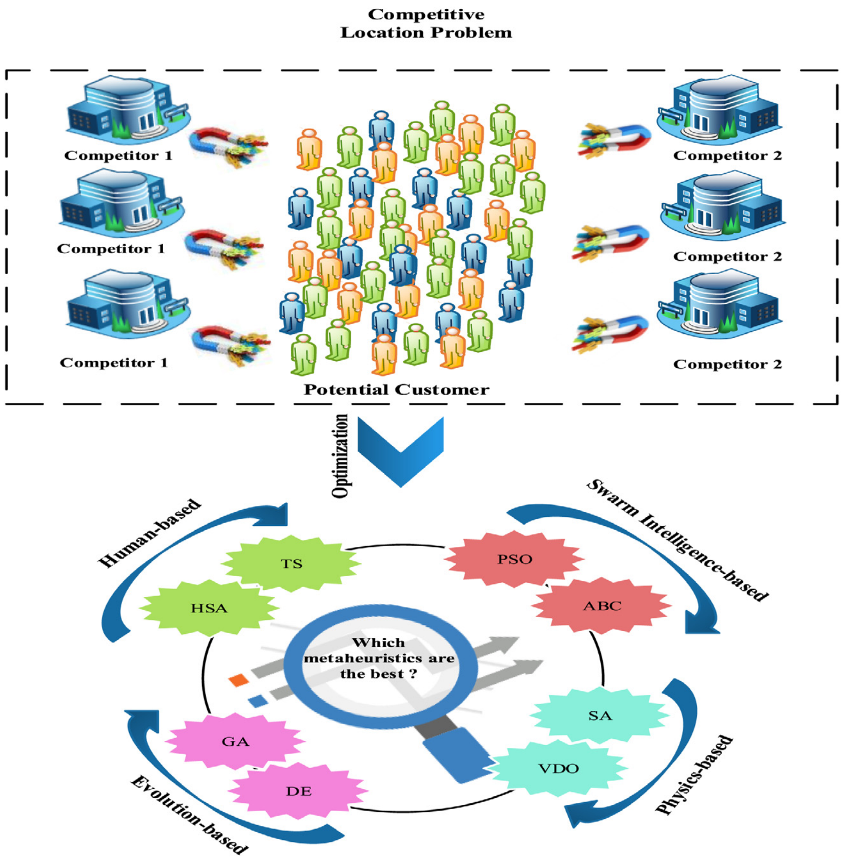

Fig. 4. Schematic of CFLP optimization using different meta-heuristics.

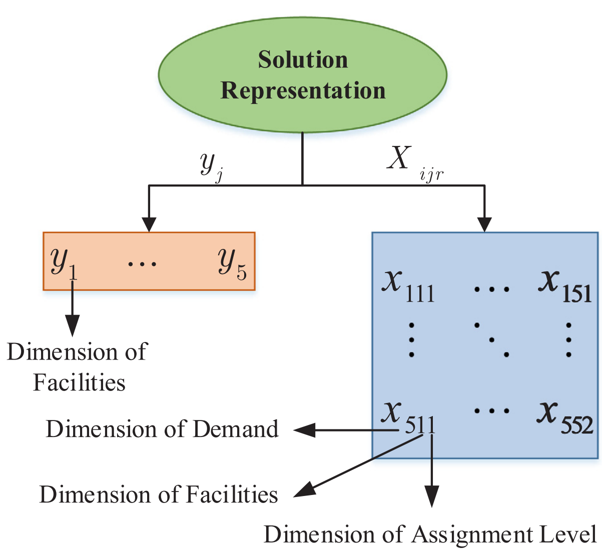

Fig. 5. Solution representation.

q	Failure probability of the servers
R	Number of assignment levels
ρ_j	Productivity rate of facility j
P_{0j}	Probability of no customer being present at facility j
Lq_j	Average queue length at facility j
Wq_j	Average waiting time at facility j
d_{ij}	Distance between demand point i and facility j
$d_{\max}$	Maximum acceptable distance
k	Maximum number of new facilities that could be established
β	Minimum acceptable percentage of market share for new facilities
f_j	Fixed installation cost to establish a new facility at node j
θ_j	Waiting cost at facility j
π_i	Penalty cost for losing the demand point i
S_{ij}	Cost of serving the customer at node i by facility j
C_{ij}	Travel cost from demand point i to facility j

Decision variables

y_j	1 If the new facility is established at node j, 0 otherwise
X_{ijr}	1 If demand point i is assigned to facility j at assignment level r, 0 otherwise

3.2. The model

A mixed-integer non-linear programming model that minimizes the total cost, including fixed and transportation costs, the cost of customer's waiting time, and service costs, is proposed in this section. To this end, the equations of the $M/M/m$ queue system are first stated as follows (Aboolian et al., 2009; Hajipour et al., 2014):

$$\lambda_j = \sum_{r=1}^{R}\sum_{i=1}^{N} h_i X_{ijr};\ \forall j = f+1,\dots,f+p \tag{1}$$

$$\rho_j = \frac{\lambda_j}{m\mu_j};\ \ \forall j = f+1,\dots,f+p \tag{2}$$

```
Initialize Population
Repeat
    Evaluate the individual fitness
    Select best pairs to reproduce
    Crossover
    Mutation
    Fitness Computation and merging populations
    Sort population and delete extra members
Until Termination Condition
```

Fig. 6. Pseudo-code of GA.

```
Create an initial population x_1,...,x_n of n random real-valued vectors;
Decode each vector into a solution;
Evaluate the fitness of each solution;
repeat:
for each vector x^j ∈ x^1,...,x^n do.
Select three other vectors randomly from the population;
Apply difference vector to base vector to create variant vector;
Combine vector x^j with variant vector to produce a new trial vector;
Evaluate the fitness of new trial vector;
If the trial vector has higher fitness than x^j then
    Replace x^j with the trial vector;
end
end
Until termination condition.
```

Fig. 7. Pseudo-code of the differential evolution algorithm.

$$P_{0j}=\left(\sum_{n=0}^{m-1}\frac{\rho_j^n}{n!}+\frac{\rho_j^m}{m!(1-\rho_j)}\right)^{-1};\quad \forall j=f+1,\dots,f+p \tag{3}$$

$$Lq_j=\frac{P_{0j}}{m!}\left(\frac{\lambda_j}{\mu_j}\right)^m\frac{\rho_j}{(1-\rho_j)^2};\quad \forall j=f+1,\dots,f+p \tag{4}$$

$$Wq_j=\frac{Lq_j}{\lambda_j};\quad \forall j=f+1,\dots,f+p \tag{5}$$

The arrival rate of customers is defined in Eq. (1). The rate of productivity for each facility, the probability that the facility is empty, the average queue length, and the average customer waiting time are defined in Eqs. (2)–(5), respectively. The mathematical model of the problem is defined as follows.

$$\begin{aligned} MinZ = & \sum_{j=f+1}^{f+p} f_j y_j + \sum_{i=1}^{N}\sum_{j=f+1}^{f+p}\sum_{r=1}^{R}\lambda_j C_{ij}\left(1-q^m\right)q^{(r-1)m}+ \\ & \sum_{i=1}^{N}\sum_{j=f+1}^{f+p}\sum_{r=1}^{R}\lambda_j\theta_j Wq_j\left(1-q^m\right)q^{(r-1)m} \\ & +\sum_{i=1}^{N}\sum_{j=f+1}^{f+p}\sum_{r=1}^{R}\lambda_j S_{ij}\left(1-q^m\right)q^{(r-1)m} \end{aligned} \tag{6}$$

$Subject\ to:$

$$\lambda_j \le m\mu_j;\quad \forall j=f+1,\dots,f+p \tag{7}$$

$$\frac{\sum_{r=1}^{R}\sum_{i=1}^{N}\sum_{j=f+1}^{f+p} h_i X_{ijr}}{\sum_{r=1}^{R}\sum_{i=1}^{N}\sum_{j=1}^{f+p} h_i X_{ijr}} \ge \beta \tag{8}$$

$$X_{ijr}=0;\quad \forall r,\forall i,\forall j\ \{i,j|\ d_{ij}>d_{\max}\} \tag{9}$$

$$\sum_{r=1}^{R}\sum_{j=1}^{f+p} X_{ijr}=1;\quad \forall i=1,\dots,N \tag{10}$$

$$\sum_{r=1}^{R} X_{ijr}\le 1;\quad \forall i,\forall j \tag{11}$$

$$X_{ijr}\le y_j;\quad \forall i,\forall j,\forall r \tag{12}$$

$$\sum_{j=f+1}^{f+p} y_j \le k \tag{13}$$

$$y_j\in\{0,1\};\quad \forall j=f+1,\dots,f+p \tag{14}$$

$$X_{ijr}\in\{0,1\};\quad \forall i,\forall j,\forall r \tag{15}$$

The model's objective function that minimizes the total cost is shown in Eq. (6). As customer allocation is level-by-level (Zarrinpoor

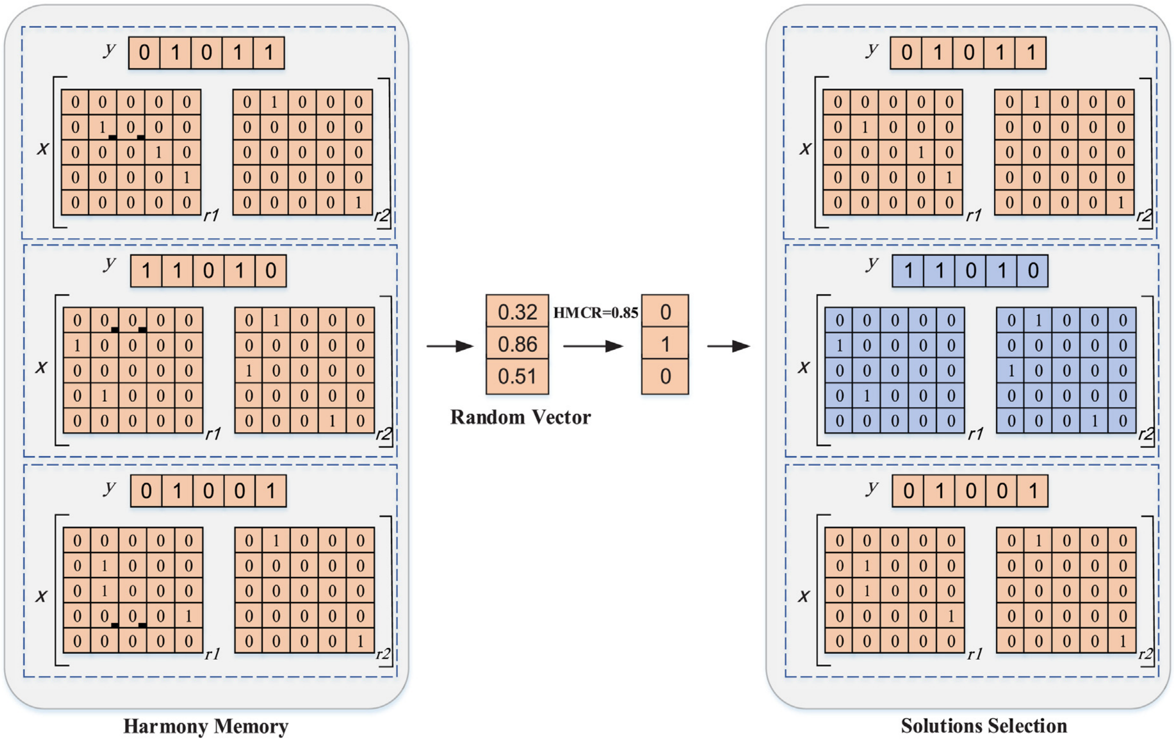

Fig. 8. Selecting solutions through HMCR.

et al., 2016), the facilities are not available in levels 1 to $r-1$, until one facility becomes available in level r to which the customer is assigned. If all the servers of a facility fail, the facility loses the customer, and the assignment level is changed. Therefore, the probability that a customer gets a service at level r is equal to $(1-q^m)\,q^{(r-1)m}$. Constraint (7) ensures that the arrival rate of customers to any facility is less than or equal to its service rate. Through Constraint (8) new facilities reach a minimum acceptable percentage of the market share. Constraint (9) allows customers to be assigned to the facilities if the maximum acceptable distance criterion is met. Constraint (10) ensures that each customer is assigned to exactly one facility at one assignment level. Constraint (11) prevents customers from being assigned to a facility at more than one assignment level. Constraint (12) enables customers to be assigned to open facilities. Constraint (13) limits the maximum number of new facilities that can be established. Constraints (14) and (15) define the type of binary decision variables.

