Bi-sided facility location problems
an efficient algorithm for k-centre, k-median, and travelling salesman problems

Davoodi, Mansoor; Rezaei, Jafar

DOI
10.1080/23302674.2023.2235814

Publication date
2023

Document Version
Final published version

Published in
International Journal of Systems Science: Operations and Logistics

INTERNATIONAL JOURNAL OF SYSTEMS SCIENCE: OPERATIONS & LOGISTICS
2023, VOL. 10, NO. 1, 2235814
https://doi.org/10.1080/23302674.2023.2235814

Taylor & Francis
Taylor & Francis Group

Check for updates

Bi-sided facility location problems: an efficient algorithm for *k*-centre, *k*-median, and travelling salesman problems

Mansoor Davoodi [a,b,c] and Jafar Rezaei [c]

[a]Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran; [b]Center for Advanced Systems Understanding (CASUS), Helmholtz-Zentrum Dresden Rossendorf (HZDR), Görlitz, Germany; [c]Faculty of Technology, Policy and Management, Delft University of Technology, Delft, The Netherlands

ABSTRACT

This study introduces a general framework, called *Bi-sided facility location*, for a wide range of problems in the area of combined facility location and routing problems such as locating test centres and designing the network of supermarkets. It is based on a multi-objective optimisation model to enhance the service quality which the clients received, called *client-side*, and enhance the interconnection quality and eligibility among the centres, called *center-side*. Well-known problems such as *k*-median and *k*-centre for the client-side and the travelling salesman problem for the centre-side are taken into account in this paper. After discussing the complexity of this kind of combination, we propose a heuristic approximation algorithm to find approximation Pareto-optimal solutions for the problem. The algorithm is an efficient local search utilising geometric objects such as the Voronoi diagram and Delaunay triangulation as well as algorithms for computing approximation travelling salesman tour. In addition to the comprehensive theoretical analysis of the proposed models and algorithm, we apply the algorithm to different instances and benchmarks, and compare it with NSGA-II based on set coverage and spacing metrics. The results confirm the efficiency of the algorithm in terms of running time and providing a diverse set of efficient trade-off solutions.

KEYWORDS

Facilities planning; local search; approximation; routing; connected facility location; travelling salesman problem

Highlights

- Introducing a general bi-side location model considering centres and clients' utilities
- Discussing and proving the NP-hardness of the model in the general framework
- Considering two instances; *k*-centre and *k*-median for client-side and TSP for centre-side
- Proposing an efficient geometric-based algorithm for solving the problems
- Implementing, testing, and comparing the proposed algorithm on several benchmarks

1. Introduction

The service quality provided by service centres like companies, hospitals, and government agencies is a primary yet crucial criterion in their success in today's competitive world. The location of service centres and the connections between them are two important interrelated issues in deploying such facility centres. Service providers (or centres) and customers (or clients) develop complex networks based on two types of connections; connections between the clients and centres (Barbati, 2013; Daskin, 2001), and the connections among the centres (Javid & Azad, 2009; Verhetsel et al., 2015). To manage those connections through high-quality services and at low costs, the service centres have to be located in an optimal position (Onstein et al., 2020). Taking into account the location of clients, the problem of finding that right is known as the facility location, e.g. the well-known problems *k*-median and *k*-centre (Oded & Hakimi, 1979; Thorup, 2005). Further, transportation planning involves an analysis of the connections among the centres and identifying the proper transportation (Bertsimas, 1989). While in many real-world problems one of the two connection types might be the focus of a decision-maker, there are problems in which the two need to be taken into account simultaneously, where focusing on either of the two could lead to sub-optimal solutions.

CONTACT Mansoor Davoodi mdmonfared@iasbs.ac.ir Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran; Center for Advanced Systems Understanding (CASUS), Helmholtz-Zentrum Dresden Rossendorf (HZDR), Görlitz, Germany; Faculty of Technology, Policy and Management, Delft University of Technology, 2628 BX, Delft, The Netherlands

In order to better understand the necessity of simultaneous consideration of the client's and centre's objectives, assumes the nowadays problem of locating some centres for COVID-19 to provide diagnostic testing centres for their citizens. For individuals to get tested, the distance or travelling cost between their home/work and the test centres is important. Thus, one might analyse the situation and formulate the goal being to deploy a set of centres (e.g. k centres) minimising the distance between each individual and its nearest test centre. This goal can be formulated as minimising the average distance between the clients and their nearest centres, i.e. k-median, or as minimising the maximum distance, i.e. k-centre. On the other hand, the test centres need to be equipped with medical and laboratory equipment and also deliver the tests regularly in a particular period of time, e.g. daily. To this end, a method is one or more vehicles with particular containers that collect all the tests from the centres one by one and deliver them to some diagnostic laboratories. So, a second objective could be formulated as minimising the travelling cost between the test centres, i.e. minimising the tour meets the centres, like the objective of the Traveling Salesman Problem (TSP). We could think of many other examples where the two types of connections need to be considered simultaneously, such as a network of a supermarket chain (see, for example, https://www.dairyfreshfarms.com), raw milk collection (Gheisariha et al., 2023), locating the cantors of a post company (Meira et al., 2017) or waste collections (Adeleke & Olukanni, 2020).

In general, there are two different types of costs and objectives when deploying a set of facility centres. The two types of objectives are (i) the objective to enhance utilities for the clients, called *client-side*, and, (ii) the objectives to improve the interconnection quality and eligibility between the centres, called *center-side*. The first type provides a utility and satisfaction level among the clients (Drezner & Hamacher, 2004; Farahani & Hekmatfar, 2009), while, the second type provides a low-cost and high-quality service for suppliers (Daskin et al., 2005).

Since, usually, the type, size, and scale of the objectives of the two sides are different, it is not sensible to integrate them into one single objective. For example, the distance between the clients and centres (e.g. from home to a post office) may be a walking distance which is not translated to monetary costs but a degree of satisfaction. However, the distance between the centres (e.g. post offices) is a cost paid by service providers as a vehicle routing cost. As the nature of objectives for the sides is different it does not make sense to develop a unifying objective function for the whole problem. Even in cases where the objective functions of the two sides seem similar (e.g. cost paid by clients and cost paid by suppliers), it is not rational to sum up the two costs as (i) each cost is paid by one actor (one side of the problem), (ii) the value of one unit of cost is not similar for both sides. For instance, while having a marginal cost of 5 euros might be perceived as a high amount of money, a service provider looks at that as almost nothing. While we argue that it is important to look at the objectives of the two sides separately (which avoids us using a unifying objective function), solving the two problems separately could lead to sub-optimal solutions. Therefore, in this paper, we bring both client-side and centre-side facility location problems together in a combined framework of a *bi-sided optimization problem*. While in general, the framework allows for other possibilities of combinations, we apply the most well-known and application-based formulation of objectives such as k-median, k-centre for the client-side, and TSP for the centre-side. This combined problem aims to open a set of k centres whose distance from the clients is minimised and simultaneously they can be connected with minimum tour length. While studying the two sides individually has a long history and many valuable studies have been conducted in recent decades, we think considering the two sides simultaneously could bring even more value to the picture. This is especially true for cases in which the objectives of the two sides are the concern of a decision-maker. For instance, in the case of locating post offices, this is the postal company that is concerned about the objectives of the two sides. That is, while the postal company is concerned about their transportation costs collecting the items from different offices, they are also concerned about their clients walking from their homes to the post offices. We think, in practice, most problems of this kind are such that the service provider is concerned about the objectives of the two sides. This necessitates considering the problem in a combined framework as it is only when the decision-maker can find solutions that make a desirable balance/tradeoff among the objectives of the two sides of the problem. Despite the significance of such a view in the formulation of the problem, unfortunately, however, we were not able to find a systematic study that focuses on such objectives simultaneously in the framework of a multi-objective optimisation problem, which is the main aim of the current study. We think this is a significant contribution to the theory and practice of facility location and routing problems.

As all of these problems belong to the complexity class of NP-hard, the model is computationally NP-hard as well. We deeply discuss the complexity of the model and show the strongly NP-hardness of the problem by proving that the size of Pareto-optimal solutions in the general version of the problem is exponential. Then, utilising observations in the search space of the problem and useful geometric objects such as the Voronoi diagram and Delaunay triangulation, and approximation algorithms, we propose an efficient and customised heuristic algorithm for solving the problem. The algorithm is presented in the structure of a multi-objective optimisation, and its outcome is a set of non-dominated solutions close to the Pareto-optimal solutions of the problem. Finally, we implement the algorithm and apply it to a set of random test problems and some benchmarks. We also compare the algorithm with NSGA-II approach based on *set coverage* metric and spacing metric. The results confirm the efficiency of the algorithm in finding a set of diverse solutions near the Pareto-optimal solutions and its effectiveness in terms of computational time. In some comparisons, the algorithm outperforms twice in terms of the mentioned metrics.

This paper is organised into seven sections. In Section 2, we comprehensively review the related studies in the literature based on the presented framework. In Section 3, we formally define the bi-sided optimisation facility location model. In Section 4, we introduce some preliminaries such as multi-objective optimisation and related concepts, geometric structures, and approximation algorithms used throughout this paper. In Section 5, we first discuss the complexity of the model for the case the client-side follows the objective of the k-median problem (min-sum) and the centre-side follows the objective of TSP (minimum tour). Then, we propose our general algorithm to solve the problem as well as the objective function of the k-centre problem. In Section 6, we implement the algorithm and apply it to several test examples and benchmarks. Finally, we conclude the study in Section 7.

2. Related work

In this section, we reviewed, the most related studies in the area of facility location and routing. Two well-known facility location problems with many real-world applications are k-median and k-centre. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ be a set of clients, Q be a set of potential facility centres. In the primitive version of the k-median, the goal is to choose $k = |C|$ centres such that the following objective function is minimised:

$$Cost(C) = \sum_{p \in \mathcal{P}} d_{PQ}(p, \delta(p)), \tag{1}$$

where $d_{PQ}(p, q)$ denotes the distance between a client and a centre and $\delta(p)$ denotes the nearest open centre to p. Similarly, the objective of the k-centre problem with the applications of emergency FL (Wang, Miao, et al., 2021) can be defined as minimising the following function.

$$Cost(C) = \max_{p \in \mathcal{P}} d_{PQ}(p, \delta(p)). \tag{2}$$

Because of the definition of $\delta(p)$, the problems of k-median and k-centre are not assignment problems (Bateni & Hajiaghayi, 2012). That is, by choosing C, each client automatically is assigned to its nearest centre and is served by it. However, the NP-hardness of both k-median and k-centres has been proved in such defined simple versions even in metric space (Hochba, 1997), numerous variations of them such as capacitated (Khuller & Sussmann, 2000), weighted and constrained (Daskin, 2001), Fault-tolerant (Fernandes et al., 2018) have been introduced and solved using greedy (Jain et al., 2002), heuristic (Drezner & Hamacher, 2004), approximation (Gupta, 2018; Krishnaswamy et al., 2018) and mathematical programming approaches (Chakrabarty & Swamy, 2017; Li, 2017). Among many variations of the k-centre problem, a related version of the k-centre is called the α-connected bi-centre problem (Huang et al., 2003). In this 2-centre problem, it is asked to serve the clients using two centres such that the length of the farthest client from its nearest centre is minimised under the constraint that the distance between the centres is less than a given specified bound.

Among the diverse variety of FL problems, connected FL (CFL) (Eisenbrand et al., 2010; Swamy & Kumar, 2004; Turkoglu & Genevois, 2020) is more related to the client-side and centre-side facility location introduced in this paper. In addition to the opening cost of the centres, CFL aims to minimise the summation of the *Steiner cost* and *connection cost* as well. Formally, the objective function is

$$CFL : Cost(C) = \sum_{c \in C} open(c) + \rho \sum_{p \in \mathcal{P}} d_{PQ}(p, \sigma(p)) + \sum_{e=(q,q') \in ST} d_{QQ}(q, q'), \tag{3}$$

where $open(c)$ is the opening cost of centre c, $\sigma(p)$ denotes a centre that serves p, $d_{PQ}(p, q)$ shows the length (cost) of the connection between p and q. Also, in CFL, the opened facilities must be connected, so, ST shows the Steiner tree connects the centres, and consequently, the third term in Eq.(3) is the total cost of ST. Finally, the parameter ρ is a balancing or priority weight of the objectives. Note that, it is possible that $\sigma(p) \neq \delta(p)$, and consequently, CFL can be interpreted as an assignment problem. CFL has several applications such as for installing a telecommunication network infrastructure (Swamy & Kumar, 2004). Figure 1 (a) shows a schematic view of the CFL problem.

