Toward unraveling multi-objective optimization problems: A hybrid approach for solving a novel facility location problem

Maryam DehghanChenary 1, Arman Ferdowsi 2,2, Fariborz Jolai 1, and Reza Tavakkoli-Moghaddam 1

1Affiliation not available
2TU Wien

November 8, 2023

Abstract

The focus of this paper is to provide an authentic approach to solving bi-objective optimization problems. The target problem is a novel extension of a multi-period $p$-mobile hub location problem, which takes into account the impact of the traveling time at the hubs’ network, the time spent at each hub for processing the flows, and the delay caused by congestion at hubs with specific capacities. We first develop a mixed-integer mathematical model corresponding to the context problem. Afterward, a hybrid meta-heuristic algorithm will be proposed to solve the unveiled model that operates based on simultaneously employing a novel evaluation procedure, a clustering technique, and a genetic approach. The experiments validate that the proposed algorithm performs significantly better than several state-of-the-art algorithms. Furthermore, the decisive effect of two considerable factors: congestion and service time, are also analyzed.
Toward unraveling multi-objective optimization problems: A hybrid approach for solving a novel facility location problem

Arman Ferdowsi · Maryam DehghanChenary · Fariborz Jolai · Reza Tavakkoli-Moghaddam

Abstract The focus of this paper is to provide an authentic approach to solving bi-objective optimization problems. The target problem is a novel extension of a multi-period p-mobile hub location problem, which takes into account the impact of the traveling time at the hubs’ network, the time spent at each hub for processing the flows, and the delay caused by congestion at hubs with specific capacities. We first develop a mixed-integer mathematical model corresponding to the context problem. Afterward, a hybrid meta-heuristic algorithm will be proposed to solve the unveiled model that operates based on simultaneously employing a novel evaluation procedure, a clustering technique, and a genetic approach. The experiments validate that the proposed algorithm performs significantly better than several state-of-the-art algorithms. Furthermore, the decisive effect of two considerable factors: congestion and service time, are also analyzed.

Keywords Multi-objective optimization · Facility location problem · Combinatorial optimization · Heuristic approach

1 Introduction

Mathematical optimization is a considerable and growing domain that provides strict procedures to achieve (near) optimal solutions to engineering, science, and finance problems. The rapid growth of technology and research in various scientific fields has led to an emergence of a wide range of new optimization problems every day. On the other hand, the complex problems we face in the new world often consist of more than one goal, forcing us to optimize multiple objectives simultaneously. Multi-objective optimization problems consider at least two conflicting optimization goals, meaning that improvement to one objective comes at the expense of another objective. This results in not having a unique optimal strategy for these problems but rather a set of designs that represent the optimal trade-offs of the optimization objectives. These trade-offs are referred to as Pareto optimal solutions. More specifically, Pareto optimality is the condition at which one objective cannot improve without worsening the other one; See [16].

In the literature, there is a large number of approaches developed for the optimization problems, such as the (non-)gradient-based [46], [38] or heuristic search methods [29]. However, turning the attention toward multi-objective optimization problems indicates that more elaborate approaches are required to achieve the Pareto optimal solution. As pointed out in [45], at a glance, these approaches can be divided into three basic categories. The most basic one is to use standard optimization methods and pass through a range of constraints, each of which demands a specific optimization, afterward. Another strategy is to consider one objective that is made up of a weighted sum of all objectives rather than a constraint. Same as the previous approach, a range of weights can be used to solve the optimization problem, yielding an approximate Pareto
front. However, both of the mentioned techniques suffer from a high computational running time since a large number of optimization runs is required. On the other hand, genetic-based methods are among the most reliable methods for seeking the Pareto optimal solutions. The privilege of genetic algorithms is that a single run of the optimization algorithm is often needed to develop a Pareto curve because these methods use a population of possible solutions rather than a single search agent. Nevertheless, it is worth mentioning that the nature of the context problem can also have a severe impact on choosing/obtaining a proper optimization technique [23]. From this perspective, we converge our focus on one of the well-known and crucial optimization problems: the Facility Location Problem.

In network and graph theory, designing a network $G$ can be referred to as the problem of selecting several nomination edges (i.e., connections) between the vertices of $G$ to curtail the sum of constructions. Roughly speaking, the Facility Location Problem (FLP) is one of the fundamental categories of network designing problems and plays a decisive role in many research areas, such as structuring postal services [65], controlling air traffic [62], scheduling manufacturing factories [35], supply chain warehouses [9], and even analyzing computer networks [1] [24], [44]. Overall, an FLP seeks to locate the number of facilities (i.e., hub vertices) that aim to satisfy the demands of clients (the remaining nodes) provided that several constraints hold, as detailed in references [58], [36]. Consequently, it is about designing a distribution flow network of hubs and customers and connecting them to each other to optimize energy/time efficiency and commodity shipping costs as accurately as possible.

The problem’s input is a set of customers and their needs. The goal is to find the facilities’ locations and how each customer is allocated to these facilities so that the services can flow optimally. In this fashion, the network scheme of an FLP needs to be designed in a way that a specific and limited number of nodes are selected as hubs. These hubs form a complete graph and have the duty to merge flows from origin nodes and deliver them to destination nodes to satisfy the desired demand of each vertex (i.e., customer) as well as to minimize the total cost and service time. Hence, the primary pursuit is to select suitable locations from the available places to establish facilities there. These hub vertices will then characterize the flow paths and organize the transit among sources and destination (demand) spots.