3.3. CFLP optimization algorithm classification

Several optimization algorithms are classified and selected to solve the NP-hard CFLP problem at hand to determine which solution algorithms display better performances. The classification is presented in Fig. 3. As shown in this figure, meta-heuristic algorithms are divided into four categories, (1) evolution-based, (2) human-based, (3) physics-based, and (4) swarm intelligence-based (Du & Swamy, 2016).

The first category is composed by evolution-based algorithms including evolutionary strategies (Rechenberg, 1973), genetic algorithm (Holland, 1975), memetic algorithm (Moscato, 1989), genetic programming (Koza, 1994), differential evolution (Storn & Price, 1997), estimation of distribution (Larrañaga & Lozano, 2001), and biogeography-based optimization (Simon, 2008). The second category includes algorithms such as simulated annealing (Kirkpatrick et al., 1983); (Ferreira & de Queiroz, 2018), quantum computing (Neil & C.L., 1998), electromagnetism-like algorithm (Birbil & Fang, 2003), big bang big crunch (Erol & Eksin, 2006), water flow-like algorithm (Yang & Wang, 2007), central force optimization (Formato, 2007), vibration damping optimization (Mehdizadeh & R., 2008), gravitational search (Rashedi et al., 2009), and charged system search (Kaveh & Talatahari, 2010), all of which are based on physics laws. The third category includes algorithms such as tabu search (Glover, 1986), cultural algorithm (Reynolds, 1994), harmony search (Geem et al., 2001), imperialist competitive algorithm (Atashpaz-Gargari & Lucas, 2007; Hosseini & Al Khaled, 2014), and teaching and learning-based optimization (Rao et al., 2011), which are based on human behavior. The final category corresponds to algorithms based on swarm intelligence. This category includes artificial immune systems (Farmer et al., 1986), ant colony optimization (Dorigo, 1992), particle swarm optimization (Eberhart & Kennedy, 1995), bacterial foraging (Passino, 2002), shuffle frog leaping (Eusuff & Lansey, 2003), artificial bee colony (Karaboga, 2005), invasive weed optimization (Mehrabian & Lucas, 2006), cat swarm optimization (Chu et al., 2006), monkey search (Mucherino & Seref, 2007), fish school search (Bastos Filho et al., 2008), firefly algorithm (Yang, 2009), dolphin partner optimization (Shiqin et al., 2009), cuckoo optimization (Yang & Deb, 2009), bat-inspired algorithm (Yang, 2010), wolf search (Tang et al., 2012), flower pollination (Yang, 2012), krill herd algorithm (Gandomi & Alavi, 2012), social spider optimization (Cuevas et al., 2013), dolphin echolocation (Kaveh & Farhoudi, 2013), forest optimization (Ghaemi & Feizi-Derakhshi, 2014), grey wolf optimizer (Luo, 2019; Mirjalili et al., 2014), and lion optimization (Yazdani & Jolai, 2016).

In this paper, two algorithms from each of the above four categories (shaded in Fig. 3) are chosen to compare their performances in solving the new CFLP problem modeled in Section 3.2. Fig. 4 displays a schematic of the current research.

Start
Set the algorithm's parameters (HMS, HMCR, PAR, MaxIteration)
Initialize the HM
Evaluate the fitness
***While** Number of iteration (t) is less than* $MaxIteration$
for $i = 1$ *to* HMS ***do** (Index of population)*
for $j = 1$ *to* n ***do** (Index of decision variables)*
***if** rand < HMCR // (Memory Consideration)*
$x_{ij}^{new} = x_{kj} \quad, k \in (1,2,\ldots,HMS)$
***if** rand < PAR // (Pitch Adjustment)*
$x_{ij}^{new} = x_{ij}^{new} + BW(2rand - 1)$
endif
else
Randomly select x_{ij}^{new} *in its domain // (Random Initialization)*
endif
endfor
Evaluate the fitness of x^{new}
Update the HM by replacing the worst HM member (x^{worst}*) with* x^{new} *if* $f(x^{new})$ *is better than* $f(x^{worst})$*, or disregard* x *otherwise*
Update the best harmony vector
Set $t = t + 1$
End

Fig. 9. Pseudo-code of the harmony search algorithm.

4. Solution algorithms

Feasible solutions are first generated to initialize the meta-heuristic algorithms designed to improve upon these initial solutions. In addition, the Dicopt solver in GAMS, which applies an approximation algorithm to solve the problem, is used to validate the model. The solution representation, general approach used, and pseudo-code for the eight solution algorithms implemented are briefly presented below. Since these algorithms have been extensively used in the literature, interested readers are referred to the research cited in Section 3.3 for additional details.

4.1. Solution representation

In this study, a one-dimensional binary vector, y_j, and a multidimensional binary matrix, X_{ijr}, both illustrated in Fig. 5, are used to encode a solution to the problem.

4.2. Genetic algorithm (GA)

In the implementation process of the GA, n random feasible solutions (chromosomes) are first generated to comprise the initial population. Then, the chromosomes of the initial population are evaluated. Next, some solutions are randomly selected as parents for reproduction using the mask crossover and inversion mutation operations. In the next step, the populations generated are merged and sorted according to the fitness value of their chromosomes. Only the n primary chromosomes of this population are kept, while the chromosomes that have less fitness and are therefore placed at the end of the sequence are eliminated (Yu et al., 2019). This process continues until the algorithm stops when a certain number of similar solutions in successive repetitions is obtained (*stall generation*). Fig. 6 shows the Pseudo-code of the GA used in this study.

Start
Set t=0
Generate an initial solution x.
Initialize the tabu lists $\tau \leftarrow \varnothing$ *and the size of tabu list L.*
Repeat:
  Set the candidate set $A(x,t) = \; x' \in N(x) \setminus \tau(x,t) \cup \tilde{\tau}(x,t)$.
  Find the best x *from* $A(x,t)$: *Set* $x' = \arg\min_{y \in A(x,t)} f(y)$.
  If $f(x')$ *is better than* $f(x), x \leftarrow x'$.
  Update the tabu lists and the aspiration criteria.
  If the tabu list τ *is full, then old features from* τ *are replaced.*
Set t = t+1.
Until termination criteria are satisfied.
End

Fig. 10. Pseudo-code of the tabu search algorithm.

Start
  Define algorithm's setting (parameters, neighborhood function, cooling structure)
  Create the initial solution
  While *the stop criteria are not met, do*
    Decrease the temperature with respect to cooling mechanism (Outer Loop)
    While *sufficient neighbors are not created (Inner Loop)*
    Create a neighbor for previous solution with respect to neighborhood function
      IF *Neighbor is better than current solution*
      Accept the Neighbor as new current solution
      Else
      Calculate the acceptance probability for Neighbor (Metropolis algorithm)
      Generate a random number
        IF *random Number generated is less than the acceptance probability*
        Accept the Neighbor as new current solution
        End
      End
    End (End of inner loop)
  End (End of outer loop)
End

Fig. 11. Pseudo-code of the simulated annealing algorithm.

4.3. Differential evolution (DE)

Population-based algorithms begin with an initial population of solutions. Similarly to GA, they have operators such as mutation and crossover, with the difference that mutation occurs before crossover and is applied regularly to each generation for the production of offspring. In GA, mutation comes after crossover; its actions are not regular and definitive but are only used occasionally. The same crossover and mutation operators used in GA are also employed in DE. Fig. 7 shows the Pseudo-code of DE (Brabazon et al., 2015).

4.4. Harmony search algorithm (HSA)

The harmony memory in HSA is similar to the elitism concept used in GA. This operator ensures that the best harmonies will not be erased when optimizing the memory. This operator is controlled at a harmony memory rate (HMCR), which describes the probability of selecting a component from the members of the harmony memory. The selection of members from the harmony memory by HMCR is described in Fig. 8. Similarly to GA, the *stall generation* approach is used to terminate the algorithm. The Pseudo-code of the harmony search algorithm is shown in Fig. 9.

Start
Define the algorithm's setting (parameters, neighborhood function, damping structure)
Create the initial solution
While *the stop criteria are not met, do*
Adjust the amplitude with respect to the damping mechanism (Outer Loop)
While *sufficient neighbors are not created (Inner Loop)*
Create a neighbor for the previous solution with respect to a neighborhood function
If the *Neighbor is better than the current solution*
Accept the Neighbor as a new current solution
Else
Calculate the acceptance probability for the Neighbor (Rayleigh distribution)
Generate a random number
If the *random number generated is less than the acceptance probability*
Accept the Neighbor as a new current solution
End
End
End (End of inner loop)
End (End of outer loop)
End

Fig. 12. Pseudo-code of the vibration damping optimization algorithm.

Set t=1
Initialize each particle in the population by randomly selecting values for its position x_i *and velocity* v_i, $i = 1,\ldots,N_p$.
Repeat:
Calculate the fitness value of each particle i.
Determine the location of the particle with the highest fitness and revise $x^g(t)$ *if necessary.*
For each particle i, calculate its velocity.
Update the location of each particle i.
Set t = t+1.
Until stopping criteria are met.

Fig. 13. Pseudo-code of the particle swarm optimization algorithm.

4.5. Tabu search algorithm (TS)

Similarly to GA, the initial population is generated randomly, based on which the same mutation operation is applied to generate new solutions in TS. Any solution that is reviewed is maintained in a tabu list, so the algorithm does not return to those positions and moves to a position with a better fitness function. This search process is repeated until the stop criteria are met (*stall generation*). In this algorithm, the best solution is always stored. In each generation, the solution obtained is compared with the best solution found up to that moment, and if it is better, it is stored as the best solution. Fig. 10 presents the Pseudo-code of the tabu search algorithm.