Swamy and Kumar (Swamy & Kumar, 2004) presented the first primal-dual approach for CFL when

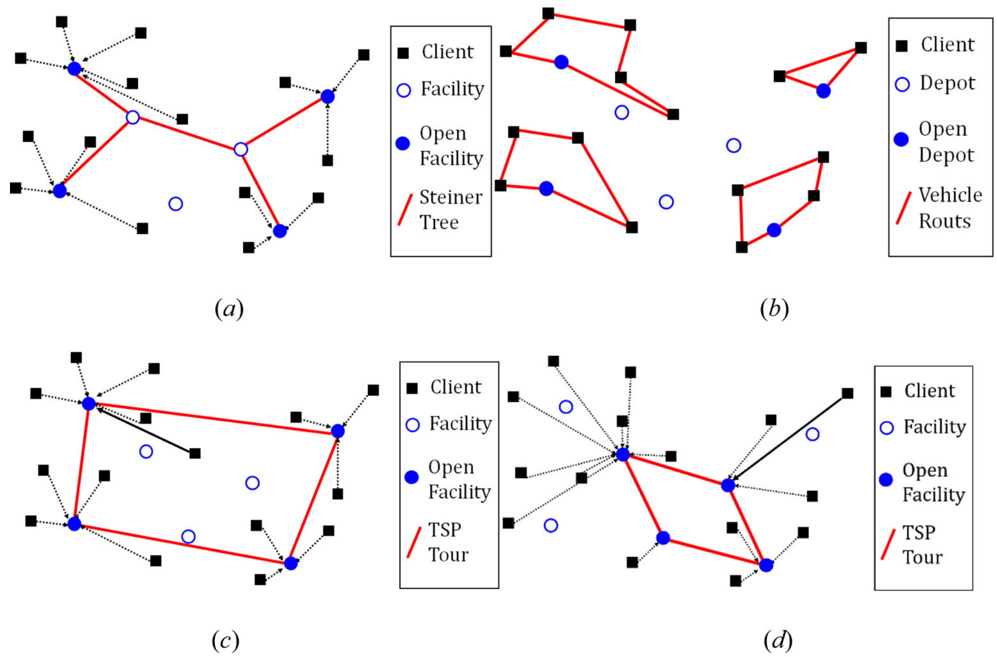


Figure 1. Connected Facility location problem (*a*), Location Routing Problem (*b*), Client-side and Center-side FL problem (*c*) and (*d*). The dashed edges in subfigures show the connection distance (cost) between each client and its assigned centre (e.g. the nearest opened centre). So, the problem can be seen as the *k-median&TSP* (for $k = 4$) in the framework of client-side and centre-side FL. However, if the maximum length (the solid black connection) is considered on the client-side, the problem will be *k-center&TSP*. Note that, subfigure (*c*) and (*d*) shows the optimal solutions for the Client-side and Center-side FL problems, respectively.

the opening costs are zero and $Q = \mathcal{P}$. They improved the previous results and proposed two approximation algorithms with the factors 8.55 and 4.55 for the particular cases of CFL. These approximation factors are also improved by Eisenbrand et al. (Eisenbrand et al., 2010) to 4 and 2.92, respectively. To this end, they used a new analytical tool, called *core detouring*, and proposed a randomised algorithmic framework. Gollowitzer and Ljubic´ (Gollowitzer & Ljubić, 2011) formulated CFL in the structure of different mixed integer programming models and proposed branch-and-cut algorithms to solve them. However, other variations of CFL such as Online CFL (San Felice et al., 2014) and Incremental CFL (Arulselvan et al., 2019), formulated the problem in one single objective using a weighted linear combination of three types of costs, opening cost, connection cost and Steiner cost. Regarding the introduced client-side and centre-side facility location problem, the opening cost and Steiner tree cost in CFL are two examples of centre-side objective functions, and the connection cost is an example of the client-side objective functions.

Vehicle Routing Problem (VRP) and its general version, Location Routing Problem (LRP), are other related problems that can be discussed as centre-side and client-side facility location problems. The goal of LRP is to open a set of centres (say *depots*) and assign a set of clients to the opened depots, however, the clients are served using a set of vehicles starting with the opened depots and visiting the clients. In general, the objective is to minimise the total cost (usually as a function of distance) of the vehicle's routes, opening costs, and the fixed costs per vehicle. LRP is a specific combination of facility location and transportation problems and has been significantly studied (Drexl & Schneider, 2015; Laporte, 1989, 2009; Nagy & Salhi, 2007; Prodhon & Prins, 2014; Schiffer et al., 2019). Both of the objectives in LRP focus on the client-sides; optimally locating the depots and optimally serving the clients by vehicle tours starting from the depots. So, the problem is an assignment problem with a combination of FL and TSP. Figure 1 (*b*) shows a schematic view of the problem. So many variations of LRP as well as multi-objective models of the problem, dynamic, online, and multi-echelon have been proposed and solved using linear programming relaxation, branch-and-cut, and heuristic approaches (Drexl & Schneider, 2015; Jaigirdar et al., 2023; Moon et al., 2020; Sluijk et al., 2023). Tordecilla et al. (Tordecilla et al., 2023) explored a version of the LRP with different capacity facilities and presented three mixed-integer linear formulations. The VRP can be seen as a specific case of LRP which concerns a routing problem for a fixed depot. One could also argue that VRP is a general case of TSP where the goal is to visit all the clients using some vehicle routes starting and ending at the depot. Similar to LRP, numerous variations of VRP as well as different algorithms have been proposed for solving them (Fahmy & Gaafar, 2023; Lin et al., 2014; Wang, Miao, et al., 2021).

The ring star problem (Calvete et al., 2016; Labbé et al., 2004; Liefooghe et al., 2010) is one of the other related problems to our study. Indeed, a particular case of the proposed model in this paper will cover the ring problem as well. The input of the ring problem is a weighted graph and the goal is to find a tour (i.e. ring) that minimises two costs: the cost of the ring and the cost of assigning the vertices to the ring. The assigning cost is computed only for the vertices out of the ring and it is the minimum distance between the vertices and a vertex on the ring. In this problem, the vertices are assumed to be the same and both the costs are of the same type and scale. Labbé et al. (2004) introduced the ring problem and mentioned some applications for the problem such as designing telecommunications network planning rapid transit systems. They formulated the problem in the framework of a mixed-integer linear programming model and proposed a branch-and-cut algorithm to solve it. Liefooghe et al. (2010) considered the ring problem in the framework of a bi-objective optimisation model and presented multi-objective heuristic algorithms to solve it. Calvete et al. (2016) inserted a depot node into the problem as the starting and ending point of the ring and presented a multi-objective evolutionary algorithm to solve it.

In addition to the mentioned well-known problems, some studies are based on LRP which is customised for a particular application by inserting an extra objective. Tavakkoli et al. (Tavakkoli–Moghaddam et al., 2010) presented an integrated bi-objective optimisation model for the problem of multi-depot location routing with the objectives of maximising demand served and minimising the cost of opening and delivery. They utilised a multi-objective scatter search algorithm to find non-dominated solutions to the problem. Martínez et al. (Martínez–Salazar et al., 2014) considered an extension of bi-stage LRP, that is, a bi-objective optimisation problem with the objectives of minimising the distribution cost and maximising the balance of workloads for drivers. They used local search and evolutionary algorithms to solve the problem. See (Pacheco et al., 2013; Rayat et al., 2017; Tavakkoli–Moghaddam et al., 2013) as some extensions of LRP. Alijani et al. (Alijani et al., 2017) presented an approximation algorithm for a matching problem between online buyers and sellers, called a two-sided facility location. Delfani et al. (Delfani et al., 2021) presented a hazardous waste location-routing model to consider both the risks of transportation and population.

3. Bi-sided facility location problem

Let $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ be a set of clients and $Q = \{q_1, q_2, \dots, q_m\}$ be a set of potential facilities. Also, suppose a fully connected network on $\mathcal{P} \cup Q$, that is, there is a distance (or cost) function $d_{\mathcal{P}\mathcal{P}} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}^{\geq 0}$ between any pair in $\mathcal{P}$, a distance (or cost) function $d_{QQ} : Q \times Q \to \mathbb{R}^{\geq 0}$ between any pair in Q, and a distance (or cost) function $d_{PQ} : P \times Q \to \mathbb{R}^{\geq 0}$ between any client in $\mathcal{P}$ and any potential centre in Q. The goal is to choose a set of facility centres $C \subseteq Q$ such that both objectives on the client-side and centre-side are optimised. Let $\delta(p) \epsilon C$ denote the centre which serves p. In the existing literature, problems such as k-median and k-centre have been significantly studied to minimise the service cost among the facility centres and clients. Also, well-known problems such as the travelling salesman problem (TSP), aim at minimising the movement cost among the facilities or customers. In designing a service system, it is possible that first a location analysis is performed and the optimal locations are determined, and then, an optimal planning strategy, i.e. a minimum TSP tour, is computed for optimal movement among the locations. Unfortunately, such a consecutive strategy is not optimal in general, and considering both locating and planning phases simultaneously is important. Among the studies in the literature, the problems of Connected FL (CFL), Vehicle Routing Problem (VRP), and Location Routing Problem (LRP) aims at addressing this issue in a single objective optimisation problem, however, as discussed, based on the application of the problem type utilities and objectives might be different in the client-side and centre-side. Thus integrating them into a single optimisation problem is not plausible in general. In the *bi-sided facility location problem*, the goal is to locate (or say openly) a set of facility centres $C \subseteq Q$ such that both objectives on the client-side and centre-side are optimised, or precisely,

$$\begin{cases} \mathit{minimize}\ F_1(C, P) \\ \mathit{minimize}\ F_2(C) \\ \mathit{subject\ to:} \\ G(C, P) \leq 0 \\ H(C) \leq 0 \\ C \subseteq Q \end{cases} \tag{4}$$

$F_1(C, P)$ shows the objective set on the client-side. For example, minimising the total setup cost, minimising the longest distance from the existing facilities, minimising fixed cost, minimising total annual operating cost, maximising service, minimising average or maximum distance travelled, minimiding the number of located facilities, and maximising responsiveness (Farahani et al., 2010). k-median and k-centre are two well-known problems of this type (Arya et al., 2004). Let $\delta(p) \epsilon C$ denote the opened centre to which client p is assigned, e.g. the nearest opened facility to p. So, the objective of k-median is to minimise $F_1(C,P) = \sum_{i=1}^{n} d(p_i,\ \delta(p_i))$, where $d(.,.)$ is a proper measurement distance function, e.g. Euclidean distance or Manhattan distance. Similarly, it can be defined as $F_1(C,P) = \max_{1\leq i\leq n} d(p_i,\ \delta(p_i))$ in the form of a k-centre problem. In addition to k-median and k-centre problems, TSP is another well-known problem on this side that has been extensively applied in VRP (Toth & Vigo, 2002) and LRP (Drexl & Schneider, 2013; Prodhon & Prins, 2014). In these problems, $F_1(C,P)$ is to minimise the length of one or more tours such that visit all the clients by starting an opened centre in C. Also, the demand weight of each client or any other objective and service priority related to the clients can be applied to $F_1(C,P)$. The objective set $F_2(C)$ shows the cost on the centre-side. Again TSP and Steiner Tree Problem (STP) (Gouveia et al., 2011) are two interesting problems on this side that have been extensively applied to Connected Facility Location (CFL) problems (Arulselvan et al., 2019; Gollowitzer & Ljubić, 2011), e.g. the goal is to minimise connection cost among the centres. Another interesting example of $F_2(C)$ is the problem of k-balance (Davoodi, 2019 Marín, 2011;), which aims at making a balance among the workloads of the centres, e.g. the difference between the maximum and minimum workload is minimised. Besides, the opening cost for each centre, which is extensively used in the literature, can be incorporated into this side of the objective. $G(C,P)$ shows the constraints on the interconnected transition between the clients and the centres. For example, maximum service capacity (Melkote & Daskin, 2001), limits on serving some particular clients using some particular centres, maximum service distance for the clients, and priority among the clients (Ravi & Sinha, 2004). Finally, $H(C)$ shows the set of constraints related to the centres. For example, the serving capacity of the centres, maximum service distance for the centres (Martínez–Salazar et al., 2014), limitations in locating the centres (Davoodi & Mohades, 2011), and maximum budget-constrained facility location (Wang et al., 2003).

Thus, it is possible to choose different objective functions for the client-side and centre-side of the introduced bi-sided facility location model. Inspired by several real-world problems, in this paper, we choose two well-known objectives *min-sum* which is the objective function of the k-median problem and *min–max* which is the objective function of the k-centre problem for the client-side of the model, and minimising the length of the visiting tour which is the objective function of the travelling salesman problem (TSP) for the centre-side of the model. Consequently, two new bi-objective optimisation location-routing problems are presented and we denote them by *k-median&TSP* and *k-center&TSP*, respectively. Figure 1 (c) and Figure 1 (d) show two different solutions to these problems. Figure 1 (c) (compared to Figure 1(d)) represents a TSP tour with a larger length but smaller service distance for the clients, which shows a trade-off between the two objectives.

It is notable that, since the type, measurement, and scale of the objectives on the client-side and the centre-side of the model may be completely different, the weighted summation of them and obtaining a single objective optimisation problem is not plausible always. For example, the cost of opening a centre is the type of cost paid by the service provider and it can be measured on the scale of the service provider's cost. While the service cost between the clients and centres is the type that may be paid by clients (like the example of the post office or waste collection). Therefore, unifying such different types of cost functions is not sensible, and the problem is a general bi-objective optimisation problem with different Pareto-optimal solutions. We show that the size of the Pareto-optimal solutions for combinations of the objectives may be exponential. As a result, the problems are strongly NP-hard. So, we propose an efficient bi-objective local search algorithm that solves both combinations of the model, denoted by *k-center&TSP* and *k-median&TSP*. We consider an uncapacitated variation of the problem and ignore the opening cost of centres. Indeed, the costs between the clients and centres and the length of the tour between the opened centres are paid regularly (e.g. daily) until the system works, however, the opening cost is paid once. The algorithm presents a set of non-dominated solutions, and at the final step, it is possible to choose one of them to apply in the real world. This can be performed using some higher-level information by decision-makers or choosing automatically using existing approaches such as the *knee point* solution (Jurgen et al., 2004), and Kalai- Smorodinsky solution (Gaudrie et al., 2018; Kalai & Smorodinsky, 1975).