There are two basic hypotheses in the study of this problem that can be summarized as follows: (i) there must be no direct route between two non-hub nodes, and (ii) each non-hub node has to be allocated to precisely one hub vertex. The latter is referred to as the single-allocation condition [20]. We suppose that the total number of nodes and the number of hubs are fixed/given and denoted by $n$ and $p$. Since each node has a demand that has to be satisfied by a responsible hub, the objective is to accurately locate the hubs, which results in fast transferring of the services so that all demands are satisfied, and the transportation cost achieves its lowest possible value.

There are various characteristics involved in FLP that can be studied individually or even in combination. More particularly, each scenario leads to a distinct variant of the problem. Below we refer to those that are considered in this article.

One crucial aspect of FLPs is that hubs can have capacities. In this case, the problem’s model needs to be equipped with capacity constraints that guarantee that the total demands of clients allocated to a hub do not exceed the hub’s capability.

Another critical factor of the $p$-hub location problem is dynamism. Clearly, in the real-world, the decision is not limited to just one period of time (e.g., yearly), meaning that many structural operations, decisions variable, problem parameters, budgets, and many other influencing variables may change over time. To be more precise, early decisions may change under various factors. For instance, demands (i.e., flows) can differ in different periods. In dynamic FLPs, new facilities are allowed to be established in each period. Dynamic FLPs consist of two categories: One is to change the location decisions in each period, and one is to make the strategic location decisions initially. In the latter case, the location decisions are not allowed to change during periods [9].

A further property, which can be investigated, is the mobility of hubs. In a nutshell, hubs can belong to two groups: fixed and mobile hubs. Mobile hubs are capable of movement. That is, they can move to other places if needed. One can trivially think of several applications and examples of mobile hub problems, including emergency services, fire stations, and blood collection centers. These queries can be created during a situation of crisis and be relocated in the case of necessity to offer services to the demand points with lower cost and shorter time [2]. One important argument is that since mobile hubs should be able to move among other vertices, proper infrastructures between nodes have to be provided in the first place. It means that these movement infrastructures have to be provided in the first period to be used in other periods.

Another crucial argument that can be arisen is that most studies about facility location problems focus mainly on reducing costs and ignore possible other goals like
delivery time. This is while there is a strong preference for the customers to receive services from a center that can provide resources with lower cost and shorter service time. Therefore, in recent years, increasing competition between service centers (e.g., factories) has been accrued in terms of paying attention to the service and waiting time for providing services. Service time is a vital and inseparable component of FLPs, and in its absence, the designed network may not be feasible in terms of the desirability of the service level. The service time can be divided into two components: (i) the time it takes to reach streams from non-hub nodes to hubs and (ii) time elapsed at hubs for proceeding the flows. We call the latter the handling time. Handling time at a hub depends on the capacity usage percentage and congestion level of that hub. Please be noted that delays may occur during the flow process at hubs. This delay is because the hub is working with almost its total capacity. In this case, the hub is congested. Consequently, having a limited capacity at hubs in transportation networks causes congestion that leads to increasing the waiting time for receiving the services and delays the service time as well. Therefore, creating a transportation network based on decreasing congestion is a fundamental issue. Generally speaking, the congestion level is calculated by the two following factors: (i) the amount of flow that hubs need to process and (ii) maximum hub capacity.

Main contribution: As the baseline model, this work first generalizes one of the basic classes of a mobile facility location problem by concurrently taking hubs’ congestion and the service time factor into account. To the best of our knowledge, this is the first time these two crucial factors are being simultaneously applied to such a problem. More precisely, this paper considers an FLP problem covering the dynamism factor, the mobility of hubs, service time limit, capacity, and congestion. In this regard, the aim is to design a transportation network and (near) optimally locate the mobile facilities such that the transportation cost, cost of creating and transporting hubs, cost of establishing mobile infrastructure, the ready time, and congestion level at hubs in three different periods all get minimized. Incorporating these considerations led us to devise a complicated bi-objective mixed-integer optimization problem. As expected, simultaneously reducing the costs and delivery time turned out to contrast with each other, resulting in a challenge to create a proper equivalence between these two issues.

On the other hand, since an FLP model falls in NP-hard problems [64], no exact algorithm likely exists that returns the optimal answer at a reasonable time. As a result, it seems plausible to rely on heuristic approaches as a fundamental way for successfully solving this problem. Hence, we propose a hybrid meta-heuristic algorithm for solving the recommended problem in this work. Our hybrid approach consists of the following four main segments:

1. Creating a set of initial feasible solutions.
2. Estimating the goodness of the solutions based on a novel evaluation procedure and converting them into a matrix embedding formed by the obtained ranks.
3. Clustering the solutions utilizing the determined matrix form.
4. Creating new batches of feasible solutions based on a genetic approach triggered by the solutions whose selecting probability is directly related to the quality of the cluster they belong to. More precisely, a probabilistic model will be taken into account for governing the selection process.

In our devised algorithm, stages 2 to 4 will be repeatedly processed until the best possible solution can be gained.

Furthermore, we managed to evaluate our algorithm under variant conditions comprehensively. We tested our algorithm based on the well-known Australia Post dataset under considering different conditions, including different numbers of nodes, different numbers of algorithm’s iterations, different numbers of initial solutions, different numbers of hubs, etc. We compared our results with those obtained by several other state-of-the-art algorithms, which are all among the well-known and commonly-used facility locations algorithms. We also compared the outcomes with the ones obtained by the very powerful solver CPLEX. As the results show, our method outperforms all the comparable algorithms in all experimental cases in terms of the running time as well as the solutions’ quality. Besides, we also investigated the role of considering the mobility of hubs, congestion, and service time.

Paper organization: Section 2 gives a brief history of FLPs and some related state-of-the-art research work. Section 3 models and formulates the proposed p-mobile hub location problem considering congestion and service time. Section 4 is dedicated to presenting our hybrid meta-heuristic algorithm. Section 5 then, brings forward the experimental results, and finally, Section 6 briefly summarizes the proposed work and points the challenges ahead and upcoming future work.