4.6. Simulated annealing algorithm (SA)

This algorithm is implemented in three steps: (1) initializing a solution with a well-founded initial answer defined randomly, (2) neighborhood search to achieve thermodynamic equilibrium, and (3) cooling step, in which the temperature decreases based on a specific

```
Start
    Set the algorithm's parameters, including the number of employed and unemployed bees
    Initialize the food sources with respect to the number of employed bees
    Calculate the fitness of food sources
    While the number of iterations is less than the maximum cycle number (Stop criterion)
        For each employed bee
            Find a new food source in its neighborhood
            Evaluate the fitness of the new food source
            If the new food source is better than the old food source
                Substitute the old food source with the new one
            Else
                Increase the abandonment limit of the old food source by one unit
            End
        End
        Calculate the selection probability for all the employed bees
        For each onlooker, bee
            Select an employed bee randomly, based on the calculated probability
            Find a food source for the onlooker bee in the neighborhood of the selected employed bee
            Evaluate the fitness of the new food source
            If the new food source is better than the old food source
                Substitute the old food source with the new one
            Else
                Increase the abandonment limit of the old food source by one unit
            End
        End
        For food sources for which the abandonment limit is greater than a predefined value
            Abandon the food source
            Consider the related employed bee as a new scout bee
            Send the scout bee to locate a new food source randomly
        End
        Memorize the best solution found so far
        Increase the Number of iterations by one unit
    End
End
```

Fig. 14. Pseudo-code of the artificial bee colony algorithm.

structure. These steps continue until the stopping criterion (*stall generation*) is met. Fig. 11 describes the Pseudo-code of the simulated annealing algorithm.

4.7. Vibration damping optimization (VDO)

Similarly to the SA algorithm, the VDO algorithm involves three steps: (1) solution initialization, (2) neighborhood search, and (3) vibration damping (decreasing oscillation amplitude). These steps continue until the stopping criterion (*Stall Generation*) is met. The Pseudo-code of this algorithm is shown in Fig. 12.

4.8. Particle swarm optimization (PSO)

As an evolutionary algorithm, PSO starts with a population of solutions. Then, it examines different areas in the solution space consisting of several birds (called particles) as feasible solutions to the optimization problem. In other words, a particle swarm is created in the space of the problem and randomly initialized. Then, as the birds search for a food source, the algorithm also searches for particles to find the best solution.

In the discrete version of this algorithm, each particle is a string of binary variables. Here, each particle assigned a random position has its path and speed. PSO identifies its path based on the best situation experienced by all particles. Particles exchange information with other particles to update their speeds and positions. The two most important parameters of PSO are cognitive and social learning, which control the balance between the concepts of exploitation and exploration. The stopping criterion of this algorithm is chosen to be the standard *Stall Generation*. The Pseudo-code of the PSO utilized in this paper is given in Fig. 13.

4.9. Artificial bee colony algorithm (ABC)

ABC involves four steps that include (1) initializing the population of forager bees in the hive randomly, (2) performing a local search around food sources using employed bees, (3) selecting employed bees by onlooker bees to determine the position, and (4) releasing a poor-quality food source by finding a new food source. One of the most interesting features of the honeybee colony is their efforts to find food sources to store in the hive for future use. These steps continue until the stopping criterion (*Stall Generation*) is met. Fig. 14 illustrates the Pseudo-code of this algorithm.

The results obtained using the above eight meta-heuristic algorithms and the one derived using GAMS are analyzed in the next section.

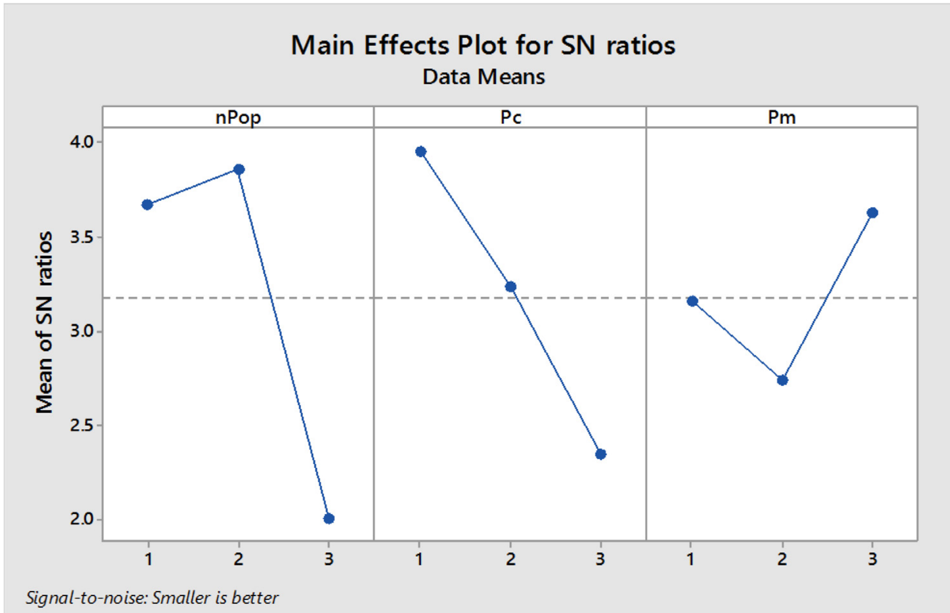

Fig. 15. SN diagram to tune the GA parameters.

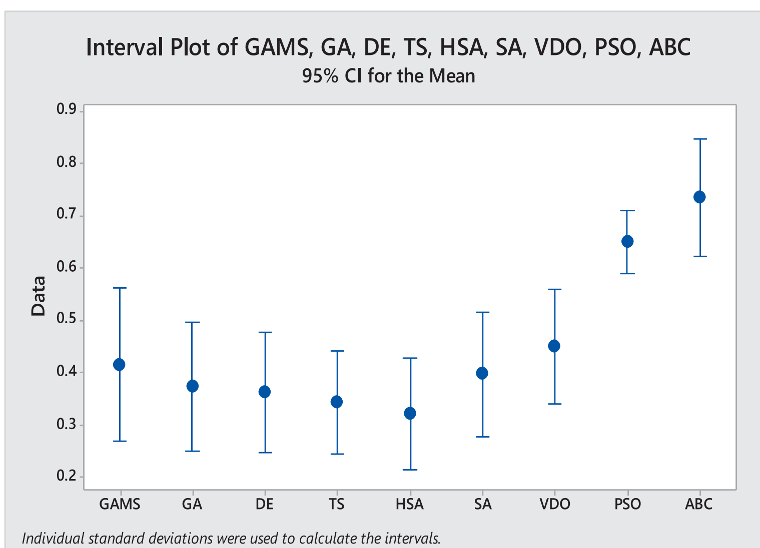

Fig. 16. Performance comparison of the meta-heuristics and GAMS.

5. Analysis of the results

Ten test problems of different sizes (regarding the number of locations and demand points) are generated randomly to analyze and compare the solutions obtained using the algorithms. The problems are listed in Table 4, and their parameters are described in Table 5. The values of the minimum acceptable market share (β), the failure probability of servers (q), the number of servers (m), and the maximum acceptable distance $\left(d_{\max}\right)$ are 0.4, 0.4, 2, and 35, respectively. In Table 5, $\mu_j \sim U$ [1015] means that the service rate of the jth facility is generated randomly using a uniform distribution in [10,15]. The same intuition applies to the other parameters composing this table.

5.1. Calibration of the algorithm parameters

Each solution algorithm has a set of initial parameters that significantly affect its performance. In this section, the Taguchi method is applied to adjust the parameters.

Table 4
Test problems.

No.	No. of facility locations (P)	No. of demand points (N)
1	5	5
2	5	10
3	5	20
4	10	20
5	10	25
6	10	30
7	10	35
8	10	40
9	12	45
10	12	50

- GA involves three parameters, including the number of chromosomes in the initial population (*nPop*), the crossover rate (*Pc*), and

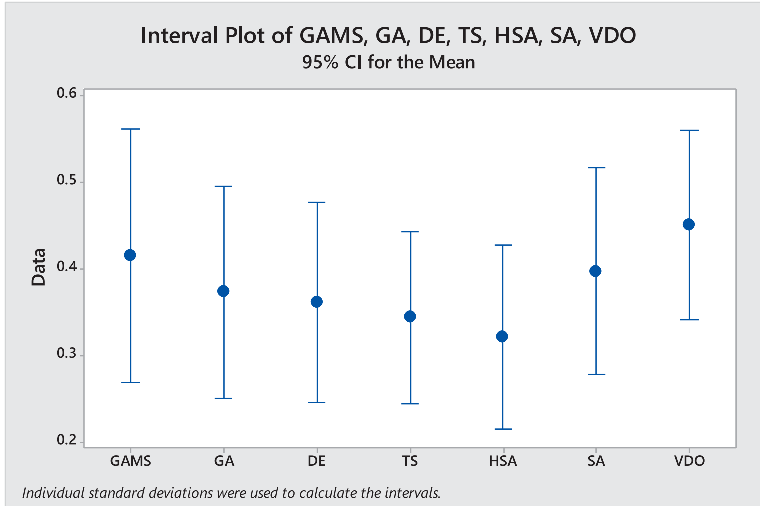

Fig. 17. Graphical comparison of the algorithms excluding PSO and ABC.

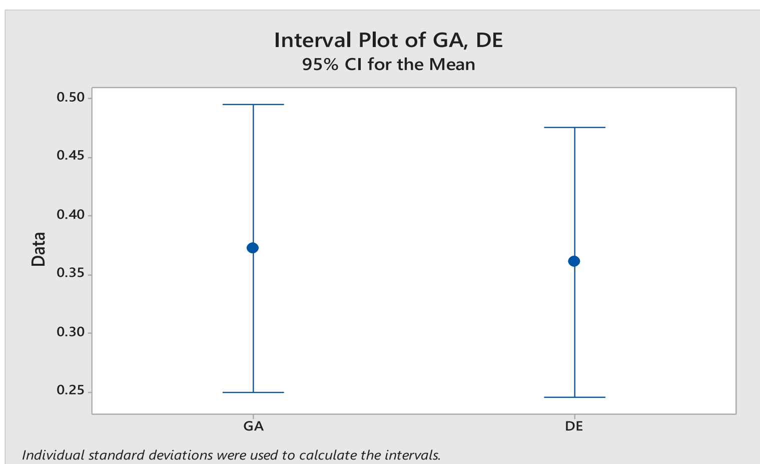

Fig. 18. Graphical performance analysis of GA and DE.

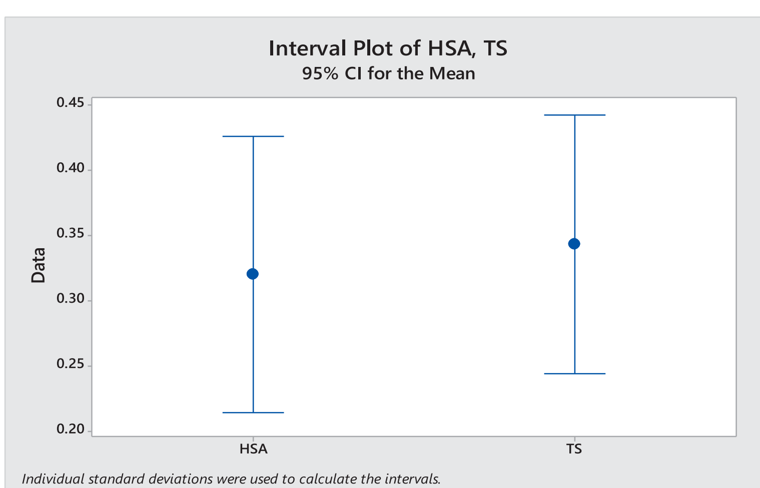

Fig. 19. Graphical performance analysis of HSA and TS.