4. Algorithmic preliminaries

This section contains three following subsections which are covered in Appendix (A) (Figure 2).

4.1. Bi-objective optimisation and Pareto-optimal Solutions
4.2. Voronoi Diagram and Delaunay Triangulation
4.3. Heuristic and Approximation Algorithms

5. Bi-objective optimisation algorithm

In this section, first, we show that the introduced client-side and centre-side facility location model is strongly

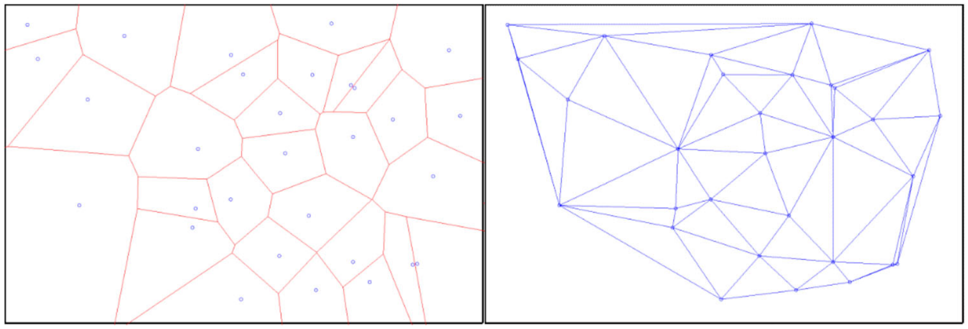


Figure 2. Voronoi diagram (left) and Delaunay Triangulation (right) of a set of 30 points

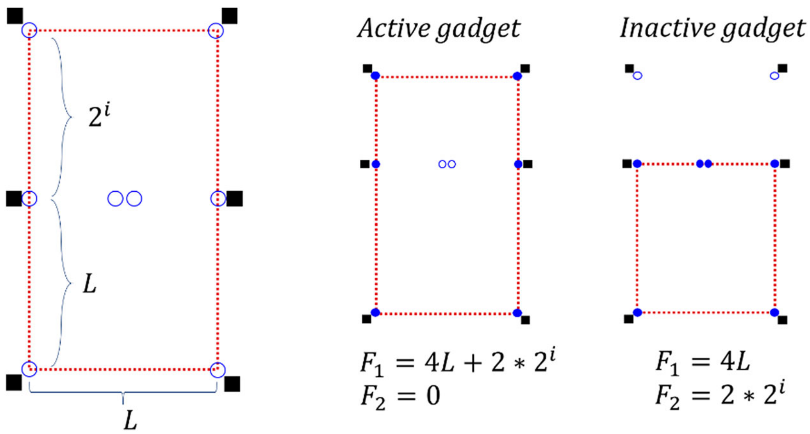


Figure 3. A gadget containing 8 potential facility points (hollow circles) and 6 demand points (solid squares). If we set $k = 6$ (pick 6 of 8 facilities), there are four different Pareto solutions. Two Pareto solutions are illustrated.

NP-hard in general. To this end, we choose k-median for the client-side and TSP for the centre-side of the model and show that the number of Pareto-optimal solutions is exponential. Then, we propose our algorithm to solve both *k-median&TSP* and *k-center&TSP* problems.

5.1. Size of Pareto-optimal solutions in k-median&TSP problem

In the following, we consider the objective of minimising the sum of the distance between the clients and their corresponding centres (which is the nearest opened centre) and minimising the length of the tour visiting the opened centres.

Theorem 5.1: *In the problem of k-median&TSP, the number of Pareto-optimal solutions in the worst case is $\Omega(2^p)$, where p is in order of the problem's size.we*

Proof: See Appendix (B) (Figures 3 and 4). ■

5.2. An heuristic approximation algorithm for k-median&TSP and k-centre&TSP problems

In this section, we propose an efficient heuristic algorithm for *FL&TSP* using Delaunay triangulation and Christofides Algorithm. Before going into details, we explain it roughly. The algorithm starts with a population of random solutions. It computes the values of the objective of the solutions. To this end, we use $\frac{3}{2}$-approximation algorithm for finding TSP tour. Then, algorithms exploit non-dominated solutions and generate a new population of solutions using the Delaunay triangulation graph of all potential facilities. This process is repeated to achieve a set of admissible non-dominated solutions.

In the initialisation step, we generate a population of N random solutions. For each solution $C = \{c_1, c_2, \dots, c_k\} \subseteq \mathcal{F}$ we determine their objective values by using VD of C and its $\frac{3}{2}$-approximation TSP tour using the Christofides algorithm. Then, we generate a new random neighbour solution $C' = DTN(C)$. So, we obtain $2N$ solutions in total. This is the exploring phase and we need to exploit and enhance the average fitness of the population by surviving N best solutions. To this end we utilise a non-dominated ranking procedure, that is, decomposing all $2N$ solutions to some non-domination fronts Γ_i, for $i = 1, 2, \dots, etc.$ By applying the domination principle, all the solutions that lie on a front are non-dominated and for each solution sol_1 in Γ_i, there is a solution sol_2 in Γ_j (for all $j < i$) so that sol_2 dominates sol_1. For a set of N solutions in a bi-dimensional objective

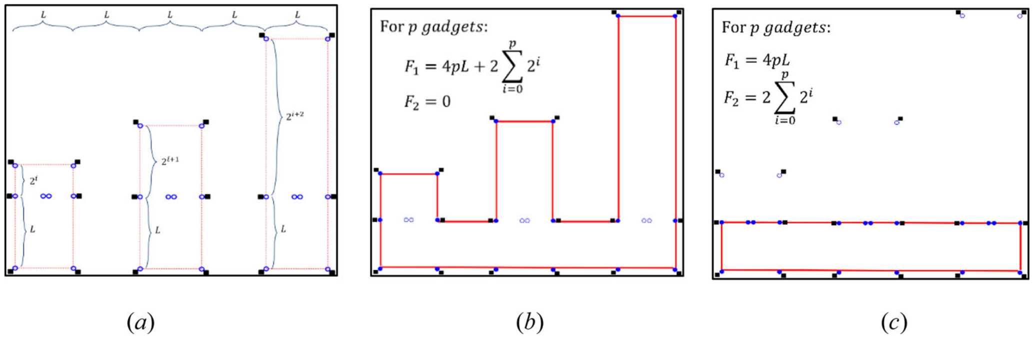


Figure 4. (*a*) A workspace with $p = 3$ gadgets which contains $n = p * 6 = 18$ demand points and $m = p * 8 = 24$ potential facility points. If we set $k = p * 4 = 12$, we have at lease $2^p = 8$ different Pareto-optimal solutions. There are two possibilities for each gadget. Two extreme Pareto solutions are shown in the figure: (*b*) all the gadgets are *active*, (*c*) all the gadgets are *inactive*.

space, the non-dominated sorting can be computed in $O(N \log N)$ time (Jensen, 2003). Finally, we select the N solutions by considering their non-domination fronts. We also apply a *crowding operator* to achieve the second goal of multi-objective optimisation problems, i.e. the *diversity* among the obtained solutions. There are several approaches to hand diversity (Coello et al., 2002), e.g. for any solution $C \in \Gamma_i$ in the objective space, we can compute the perimeter of the largest axis-aligned bounding box that contains C and no other solution of Γ_i. The value of the perimeter shows the distance of solution C from its left and right nearest solutions. This step can be also handled in $O(N \log N)$ time by having the ordered lists of the solutions according to the values of objectives.

After exploiting and reducing the size of the population to N, we iterate the above process until the termination condition is met, i.e. iterating the process for a predefined maximum number of generations or iterating until the set of non-dominated solutions of the population (Γ_1) does not change after some generations. The pseudo-code of the algorithm is proposed below.

Bi-Objective TSP Facility Location Problem

Input: Set of clients $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, set of potential facilities $\mathcal{F} = \{q_1, q_2, \ldots, q_m\}$ and an integer $1 < k < m$.
Output: Set of non-dominated solutions for the problem of FL&TSP. Each solution is a set $C = \{c_1, c_2, \ldots, c_k\} \subseteq Q$.

Step 0. (*Parameter setting)** Set the number of solutions (N) and the maximum number of iterations (M), and set $t = 0$ as the iteration counts. Also, Compute $DT(Q)$.
Step 1. (*Initialization*) Let S_t be a set of N randomly selected approximation solutions.
Step 2. (*Computing objective values*)
2(*a*). For each solution $C \in S_t$, compute $VD(C)$ and objective value $F_1(C)$.
2(*b*). For each solution $C \in S_t$, find a $\frac{3}{2}$-approximation TSP tour using Christofides algorithm. Set the length of the tour as $F_2(C)$.
Step 3. (*Local search*) generate S'_t using S_t.
For any solution $C \in S_t$, generate a random solution $C' = DTN(C)$ and insert C' to S'_t. Compute $VD(C')$ using $VD(C)$ and compute (update) the objective values of C'.

(*continued*).

Step 4. (*Selection*)
4(*a*). Perform a non-dominated sorting to $S'_t \cup S_t$, and identify different fronts Γ_i, for $i = 1, 2, \ldots, etc.$
4(*b*). Set a new population $S_{t+1} = \emptyset$ and $i = 1$.
Until $|S_{t+1}| + |\Gamma_i| \leq N$ perform $S_{t+1} = S_{t+1} \cup \Gamma_i$ and $i = i + 1$.
4(*c*). For reminder capacity in S_{t+1}, perform a crowding operator and fill it using the most diverse solutions in Γ_i.
Step 5: (*Termination Condition*) Set $t = t + 1$. If $t > M$, terminates the algorithm and report the non-dominated solutions of S_t. Otherwise go to Step 3.

**Based on our results, an efficient setting is $N \cong 2(m + k)$ and $M \cong km$.*

5.3. *Analysis of the algorithm*

See Appendix (*C*) (Table 1).

6. Simulation and experimental results

In this section, we first apply the proposed algorithm to some randomly generated instances of k-centre&TSP and k-median&TSP problems. Then, we use a set of benchmarks in the literature and show the efficiency of the algorithm. All the tests are simulated using *C#* programming language and run on a computer with 4.00 GB RAM and 3.30 GHz CPU. First, we consider a rectangular workspace with a size 1500×1000. We randomly pick $n = 1000$ clients and $m = 50$ potential points. Figure 5 shows the workspace. We run the proposed algorithm with $N = 2(m + k)$, and $M = mk$ for different values of

Table 1. Complexity time of the proposed algorithm.

Step	Worst case	Expected case
Step 1	$N * O(k)$	$N * O(k)$
Step 2(*a*)	$N * O(k^{1.5} log^5 k)$	$N * O(k^{1.5} log^5 k)$
Step 2(*b*)	$N * O(klogk + nlogk)$	$N * O(klogk + nlogk)$
Step 3	$N * O(k^{1.5} log^5 k + n \log k)$	$N * O(k + n)$
Step 4(*a*)	$O(N \log N)$	$O(N \log N)$
Step 4(*b*)	$O(N)$	$O(N)$
Step 4(*c*)	$O(N \log N)$	$O(N \log N)$
Total complexity for M iterations	$O(MN(k^{1.5} log^5 k + n \log k + \log N))$	$O(MN(k + n + \log N))$

Table 2. Obtained non-dominated solutions for $k = 5$ for the workspace displayed in Figure 5.

	sol 1	sol 2	sol 3	sol 4	sol 5	sol 6	sol 7	sol 8	sol 9	sol 10	sol 11	sol 12	sol 13
F1	1161.5	1154.6	1126.6	1124.0	789.1	785.1	701.0	684.3	674.3	632.8	597.8	577.5	574.0
F2	400.9	405.6	416.0	601.7	842.8	1157.5	1187.8	1223.7	1250.6	1477.1	1560.8	1618.0	1694.3
	sol 14	sol 15	sol 16	sol 17	sol 18	sol 19	sol 20	sol 21	sol 22	sol 23	sol 24	sol 25	
F1	**567.2**	566.7	566.7	559.4	552.4	541.1	493.9	488.6	476.7	466.6	458.3	453.0	
F2	**1712.5**	1733.4	1837.3	1898.1	1937.0	2031.7	2315.1	2356.7	2523.6	2682.5	2755.5	2811.1	

Table 3. Obtained non-dominated solutions for $k = 10$ for the workspace displayed in Figure 5.

	sol 1	sol 2	sol 3	sol 4	sol 5	sol 6	sol 7	sol 8	sol 9	sol 10	sol 11	sol 12
F1	880.6	845.7	674.8	632.8	577.5	567.2	566.7	566.7	559.4	552.4	513.7	510.4
F2	1361.5	1533.0	1672.7	1676.4	1713.3	1759.8	1767.1	1856.6	1912.2	1974.2	2055.1	2059.4
	sol 13	sol 14	sol 15	sol 16	sol 17	sol 18	sol 19	sol 20	sol 21	sol 22	sol 23	sol 24
F1	493.9	488.6	488.5	486.6	476.7	455.0	454.2	450.5	**436.8**	397.0	396.5	390.0
F2	2330.9	2366.0	2410.7	2439.8	2530.5	2691.1	2695.0	2716.2	**2824.3**	2885.5	3084.9	3225.8
	sol 25	sol 26	sol 27	sol 28	sol 29	sol 30	sol 31	sol 32	sol 33	sol 34	sol 35	
F1	383.7	374.1	370.9	350.5	348.6	344.6	341.2	337.5	336.8	335.8	324.0	
F2	3227.9	3234.3	3291.5	3313.8	3323.9	3341.0	3341.8	3380.7	3384.3	3385.4	3448.0	