2 Literature review

In 1987, O’kelly introduced the single-allocation p-median location problem [52], which attracted many other re-
searchers to study, extend and develop this subject. After that, Campbell [13] introduced a mathematical model for the single-allocation p-hub median problem in which the limited capacity of incoming flow to hubs was considered. Capacity is an essential property in hub location problems (HLP) that can be assumed for both hubs and flow transfer edges. One can think of considering some different capacity levels for each hub so that each of them has to pick one capacity level. In this fashion, Correia et al. [18] have developed a mathematical model for the single-allocation HLP with capacity constraints in terms of several capacity levels.

On the other hand, another argument might be that one can face a congestion situation in transportation systems since high or increased demand for services may occur. This case happens when an intended hub works near its full capacity, and in this situation, we say that the hub is congested. The congestion issue was introduced by Grove and O'Kelly [27]. It is known that the queue theory has been used for considering congestion in most papers. Karimi et al. [37] addressed the single allocation hub-and-spoke network design under hub congestion. They modeled the network through a bi-objective non-linear integer programming. The model was developed based on a general GI/G/G queuing system. They managed to solve their proposed model using a learning-based meta-heuristic based on NSGA-II, k-means clustering method, and an Iterated Local Search. Ghodratnama et al. [25], on the other hand, developed a new hub location with considering congestion and production schedule. They developed a bi-objective hub location model to deal with the problem. In their consideration, the first objective is to minimize the total cost, and the second objective is to minimize the waiting time for processing goods by modeling the system as a queuing network. In order to solve the bi-objective problem, they devised a novel hybrid Goal Attainment and LP Metric approach.

On the other hand, however, Alumur et al. [5] modeled the service time and congestion constraints in single and multi-allocation HLP by avoiding the queue theory. Rodriguez et al. [59] stated that service time consists of traveling time through network and time spent in hubs. In most cases, time spent in hubs is considered by the waiting time in a queue. However, like what Alumur et al. [5] explained, we calculate the time spent at hubs by considering the capacity and congestion levels of hubs and also the hub’s capacity percentage of use.

Another important argument is that in real-world cases, decisions are not made for only one specific period. During the implementation of extensive structural operations, sometimes planning horizons needs to be considered and predicted by particular periods. For instance, demand (i.e., flow) may differ in the transportation problem in various periods. The subject of considering different periods in this kind of problem was first introduced by Melo et al. [47]. Alumur et al. [4] presented a mathematical model for a multi-period (i.e., dynamic) HLP with single and multiple allocations and considering modular capacities. Correia et al. [19] brought forward a mathematical model for a multi-period HLP considering stochastic flows and limited capacity.

One can also investigate this problem from another critical viewpoint: mobility of hubs. The intuition is that mobile facilities can move to other locations without any establishment costs to satisfy the client’s demands. Halper and Raghavan [30] first introduced the mobile hub problem. Lei et al. [42] presented a two-stage mixed-integer stochastic approach for routing mobile facilities with stochastic demands in a discrete number of scenarios. In the first stage, a decision about whether the mobile facilities are used or not and where and when they are used is made. In the second stage, a decision about how clients allocate to these facilities is made. Güden and Sürål [28] studied a mobile FLP in the field of railway construction so that mobile facilities can move through railways. Furthermore, Bashiri et al. [8] presented and studied the single-allocation p-mobile hub location problem in the dynamic environment. This was a tremendous breakthrough because of combining two significant factors: dynamism and mobility of hubs. In their model, hubs can move instead of opening and closing. They compared their proposed model with the classical dynamic hub location problem. They were successful in showing that the model is much more efficient, and the total cost significantly decreases whenever hubs can move easily. It is worth mentioning that they used the genetic algorithm and the local search approach to solve their model. Qi et al. [57] introduced a mobile hub routing problem that considers service time related to the demand rate and fairness constraints. They developed a mixed-integer programming model for their problem and used a meta-heuristic algorithm for solving it. Beside, Mokhtazardeh et al. [50] investigated a multi-objective p-mobile hub location problem with the depreciation cost of hub facilities. The authors then developed a hybrid of k-medoids, a clustering algorithm, with two meta-heuristic algorithms to solve the presented problem.

We refer the reader to [15], [22], [8], and [14] to see comprehensive surveys on this problem and its different aspects.

Let us point out that one can also investigate the context problem from the perspective of the solution approach. For instance, there is a small number of approaches that are based on optimally solving the math-
A heuristic approach for solving a new model of facility location problem

Mathematical programming model; See [53], and [26, 61, 51]. Nevertheless, as mentioned, since the under-study problem is an NP-hard one, most of the solution approaches that have been proposed so far follow (meta) heuristic manners; See, for example, [63], [32], and [7].

On the other hand, it turned out to be unlikely to find a straightforward optimization since the problem lies in multi-objective optimization problems. To be more precise, in this kind of problem, different objective functions usually conflict with each other, meaning that minimizing one of them can lead to maximizing the other one. This obstacle led to the emerging a new concept of the optimal solution, the Pareto solutions. As we will figure out later, this concept will help us to deal with multi-objective optimization. In consequence, although all the heuristic approaches, which have been developed so far, work based on different methodologies, they almost aim at finding the Pareto optimal solutions. For instance, [31] provides a genetic-based for obtaining Pareto solutions, while [34] proposes a dynamic aggregation technique for doing this. However, one of the recent approaches that some researchers have recently noticed and used is to apply a clustering method to somehow evaluate solutions with simultaneous respect to each objective function: See [12] and [33].

We end this section by referring the interested reader to [15] for attaining a comprehensive vision of this problem.