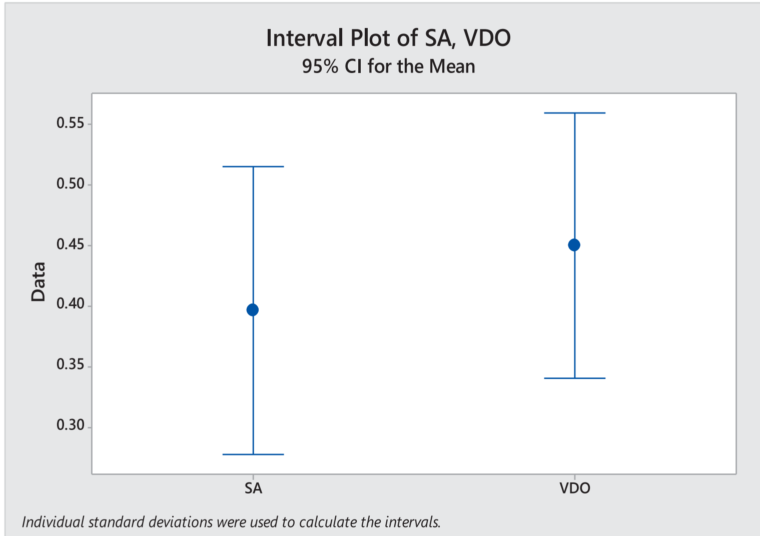

Fig. 20. Graphical performance analysis of SA and VDO.

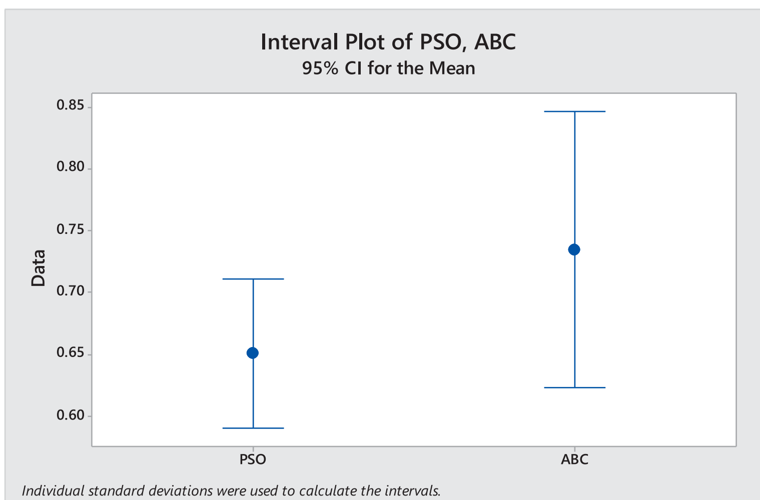

Fig. 21. Graphical performance analysis of PSO and ABC.

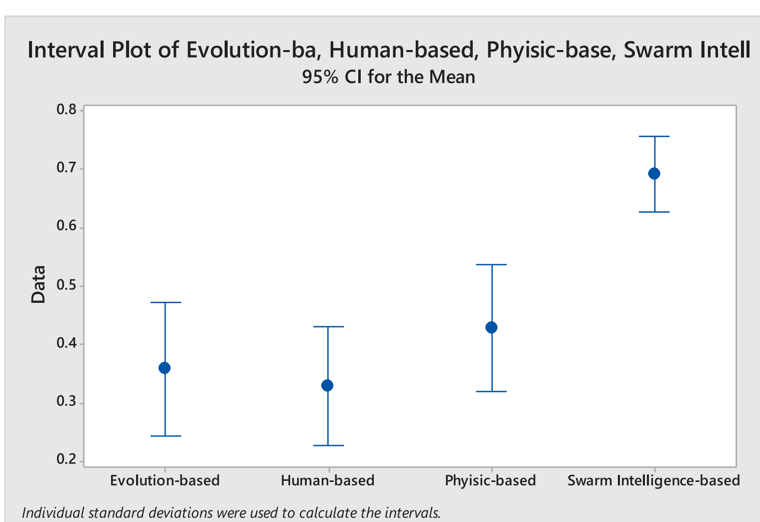

Fig. 22. Graphical performance of evolution-, human-, physics-, and swarm intelligence-based meta-heuristics.

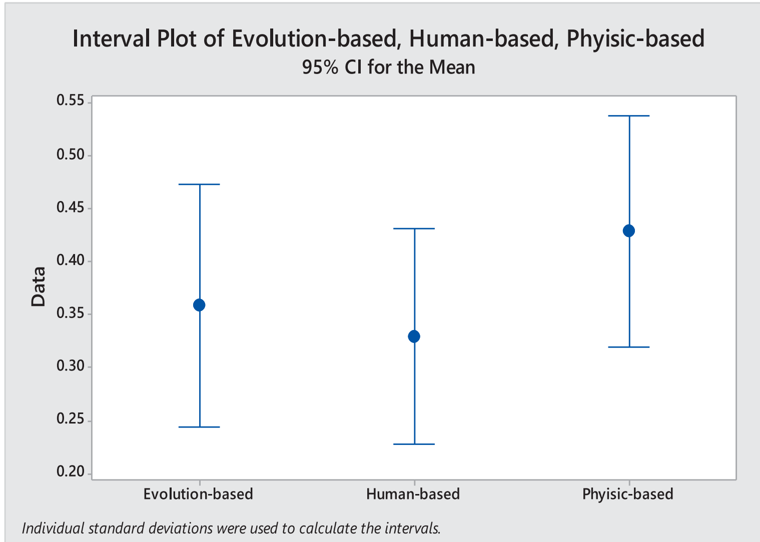

Fig. 23. Graphical performance of evolution-, human-, and physics-based meta-heuristics.

Table 5
Problem parameters.

Parameters	Values
μ_j	$\mu_j \sim U[10\ 15]$
h_i	$h_i \sim U[2\ 6]$
d_{ij}	$d_{ij} \sim U[10\ 55]$
f_j	$f_j \sim U[380000\ 420000]$
θ_j	$\theta_j \sim U[2800\ 4000]$
π_i	$\pi_i \sim U[48000\ 53000]$
S_{ij}	$S_{ij} \sim U[14000\ 16500]$
C_{ij}	$C_{ij} \sim U[4000\ 6500]$

the mutation rate (*Pm*). DE has two parameters: initial population size (nPop) and mutation rate (*Pm*).

- HSA contains the harmony memory rate (*HMCR*), pitch-adjusting rate (*PAR*), and initial harmony population (*nPop*). The parameters of the TS algorithm are the number of neighbors (*NN*) and the tabu list (LT) limitation.
- The SA algorithm includes the initial population size (*nPop*), initial temperature (*T0*), final temperature (*Tf*), and the number of neighbors in the inner loop (*nMove*). The parameters of the VDO algorithm are the initial oscillation amplitude (*A0*), length of the inner loop (*nMove*), damping coefficient (*Gama*), and standard deviation of the Rayleigh distribution function.
- The parameters of the PSO algorithm are the swarm size (*Swarmsize*), cognitive learning (*C1*), and social learning (*C2*), with C2 = 4-C1. Finally, the parameters of the ABC algorithm are the population size of the honeybee hive (*nHive*) and the limitation of releasing food source (*LNF*) (Hajipour, Farahani, & Fattahi, 2016; Hajipour, Fattahi, et al., 2016; Hajipour et al., 2014; Nasiri et al., 2018; Saif et al., 2014).

Table 6
Levels of the algorithms' parameters.

Algorithm	Parameters	Level 1	Level 2	Level 3
GA	*nPop*	25	50	100
	Pc	0.6	0.8	0.99
	Pm	0.01	0.2	0.4
DE	*nPop*	25	50	100
	Pc	0.6	0.8	0.99
HSA	*HMCR*	0.75	0.85	0.95
	PAR	0.1	0.3	0.5
	nPop	25	35	50
TS	*NN*	30	50	70
	LT	3	5	7
SA	*nPop*	5	15	30
	T0	500	750	1000
	Tf	5	10	20
	nMove	10	30	50
VDO	*A0*	6	8	10
	nMove	4	8	12
	Gama	0.005	0.05	0.5
	Sigma	1	1.5	2
PSO	*Swarmsize*	60	80	100
	C1	1	1.5	2
ABC	*nHive*	200	400	600
	LNF	10	20	30

Table 6 presents the above set of parameters, each of them defined across three different levels. The *Stall Generation* is assumed to be 30 in all algorithms.

The problem at hand is solved using meta-heuristics, where each technique is applied considering different combinations of its parameter levels. The responses obtained are the objective function value and the required computational time in seconds. For the sake of comparison, we combine both responses into a single one through the weighted sum defined in Eq. (16), where W_1 and W_2 are the weights assigned to each response and assumed equal to 0.4 and 0.6, respectively. This combined variable is used as the response in the Taguchi method.

$$Combined\,Variable = \frac{W_1 * Time}{Maximum\ Time} + \frac{W_2 * Objective\ Function}{Maximum\ Objective\ Function} \tag{16}$$

To illustrate the use of the Taguchi method, consider the GA algorithm. The values of the two responses based on different combinations of its parameters alongside the corresponding combined variable are described in Table 7 and Fig. 15. This latter figure illustrates the Signal to Noise (*SN*) ratio.

According to Fig. 15, the appropriate levels for *nPop*, *Pc*, and *Pm* are level two (50), level one (0.6), and level three (0.4), respectively. The parameters of the other seven meta-heuristics are tuned similarly. Table 8 contains the tuned values of all the solution algorithms.

Table 7
Values of the combined variable defined to tune the GA parameters.