Table 4. Obtained non-dominated solutions for $k = 20$ for the workspace displayed in Figure 5.

	sol 1	sol 2	sol 3	sol 4	sol 5	sol 6	sol 7	sol 8	sol 9	sol 10	sol 11
F1	**674.2589**	566.746	476.74	454.2345	439.9364	436.8352	402.8809	396.9887	371.8494	367.143	335.7737
F2	**2637.432**	2796.016	2976.092	3048.555	3102.039	3107.795	3342.441	3396.075	3558.751	3615.96	3627.931
	sol 12	sol 13	sol 14	sol 15	sol 16	sol 17	sol 18	sol 19	sol 20	sol 21	
F1	323.966	309.8209	306.6464	296.1689	295.6011	292.82 76	280.9306	273.8759	271.4498	270.3128	
F2	3692.823	3794.794	3831.388	3859.185	3912.321	3915.954	4016.856	4144.509	4230.076	4255.606	

Table 5. Obtained non-dominated solutions for $k = 25$ for the workspace displayed in Figure 5.

	sol 1	sol 2	sol 3	sol 4	sol 5	sol 6	sol 7	sol 8	sol 9	sol 10
F1	510.3577	493.9119	476.74	470.8652	436.8352	418.5272	402.8809	396.9887	375.2612	362.8168
F2	3185.409	3292.162	3361.909	3391.134	3427.32	3703.701	3704.675	3764.5	4006.933	4010.566
	sol 11	sol 12	sol 13	sol 14	sol 15	sol 16	sol 17	sol 18	sol 19	sol 20
F1	323.6109	**315.192**	306.6464	296.1689	296.0152	295.6011	292.8276	280.9306	273.8759	270.3128
F2	4019.666	**4023.299**	4034.5	4058.835	4088.661	4158.781	4162.414	4263.316	4334.482	4413.412

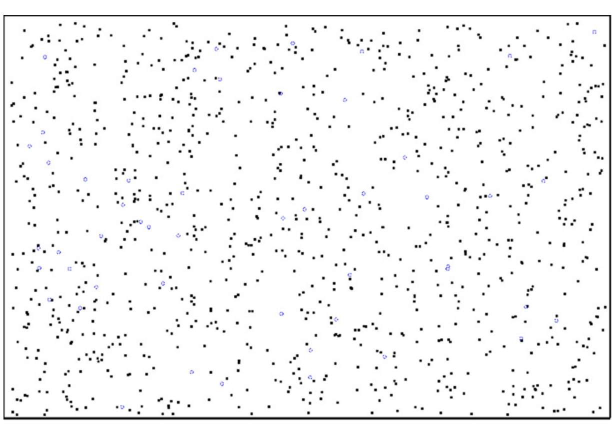


Figure 5. A workspace of the *k-center&TSP* and *k-median&TSP* problems containing $n = 1000$ demand points (black squares) and $m = 50$ potential facility points (blue circles).

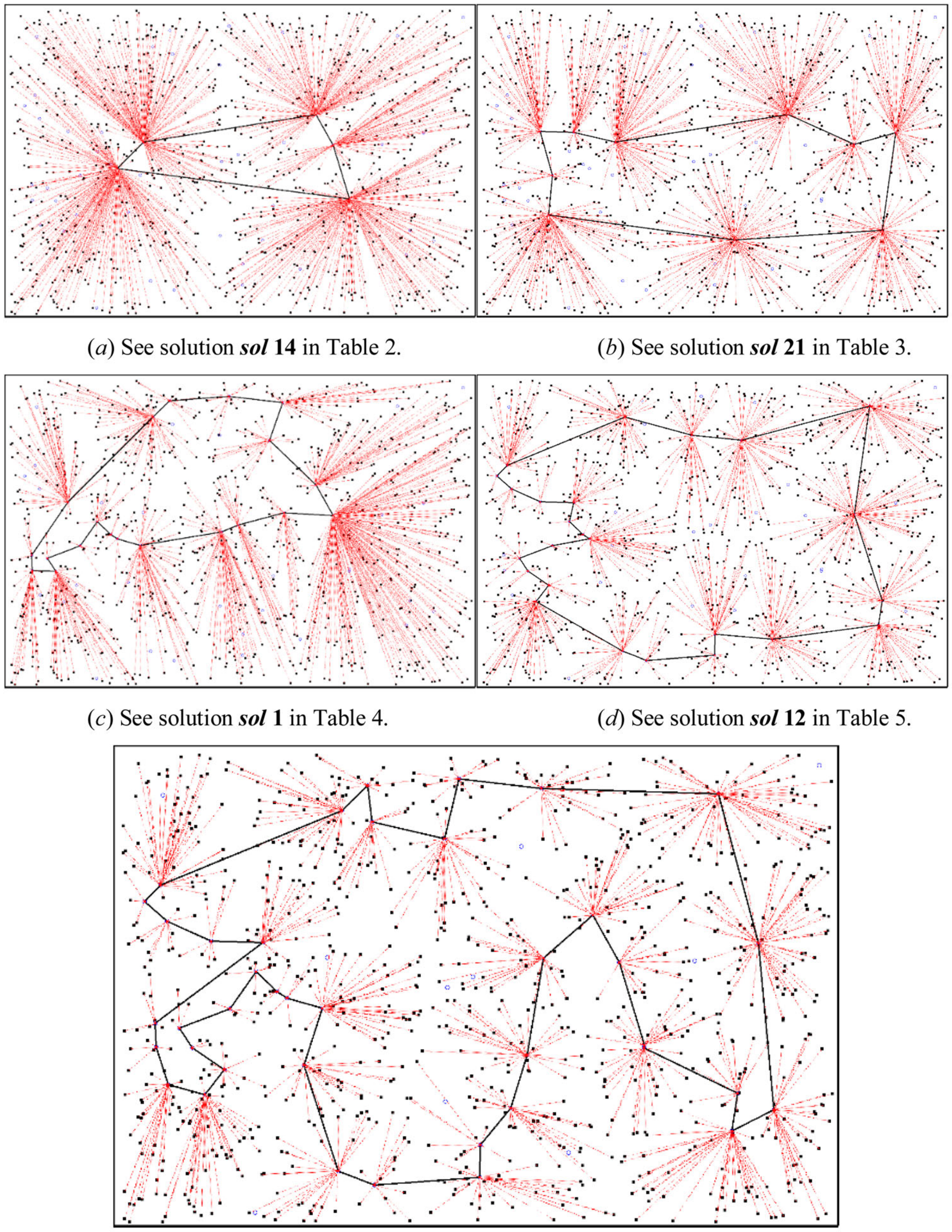
$k = 5, 10, 20, 25$ and 40. The running time for each case is about $7k$ seconds. Tables 2–6 show the obtained non-dominated solutions in each instance of the problem.

Table 6. Obtained non-dominated solutions for $k = 40$ for the workspace displayed in Figure 5.

	sol 1	sol 2	sol 3	sol 4
F1	396.9887	**296.0152**	288.7352	270.3128
F2	5147.321	**5371.345**	5392.445	5600.582

Each obtained solution has two objective values; F1 which denotes the objective value of the client-side (i.e. maximum service distance in k-centre&TSP problem or summation of service distances in k-median&TSP problem), and F2 which denotes the length of the tour which visits the open centres. Figure 6 displays a randomly obtained solution for $k = 5, 10, 20, 25$, and 40. Also, Figure 7 shows the normalised diagram of the obtained objective values.

As can be observed from the tables and also the figures, the proposed algorithm is able to find a set of diverse trade-off solutions in a reasonable running time. This helps to decision-maker to choose one of them based



(*a*) See solution ***sol* 14** in Table 2. (*b*) See solution ***sol* 21** in Table 3.

(*c*) See solution ***sol* 1** in Table 4. (*d*) See solution ***sol* 12** in Table 5.

(*e*) See solution ***sol* 2** in Table 6.

Figure 6. A randomly obtained solutions for the workspace is shown in Figure 5. Each demand point is assigned to its closest opened centre and showed by a red line. (*a*) solution for $k = 5$, (*b*) solution for $k = 10$, (*c*) solution for $k = 20$, (*d*) solution for $k = 25$ and (*e*) solution for $k = 40$.

on the high-level information and parameters. However, as a tool, it is possible to cluster the obtained solutions in some (e.g. 3–5) clusters and choose a proper solution (usually the centre of the cluster) from each cluster. The other approach to choosing a solution from the obtained solutions is choosing the knee point solution (Jurgen et al., 2004), and Kalai- Smorodinsky solution (Gaudrie et al., 2018; Kalai & Smorodinsky, 1975). Since it is out of the scope of this paper, we do not further explain it here and follow the evaluation of the algorithm with some benchmarks.

As the problem statement in this paper is new, there are no well-defined benchmarks for it. The most similar problem with benchmarks is available for the LRP. So, in this part, we use such benchmarks and apply the proposed algorithm to them. These problems are available at (http://sweet.ua.pt/sbarreto/_private/SergioBarretoHomePage.htm; Baldacci et al., 2011; Harks et al., 2013). They are some difficulties for the facility location problem approaches and LRP solvers. We use the position of clients and potential facility location centres. Table 8 shows the number of clients and potential facility points

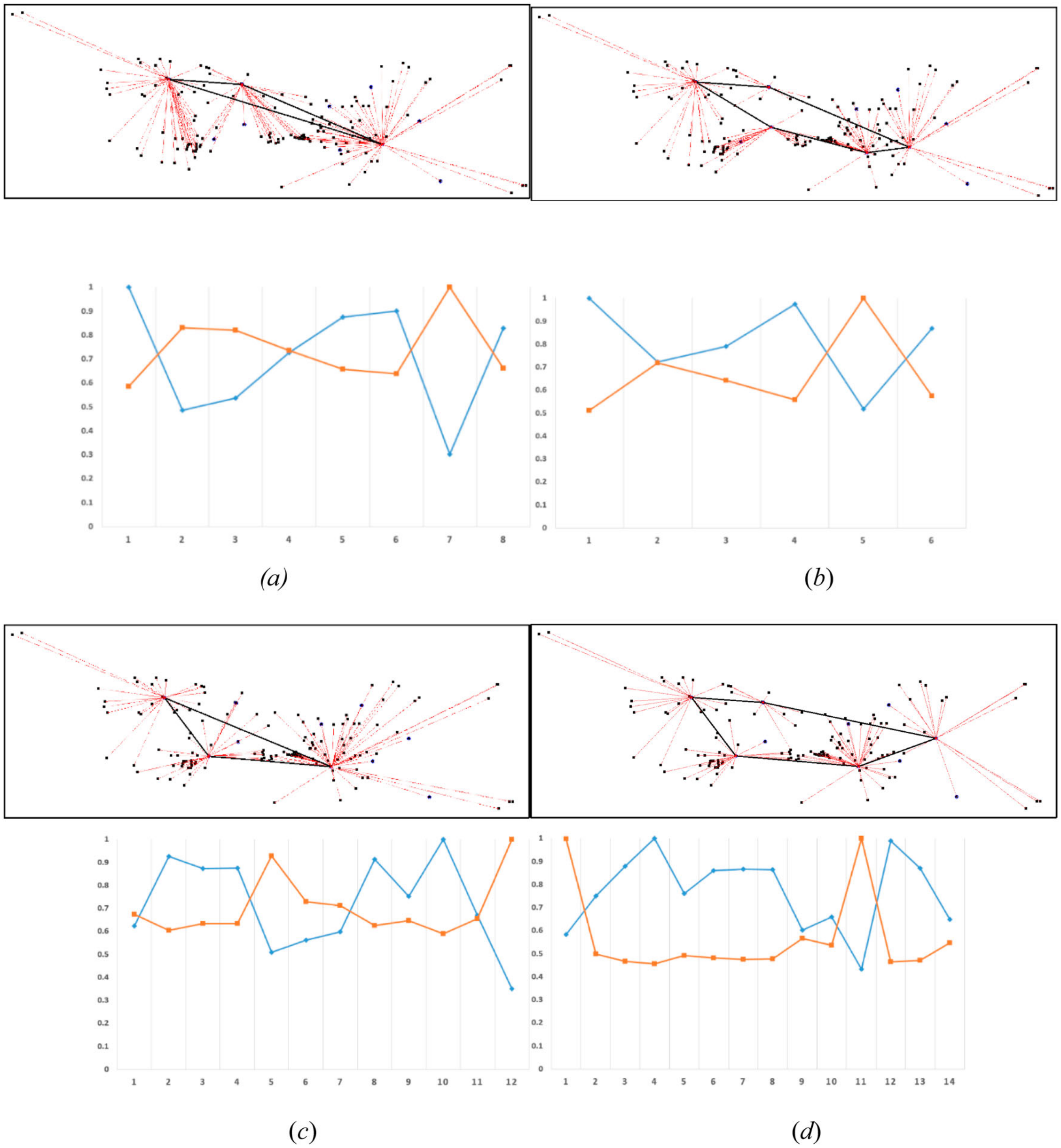


Figure 7. Obtained non-dominated solutions for the benchmark Daskin95. (*a*) *k*-centre&TSP for $k = 3$, and (*b*) *k*-centre&TSP for $k = 5$, (*c*) *k*-median&TSP for $k = 3$, and (*d*) *k*-median&TSP for $k = 5$.