3 Problem definition and the proposed model

In this section, we introduce the basic assumptions correlated to our under-study FLP. Afterward, we bring forward the mathematical formulation of the devised model.

3.1 Problem assumptions

The classic form of a hub location problem consists of the three following basic assumptions:

- No direct route between an origin and a destination point is allowed to exist.
- The connection between hubs is complete, i.e., there is a direct route between each pair of hubs.
- Flow transfer between each pair of hubs includes a discount factor α which causes to save the cost of mass transit of goods.

In addition to the above basic assumptions, this work also takes the following premises into account [5, 8]:

- Single-allocation strategy is considered.
- The number of hubs in each period is fixed and equal to p.
- Each node has the potential of being a hub.
- Hubs mobility infrastructures are created in the first period and can be used in the next periods.
- If there is any mobile infrastructure, hubs can move to other vertices in the next periods; otherwise, a new hub node should be set up.
- Each hub can tolerate a certain amount of incoming flow, meaning that each hub has a specific capacity level.
- A service time limit exists for sending flow among origin and destination nodes. That is to say, each terminus node has to gain services within a given specific service time limit.
- A generic service time limit exists for the entire network.
- Service time between two nodes is obtained by the travel time on the network plus the handling times at hubs.
- The total cost for locating hubs with specific capacities, flow transfer, creating mobile infrastructure, and hub transfer is calculated.
- Transportation costs satisfy the triangle inequality, resulting in that each transition from an origin to a destination can be directed through at most two hubs since the hub network forms a complete graph.
- Traveling time is symmetric.

3.2 Mathematical formulation

As mentioned, this paper aims at providing a bi-objective mixed-integer mathematical programming model for a single-allocation multi-period mobile hub location problem at which the hubs are equipped by capacity and congestion levels. The proposed model is heavily inspired by the model which Bashiri et al. [8] and Alumur et al. [5] have introduced. For doing this, in the sequel, we introduce the model’s variables and parameters, and afterward, we provide the mathematical model followed by its description.

3.2.1 Sets, decision variables, and parameters

Sets:

-N Set of vertices
-C Set of capacity levels of hubs
-G Set of congestion levels of hubs
-T Set of periods

Decision variables:

Our model consists of seven 0 – 1-valued variables and one real-valued variable, leading us to face a mixed-integer programming problem.
$X_{i,k}^t$ if in period $t$ vertex $i$ is allocated in hub $k$; 0, otherwise
$Z1_{i,k}^t$ if in period $t$ vertex $k$ is a non-mobile vertex; 0, otherwise
$Z2_{i,k}^t$ if in period $t$ hub $i$ transfers to hub $k$; 0, otherwise
$Y_{e_{k,c,g}}^t$ if in period $t$ hub $k$ has capacity $c$ and congestion level $g$; 0, otherwise
$Y_{i,k,l,i,j}^t$ if in period $t$ the flow from node $i$ and node $j$ is established from hubs $k$ and $l$; 0, otherwise
$R_{k,t}$ 1 if mobile infrastructure between nodes $k$ and $l$ is established; 0, otherwise
$Y_{e_{k,c,g}}^{t-1}$ if node $k$ is in periods $t-1$ and $t$; 0, otherwise
$r_k^t$ ready time in hub $k$ in period $t$

**Parameters:**
- $\alpha$ Discount rate
- $p$ Number of hubs
- $W_{ij}$ Flow sending from node $i$ to node $j$
- $O_i = \sum_{j \in N} W_{ij}$ Total flow emanate from $i$
- $D_j = \sum_{i \in N} W_{ij}$ Total flow entering $j$
- $F_{k,c}^t$ Fixed cost for constructing hub $k$ with capacity $c$ in period $t$
- $F_{k,c}^{t-1}$ Cost of transferring a mobile hub from node $i$ to node $k$ in period $t$
- $C_{k}^{1}$ Cost of flow transfer between node $i$ and hub $k$ in period $t$
- $C_{k,l}^{2}$ Cost of constructing a route between hubs $k$ and $l$
- $t_{ij}$ Traveling time between nodes $i$ and $j$
- $\Delta^c$ Handling time of a hub with a capacity level $c \in C$
- $\sigma^t$ Service time limit in period $t$
- $\Gamma^c$ Capacity of a hub with a capacity level $c \in C$
- $\gamma^g$ Congestion factor (delay in time, associated with the congestion level $g \in G$)
- $\tau_g$ Congestion factor (delay in time, associated with the congestion level $g \in G$)

Be advised that the service time between two nodes depends on the traveling time through the network plus the handling time at hubs, where the handling time at hub $k$ depends on $k$’s capacity and congestion level. On the other hand, congestion level at $k$ depends on the percentage of $k$’s capacity utilization. In this sense, let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of capacity levels of hubs where, capacity level $c_i$ is higher than the capacity level $c_{i-1}$ for every $i \in \{1, 2, \ldots, m\}$. Then $\Gamma = [\Gamma^{c_1}, \Gamma^{c_2}, \ldots, \Gamma^{c_m}]$ and $\Delta = [\Delta^{c_1}, \Delta^{c_2}, \ldots, \Delta^{c_m}]$ are two vectors such that $\Gamma^{c_i} > \Gamma^{c_{i-1}}$ and $\Delta^{c_i} > \Delta^{c_{i-1}}$ ($1 \leq i \leq m$). By this notation, we mean that a hub with a higher capacity level has to contain a higher capacity and handling time. Furthermore, let $G = \{1, 2, \ldots, l\}$ be the set of congestion levels. $\gamma = [\gamma^1, \gamma^2, \ldots, \gamma^l]$ is a vector, where $\gamma^i > \gamma^{i-1}$ ($1 \leq i \leq l$). That is, a hub with a higher congestion level can use the higher amount of its capacity rather than a hub with a lower congestion level. On the other hand, $\tau_i > \tau_{i-1}$ ($1 \leq i \leq l$) and it means, the higher the congestion level of a hub is, the more delay in the hub for processing flow occurs. $r_k$ represents the ready time for hub $k$ and it is equal to the latest arrival or departing time of all flow arriving at or exiting from $k$.