Test problems	Parameters			Objective function	Time (s)	Combined variable
	nPop	*Pc*	*Pm*			
1	1	1	1	1462933.3589	0.457020	0.606572811
2	1	2	2	1658023.6545	0.429390	0.67250589
3	1	3	3	1365400.5866	1.158187	0.68967555
4	2	1	2	1462934.1756	0.538897	0.62039868
5	2	2	3	1364583.3459	0.694750	0.61112482
6	2	3	1	1557480.4202	0.774318	0.69436545
7	3	1	3	1406154.0715	0.995525	0.67695653
8	3	2	1	1107287.1733	2.341994	0.79616548
9	3	3	2	1462934.5782	2.368856	0.92940182

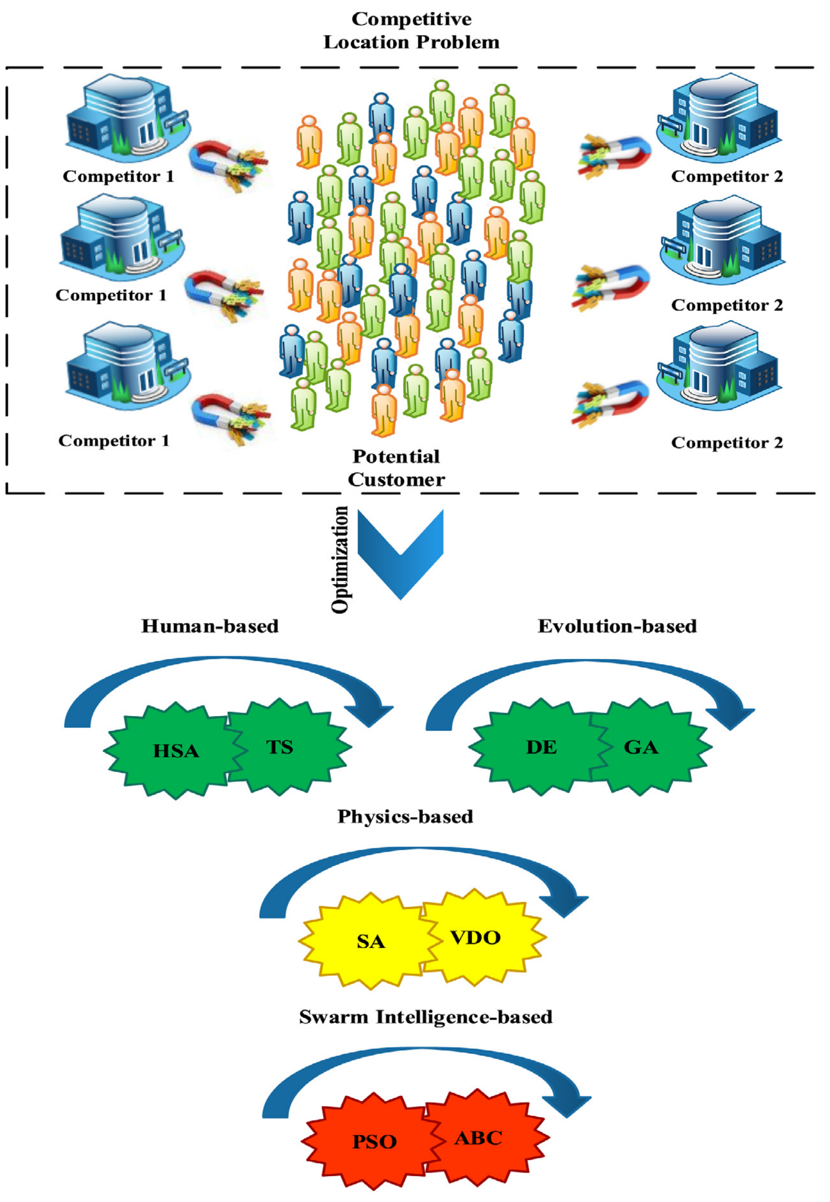

Fig. 24. An overview of meta-heuristics performances when solving CFLPs.

Table 8
Tuned levels of the algorithms' parameters.

Algorithms	Parameters	Tuned value
GA	*nPop*	50
	Pc	0.6
	Pm	0.4
DE	*nPop*	25
	Pc	0.6
HSA	*HMCR*	0.85
	PAR	0.5
	nPop	35
TS	*NN*	30
	LT	3
SA	*nPop*	5
	T0	750
	Tf	10
	nMove	10
VDO	*A0*	8
	nMove	4
	Gama	0.05
	Sigma	2
PSO	*Swarmsize*	60
	C1	2
ABC	*nHive*	400
	LNF	20

5.2. Comparison analysis

The tuned parameters determined in Section 5.1 are used in each of the eight meta-heuristics to optimize the test problems. GAMS software is also employed to solve the problems. Tables A.1 to A.5 within the appendix section present the outputs derived from GAMS, the GA and DE, HSA and TS, SA and VDO, and PSO and ABC algorithms, respectively. The non-parametric Kruskal–Wallis test, which does not require the normality of the data, is used to compare the solution algorithms statistically. In particular, we analyze and compare the values of the combined variable. Fig. 16 illustrates graphically the differences between the values obtained. Table A.6 formally complements the intuition provided by this figure, describing how the solution means obtained from these algorithms differ significantly (P-value is 0.000). Clearly, PSO and ABC are incompatible with the other algorithms. We have therefore eliminated these two algorithms from the analysis. The resulting output is illustrated in Fig. 17, and the corresponding numerical values are reported in Table A.7.

The results in Table A.7 do not present any significant difference between the solution methods in terms of the mean of the combined variable (P-value is 0.402). In addition, according to Fig. 17, even though the performances of the different methods are close to each other, HSA, TS, DE, and GA, respectively, display better performances than GAMS, SA, and VDO. Note that the best-performing algorithms, HSA and TS, belong to the human-based class. We elaborate on this feature below.

We start by testing whether there is a significant difference between the two algorithms that belong to the same group. The performances of GA and DE within the evolution-based algorithms are reported in Table A.8 and graphically illustrated in Fig. 18.

The results in Table A.8, as well as those in Fig. 18, show no significant difference between GA and DE (P-value is 0.940). The same analysis has been performed between HSA and TS, which belong to the class of human-based algorithms. The corresponding results are reported in Table A.9 and Fig. 19, from which the same conclusion can be drawn, i.e., HSA and TS do not differ significantly (P-value is 0.496).

The same conclusions can be drawn from the performances of the physics-based SA and VDO algorithms. These algorithms do not differ significantly regarding the average combined variable derived from the facility location problems. Table A.10, displaying a P-value of 0.406, and Fig. 20 support this claim. A similar intuition applies to the performances of the intelligence-based PSO and ABC algorithms, compared in Table A.11 (P-value is 0.545) and Fig. 21.

Finally, we compare the performance of the different algorithm classes through the average of the combined variable obtained for each group. Tables A.12 and A.13 present the average values of the combined variable for the different classes of solution algorithms. The results described in these tables are statistically analyzed in Table A.14 and graphically illustrated in Fig. 22. The P-value displayed in Table A.14 is smaller than 0.05, implying a significant difference among the four solution algorithms.

In this regard, Fig. 22 illustrates how the swarm intelligence-based algorithms are incompatible with the other classes. Thus, we eliminate these algorithms from the subsequent analysis, where the other three classes are further compared. The results obtained are reported in Table A.15, while Fig. 23 demonstrates their performance graphically. Note how these three classes do not differ statistically (P-value is 0.217). Based on Fig. 23, it can be concluded that while the results obtained are close to each other, the human- and evolution-based algorithms display better performances than the physics-based ones. Furthermore, as illustrated in Table A.7 and Fig. 17, HSA performs better than TS within the human-based class, defining the best potential choice when facing the type of CFLP framework described in this paper.

5.3. Discussion

Fig. 24 provides a graphical representation of the conclusions derived in Section 5.2. As shown in this figure, the human- and evolution-based algorithms display the best performances in solving the CFLP analyzed in the current paper. On the other hand, the physical- and swarm intelligence-based algorithms have the lowest performances.

Being human-based algorithms, the harmony search and tabu search algorithms require fewer computations, are based on simple concepts and few parameters, and are easy to run, making them better fits for optimizing CFLPs. Although many researchers have utilized the tabu search algorithm to solve CFLPs, the harmony search algorithm, which performs considerably well on these problems, has not been implemented. These results should encourage its use when optimizing CFLPs.

The genetic algorithm and differential evolution are evolution-based algorithms, which, according to the analysis, display good performances when solving CFLPs. These algorithms have substantial search power in the solution space. Namely, they do not seek only part of the space when finding the best solution but randomly select points from the whole solution space. Furthermore, these algorithms are highly flexible and can be applied to a variety of problems. Many studies have applied genetic algorithms within the literature on CFLPs, while the

Table A.1
GAMS results.

NO.	GAMS		
	Objective function	Time (s)	Combined variable
1	463935.448	0.309	0.62699974
2	885707.654	2.837	0.750882185
3	930400.187	0.981	0.511229562
4	997323.672	0.959	0.63218509
5	987918.69	1.342	0.299610862
6	1867649.401	8.834	0.434938894
7	1476416.444	5.956	0.221709644
8	2471083.448	2.45	0.242333123
9	2570437.925	1.515	0.184997277
10	3955736.316	2.233	0.626899974

Table A.2
GA and DE results.

NO.	GA			DE		
	Objective function	Time (s)	Combined variable	Objective function	Time (s)	Combined variable
1	385200.0148	1.540104	0.63224642	398000.0148	0.921090	0.594912
2	489671.2044	0.518648	0.35929886	553486.4568	0.795816	0.41727
3	1072768.9039	0.603568	0.57296319	1072771.0933	0.962578	0.584189
4	775630.5476	0.992898	0.49994311	808948.4954	1.536440	0.538226
5	769062.8308	1.976647	0.25758521	640932.8705	1.916512	0.221703
6	854259.5328	1.152710	0.15101731	988029.8686	3.936455	0.217864
7	1588530.5906	1.845624	0.195029	1236376.3413	4.389603	0.179959
8	3332378.9508	1.608730	0.31275517	2351353.2766	3.440795	0.23783
9	3078207.0677	1.849930	0.2175293	3699620.697	4.018755	0.277141
10	5899626.0119	2.606391	0.35876644	5265677.0463	5.899738	0.338482

Table A.3
HSA and TS results.

NO.	HSA			TS		
	Objective function	Time (s)	Combined variable	Objective function	Time (s)	Combined variable
1	398000.0148	0.936717	0.596273	385200.0148	0.983200	0.583765
2	442082.4243	1.063148	0.35602	480505.1206	0.598926	0.357359
3	592909.3268	1.535219	0.354241	785819.343	1.901903	0.465345
4	823205.2194	1.484000	0.545043	787103.0775	1.576521	0.526428
5	1055644.076	0.678177	0.246847	875677.3937	2.985747	0.31299
6	854266.148	0.703129	0.143558	1270256.8058	3.560210	0.255193
7	1681248.0485	1.223945	0.199456	1259277.4025	2.945479	0.168744
8	2228507.3402	0.935710	0.207992	2387503.4178	4.456171	0.251442
9	3285990.152	0.802730	0.229787	2822123.5126	3.701960	0.215172
10	4576713.4131	1.320705	0.266638	4917300.3436	2.755516	0.296155

Table A.4
SA and VDO results.

NO.	SA			VDO		
	Objective function	Time (s)	Combined variable	Objective function	Time (s)	Combined variable
1	385200.0148	0.755788	0.563968	385200.0148	2.391468	0.706362
2	476617.1087	1.302520	0.392145	489671.4763	2.839987	0.482756
3	1161650.4524	2.007820	0.662776	908061.396	3.720682	0.58535
4	805054.6802	3.716447	0.609032	774856.6296	5.696973	0.657319
5	1145165.3177	3.773017	0.405232	999402.9757	6.709794	0.442924
6	1246954.2092	2.747810	0.238115	852444.8438	10.606521	0.307615
7	1140570.7871	3.940148	0.164972	2026030.9874	9.471612	0.316628
8	3547466.1111	3.523345	0.347942	2961388.7005	10.611692	0.354
9	3436428.8891	5.387180	0.1267213	4339711.145	12.403014	0.370581
10	4594507.5227	7.587707	0.313	3075525.0841	14.050953	0.358717

Table A.5
PSO and ABC results.