Table 7. Benchmark instances.

Benchmark	*n*: # of clients	*m*: # of potential centres	*k*: # of centres	Running Time (*Second*)
Daskin95	150	10	3, 5	6, 9
Tuzun	200	20	5, 10	13, 18
Harks	10000	1000	300, 500	192, 267

in each benchmark. We run the proposed algorithm for different numbers of *k* determined in the fourth column of Table 7 on each benchmark for both the problems *k*-centre&TSP and *k*-median&TSP. Further, the last column of the table shows the average running time of the algorithm on each instance. Figures 7–9 show the normalised objective values of the obtained non-dominated solutions and a randomly selected solution for each problem. We ran the algorithm for the different number of centres, *k*. Clearly, in all of the cases, the obtained solution fronts provide good trade-off sets. Each set provides a smooth change of the objective values among the solutions. For example, in Figure 7 (a), solutions 1, 4, and 7, and in Figure 7(b) solutions 1, 2, and 5 represent solutions that are most favourable from the client-side up to the centre-side. Besides, in Figure 7 (d) there are significant improvements among the solutions 4, 5, and 6 on the client-side while the objective values on the centre-side change a little. There are similar cases in Figure 8

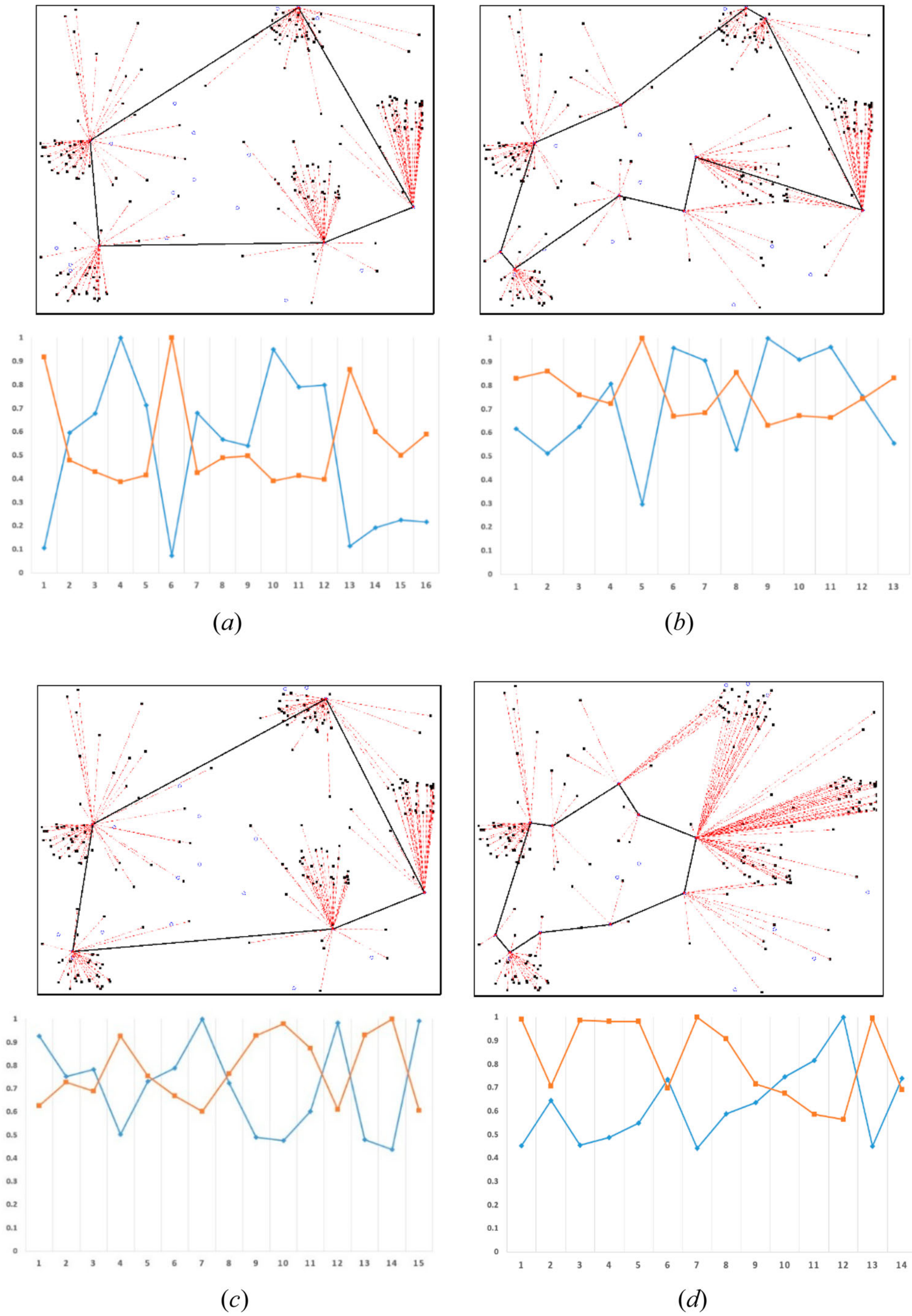


Figure 8. Solutions for the benchmark Tuzun (a) k-centre&TSP for $k = 5$, and (b) k-centre&TSP for $k = 10$, (c) k-median&TSP for $k = 5$, and (d) k-median&TSP for $k = 10$.

and Figure 9. Therefore, considering the scale and type of the cost and objective, one of them may be efficiently selected by the decision-maker.

Finally, to compare the proposed algorithm, we implement a standard version of the robust and popular Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2000). NSGA-II as a fast and efficient algorithm has been extensively applied to many intractable problems. We represent each chromosome by a binary vector with size m and utilise a one-point crossover operator. We always force the number of opened centres to be exactly m in each chromosome. To this end, we randomly close or open centres when a child is generated by crossing bi-parent solutions. We also use the exchange mutation operator with a probability 0.1. To have a fair comparison in terms of the number of fitness evaluations in both algorithms, we set the population size to 40 and the number of generations to 100 and run them

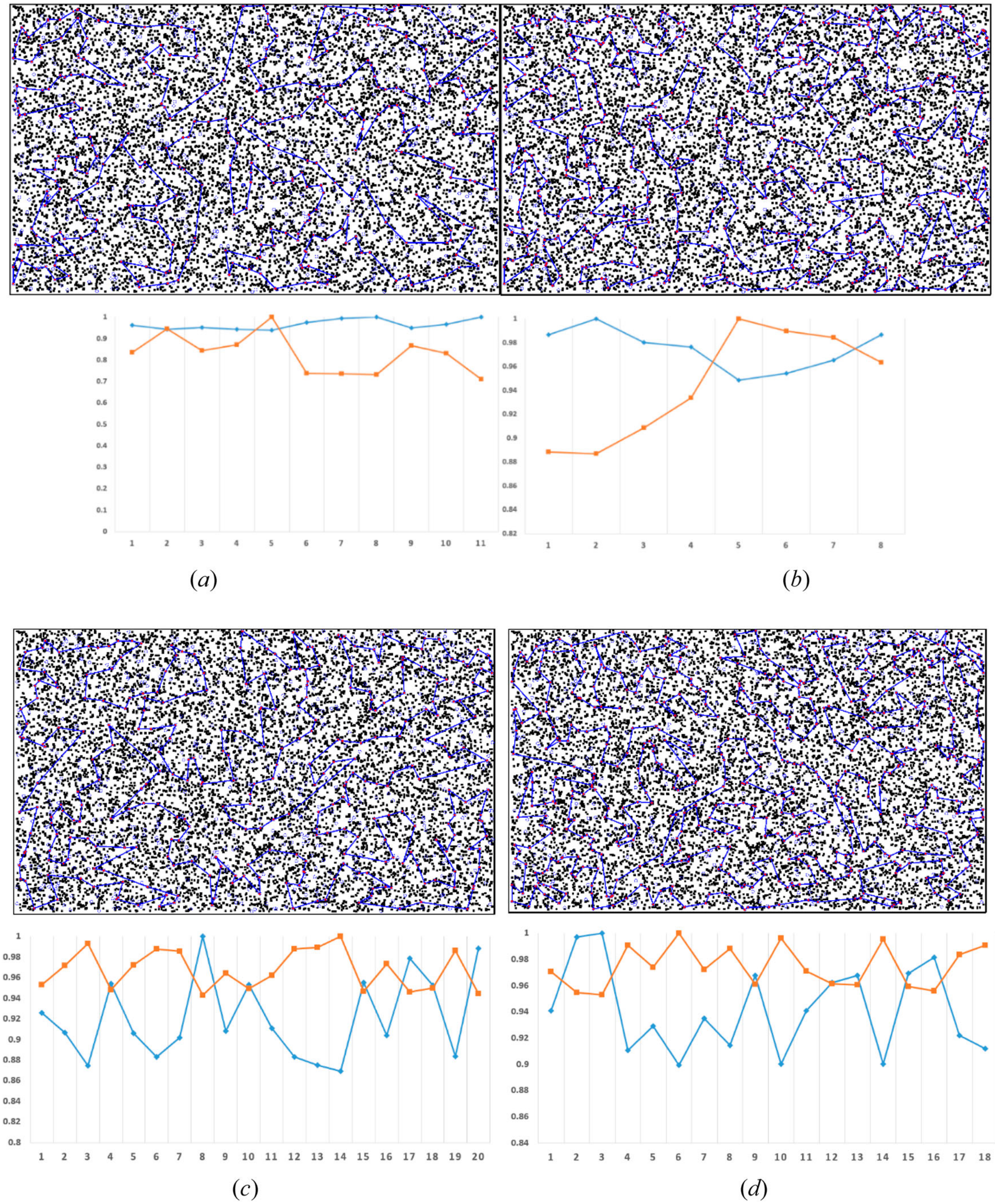


Figure 9. Solutions for the benchmark Herks. The connection between the clients and the centres are skipped anf the TSP tour is shown with blue. (*a*) k-centre&TSP for $k = 300$, and (*b*) k-centre&TSP for $k = 500$, (*c*) k-median&TSP for $k = 300$, and (*d*) k-median&TSP for $k = 500$.

on a randomly generated test problem with 1000 clients and 50 potential facility centres with $k = 10, 15, 20, 25$. Finally, to measure the efficiency of the algorithms we apply the set coverage metric (*scm*), (Deb, 2001) to compare the Pareto-optimality of the final obtained solutions, and the spacing metric (*sm*) (Deb, 2001) to compare the diversity goal. In each run, we separately compute the set coverage metric and spacing metric. Tables 8 and 9 show the average values of the set coverage and the spacing metrics, respectively. Based on these results, the proposed algorithm outperforms NSGA-II in terms of achieving solutions close to the Pareto-optimal solutions, however, its running time is almost 1.2 times more than NSGA-II's running time. Based on the multi-objective optimisation literature, NSGA-II is an efficient algorithm for finding the Pareto-optimal solutions, however, the obtained set of solutions by the proposed algorithm in our study dominates the ones by NSGA-II. For example,

Table 8. Average comparison results of 30 runs between NSGA-II and the proposed algorithm under the set coverage metric. A is the obtained non-dominated solutions by the proposed algorithm and B is the obtained non-dominated solutions by NSGA-II.

		$C(A,B)$	$C(B,A)$
k-centre&TSP	$k=10$	0.84	0.27
	$k=15$	0.82	0.18
	$k=20$	0.79	0.26
	$k=25$	0.85	0.19
k-median&TSP	$k=10$	0.81	0.22
	$k=15$	0.80	0.20
	$k=20$	0.77	0.27
	$k=25$	0.80	0.23

Table 9. Average comparison results of 30 runs between NSGA-II and the proposed algorithm under the spacing metric.

		Proposed Algorithm	NSGA-II
k-centre&TSP	$k=10$	21.16	19.63
	$k=15$	25.81	26.16
	$k=20$	22.49	22.90
	$k=25$	26.08	24.35
k-median&TSP	$k=10$	23.34	20.95
	$k=15$	24.81	24.08
	$k=20$	19.26	21.27
	$k=25$	27.53	27.36

in case $k=10$, 84 percent of the solutions obtained by the proposed algorithm dominate the solutions obtained by NSGA-II, while only 27 percent of the obtained solutions by NSGA-II dominate the solutions by the proposed algorithm. Note that, this does not mean the proposed algorithm is strictly better than NSGA-II also for other problems, but it shows that our proposed algorithm is very well customised to the presented bi-sided facility location problem with objectives k-centre and k-median for the client-side and TSP for the centre-side of the problem.

In general, based on the simulation results of the randomly generated test problems, benchmarks and comparison with NSGA-II, it can be concluded the proposed Delaunay triangulation heuristic search algorithm, is simple to implement and efficient for both k-centre and k-median objectives. Also, since it uses $\frac{3}{2}$-approximation Christofieds algorithm for computing TSP tour, it efficiently optimises the objective in the centre-side. Note that $\frac{3}{2}$ is a guaranteed factor in the worst-case and it is much better in the average case.