### 3.2.2 Mathematical modeling

By inspiring from [6,8], in what follows, we present the mathematical formulation of our proposed model, which explains a p-mobile hub location problem that considers service time and congestion at hubs. We name the model PMHM that stands for p-Mobile Hub Model. Let $i, l, k \in N, c \in C, g \in G$ and $t \in T$. PMHM consists of two objective functions along with 20 constraints as follows:

$$\begin{align*}
\text{Min} & \quad \sum_{t,i,k} C_{i,k}^t W_{i,j} X_{i,k}^t + \sum_{t,i,k,l,j} \alpha C_{i,k}^t W_{i,j} Y_{i,k,l,j}^t \\
& + \sum_{t,k,c,g} F_{k,c}^t Y_{e_{k,c,g}}^t + \sum_{k,l} C_{k,l}^2 R_{k,l} + \sum_{t,k} F_{k,c}^{t-1} Z_{k,c}^{t-1} \\
\text{subject to:} & \\
Z_{21,k}^t = 0, \quad \forall i, k & \quad (3) \\
X_{i,k}^t = Z_{11,k}^t + \sum_{i, i \neq k} Z_{22,k}^t, \quad \forall t, k & \quad (4) \\
\sum_k X_{i,k}^t = p, \quad \forall t & \quad (5) \\
X_{i,k}^t \leq X_{i,k}^1, \quad \forall t, i, k & \quad (6) \\
\sum_k X_{i,k}^t = 1, \quad \forall t, i & \quad (7) \\
Z_{22,k,i}^t \leq X_{i,k}^{t-1}, \quad \forall t, k, i, k \neq i & \quad (8) \\
Z_{22,k,i}^t \leq R_{k,i}, \quad \forall t, k, i, k \neq i & \quad (9) \\
X_{i,k}^t + X_{j,k}^t - 1 \leq Y_{i,k,l,j}^t, \quad \forall t, i, k, l, j & \quad (10) \\
Y_{i,k,l,j}^t \leq X_{i,k}^1, \quad \forall t, i, k, l, j & \quad (11) \\
Y_{i,k,l,j}^t \leq X_{j,k}^t, \quad \forall t, i, k, j & \quad (12) \\
Y_{e_{k,c,g}}^t = Z_{11,k}^t - Y_{22,k}^{t-1}, \quad \forall t, k & \quad (13) \\
Z_{11,k}^t + Z_{11}^{t-1} \leq Y_{22,k}^{t-1}, \quad \forall t, k & \quad (14) \\
Y_{22,k}^{t-1} \leq Z_{11,k}, \quad \forall t, k & \quad (15)
\end{align*}$$


\[ Y_{2,t-1} \leq Z_{1,t-1}, \forall t, k \]  
\[ \sum_{c,g} Y_{c,k,g} = X_{k,k}, \forall t, k \]  
\[ Z_{1,t} + \sum_{i \neq k} Z_{2,i,t} \leq 1, \forall t, k \]  
\[ \sum_{i} d_{i} X_{i,k} \leq \sum_{c,g} I_{c,g} Y_{c,k,g}, \forall t, k \]  
\[ r_{t} + \sum_{c,g} \Delta_{c} Y_{c,k,g} + t_{k,l} + \sum_{c,g} \Delta_{c} Y_{c,l,g} + r_{l} \leq \sigma^{t} \]  
\[ X_{i,k}, Z_{1,t}, Z_{2,i,t}, Y_{1,i,k,l,j}, Y_{c,k,g}, R_{k,l}, Y_{2,i,t-1} \in \{0,1\} \]  
\[ r_{k} \geq 0: \forall t, i, k, l,j \]  

The objective function (1) minimizes the total cost of transportation, constructing mobile infrastructures, establishing hubs, and transportation at hubs. The objective function (2) minimizes hubs’ congestion level and ready time. Equations (3)-(22) provide the constraints of PMHM. Constraints (3) guarantee that no hubs are transported in the first period. Constraints (4) make sure that in each period and vertex, one hub can be constructed or transferred to another vertex. Constraints (5) show the number of hubs in each period. Constraints (6) say that neither of the non-hub nodes is adjacent. Constraints (7) emphasize that the single allocation strategy is considered. Constraints (8) show that if \( k \) is a hub in period \( t-1 \), it can move to hub \( i \) in period \( t \). Constraints (9) imply that if a mobile infrastructure is available between two nodes, a hub can move between these nodes. Constraints (10)-(12) express the route between pair of nodes. Constraints (13)-(16) are used for hub infrastructure differentiation. Constraints (17) explain that each hub can only have one capacity level and one congestion level. Constraints (18) say that in each period, a hub can be established or moved to other vertices. Capacity constraints are expressed by constraints (19). For each hub in each period, the congestion level is determined simultaneously by these constraints. The higher the usage percentage of a hub’s capacity (i.e., the maximum inflow that this hub processes), the higher its congestion level is. Constraints (20) consider an upper bound for the ready time, in the sense that the value of the ready time for a hub in each period has to be greater than or equal to the longest time for collecting flows from all the nodes allocated to that hub. Constraints (21) guarantee that the service time in each period does not exceed service time limit \( \sigma \). These constraints apply to the following two components of the total service time: (i) transportation time, which is independent from congestion level in hubs, and (ii) the handling time at hubs, which depends on congestion. Traveling time between pairs of origin and destination nodes is obtained by the sum of the ready time at each hub, handling time at hubs, and traveling time among the hubs. It’s worth mentioning that delay throughout congestion only sticks at the first hub on the route since capacity constraints are only inflicted on the inflow. Constraints (22) show the domain constraints (see [58]).