NO.	PSO			ABC		
	Objective function	Time (s)	Combined variable	Objective function	Time (s)	Combined variable
1	385200.0148	1.954129	0.668289	385200.0148	4.5948	0.898173
2	480505.1747	2.159404	0.440351	410062.0295	7.5211	0.677786
3	978730.3407	2.706251	0.590134	1085186.14	12.7935	0.960506
4	992669.9115	3.850869	0.726413	983591.2984	11.921	0.991738
5	2240602.2033	4.666431	0.721915	1123217.0423	15.3105	0.700781
6	3886263.3851	6.848124	0.173639	1571087.0426	24.1049	0.64256
7	5371912.3086	7.990135	0.676206	1603226.8143	41.9395	0.579068
8	6677478.1658	7.163850	0.659345	2156658.0029	48.2861	0.593785
9	8761156.1413	8.074459	0.647771	3002263.6333	67.6104	0.60507
10	10681675.4552	8.933362	0.664662	5291351.5743	55.2615	0.69722

Table A.6
Comparative performance analysis of the Meta-Heuristics and GAMS.

Algorithms	N	Median	Ave. Rank	P-value
ABC	10	0.6875	77.2	
DE	10	0.3078	33.3	
GA	10	0.3360	35.1	
GAMS	10	0.3673	40.7	
HSA	10	0.2835	28.4	
PSO	10	0.6665	74.7	0.000
SA	10	0.3700	40.0	
TS	10	0.3043	31.6	
VDO	10	0.4068	48.5	
Overall	90		45.5	

Table A.7
Comparative performance analysis of the algorithms excluding PSO and ABC.

Algorithms	N	Median	Ave. Rank	P-value
DE	10	0.3078	32.6	
GA	10	0.3360	34.3	
GAMS	10	0.3673	37.9	
HSA	10	0.2835	27.9	0.402
SA	10	0.3700	38.6	
TS	10	0.3043	31.2	
VDO	10	0.4068	46	
Overall	70		35.5	

Table A.8
Comparative performance analysis of GA and DE.

Algorithms	N	Median	Ave. Rank	P-value
DE	10	0.3078	10.4	
GA	10	0.3360	10.6	0.940
Overall	20		10.5	

Table A.9
Comparative performance analysis of HSA and TS.

Algorithms	N	Median	Ave. Rank	P-value
HSA	10	0.2835	9.6	
TS	10	0.3043	11.4	0.496
Overall	20		10.5	

Table A.10
Comparative performance analysis of SA and VDO.

Algorithms	N	Median	Ave. Rank	P-value
SA	10	0.3700	9.4	
VDO	10	0.4068	11.6	0.406
Overall	20		10.5	

Table A.11
Comparative performance analysis of PSO and ABC.

Algorithms	N	Median	Ave. Rank	P-value
ABC	10	0.6875	11.3	
PSO	10	0.6665	9.7	0.545
Overall	20		10.5	

differential evolution algorithm remains mainly unused. As was the case with the harmony search algorithm, the results obtained should encourage its use when optimizing CFLPs.

Finally, simulated annealing and vibration damping optimization are physics-based algorithms. These algorithms have the advantage of simplicity in implementation compared to similar methods and the ability to escape local optima. Even though a limited number of studies have applied simulated annealing to solve CFLPs, the vibration damping optimization algorithm remains unused. Given the fairly good performance of both algorithms, they should constitute potential alternatives when solving CFLPs.

6. Conclusion and potential extensions

In today's competitive environment, facilities are generally similar in terms of services provided. Hence, they must compete to attract customers and increase their profits. One of the main features affecting competition between similar facilities is their location, which considerably impacts their capacity to attract customers. A wide variety of models displaying different sets of real-world features have been proposed to select the location of competitive facilities. Given the important role played by the solution methodology, we have compared the relative performances of several meta-heuristic algorithms. These algorithms have been applied to solve a new CFLP modeled in this paper, which considers the static competition among facilities subject to reliability and congestion constraints. GAMS was also applied to validate and optimize the model.

According to our analysis, human-based and evolutionary algorithms display better performance in optimizing CFLPs. Physics-based algorithms follow while GAMS is able to perform relatively close to these groups of meta-heuristics. On the other hand, swarm intelligence-based algorithms such as particle swarm optimization and artificial bee colony were among the less efficient methods in optimizing CFLPs.

Based on the results obtained in this research, the following suggestions are presented as potential extensions:

- Investigate the capacity of the solution algorithms to solve Nash and Stackelberg equilibrium models and determine how the type of competition affects the conclusions obtained.
- Investigate the performance of the algorithms when solving location problems that are not competitive and compare their behavior with the results of this research.

CRediT authorship contribution statement

Vahid Hajipour: Conceptualization, Methodology, Formal analysis, Writing – review & editing, Validation. **Seyed Taghi Akhavan Niaki:** Investigation, Formal analysis, Editing, Visualization, Validation. **Madjid Tavana:** Methodology, Formal analysis, Writing – review & editing. **Francisco J. Santos-Arteaga:** Methodology, Formal analysis, Writing – review & editing. **Sanaz Hosseinzadeh:** Methodology, Formal analysis, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

Dr. Francisco J. Santos-Arteaga is grateful for the support received from the María Zambrano contract from the Universidad Complutense de Madrid financed by the Ministerio de Universidades with funding from the European Union Next Generation program.

Table A.12
Average of the combined variable in evolution-based and human-based meta-heuristics.

Based on evolution			Based on human		
GA	DE	Average	HSA	TS	Average
0.63224642	0.594912	0.613579325	0.596273	0.583765	0.590019
0.35929886	0.41727	0.388284276	0.35602	0.357359	0.356689
0.57296319	0.584189	0.578576133	0.354241	0.465345	0.409793
0.49994311	0.538226	0.51908438	0.545043	0.526428	0.535736
0.25758521	0.221703	0.239644022	0.246847	0.31299	0.279673
0.15101731	0.217864	0.184440631	0.143558	0.255193	0.199376
0.195029	0.179959	0.187494239	0.199456	0.168744	0.1841
0.31275517	0.23783	0.276268862	0.207992	0.251442	0.229717
0.2175293	0.277141	0.252460458	0.229787	0.215172	0.224793
0.35876644	0.338482	0.344367924	0.266638	0.296155	0.281396

Table A.13
Average of the combined variable in physics-based and swarm intelligence-based meta-heuristics.

Based on physics			Based on swarm intelligence		
SA	VDO	Average	PSO	ABC	Average
0.563968	0.706362	0.635165	0.668289	0.898173	0.783231
0.392145	0.482756	0.437451	0.440351	0.677786	0.559069
0.662776	0.58535	0.624063	0.590134	0.960506	0.77532
0.609032	0.657319	0.633175	0.726413	0.991738	0.859076
0.405232	0.442924	0.424078	0.721915	0.700781	0.711348
0.238115	0.307615	0.272865	0.173639	0.64256	0.678099
0.164972	0.316628	0.2408	0.676206	0.579068	0.627637
0.347942	0.354	0.350971	0.659345	0.593785	0.626565
0.1267213	0.370581	0.328031	0.647771	0.60507	0.63835
0.313	0.358717	0.335858	0.664662	0.69722	0.680941

Table A.14
Results of the statistical analysis across groups of meta-heuristics.

Algorithms	N	Median	Ave. Rank	P-value
Evolution-Based	10	0.3103	14.4	
Human-Based	10	0.2805	13.0	
Physics-Based	10	0.3875	20.3	0.000
Swarm Intelligence-Based	10	0.6795	34.3	
Overall	40	0.3103	20.5	

Table A.15
Statistical analysis of evolution-, human-, and physics-based meta-heuristics.

Algorithms	N	Median	Ave. Rank	P-value
Evolution-Based	10	0.3103	14.2	
Human-Based	10	0.2805	12.9	0.217
Physics-Based	10	0.3875	19.4	
Overall	30	0.3103	15.5	

Appendix. Numerical results from Section 5.2

We provide below the set of tables summarizing the numerical results presented in Section 5.2.

References

Aboolian, R., Berman, O., & Krass, D. (2007). Competitive facility location and design problem. *European Journal of Operational Research*, *182*(1), 40–62.

Aboolian, R., Berman, O., & Krass, D. (2008). Optimizing pricing and location decisions for competitive service facilities charging uniform price. *Journal of the Operational Research Society*, *59*(11), 1506–1519.

Aboolian, R., Sun, Y., & Koehler, G. J. (2009). A location–Allocation problem for a web services provider in a competitive market. *European Journal of Operational Research*, *194*(1), 64–77.

Adamu, A., Abdullahi, M., Junaidu, S. B., & Hassan, I. H. (2021). An hybrid particle swarm optimization with crow search algorithm for feature selection. *Machine Learning with Applications*, *6*, Article 100108. http://dx.doi.org/10.1016/j.mlwa.2021.100108.

Ahmadi, Z., & Ghezavati, V. (2020). Developing a new model for a competitive facility location problem considering sustainability using Markov chains. *Journal of Cleaner Production*, *273*, Article 122971.

Asani, E., Vahdat-Nejad, H., & Sadri, J. (2021). Restaurant recommender system based on sentiment analysis. *Machine Learning with Applications*, *6*, Article 100114. http://dx.doi.org/10.1016/j.mlwa.2021.100114.

Ashtiani, G., Makui, A., & Ramezanian, R. (2013). A robust model for a leader–follower competitive facility location problem in a discrete space. *Applied Mathematical Modelling*, *37*(1–2), 62–71.

Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *Evolutionary computation,. CEC 2007. IEEE congress on* (pp. 4661–4667).

Bagherinejad, J., & Niknam, A. (2018). Solving the competitive facility location problem considering the reactions of competitor with a hybrid algorithm including Tabu search and exact method. *Journal of Industrial Engineering International*, *14*(1), 171–183.

Basciftci, B., Ahmed, S., & Shen, S. (2021). Distributionally robust facility location problem under decision-dependent stochastic demand. *European Journal of Operational Research*, *292*(2), 548–561.

Bastos Filho, C. J., de Lima Neto, F. B., Lins, A. J., Nascimento, A. I., & Lima, M. P. (2008). A novel search algorithm based on fish school behavior. In *2008 IEEE international conference on systems, man and cybernetics* (pp. 2646–2651). IEEE.

Bell, D. R., Ho, T. H., & Tang, C. S. (1998). Determining where to shop: Fixed and variable costs of shopping. *Journal of Marketing Research*, *35*(3), 352–369.

Beresnev, V. L. (2009). Upper bounds for objective functions of discrete competitive facility location problems. *Journal of Applied and Industrial Mathematics*, *3*(4), 419–432.

Beresnev, V. L., & Mel'nikov, A. A. (2011). Approximate algorithms for the competitive facility location problem. *Journal of Applied and Industrial Mathematics*, *5*(2), 180–190.