7. Conclusion

Enhancing the efficiency and quality of service centres and increasing the satisfiability and utility of the clients are two important sides of designing service provider systems. Numerous studies focus on either the clients or on the centre side individually. We introduced a bi-side optimisation model, called *client-side and center-side facility location*, as a general framework to consider both increasing the efficiency of service centres and increasing the utility of the clients. It takes into account major problems such as 'Where to locate the centers?' and 'Which clients are assigned to which centers?', and helps decision-makers by providing a diverse set of solutions. The model includes a wide range of well-known problems such as facility location, connected facility location, vehicle routing, and location-routing problems as well as other variations of these problems. In this paper, we considered well-known problems k-median and k-centre for the client-side and travelling salesman for the centre-side of the model. Thus, two bi-objective optimisation versions of the model, denoted by *k-median&TSP* and *k-center&TSP*, were constructed. In addition to discussing the complexity of the problems, we proposed an efficient heuristic approximation algorithm to solve the problems. The algorithm uses the geometric properties of the problem and it unlike heuristic algorithms does not need any user predefined parameter. Finally, we test the algorithm on randomly generated instances of the problems as well as some benchmarks. The results confirm the efficiency of the algorithm in terms of finding a set of diverse trade-off solutions and running time. Considering different problems in the real world and choosing different objective functions for the client-side and centre-side of the model are future extensions of this study.

Acknowledgment

This research was financinsally supported by *Dutch Research Council* (*NWO*), *Social Sciences and Humanities:* grant number 040.11.7421.

Data availability statement

Two classes of data have been used in this study: randomly generated instances and benchmarks. This first class is simply generated by a C# language code and the second class, which is mentioned in Table 7, is freely available at the following website and references. http://sweet.ua.pt/sbarreto/_private/SergioBarretoHomePage.htm; Baldacci et al., 2011; Harks et al., 2013.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Mansoor Davoodi is an expert in modeling real-world problems, optimization algorithms and operations research. He is an assistant professor and currently, he is a senior researcher at the Center for Advanced Systems Understanding, Helmholtz-Zentrum Dresden Rossendorf (HZDR), Germany.

Jafar Rezaei is an associate professor of operations and supply chain management at Delft University of Technology, The Netherlands, where he obtained his Ph.D. in 2012. His main research interests are in the areas of logistics and supply chain management, and decision science.

ORCID

Mansoor Davoodi http://orcid.org/0000-0003-1010-4121
Jafar Rezaei http://orcid.org/0000-0002-7407-9255

References

Adeleke, O. J., & Olukanni, D. O. (2020). Facility location problems: Models, techniques, and applications in waste management. *Recycling*, *5*(2), 10. https://doi.org/10.3390/recycling5020010

Alijani, R., Banerjee, S., Gollapudi, S., Kollias, K., & Munagala, K. (2017). Bi-sided facility location. arXiv preprint arXiv:1711.11392.

Arulselvan, A., Bley, A., & Ljubić, I. (2019). The incremental connected facility location problem. *Computers & Operations Research*, *112*, 104763. https://doi.org/10.1016/j.cor.2019.104763

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., & Pandit, V. (2004). Local search heuristics for k–median and facility location problems. *SIAM Journal on Computing*, *33*(3), 544–562. https://doi.org/10.1137/S0097539702416402

Bajaj, A., & Dhodiya, J. (2023). Multi-objective quasi oppositional Jaya algorithm to solve multi-objective solid travelling salesman problem with different aspiration level. *International Journal of Systems Science: Operations & Logistics*, *10*(1), 2127340. https://doi.org/10.1080/23302674.2022.2127340

Baldacci, R., Mingozzi, A., & WolflerCalvo, R. (2011). An exact method for the capacitated location–routing problem. Operations Research. 59:1284–1296 (downloaded at http://claudio.contardo.org/datasets–source–code/).

Barbati, M. (2013). Models and Algorithms for Facility Location Problems with Equity Considerations, PhD thesis.

Bateni, M. H., & Hajiaghayi, M. T. (2012). Assignment problem in content distribution networks: Unsplittable hard–capacitated facility location. *ACM Transactions on Algorithms (TALG)*, *8*(3), 1–19. https://doi.org/10.1145/2229163.2229164

Bertsimas, D. J. (1989). Traveling salesman facility location problems. *Transportation Science*, *23*(3), 184–191. https://doi.org/10.1287/trsc.23.3.184

Braekers, K., Caris, A., & Janssens, G. K. (2014). Bi-objective optimization of drayage operations in the service area of intermodal terminals. *Transportation Research Part E: Logistics and Transportation Review*, *65*, 50–69. https://doi.org/10.1016/j.tre.2013.12.012

Calvete, H. I., Galé, C., & Iranzo, J. A. (2016). MEALS: A multiobjective evolutionary algorithm with local search for solving the bi-objective ring star problem. *European Journal of Operational Research*, *250*(2), 377–388. https://doi.org/10.1016/j.ejor.2015.09.044

Chakrabarty, D., & Swamy, C. (2017). Interpolating between k–Median and k–Center: Approximation Algorithms for Ordered k –Median. arXiv preprint arXiv:1711.08715.

Charikar, M., Guha, S., Tardos, É., & Shmoys, D. B. (2002). A constant–factor approximation algorithm for the k–median problem. *Journal of Computer and System Sciences*, *65*(1), 129–149. https://doi.org/10.1006/jcss.2002.1882

Christofides, W. (1976). *Worst–case analysis of a new heuristic for the traveling salesman problem. Tech. Rep. 388, graduate school of industrial administration*. Carnegie Mellon University.

Coello, C. C. A., van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms For solving multi–objective problems. (242)*. Kluwer Academic.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT press.

Daskin, M. S. (2001). *Network and discrete location: Models, algorithms and applications*. John Wiley.

Daskin, M. S., Snyder, L. V., & Berger, R. T. (2005). Facility location in supply chain design. In *Logistics systems: Design and optimization*, Langevin, A., Riopel, D. (eds), (pp. 39–65). Springer.

Davoodi, M. (2019). K–Balanced center location problem: A new multi–objective facility location problem. *Computers & Operations Research*, *105*, 68–84. https://doi.org/10.1016/j.cor.2019.01.009

Davoodi, M., & Mohades, A. (2011). Solving the constrained coverage problem. *Applied Soft Computing*, *11*(1), 963–969. https://doi.org/10.1016/j.asoc.2010.01.016

Deb, K. (2001). *Multi–objective optimization using evolutionary algorithms*. Wiley.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non–dominated sorting genetic algorithm for multi–objective: NSGA–II. in: *Proceedings of the Parallel Problem Solving from Nature VI Conference* (pp. 846–858). Springer–Verlag.

de Berg, M., Cheong, O., van Kreveld, M., & Overmans, M. (2008). *Computational geometry: Algorithms and applications* (3rd ed.). Springer–Verlag.

Delfani, F., Kazemi, A., SeyedHosseini, S. M., & Niaki, S. T. A. (2021). A novel robust possibilistic programming approach for the hazardous waste location-routing problem considering the risks of transportation and population. *International Journal of Systems Science: Operations & Logistics*, *8*(4), 383–395. https://doi.org/10.1080/23302674.2020.1781954

DePuy, G. W., Moraga, R. J., & Whitehouse, G. E. (2005). Meta-RaPS: A simple and effective approach for solving the traveling salesman problem. *Transportation Research Part E: Logistics and Transportation Review*, *41*(2), 115–130. https://doi.org/10.1016/j.tre.2004.02.001

Drexl, M., & Schneider, M. (2013). A survey of location–routing problems. *Technical Report LM*. 2013–03.

Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location–routing problem. *European Journal of Operational Research*, *241*(2), 283–308. https://doi.org/10.1016/j.ejor.2014.08.030

Drezner, Z., & Hamacher, H. W. (2004). *Facility location: Applications and theory*. Springer.

Eisenbrand, F., Grandoni, F., Rothvoß, T., & Schäfer, G. (2010). Connected facility location via random facility sampling and core detouring. *Journal of Computer and System Sciences*, *76*(8), 709–726. https://doi.org/10.1016/j.jcss.2010.02.001

Fahmy, S. A., & Gaafar, M. L. (2023). Modelling and solving the split-delivery vehicle routing problem, considering loading constraints and spoilage of commodities. *International Journal of Systems Science: Operations & Logistics*, *10*(1): 2074566. https://doi.org/10.1080/23302674.2022.2074566
Farahani, R. Z., & Hekmatfar, M. (2009). *Facility location: Concepts, models, algorithms and case studies*. Springer Science & Business Media.
Farahani, R. Z., SteadieSeifi, M., & Asgari, N. (2010). Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, *34*(7), 1689–1709. https://doi.org/10.1016/j.apm.2009.10.005
Fernandes, C. G., de Paula, S. P., & Pedrosa, L. L. C. (2018). Improved approximation algorithms for capacitated fault–tolerant k–center. *Algorithmica*, *80*(3), 1041–1072. https://doi.org/10.1007/s00453-017-0398-x
Gaudrie, D., Riche, L., Picheny, V., Enaux, B., & Herbert, V. (2018). Budgeted Multi–Objective Optimization with a Focus on the Central Part of the Pareto Front. arXiv preprint arXiv:1809.10482.
Gheisariha, E., Etebari, F., Vahdani, B., & Tavakkoli-Moghaddam, R. (2023). Scheduling and routing of multiple heterogeneous vehicles in a milk collection problem with blending in compartments and time windows. *International Journal of Systems Science: Operations & Logistics*, *10*(1), 2190852. https://doi.org/10.1080/23302674.2023.2190852
Gollowitzer, S., & Ljubić, I. (2011). MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, *38*(2), 435–449. https://doi.org/10.1016/j.cor.2010.07.002
Gouveia, L., Simonetti, L., & Uchoa, E. (2011). Modeling hop–constrained and diameter–constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming*, *128*(1–2), 123–148. https://doi.org/10.1007/s10107-009-0297-2
Gowda, D. K., Lee, D., & Naamad, A. (1983). Dynamic Voronoi diagrams. *IEEE Transactions on Information Theory*, *29*(5), 724–731. https://doi.org/10.1109/TIT.1983.1056738
Guan, C., Zhang, Z., Liu, S., & Gong, J. (2019). Multi–objective particle swarm optimization for multi–workshop facility layout problem. *Journal of Manufacturing Systems*, *53*, 32–48. https://doi.org/10.1016/j.jmsy.2019.09.004
Gupta, S. (2018). *Approximation algorithms for clustering and facility location problems. PhD diss.* University of Illinois at Urbana–Champaign.
Hakli, H., & Ortacay, Z. (2019). An improved scatter search algorithm for the uncapacitated facility location problem. *Computers & Industrial Engineering*, *135*, 855–867. https://doi.org/10.1016/j.cie.2019.06.060
Harks, T., König, F., & Matuschke, J. (2013). Approximation algorithms for capacitated location routing. *Transportation Science*, *47*(1), 3–22. https://doi.org/10.1287/trsc.1120.0423. https://www.coga.tu–berlin.de/v–menue/download_media/clrlib
He, P., & Hao, J. K. (2023). Memetic search for the minmax multiple traveling salesman problem with single and multiple depots. *European Journal of Operational Research*, *307*(3), 1055–1070. https://doi.org/10.1016/j.ejor.2022.11.010
Hochba, D. S. (1997). Approximation algorithms for NP–hard problems. *ACM Sigact News*, *28*(2), 40–52. https://doi.org/10.1145/261342.571216
Huang, P. H., Tsai, Y. T., & Tang, C. Y. (2003). A fast algorithm for the alpha–connected two–center decision problem. *Information Processing Letters*, *85*(4), 205–210. https://doi.org/10.1016/S0020-0190(02)00402-7
Jaigirdar, S. M., Das, S., Chowdhury, A. R., Ahmed, S., & Chakrabortty, R. K. (2023). Multi-objective multi-echelon distribution planning for perishable goods supply chain: A case study. *International Journal of Systems Science: Operations & Logistics*, *10*(1), 2020367. https://doi.org/10.1080/23302674.2021.2020367
Jain, K., Mahdian, M., & Saberi, A. (2002). A new greedy approach for facility location problems. *In Proceedings of the thiry–fourth annual ACM symposium on Theory of computing*, 731–740.
Javid, A. A., & Azad, N. (2009). Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E: Logistics and Transportation Review*, *46*(5), 582–597. https://doi.org/10.1016/j.tre.2009.06.005
Jensen, M. (2003). Reducing the run–time complexity of multi objective EAs: The NSGA–II and other algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(5), 503–515. https://doi.org/10.1109/TEVC.2003.817234
Jurgen, B., Deb, K., Dierolf, H., & Osswald, M. (2004). Finding knees in multi–objective optimization. In International conference on parallel problem solving from nature, Springer, 722–731.
Kalai, E., & Smorodinsky, M. (1975). Other solutions to Nash's bargaining problem. *Econometrica: Journal of the Econometric Society*, 513–518. https://doi.org/10.2307/1914280
Karlin, A. R., Klein, N., & Gharan, S. O. (2020). A (Slightly) Improved Approximation Algorithm for Metric TSP. arXiv preprint arXiv:2007.01409.
Khuller, S., & Sussmann, Y. J. (2000). The capacitated k–center problem. *SIAM Journal on Discrete Mathematics*, *13*(3), 403–418. https://doi.org/10.1137/S0895480197329776
Krishnaswamy, R., Li, S., & Sandeep, S. (2018). Constant approximation for k–median and k–means with outliers via iterative rounding. In *Proc. of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 646–659.
Labbé, M., Laporte, G., Rodríguez-Martín, I., & Salazar-González, J. J. (2004). The Ring Star problem: Polyhedral analysis and exact algorithm. *Networks*, *43*(3), 177–189. https://doi.org/10.1002/net.10114
Laporte, G. (1989). A survey of algorithms for location–routing problems. *Investigación Operativa*, *1*(2), 93–118.
Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(2), 231–247. https://doi.org/10.1016/0377-2217(92)90138-Y
Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, *43*(4), 408–416. https://doi.org/10.1287/trsc.1090.0301
Li, S. (2017). On uniform capacitated k–median beyond the natural LP relaxation. *ACM Transactions on Algorithms (TALG)*, *13*(2), 1–18. https://doi.org/10.1145/2983633
Liefooghe, A., Jourdan, L., & Talbi, E.-G. (2010). Metaheuristics and cooperative approaches for the Bi-objective Ring Star problem. *Computers & Operations Research*, *37*(6), 1033–1044. https://doi.org/10.1016/j.cor.2009.09.004