4 Solution approach

In this section, we are going to propose our hybrid meta-heuristic algorithm that aims at discovering a (near) optimal solution to PMHM. In a nutshell, our algorithm first creates a set of random feasible solutions to the model. It then performs an evaluation procedure to embed solutions’ qualities into a matrix representation. Next, employing the well-known K-means algorithm accomplishes a clustering task over the determined matrix. This batching helps us cluster solutions so that the solutions with approximately similar ranks belong to the same classes. In this fashion, a large number of solutions will be pruned instantly, and many of them belong to low-quality clusters will be omitted. Afterward, a genetic approach will be employed to generate new solutions based on the previously obtained high-quality remained solutions. By repeating this procedure, it is likely to expect to move toward finding better solutions. In the sequel, we dive into elaborating on each part of our method.

4.1 Representing and creating random feasible solutions to PMHM

Please be noted that this work considers three levels for both capacity and congestion associated with each hub. These levels are represented by the integer values 1, 2, and 3. Moreover, three periods of time \( T = \{1, 2, 3\} \) will also be considered. In what follows, we present a way for expressing a solution as well as creating a random feasible solution of PMHM for each period \( t \in T \).

For a specific time period \( t \in T \), we represent a feasible solution \( s^{(t)} \) to PMHM by an \( 5 \times (n + p) \) matrix where \( n \) and \( p \) are the number of nodes and the number of hubs, respectively. In this fashion \( s^{(t)}_{i,j} \) is said to be the element placed in the row \( i \) and column \( j \) of \( s^{(t)} \). The first row of this matrix is involved in creating a random feasible solution and the second to the fifth rows help us present it.
We first express the way we create this matrix. Afterward, we explain how this obtained matrix can perfectly represent a random feasible solution \( s(t) \).

- First row: the first \( n \) entries are filled by \( n \) random real numbers between 0 and 1. Elements located at positions (columns) \( n+1 \) to \( n+p-1 \) are filled out with random integer numbers between 1 and \( n-1 \) and the last element \(((n+p)-\text{th} \) column) gets the value of \( n \).

- Second row: suppose that, \( j_1 \) is the index of a column that contains the maximum element among the first \( n \) columns of \( s(t) \’s \) first row and let \( s_{2,j_1}^{(t)} = 1 \). Now assume that \( j_2 \) is the index of a column that contains the next maximum element among the first \( n \) columns of \( s(t) \’s \) first row and let \( s_{2,j_2}^{(t)} = 2 \). We continue this procedure until all the first \( n \) columns of the second row are set by the integer numbers 1 to \( n \). The \((n+1)-\text{th} \) to \((n+p)-\text{th} \) columns of this row get exactly the values of the \((n+1)-\text{th} \) to \((n+p)-\text{th} \) columns of the first row.

- Third and Fourth row: assume that \( A = \bigcup_{j=n+1}^{n+p} s_{1,j}^{(t)} \) and \( s_{3,j}^{(t)} \) and \( s_{4,j}^{(t)} \) for all \( j \in \{1,2,\ldots,n+p\} \) are defined as follows:

\[
s_{3,j}^{(t)} = \begin{cases} 
\text{random integer number between 1 and 3 , if } j \in A \\
0 & \text{, otherwise}
\end{cases}
\]

\[
s_{4,j}^{(t)} = \begin{cases} 
\text{random integer number between 1 and 3 , if } j \in A \\
0 & \text{, otherwise}
\end{cases}
\]

- Fifth row: we define entries of this row as follows:

\[
s_{5,j}^{(t)} = \begin{cases} 
\text{if } j \in A \\
0 & \text{, otherwise}
\end{cases}
\]

where \( A \) is the set defined in the previous case and \( j \in \{1,2,\ldots,n+p\} \).

We claim that the introduced matrix can perfectly represent a feasible solution: The second row of \( s(t) \) specifies hub nodes and the allocation of non-hub nodes to hubs. In this regard, \( s_{2,n+1}^{(t)},\ldots,s_{2,n+p}^{(t)} \) represent the indices of hub node. To be more specific, \( s_{2,j}^{(t)} \) is a hub vertex for every \( j \in \{n+1,\ldots,n+p\} \). On the other hand, each of the first \( n \) elements of this row, which is not a hub, is allocated to the first hub placed at the right-side of that element. The third and fourth rows respectively express the capacity and congestion levels of hubs. Furthermore, the fifth row represents the ready time value for each hub. With possession of all solution matrices \( s(t) \ (t \in T) \), all the decision variables introduced in the previous section can be derived.

Fig. 1 shows a sample of a solution matrix of a specific time period that contains 7 nodes and 3 hubs. As we can figure out, nodes 5, 2, and 3 are hubs. Node 6 is allocated to hub 5. Node 7 is allocated to hub 2, and nodes 1 and 4 are allocated to hub 3. In addition, for example, hub 5 has the capacity and congestion level 1.

Fig. 2 represents the network scheme corresponding to the solution shown in Fig. 1.

Suppose that the weights over the edges in Fig. 2 show the traveling time between each pair of vertices, and also, weights over the hub vertices show the handling time. In this case, ready time, for example, for hub 3 (i.e., \( r_3 \)) equals 5. The longest service time in this network is 14.5 h.