Beresnev, V. L., & Mel'nikov, A. A. (2014). The branch-and-bound algorithm for a competitive facility location problem with the prescribed choice of suppliers. *Journal of Applied and Industrial Mathematics*, *8*(2), 177–189.

Beresnev, V. L., & Mel'nikov, A. A. (2016). A capacitated competitive facility location model. *Diskretnyi Analiz I Issledovanie Operatssi*, *23*, 35–48.

Beresnev, V. L., & Mel'nikov, A. A. (2018a). Cut generation algorithm for the discrete competitive facility location problem. In *Doklady Mathematics, vol. 97, no. 3* (pp. 254–257). Pleiades Publishing.

Beresnev, V., & Mel'nikov, A. (2018b). Exact method for the capacitated competitive facility location problem. *Computers & Operations Research*, *95*, 73–82.

Beresnev, V., & Melnikov, A. (2020). ε-Constraint method for bi-objective competitive facility location problem with uncertain demand scenario. *EURO Journal on Computational Optimization*, *8*(1), 33–59.

Berman, O., Drezner, T., Drezner, Z., & Krass, D. (2009). Modeling competitive facility location problems: New approaches and results. In *Decision technologies and applications* (pp. 156–181). INFORMS.

Biesinger, B., Hu, B., & Raidl, G. (2016). Models and algorithms for competitive facility location problems with different customer behavior. *Annals of Mathematics and Artificial Intelligence*, *76*(1–2), 93–119.

Bilir, C., Ekici, S. O., & Ulengin, F. (2017). An integrated multi-objective supply chain network and competitive facility location model. *Computers & Industrial Engineering*, *108*, 136–148.

Birbil, Ş. I., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, *25*(3), 263–282.

Brabazon, A., O'Neill, M., & S., McGarraghy (2015). *Natural computing series*, *Natural computing algorithms*. New York, Dordrecht, London: Springer Heidelberg.

Chong, E. K., & Zak, S. H. (2013). *An introduction to optimization, vol. 76*. John Wiley & Sons.

Chu, S. C., Tsai, P. W., & Pan, J. S. (2006). Cat swarm optimization. In *Pacific rim international conference on artificial intelligence* (pp. 854–858). Berlin, Heidelberg: Springer.

Cuevas, E., Cienfuegos, M., Zaldívar, D., & Pérez, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, *40*(16), 6374–6384.

Di Caprio, D., & Santos-Arteaga, F. J. (2022). Enhancing the pattern recognition capacity of machine learning techniques: The importance of feature positioning. *Machine Learning with Applications*, *7*, Article 100196. http://dx.doi.org/10.1016/j.mlwa.2021.100196.

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. (Ph.D. thesis), Politecnico di Milano.

Drezner, T., & Drezner, Z. (2002). Validating the gravity-based competitive location model using inferred attractiveness. *Annals of Operations Research*, *111*(1–4), 227–237.

Drezner, T., & Drezner, Z. (2004). Finding the optimal solution to the Huff based competitive location model. *Computational Management Science*, *1*(2), 193–208.

Drezner, T., Drezner, Z., & Kalczynski, P. (2011). A cover-based competitive location model. *Journal of the Operational Research Society*, *62*(1), 100–113.

Drezner, T., Drezner, Z., & Kalczynski, P. (2012). Strategic competitive location: Improving existing and establishing new facilities. *Journal of the Operational Research Society*, *63*, 1720–1730.

Drezner, T., Drezner, Z., & Kalczynski, P. (2015). A leader–follower model for discrete competitive facility location. *Computers & Operations Research*, *64*, 51–59.

Drezner, Z., Suzuki, A., & Drezner, T. (2007). Locating multiple facilities in a planar competitive environment. *Journal of the Operations Research Society of Japan*, *50*(3), 250–263.

Du, K.-L., & Swamy, M. N. S. (2016). *Search and optimization by metaheuristics techniques and algorithms inspired by nature*. Switzerland: Springer International Publishing.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro machine and human science, Proceedings of the sixth international symposium on* (pp. 39–43). IEEE.

Erol, O. K., & Eksin, I. (2006). A new optimization method: Big bang–big crunch. *Advances in Engineering Software*, *37*(2), 106–111.

Esmaeili, M., & Hamedani, S. G. (2022). A competitive facility location problem using distributor Stackelberg game approach in multiple three-level supply chains. *International Journal of Applied Management Science*, *14*(3), 205–220.

Eusuff, M. M., & Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, *129*(3), 210–225.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Non-Linear Phenomena*, *22*(1–3), 187–204.

Fernández, J., Boglárka, G., Redondo, J. L., & Ortigosa, P. M. (2019). The probabilistic customer's choice rule with a threshold attraction value: Effect on the location of competitive facilities in the plane. *Computers & Operations Research*, *101*, 234–249.

Fernández, P., Pelegrín, B., Lančinskas, A., & Žilinskas, J. (2017). New heuristic algorithms for discrete competitive location problems with binary and partially binary customer behavior. *Computers & Operations Research*, *79*, 12–18.

Fernández, J., Salhi, S., & Boglárka, G. (2014). Location equilibria for a continuous competitive facility location problem under delivered pricing. *Computers & Operations Research*, *41*, 185–195.

Fernández, J., Tóth, B. G., Redondo, J. L., Ortigosa, P. M., & A.G., Arrondo (2017). A planar single-facility competitive location and design problem under the multi-deterministic choice rule. *Computers & Operations Research*, *78*, 305–315.

Ferreira, K. M., & de Queiroz, T. A. (2018). Two effective simulated annealing algorithms for the location-routing problem. *Applied Soft Computing*, *70*, 389–422.

Fischer, K. (2002). Sequential discrete p-facility models for competitive location planning. *Annals of Operations Research*, *111*(1–4), 253–270.

Fletcher, R. (1972). Methods for the solution of optimization problems. *Computer Physics Communications*, *3*(3), 159–172.

Formato, R. A. (2007). Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in Electromagnetics Research*, *77*, 425–491.

Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, *17*(12), 4831–4845.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, *76*(2), 60–68.

Gentile, J., Pessoa, A. A., Poss, M., & Roboredo, M. C. (2018). Integer programming formulations for three sequential discrete competitive location problems with foresight. *European Journal of Operational Research*, *265*(3), 872–881.

Ghaemi, M., & Feizi-Derakhshi, M. R. (2014). Forest optimization algorithm. *Expert Systems with Applications*, *41*(15), 6676–6687.

Ghaffarinasab, N., Motallebzadeh, A., Jabarzadeh, Y., & Kara, B. Y. (2018). Efficient simulated annealing based solution approaches to the competitive single and multiple allocation hub location problems. *Computers & Operations Research*, *90*, 173–192.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549.

Hajipour, V., Farahani, R. Z., & Fattahi, P. (2016). Bi-objective vibration damping optimization for congested location–pricing problem. *Computers & Operations Research*, *70*, 87–100.

Hajipour, V., Fattahi, P., Tavana, M., & Di Caprio, D. (2016). Multi-objective multi-layer congested facility location–allocation problem optimization with Pareto-based meta-heuristics. *Applied Mathematical Modelling*, *40*(7–8), 4948–4969.

Hajipour, V., Khodakarami, V., & Tavana, M. (2014). The redundancy queuing-location–Allocation problem: A novel approach. *IEEE Transactions on Engineering Management*, *61*(3), 534–544.

Holland, J. H. (1975). *Adaptation in natural and artificial systems* (2nd ed.). University of Michigan Press, MIT Press, 1992.

Hosseini, S., & Al Khaled, A. (2014). A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research. *Applied Soft Computing*, *24*, 1078–1094.

Hotelling, H. (1990). Stability in competition. In *The collected economics articles of Harold hotelling* (pp. 50–63). New York, NY: Springer.

Huff, D. L. (1964). Defining and estimating a trade area. *Journal of Marketing*, *28*, 34–38.

Ivanov, S. V., & Morozova, M. V. (2016). Stochastic problem of competitive location of facilities with quantile criterion. *Automation and Remote Control*, *77*(3), 451–461.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization, vol. 200*: *Technical report-tr06*, Erciyes university, engineering faculty, computer engineering department.

Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, *59*, 53–70.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mechanica*, *213*(3–4), 267–289.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Konak, A., Kulturel-Konak, S., & Snyder, L. (2017). A multi-objective approach to the competitive facility location problem. *Procedia Computer Science*, *108*, 1434–1442.

Konur, D., & Geunes, J. (2012). Competitive multi-facility location games with non-identical firms and convex traffic congestion costs. *Transportation Research Part E: Logistics and Transportation Review*, *48*(1), 373–385.

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, *4*(2), 87–112.

Küçükaydın, H., & Aras, N. (2020). Gradual covering location problem with multi-type facilities considering customer preferences. *Computers & Industrial Engineering*, *147*, Article 106577.

Küçükaydın, H., Aras, N., & Altınel, I. K. (2011). A discrete competitive facility location model with variable attractiveness. *Journal of the Operational Research Society*, *62*(9), 1726–1741.

Küçükaydın, H., Aras, N., & Altınel, I. K. (2012). A leader–follower game in competitive facility location. *Computers & Operations Research*, *39*(2), 437–448.

Kung, L. C., & Liao, W. H. (2018). An approximation algorithm for a competitive facility location problem with network effects. *European Journal of Operational Research*, *267*(1), 176–186.

Lančinskas, A., Fernández, P., Pelegín, B., & Žilinskas, J. (2017). Improving solution of discrete competitive facility location problems. *Optimization Letters*, *11*(2), 259–270.

Lančinskas, A., Fernández, P., Pelegrín, B., & Žilinskas, J. (2020). Discrete competitive facility location by ranking candidate locations. In G. Dzemyda, J. Bernatavičienė, & J. Kacprzyk (Eds.), *Studies in computational intelligence*: *vol. 869*, *Data science: new issues, challenges and applications*. Cham: Springer, http://dx.doi.org/10.1007/978-3-030-39250-5_8.

Lančinskas, A., Ortigosa, P. M., & Žilinskas, J. (2015). Parallel optimization algorithm for competitive facility location. *Mathematical Modelling and Analysis*, *20*(5), 619–640.

Lančinskas, A., Žilinskas, J., Fernández, P., & Pelegrín, B. (2020). Solution of asymmetric discrete competitive facility location problems using ranking of candidate locations. *Soft Computing*, *24*(23), 17705–17713.

Larrañaga, P., & Lozano, J. A. (2001). *Estimation of distribution algorithms: a new tool for evolutionary computation*. Berlin, Germany: Springer Science & Business Media.

Latifi, S. E., Tavakkoli-Moghaddam, R., Fazeli, E., & Arefkhani, H. (2022). Competitive facility location problem with foresight considering discrete-nature attractiveness for facilities: Model and solution. *Computers & Operations Research*, Article 105900.

Lee, G., & O'Kelly, M. E. (2009). Exploring locational equilibria in a competitive broadband access market: Theoretical modeling approach. *Journal of Regional Science*, *49*(5), 953–975.

Levanova, T. V., & Gnusarev, A. Y. (2020). Variable neighborhood search algorithms for a competitive location problem with elastic demand. *Journal of Applied and Industrial Mathematics*, *14*(4), 693–705.