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, *41*(4), 1118–1138. https://doi.org/10.1016/j.eswa.2013.07.107
Liu, Y., Xu, L., Han, Y., Zeng, X., Yen, G. G., & Ishibuchi, H. (2023). Evolutionary multimodal multiobjective optimization for traveling salesman problems. *IEEE Transactions on Evolutionary Computation*. https://doi.org/10.1109/TEVC.2023.3239546
Marín, A. (2011). The discrete facility location problem with balanced allocation of customers. *European Journal of Operational Research*, *210*(1), 27–38. https://doi.org/10.1016/j.ejor.2010.10.012
Martínez–Salazar, I. A., Molina, J., Ángel–Bello, F., Gómez, T., & Caballero, R. (2014). Solving a bi–objective transportation location routing problem by metaheuristic algorithms. *European Journal of Operational Research*, *234*(1), 25–36. https://doi.org/10.1016/j.ejor.2013.09.008
Meira, L. A. A., Paulo, S. M., Menzori, M., & Zeni, G. A. (2017). Multi-objective vehicle routing problem applied to large scale post office deliveries. *arXiv preprint arXiv:1801.00712.*
Melkote, S., & Daskin, M. S. (2001). Capacitated facility location/network design problems. *European Journal of Operational Research*, *129*(3), 481–495. https://doi.org/10.1016/S0377-2217(99)00464-6
Moon, I., Salhi, S., & Feng, X. (2020). The location–routing problem with multi-compartment and multi-trip: Formulation and heuristic approaches. *Transportmetrica A: Transport Science*, *16*(3), 501–528. https://doi.org/10.1080/23249935.2020.1720036
Nagy, G., & Salhi, S. (2007). Location–routing: Issues, models and methods. *European Journal of Operational Research*, *177*(2), 649–672. https://doi.org/10.1016/j.ejor.2006.04.004
Oded, K., & Hakimi, S. L. (1979). An algorithmic approach to network location problems. ii: The p–medians. *SIAM Journal on Applied Mathematics*, *37*(3), 539–560. https://doi.org/10.1137/0137041
Onstein, A. T., Ektesaby, M., Rezaei, J., Tavasszy, L. A., & van Damme, D. A. (2020). Importance of factors driving firms' decisions on spatial distribution structures. *International Journal of Logistics Research and Applications*, *23*(1), 24–43. https://doi.org/10.1080/13675567.2019.1574729
Pacheco, J., Caballero, R., Laguna, M., & Molina, J. (2013). Bi–objective bus routing: An application to school buses in rural areas. *Transportation Science*, *47*(3), 397–411. https://doi.org/10.1287/trsc.1120.0437
Prodhon, C., & Prins, C. (2014). A survey of recent research on location–routing problems. *European Journal of Operational Research*, *238*(1), 1–17. https://doi.org/10.1016/j.ejor.2014.01.005
Ravi, R., & Sinha, A. (2004). *Multicommodity facility location. In Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, 342–349.
Rayat, F., Musavi, M., & Bozorgi–Amiri, A. (2017). Bi–objective reliable location–inventory–routing problem with partial backordering under disruption risks: A modified AMOSA approach. *Applied Soft Computing*, *59*, 622–643. https://doi.org/10.1016/j.asoc.2017.06.036
Redi, A. A. N., Jewpanya, P., Kurniawan, A. C., Persada, S. F., Nadlifatin, R., & Dewi, O. A. C. (2020). A simulated annealing algorithm for solving Two–echelon vehicle routing problem with locker facilities. *Algorithms*, *13*(9), 218. https://doi.org/10.3390/a13090218
Roostapour, V., Kiarazm, I., & Davoodi, M. (2016). Deterministic algorithm for 1–Median 1–Center Two–objective optimization problem. *Topics in Theoretical Computer Science, LNCS. 9541*, 164–178. https://doi.org/10.1007/978-3-319-28678-5_12
San Felice, M. C., Williamson, D. P., & Lee, O. (2014). The online connected facility location problem. In Latin American Symposium on Theoretical Informatics, Springer, Berlin, Heidelberg. 574–585.
Schiffer, M., Schneider, M., Walther, G., & Laporte, G. (2019). Vehicle routing and location routing with intermediate stops: A review. *Transportation Science*, *53*(2), 319–343. https://doi.org/10.1287/trsc.2018.0836
Shamos, M. I., & Hoey, D. (1975). Closest–point problems. *Foundations of Computer Science, 16th Ann. Symp*. IEEE.
Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., & Van Woensel, T. (2023). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, *304*(3), 865–886. https://doi.org/10.1016/j.ejor.2022.02.022
Svensson, O., Tarnawski, J., & Végh, L. A. (2020). A constant–factor approximation algorithm for the asymmetric traveling salesman problem. *Journal of the ACM (JACM)*, *67*(6), 1–53. https://doi.org/10.1145/3424306
Swamy, C., & Kumar, A. (2004). Primal–dual algorithms for connected facility location problems. *Algorithmica*, *40*(4), 245–269. https://doi.org/10.1007/s00453-004-1112-3
Tavakkoli–Moghaddam, R., Forouzanfar, F., & Ebrahimnejad, S. (2013). Incorporating location, routing, and inventory decisions in a bi–objective supply chain design problem with risk–pooling. *Journal of Industrial Engineering International*, *9*(1), 19. https://doi.org/10.1186/2251-712X-9-19
Tavakkoli–Moghaddam, R., Makui, A., & Mazloomi, Z. (2010). A new integrated mathematical model for a bi–objective multi–depot location–routing problem solved by a multi–objective scatter search algorithm. *Journal of Manufacturing Systems*, *29*(2–3), 111–119. https://doi.org/10.1016/j.jmsy.2010.11.005
Thorup, M. (2005). Quick k–median, k–center, and facility location for sparse graphs. *SIAM Journal on Computing*, *34*(2), 405–432. https://doi.org/10.1137/S0097539701388884
Tordecilla, R. D., Montoya-Torres, J. R., Quintero-Araujo, C. L., Panadero, J., & Juan, A. A. (2023). The location routing problem with facility sizing decisions. *International Transactions in Operational Research*, *30*(2), 915–945. https://doi.org/10.1111/itor.13125
Toth, P., & Vigo, D. (2002). The vehicle routing problem. *Society for Industrial and Applied Mathematics*, *2002*.
Turkoglu, D. C., & Genevois, M. E. (2020). A comparative survey of service facility location problems. *Annals of Operations Research*, 292, 399–468.
Varadarajan, K. R. (1998). *A divide–and–conquer algorithm for min-cost perfect matching in the plane*. Foundations of Computer Science, Proceedings. 39th Annual Symposium on. IEEE.
Vazirani, V. V. (2013). *Approximation algorithms*. Springer Science & Business Media.
Verhetsel, A., Kessels, R., Goos, P., Zijlstra, T., Blomme, N., & Cant, J. (2015). Location of logistics companies:

A stated preference study to disentangle the impact of accessibility. *Journal of Transport Geography*, *42*, 110–121. https://doi.org/10.1016/j.jtrangeo.2014.12.002

Wang, M., Miao, L., & Zhang, C. (2021). A branch-and-price algorithm for a green location routing problem with multi-type charging infrastructure. *Transportation Research Part E: Logistics and Transportation Review*, *156*, 102529. https://doi.org/10.1016/j.tre.2021.102529

Wang, Q., Batta, R., Bhadury, J., & Rump, C. M. (2003). Budget constrained location problem with opening and closing of facilities. *Computers & Operations Research*, *30*(13), 2047–2069. https://doi.org/10.1016/S0305-0548(02)00123-5

Williamson, D. P., & Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge university press.

Appendices

Appendix (*A*)

4.1. Bi-objective optimisation and Pareto-optimal solutions

While single-objective optimisation problems have one optimal solution, multi-objective optimisation problems have several and possibly infinite optimal solutions, called *Pareto-optimal solutions*. Let s and s' be two (feasible) solutions to a multi-objective optimisation problem. We say *s dominates s'*, if s is not worse than s' in all objectives, and also there is some objective in which s is better than s'. Thus, for two solutions s and s', three cases may happen: (*i*) *s dominates* s', (*ii*) *s' dominates s*, and (*iii*) neither of s and s' dominates the other one. In the third case, s and s' are called *non-dominated* solutions. *Pareto-optimal fronts* are the image of all *non-dominated* feasible solutions in the objective space. Finding all Pareto-optimal solutions is the main goal of solving a multi-objective optimisation problem (as we are not interested in *dominated* solutions). However, there are problems such that the size of such solutions is exponential in the size of the problem or even infinite in continuous search spaces. Since reporting all of such solutions is inefficient, the second goal in multi-objective optimisation is reporting a set of Pareto-optimal solutions that are as diverse as possible in the objective space (Deb, 2001).

A classical approach to solving a multi-objective optimisation problem is to convert it to a single-objective one aggregating the objectives. To this end, we need to choose a priority weight for each objective and combine them into a single weighted objective. So, different priority weights result in different Pareto-optimal solutions. There are three main issues in this approach. First, in most real-world optimisation problems, such priority weights are unavailable, and they depend on some high-level information. Second, some Pareto-optimal solutions cannot be found using this approach, e.g. when Pareto fronts are non-convex (Coello et al., 2002). Third, if the objective set represents independent problem-owners, there is a need to find solutions that represent the tradeoff between the (conflicting) preferences of different problem-owners. To tackle these issues, several approaches such as multi-objective evolutionary algorithms and other heuristic algorithms have been developed in the last two decades. Most of these approaches work based on the domination principle and try to explore the search space using generating a random neighbour of a solution. Since such algorithms work based on making a balance between exploration and exploitation in the search space, unfortunately, their efficiency depends on the parameters which are defined by the user. For example, in the evolutionary algorithms, in addition to the size of the population and the number of generations, other parameters such as crossover and mutation rates need to be predefined and tuned. Thus, proposing algorithms that have the least number of parameters to solve multi-objective optimisation problems is valuable. In this section, we propose an efficient heuristic approach based on the geometric facts in the search space of the problem.

4.2. Voronoi diagram and Delaunay Triangulation

Voronoi Diagram (VD) is a well-known decomposition of the space for a set of sites such as points, segments, or circles. In a bi-dimensional space, it partitions the plane into a set of regions such that the points that lie in each region have the same nearest site. The diagram's edges are the points with more than one nearest site. So, each point in the plane is assigned to its nearest site. Let $C = \{c_1, c_2, \ldots, c_m\}$ be a set of k sites (i.e. points) in the plane. The *Voronoi region* of each site, c_i, denoted by VR_i, is a convex polygon such that all points that lie in it, are closer to c_i compared to the other sites of C. Formally, $VR_i = \{p \in R^2 : d(p, c_i) \leq d(p, c_j), \forall j \neq i\}$, where $d(.,.)$ is the distance function. *Voronoi Diagram* (VD) is the boundary of Voronoi regions, and it can be computed in $O(m \log m)$ time by a sweep line algorithm named Fortune's algorithm (de Berg et al., 2008). Having VD, the nearest neighbour query is founded in $O(\log m)$ time.

Delaunay Triangulation (DT) is the geometric dual of the Voronoi diagram. It is constructed by connecting a line segment between any two neighbour sites in VD. It can also be constructed directly by checking the *empty circle property* (de Berg et al., 2008) between any three sites. That is, any three sites of C form a triangle of DT if and only if the (unique) circle passing through the three sites is empty, there is no other site that lies interior of the circle. Figure 2 shows the Voronoi diagram of 30 randomly selected sites and their corresponding Delaunay triangulation. For a set of m given sites, DT is a planar graph with at most $3m$ edges. The maximum degree of each site in the DT graph is $O(m)$ in the worst case, however, it is constant on average. Since in the problem *FL&TSP* each client is assigned to its closest opened centre, we use DT as an efficient adjacency structure to generate neighbours of a solution in the proposed local search algorithm.

Let $\mathcal{F}$ be a set of m potential facility centres. Suppose $C = \{c_1, c_2, \ldots, c_k\} \subseteq \mathcal{F}$ is a solution, a set of k opened centres, and let $DT(c_i), i = 1, 2, \ldots, m$, be the neighbours of the site c_i in the Delaunay Triangulation graph of $\mathcal{F}$. A new random solution $C' = \{c_1, c_2, \ldots, c_{i-1}, c_i', c_{i+1}, \ldots, c_k\}$ can be simply obtained using C by changing c_i to a random non-opened centre $c_i' \in DT(c_i)$. This relocation is a *local change* and can be handled efficiently (Gowda et al., 1983). Let denote this simple process by $C' = DTN(C)$.