One important issue to note is that since our model is supposed to consider three periods for PMHM \((T = \{1,2,3\})\), each solution has to contain all the three periods’ information. Accordingly, we consider each solution to PMHM as a set \( s = \{s^{(1)},s^{(2)},s^{(3)}\} \) where, each \( s(t) \ (t \in \{1,2,3\}) \) is a solution corresponding to the period \( t \). Furthermore, recall that the first row is responsible for creating a solution by which the other rows will be defined. Accordingly, for each solution \( s \), we denote the first row of the matrix representation of \( s \) as the leader row and denote it by \( l \).

4.2 Evaluating the solutions

This section discuss a core part in our work. First of all, please recall that since we are dealing with a bi-objective problem, it is impossible to find a feasible solution that could simultaneously minimize all the objectives. Therefore, we need a new prospect to express and obtain the optimal solution. In this regard, by a solution optimal, one can refer to as the Pareto optimal solution: a solution that cannot be improved in any of the objectives without degrading at the other objectives. Accordingly, in this kind of problem, one might think of looking for solutions that hold the respective balance between goals. To be more precise, solutions must be found that do not limit us to optimizing just one objective function. Consequently, in this fashion, most of the research work in this area utilizes the notion of domination. [21] Let \( s_1 \) and \( s_2 \) be two solutions to a bi-objective problem. \( s_1 \) is said to dominate \( s_2 \) if the two following conditions hold:

1. Neither of the objective functions’ values for \( s_1 \) shouldn’t have worse value of the objective functions for \( s_2 \).
A heuristic approach for solving a new model of facility location problem

1. Fig. 1: Representation of a random solution $s^{(t)}$

2. Fig. 2: Network scheme of the solution $s^{(t)}$ shown in Fig. 1. Nodes 2, 3 and 5 are hubs and are fully connected.

3. Fig. 3: A sample of Pareto solution in minimization problem. Blue points are feasible solutions, where those that are not dominated by any other solution are referred to as Pareto points (alloy orange). Together they form the Pareto front. The lower-left area points represent desired but not feasible (light blue/green). As will be discussed below, we take advantage of this notion for further evaluating the solutions to ultimately propose a method that aims to discover (near) Pareto optimal solutions.

In what follows, we first introduce a new function that enables us to evaluate each individual solution to another based on the well-known Chi-square distribution \[40\]. Utilizing the proposed function and the domination notion, we then devise vector representation for each solution, which further helps us evaluate solutions more accurately using a clustering technique. We coined the term VRCE for our Vector Representation Clustering-based Evaluation method.

Regarding the evaluation vector, we first let $f : S \rightarrow \mathbb{R}$ to be a monotonically non-decreasing function, such that $f(s(i))$ considers the number of solutions that $s(i)$ dominates and corresponds a real value number between 0 and 1 to that number. A very naive function $f$ could be defined in a way that for each solution $s(i)$, $f(s(i))$ is the normalized value of the number of solutions that $s(i)$ dominates. Nevertheless, we introduce a novel function, which further leads to much more accurate outcomes. We employ the Chi-squared distribution, the cumulative distribution function for defining the function $f$. Mathematically speaking, given a solution $s(i) \in S$, where $|S| = n$, assume that $m$ is the number of solutions $s(i)$ dominates. We define:

$$f(s(i)) = \frac{\gamma(1, \frac{\alpha \cdot m}{2})}{\Gamma(1)},$$

where $\Gamma(x)$ and $\gamma(y, z)$ are respectively the gamma and the lower incomplete gamma function \[40\], and $\alpha$ is an integer parameter ranging from 1.

To motivate our discussion, we first draw your attention to Fig. 4 which illustrates the shape of the function $f(s(i))$ with respect to the different number is solutions that the solution $s(i)$ dominates, for five different values for the parameter $\alpha$. $f$ is a monotonically non-decreasing function with the range $[0, 1]$. Therefore, it is apparent that the more solutions are dominated by the solution $s(i)$, the higher the value of $f(s(i))$, meaning that $s(i)$ is a high-quality solution. A closer look at the figure turns out that function $f(s(i))$ exhibits a steeper growth for the initial $m$’s values, followed by a smoother increase corresponding to the larger $m$’s values. In particular, this increment becomes even more severe for a more considerable $\alpha$ value. This nature is precisely in line with our desire which indicates the relative higher privilege of a solution that dominates at least a bunch of other solutions. Our investigations and experiments turned out that although the initial rate of dominations is very critical, from a certain point onwards, the number of domination should not significantly affect the quality of a solution because it will be at the cost of insufficient improvement in the value of objective functions.
Now, following this insight, assume that \( v_i \) is our proposed three-element vector representation corresponding to the solution \( s(i) \in S \). We let \( v_i(1) = f(s(i)) \). For obtaining \( v_i(2) \) and \( v_i(3) \), we turn our attention to the following argument. Considerably notable is that as crucial as the degree of dominance in measuring a solution’s superiority is as vital as the rate of improvement it causes to each of the objective functions. Interestingly, the simultaneous use of these two alternatives is a missing link in most research conducted in this field. Therefore, we try also to take the objective functions’ improvement rate into account. To be more precise, for the solution \( s(i) \), intuitively, \( v_i(2) \) and \( v_i(3) \) show how much the first resp. second objective function’s value can be improved (i.e., minimized) by \( s(i) \) over its dominated solutions. Mathematically speaking, we have:

\[
\begin{align*}
v_i(2) &= \frac{\sum_{s(j) \in D} obj_j^{(1)}(s(j)) - |D| \cdot obj_j^{(1)}(s(i))}{|D|}, \\
v_i(3) &= \frac{\sum_{s(j) \in D} obj_j^{(2)}(s(j)) - |D| \cdot obj_j^{(2)}(s(i))}{|D|},
\end{align*}
\]

where \( D \subseteq S \) is the set of solutions that are dominated by \( s(i) \), and furthermore, \( obj_j^{(1)}(s(i)) \), for example, means the first objective value in terms of the solution \( s(i) \). The higher the value of \( v_i(2) \) and \( v_i(3) \), the higher the amount of decrease yielded in objective values by considering the solution \( s(i) \), resulting in that \( s(i) \) is a better solution.