Li, X., Zhang, T., Wang, L., Ma, H., & Zhao, X. (2020). A minimax regret model for the leader–follower facility location problem. *Annals of Operations Research*, 1–22.

Lin, Y. H., & Tian, Q. (2021a). Branch-and-cut approach based on generalized benders decomposition for facility location with limited choice rule. *European Journal of Operational Research*, *293*(1), 109–119.

Lin, Y. H., & Tian, Q. (2021b). Generalized benders decomposition for competitive facility location with concave demand and zone-specialized variable attractiveness. *Computers & Operations Research*, *130*, Article 105236.

Lin, Y. H., & Tian, Q. (2021c). Exact approaches for competitive facility location with discrete attractiveness. *Optimization Letters*, *15*(2), 377–389.

Ljubić, I., & Moreno, E. (2018). Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research*, *266*(1), 46–56.

Lüer-Villagra, A., & Marianov, V. (2013). A competitive hub location and pricing problem. *European Journal of Operational Research*, *231*(3), 734–744.

Luo, K. (2019). Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey. *Applied Soft Computing*, *77*, 225–235.

Ma, H., Guan, X., & Wang, L. (2020). A single-facility competitive location problem in the plane based on customer choice rules. *Journal of Data, Information and Management*, *2*(4), 323–336.

Mai, T., & Lodi, A. (2020). A multicut outer-approximation approach for competitive facility location under random utilities. *European Journal of Operational Research*, *284*(3), 874–881.

Marianov, V., Eiselt, H. A., & Lüer-Villagra, A. (2020). The follower competitive location problem with comparison-shopping. *Networks and Spatial Economics*, *20*(2), 367–393.

Marianov, V., Ríos, M., & Icaza, M. J. (2008). Facility location for market capture when users rank facilities by shorter travel and waiting times. *European Journal of Operational Research*, *191*(1), 32–44.

McGarvey, R. G., & Cavalier, T. M. (2005). Constrained location of competitive facilities in the plane. *Computers & Operations Research*, *32*(2), 359–378.

Mehdizadeh, E., & R., Tavakkoli-Moghaddam (2008). Vibration damping optimization. In *Proceedings of the international conference of operations research and global business, Germany, 3–5, September*.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, *1*(4), 355–366.

MirHassani, S. A., Raeisi, S., & Rahmani, A. (2015). Quantum binary particle swarm optimization-based algorithm for solving a class of bi-level competitive facility location problems. *Optimization Methods & Software*, *30*(4), 756–768.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Moscato, P. (1989). *On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms*: *Tech. Rep. Caltech concurrent computation program 826*, California Institute of Technology, Pasadena, California, USA.

Mucherino, A., & Seref, O. (2007). Monkey search: a novel metaheuristic search for global optimization. In *AIP conference proceedings, vol. 953, no. 1* (pp. 162–173). AIP.

Nasiri, M. M., Mahmoodian, V., Rahbari, A., & Farahm, S. H. (2018). A modified genetic algorithm for the capacitated competitive facility location problem with the partial demand satisfaction. *Computers & Industrial Engineering*, *124*, 435–448.

Neil, G., & C.L., Isaac (1998). *Quantum computing with molecules*. Scientific American.

Niknamfar, A. H., Niaki, S. T. A., & Niaki, S. A. A. (2017). Opposition-based learning for competitive hub location: A bi-objective biogeography-based optimization algorithm. *Knowledge-Based Systems*, *128*, 1–19.

Panin, A. A., Pashchenko, M. G., & Plyasunov, A. V. (2014). Bilevel competitive facility location and pricing problems. *Automation and Remote Control*, *75*(4), 715–727.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, *22*(3), 52–67.

Pelegrín-Pelegrín, B., Dorta-González, P., & Fernández-Hernández, P. (2011). Finding location equilibria for competing firms under delivered pricing. *Journal of the Operational Research Society*, *62*(4), 729–741.

Qi, M., Jiang, R., & Shen, S. (2022). Sequential competitive facility location: Exact and approximate algorithms. *Operations Research*, http://dx.doi.org/10.1287/opre.2022.2339.

Qi, M., Xia, M., Zhang, Y., & Miao, L. (2017). Competitive facility location problem with foresight considering service distance limitations. *Computers & Industrial Engineering*, *112*, 483–491.

Rahmani, A. (2016). Competitive facility location problem with attractiveness adjustment of the follower on the closed supply chain. *Cogent Mathematics*, *3*(1), Article 1189375, 1-19.

Rahmani, A., & Hosseini, M. (2021). A competitive stochastic bi-level inventory location problem. *International Journal of Management Science and Engineering Management*, *16*(3), 209–220.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*(3), 303–315.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248.

Rechenberg, I. (1973). *Evolutionsstrategie–optimierung technisher systeme Nach Prinzipien der biologischen evolution*: *[Evolution strategy: optimization of technical systems according to the principles of biological evolution]*, Stuttgart: Frommann-Holzboog Verlag.

Redondo, J. L., Fernandez, J., Garcia, I., & Ortigosa, P. M. (2008). Parallel algorithms for continuous competitive location problems. *Optimisation Methods & Software*, *23*(5), 779–791.

Redondo, J. L., Fernández, J., García, I., & Ortigosa, P. M. (2009). Solving the multiple competitive facilities location and design problem on the plane. *Evolutionary Computation*, *17*(1), 21–53.

Redondo, J. L., Fernández, J., Hervás, J. D. Á., Arrondo, A. G., & Ortigosa, P. M. (2015). Approximating the Pareto-front of a planar bi-objective competitive facility location and design problem. *Computers & Operations Research*, *62*, 337–349.

ReVelle, C. (1986). The maximum capture or sphere of influence location problem: Hotelling revisited on a network. *Journal of Regional Science*, *26*(2), 343–358.

Reynolds, R. G. (1994). An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming* (pp. 131–139). River Edge, NJ: World Scientific.

Rohaninejad, M., Navidi, H., Nouri, B. V., & Kamranrad, R. (2017). A new approach to cooperative competition in facility location problems: Mathematical formulations and an approximation algorithm. *Computers & Operations Research*, *83*, 45–53.

Sadjadi, S. J., Ashtiani, M. G., Ramezanian, R., & Makui, A. (2016). A firefly algorithm for solving competitive location-design problem: A case study. *Journal of Industrial Engineering International*, *12*(4), 517–527.

Saidani, N., Chu, F., & Chen, H. (2012). Competitive facility location and design with reactions of competitors already in the market. *European Journal of Operational Research*, *219*(1), 9–17.

Saif, U., Guan, Z., Liu, W., Zhang, C., & Wanga, B. (2014). Pareto based artificial bee colony algorithm for multi objective single model assembly line. *Computers & Industrial Engineering*, *76*, 1–15.

Santos-Peñate, D. R., Campos-Rodríguez, C. M., & Moreno-Pérez, J. A. (2020). A kernel search matheuristic to solve the discrete leader-follower location problem. *Networks and Spatial Economics*, *20*(1), 73–98.

Sasaki, M., Campbell, J. F., Krishnamoorthy, M., & Ernst, A. T. (2014). A Stackelberg hub arc location model for a competitive environment. *Computers & Operations Research*, *47*, 27–41.

Shan, W., Yan, Q., Chen, C., Zhang, M., Yao, B., & Fu, X. (2019). Optimization of competitive facility location for chain stores. *Annals of Operations Research*, *273*(1), 187–205.

Shiode, S., Yeh, K. Y., & Hsia, H. C. (2012). Optimal location policy for three competitive facilities. *Computers & Industrial Engineering*, *62*(3), 703–707.

Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009). A dolphin partner optimization. In *Global congress on intelligent systems, vol. 12* (pp. 4–128). IEEE.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713.

Snyder, L. V., & Daskin, M. S. (2005). Reliability models for facility location: The expected failure cost case. *Transportation Science*, *39*(3), 400–416.

Stepinski, T. F., & Dmowska, A. (2022). Machine-learning models for spatially-explicit forecasting of future racial segregation in US cities. *Machine Learning with Applications*, *9*, Article 100359. http://dx.doi.org/10.1016/j.mlwa.2022.100359.

Storn, R., & Price, K. (1997). Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Suárez-Vega, R., Santos-Peñate, D. R., & Dorta-González, P. (2004). Competitive multifacility location on networks: The (r| Xp)-medianoid problem. *Journal of Regional Science*, *44*(3), 569–588.

Tang, R., Fong, S., Yang, X. S., & Deb, S. (2012). Wolf search algorithm with ephemeral memory. In *Digital information management (ICDIM), 2012 seventh international conference on* (pp. 165–172). IEEE.

Wang, S. C., & Chen, T. C. (2017). Multi-objective competitive location problem with distance-based attractiveness and its best non-dominated solution. *Applied Mathematical Modelling*, *47*, 785–795.

Wang, S. C., & Chen, T. C. (2021). Using NSGA-II to solve multi-objective competitive location problem with cooperative coverage for distance-based attractiveness. *Journal of Intelligent & Fuzzy Systems*, *40*(4), 7723–7734.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169–178). Berlin, Heidelberg: Springer.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Studies in computational intelligence*: *vol 284*, *Nature inspired cooperative strategies for optimization*. Berlin, Heidelberg: Springer.

Yang, X. S. (2012). Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation* (pp. 240–249). Berlin, Heidelberg: Springer.

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *Nature & biologically inspired computing, NaBIC 2009. World congress on* (pp. 210–214). IEEE.

Yang, F. C., & Wang, Y. P. (2007). Water flow-like algorithm for object grouping problems. *Journal of the Chinese Institute of Industrial Engineers*, *24*(6), 475–488.

Yang, H., & Wong, S. C. (2000). A continuous equilibrium model for estimating market areas of competitive facilities with elastic demand and market externality. *Transportation Science*, *34*(2), 216–227.

Yazdani, M., & Jolai, F. (2016). Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*, *3*(1), 24–36.

Yu, W. (2020). Robust model for discrete competitive facility location problem with the uncertainty of customer behaviors. *Optimization Letters*, *14*(8), 2107–2125.

Yu, W. (2022). Robust competitive facility location model with uncertain demand types. *PLoS One*, *17*(8), Article e0273123.

Yu, X., Zhou, Y., & Liu, X.-F. (2019). A novel hybrid genetic algorithm for the location routing problem with tight capacity constraints. *Applied Soft Computing*, *85*, Article 105760.

Zarrinpoor, N., Fallahnezhad, M., & Pishvaee, M. (2016). The reliable hierarchical location–allocation model under heterogeneous probabilistic disruptions. *International Journal of Engineering-Transactions A: Basics*, *29*(10), 1401–1411.

Zarrinpoor, N., & Seifbarghy, M. (2011). A competitive location model to obtain a specific market share while ranking facilities by shorter travel time. *International Journal of Advanced Manufacturing Technology*, *55*(5–8), 807–816.

Zeigler, D. (2014). *Evolution: components and mechanisms*. Academic Press. Amsterdam and Boston (Massachusetts), Elsevier. Inc..

Zhang, Y., Snyder, L. V., Ralphs, T. K., & Xue, Z. (2016). The competitive facility location problem under disruption risks. *Transportation Research Part E: Logistics and Transportation Review*, *93*, 453–473.