4.3. Heuristic and approximation algorithms

When the size of a problem, e.g. the number of cities in TSP or the number of clients and potential facilities in FL problems, becomes large, there is no efficient algorithm to find the exact optimal solution. More precisely, for the problems belonging to the class of NP-hard, no polynomial-time algorithm has been presented to find their optimal solution. While, most of the optimisation problems arising in application areas such as TSP, clustering, and FL problems are NP-hard. So, to handle this issue, two general approaches have been suggested: heuristic algorithms and approximation algorithms. Both of these approaches are efficient in time but they find an approximation solution instead of the optimal one. Heuristic algorithms are popular and usually simple as a general framework to optimise a function, however, they cannot guarantee the extent to which the solution is near to the optimal solution, the final obtained solution.

Heuristic approaches, such as evolutionary algorithms (Davoodi & Mohades, 2011), particle swarm optimisation (Guan et al., 2019), simulated Annealing (Redi et al., 2020), meta-heuristics (DePuy et al., 2005) have been widely applied to real-world optimisation problems such as facility location (Hakli & Ortacay, 2019) and vehicle routing and TSP (Bajaj & Dhodiya, 2023; Braekers et al., 2014). All of these approaches are objective-based random search algorithms. They include two general phases, *exploring* the search space and *exploiting* the obtained solutions. They usually have an operator to increase the rank of a solution with high fitness, and another operator to generate a new neighbour solution using some solutions. One of the main disadvantages of heuristic approaches is that some predefined parameters need to be tuned by the user. For example, the size of the solution population, termination conditions, crossover and mutation rates in EA, or cognition and social weights in Particle Swarm Optimization (PSO).

Approximation algorithms can guarantee that their obtained solution is not worse than an approximation ratio α in comparison to the optimal solution of a problem. More precisely, let Π be a minimisation problem and *Alg* be an approximation algorithm for Π. Also, let I be an instance of Π with optimal value $opt(I)$. If $Alg(I)$ denotes the value of the obtained solution by *Alg* for *I*, the approximation factor $\alpha(alg)$ is

$$\alpha(Alg) = \max_{\text{all valid instances } I \text{ of } \Pi} \frac{Alg(I)}{opt(I)}. \quad (5)$$

Indeed, $\alpha(alg)$ shows the worst approximation factor of *Alg* among all possible instances of the problem. So, it guarantees the output solution of the algorithm is never worse than α times of the optimal solution (Williamson & Shmoys, 2011). There have been a bunch of heuristic and approximation algorithms for the TSP and FL problems, particularly for metric TSP, k-median and k-centre problems discussed in this paper (see the surveys Charikar et al., 2002; Laporte, 1992; Svensson et al., 2020; Vazirani, 2013).

There are also so many heuristics and approximation algorithms for many variations of TSP. While the TSP problem is not approximable in the general case, the Christofides algorithm is a $\frac{3}{2}$-approximation algorithm for solving TSP in metric space (Christofides, 1976). This algorithm first computes a minimum spanning tree (MST) of the cities in the problem and then inserts a minimum matching on the odd-degree nodes of the MST. Consequently, it obtains a graph of all even-degree nodes. Finally, it computes an Euler tour on the graph. It may also use the *shortcut* edges in the metric space. Since the optimal solution for TSP is always greater than MST's length, and greater than half of the minimum matching, the approximation ratio of the Christofides algorithm is $\frac{3}{2}$. More details can be found in (Vazirani, 2013). The minimum matching can be computed in $O(k^{1.5} log^5 k)$ time in Euclidean space, where k is the number of cities (Varadarajan 1998), and computing MST and Euler Tour in the plane takes $O(k \log k)$ time (Cormen et al., 2009; Shamos & Hoey, 1975). So, the complexity time of the Christofides algorithm is $O(k^{1.5} log^5 k)$. Recently, Karlin et al. (Karlin et al., 2020) succeed in slightly improving this bound to $\frac{3}{2} - \varepsilon$ for $\epsilon > 10^{-36}$. Also, different variation of TSP including Multimodal Multiobjective variation (Liu et al., 2023) and a minimax multiple depots variation (He & Hao, 2023) have been studied recently.

To solve a problem using heuristic algorithms, two prerequisites are needed, first represent each solution of the problem as a well-defined structure, and second, define an evaluation method for each structure (or at least a method to compare two structures). So, they are simple and general purpose. To enhance their efficacy, it is necessary to customise them using facts and observations in the search space of the problem. In the next section, we propose a new heuristic approach based on the Delaunay triangulation. Our algorithm has not had many parameters to tune by the user. It also applies approximation algorithms for TSP and k-centre in generating solutions. This helps the algorithm benefits from approximation algorithms' advantages as well.

Appendix (*B*)

Theorem A.1: *In the problem of k-median&TSP, the number of Pareto-optimal solutions in the worst case is $\Omega(2^p)$, where p is in order of the problem's size.*

Proof: The idea behind the proof is to construct an instance of the problem with n demand points that contains at least $\Omega(2^p)$ Pareto-optimal solutions, where $p = O(n)$. ■

Definition A.1: Let $gadget(i)$ be a set of 6 demand points located on the positions $(2Li, 0), (2Li, L), (2Li, L + 2^i)$, $(2Li + L, 0), (2Li + L, L), (2Li + L, L + 2^i)$, and 8 potential facility points located on the positions $(2Li, 0), (2Li, L), (2Li, L + 2^i)$, $(2Li + L, 0), (2Li + L, L), (2Li + L, L + 2^i), \left(2Li + \frac{L}{2}, \frac{L}{2}\right)$, $\left(2Li + \frac{L}{2}, \frac{L}{2}\right)$, where L is a sufficiently large positive value (see Figure 3).

There are at least two different ways to solve the k-median&TSP problem for a $gadget(i)$ with $k = 6$. We may pick the facility centres $(2Li, L + 2^i)$ and $(2Li + L, L + 2^i)$ or not. If we pick them, we call such a structure an *active gadget*, otherwise, we call it an *inactive gadget*. See Figure 3. Let denote the objective value for k-median (sum of the distance between the clients and centres) with F_1, and the length of the TSP tour with F_2. So, for an inactive gadget(i), $F_1 = 4$ and $F_2 = 2 * 2^i$ while for an active gadget(i), $F_1 = 0$ and $F_2 = 4L + 2 * 2^i$. Therefore, $F_1 + F_2 = 4L + 2 * 2^i$.

Now, we are ready to show a workspace containing p gadgets $gadget(0), gadget(1), \ldots, gadget(p)$ with a large value $L > 2^p$. For $k = 6p$, we have two extreme Pareto-optimal solutions, the solution with minimum TSP tour and the solution with minimum k-median value. Figure 4 shows such a workspace for $p = 3$ and the two extreme Pareto-optimal solutions. If all the gadgets are active, we find the Pareto-optimal solution $F_1 = 0$ and $F_2 = 4pL + 2\sum_{i=0}^{p} 2^i$, while if all the gadgets are inactive, we find the Pareto-optimal solution and $F_1 = 2\sum_{i=0}^{p} 2^i$ and $F_2 = 4Lp$. Also, for constructing the other Pareto-optimal solutions, we can activate or inactivate $gadget(i)$, for $i = 1, 2, \ldots, p$. Note that, the value $F_1 + F_2 = 2\sum_{i=0}^{p} 2^i + 4pL$ holds for all such solutions. Therefore, if s_1 and s_2 are two different such solutions, since $F_1(s_1) + F_2(s_1) = F_1(s_2) + F_2(s_2)$, so they have different values of the objectives; consequently, they are non-dominated. On the other hand, it is straightforward to show by induction on p, there is no solution s such that $F_1(s) + F_2(s) < 4pL + 2\sum_{i=0}^{p} 2^i$. Consequently, by activating or inactivating the gadgets it is possible to have $\Omega(2^p)$ different Pareto-optimal solutions. ■

Appendix (*C*)

Table 1 shows the time complexity of the proposed algorithm in the worst and expected cases. As can be seen, the bottleneck of the algorithm is computing the objective functions, the length of TSP tour and k-centre or k-median objective values. In Step 2, we compute VD and $\frac{3}{2}$-approximated TSP tour for a solution of k centres. As mentioned before, Christofides algorithm for k cities runs in $O(k^{1.5} log^5 k)$ time in the Euclidean space in Step 2(a). Running it for all N solutions takes $O(N(k^{1.5} log^5 k))$ time. In Step 2(b) VD can be computed in $O(klogk)$ time. Having the VD, helps to demand points takes $O(logk)$ time. So, computing the nearest centre for all demand points needs to $O(nlogk)$ time together. So, Step 2(b) takes $O(N(klog\, k + nlog\, k))$ time for all N solutions. Note that Step 1 and Step 2 run just one time in the algorithm, and Step 3 and step 4 iterate M times. As explained above, generating new solutions is simply performed using Delaunay triangulation and the procedure of reducing the size of the population from $2N$ to N can be done in $O(NlogN)$ time. Therefore, the most critical step is to compute objective values of generated solutions which iterates MN times in general. Notably since each solution C' is constructed locally (changing between two neighbouring potential facilities in DT) using a solution C, we can also obtain (update) objectives value of C' using objective values of C instead of re-computing. To this end, note that the degree of each vertex in DT (or the number of neighbouring cells) is 6 on average. Besides, the nearest opened centre of $O\left(\frac{n}{k}\right)$ demand points are changed in the expected case. So, the objective value $F_1(C')$ can be computed in $O(n)$ time in the expected case. Similarly, updating MST, minimum matching and Euler tour using the depth-first search idea results in $O(k)$ expected $\frac{3}{2}$-approximated TSP tour for C'.

The size of the search space in the problem of FL&TSP is $O\left(k!\begin{pmatrix} m \\ k \end{pmatrix}\right)$. Indeed, there are $\begin{pmatrix} m \\ k \end{pmatrix}$ choices for picking k centres from m centres and each of them has $k!$ possible TSP tours. So, we tune the number of solutions (N) and the maximum number of iterations (M) concerning k and m. That is, the number of demand points does not play a role in the value of N and M. Clearly, increasing N and M increases the probability of achieving real Pareto-optimal solutions, however, it also increases the complexity time of the algorithm. This issue always happens in heuristic search algorithms and there is a trade-off between the efficiency and complexity of such algorithms. On the other hand, when a population convergence, it (almost) does not progress anymore. So, there needs to have a balance between the size of the population and the number of generations. Thus, as a strong suggestion and to let the algorithm be user-defined parameter-free, we set $N \cong 2(m + k)$ and $M \cong km$. This setting is based on our simulation results, and such a setting works efficiently in general and achieves admissible solutions in a reasonable running time.

Most of the popular heuristic search algorithms such as Genetic Algorithm (GA) and PSO are population-based algorithms, that is, a set of solutions have been initialised and an operator, e.g. crossover, tries to explore the search space by generating a new solution, called *child*, using randomly combining two or more solutions, called *parents*. In the proposed algorithm we have a set of solutions but exploration or generating a new solution is just based on one solution, say a parent slightly evolves to a child. So, in this regard, our proposed algorithm is an individual-based algorithm. However, we use the power of population information for enhancing exploitation, that is, in each iteration, only half of the combined parents and child solutions survive considering their fitness value in the population.

The proposed algorithm involves two main phases, exploration and exploitation. Step 3 is a smooth local exploration using connectivity information in the search space of the problem. We may image the search space of the problem as an m-vector in a binary m-dimensional space. Each feasible solution is a $0\backslash 1$ vector with length m that contains exactly k entries 1. Generating a new solution C' using $DTN(C)$ in Step 3, is exploring a new solution in the search space. The *Hamming* distance between C' and C is exactly 2. So, this is a local search that smoothly results in a global search. Note that, we always admit C' as a new solution in the population, even if it is dominated by C or any other solution. Indeed, one greedy way to generate a new solution is generating a solution that is not worse than C, that is, at least it is not dominated by C. Based on our simulation results and theories in the literature of heuristic algorithms, it is better to let C' survive even if it is dominated by C. This idea helps to escape from the local optima and to perform a global search. However, for applying exploitation, we use a non-dominated sorting procedure in Step 4 and pick half of the solutions. This method lets us implement a global elitism strategy in each iteration of the algorithm.

Note that, the idea of VD is useful for both k-median&TSP and k-centre&TSP problems. Also, the proposed algorithm can be extended easily to weighted demand points. Weight on the demand point just affects the second objective and the length of the TSP tour remains unchanged. So, this can be automatically handled when we compute the second objective value without increasing the complexity time.

The proposed algorithm is a bi-objective algorithm inherently. However, it can be extended for one or more than two objectives. For example, we might be interested to find Pareto-optimal solutions of a three-objective optimisation problem with objectives *min-sum* and *min-max* in the client-side simultaneously (Roostapour et al., 2016). Also, as mentioned before, if in some applications priority weights among the objectives provided by the decision-maker, we may convert the problem to a single objective optimisation using a weighted linear aggregation of the objectives. The only difference, in this case, is that the non-dominated sorting procedure in Step 4(a) reports a sorted list of $2N$ solutions concerning their combined objective value, and N bests of them will easily be selected for the next generation in Step 4(b). So, the proposed algorithm is adaptable for such priority weights as well.