Once we obtained vector \( v_i \) for each solution \( s(i) \in S \), we let \( M = (v_1, \ldots, v_n)^T \). Instinctively, matrix \( M \) can be thought of as a dataset matrix, equipped with three features, whose each row is a vector representation of a solution that can perfectly reflect the goodness of that solution.

Right after achieving this matrix representation, the \( K \)-means algorithm will be applied in order to cluster the obtained solutions in terms of their similarities. To be more specific, noticing matrix \( M \) reveals that the similarity between two solutions can be expressed based on:

- Similarity between the number of solutions they can dominate.
- Similarity between the amount of improvement they can establish on the first objective function.
- Similarity between the amount of improvement they can establish on the second objective function.

In this vain, solutions with the same similarities are likely to belong to the same colonies. Considerably promising is that one can immediately conclude that two solutions in a same cluster are likely to dominate almost the same number of solutions, and moreover, they can decrease the objective functions by almost the same amount. To be more precise, this clustering can be thought of as an evaluation procedure. We end up this section with providing three following notations. Let \( C = \{C_1, \ldots, C_k\} \) be \( k \) clusters of solutions achieved by the \( K \)-means algorithm.

- We define the rank of a solution \( s_i \in S \) as \( r(i) = \sum_{j=1}^3 v_i(j) \).
- We refer to \( w(j) = \sum_{i \in C_j} r(i) / |C_j| \) as the rank of the cluster \( C_j \in C \).
- \( w^d \) denotes the rank of a cluster that contains the solution \( s_i \).

### 4.3 Creating new generations of solutions

To generate a new set of feasible solutions, which inherit the superior genomes of the parents (already obtained feasible solutions), a genetic approach is employed. However, considerably promising is that, unlike the naive genetic method, we wisely use the clusters obtained in Section 4.2 to generate more accurate solutions. In a nutshell, the idea is that the probability of generating a new solution based on two parents (two solutions) belonging to a high-quality cluster shall be more than when they belong to a lower quality cluster. Consequently, since the parents are often selected from high-quality clusters, it is very likely to expect the new solutions to get even better during the procedure.

The following repeatable procedure present the employed genetic algorithm. Be noted that two solutions
will be picked as parents in each iteration, and two new solutions will be generated afterward.

- Repeat the following steps for a desired number of iterations.
  - Pick two solutions \( s, s' \in S \) in such a way that the probability of choosing a solution \( s_i \in S \) is equal to
    \[
    \frac{w_i}{\sum_{j=1}^{n} w_j}.
    \]
  - Consider the leader rows of these solutions (i.e., \( l_s, l_{s'} \)).
  - Select one of the three following procedures with equal probability:
    - **Single-point crossover:**
      - A point on both \( l_s \) and \( l_{s'} \) is picked randomly. We coin the term a crossover point for it. Genes to the right of that point are swapped between the two parent solutions.
      - In this fashion we will have two new leader rows.
      - Based on them and also the procedure explained in Section 4.1, we create two new solutions and add them to \( S \).
    - **Two-point crossover:**
      - Two crossover points are picked randomly on both \( l_s \) and \( l_{s'} \). The hits in between the two points are swapped between the parent organisms.
      - Same as what we described for the case of single-point crossover, we create two new solutions and add them to \( S \).
    - **Mutation:**
      - We randomly choose two integer numbers \( r_1 \) and \( r_2 \) between 0 and \( n \) so that to change \( r_1 \) genomes of the \( l_s \) and \( r_2 \) genomes of \( l_{s'} \).
      - Randomly choose \( r_1 \) locations of the \( l_s \) and \( r_2 \) locations of \( l_{s'} \).
      - Replace each of the selected location by a random integer numbers between 1 and \( n \).
      - Create two new solutions based on the new leader rows and the procedure introduced in Section 4.1.
      - Add the obtained solutions to \( S \).

4.4 KGPMH algorithm

Gathering all the components together, we are now able to depict our main algorithm in an integrated way.

Our algorithm is named KGPMH, which stands for the Kmean-Genetic-P-Mobile-Hub algorithm. Algorithm 1 declares the pseudocode of KGPMH method. Please note that the number of nodes (\( n \)) and hubs (\( p \)) must be fixed, and given. In addition, be advised that three periods have been considered for the PMHM, meaning that \( T = \{1, 2, 3\} \).

Considerably remarkable is that despite KGPMH being described in the sense of an approach for solving bi-objective optimization problems and, in particular, a facility location problem, since the concept of domination can straightforwardly be generalized for multi-objective problems, KGPMH can perfectly be expanded to operate on any multi-objective problem.

5 Experimental results

The goal of this section is to explore the efficiency of our proposed method from different perspectives. To be more specific, our investigation is broken into the following six main parts:

- Investigating the role of the VRCE in improving the results.
- Comparing our results with ones obtained by two other state-of-the-art clustering-based facility location algorithms.
- Evaluating our algorithm against two commonly used multi-objective meta-heuristic algorithms NSGA-II and MOPSO.
- Analyzing our results with the very accurate solutions obtained by the CPLEX 12.90 solver.
- Exploring the effect of considering the mobility of hubs by analyzing the sensitivity over the related parameters \( C_2 \) and \( F_2 \).
- Inquiring the effect of considering service time on the total cost.

It is worth noting that we employed the well-known and commonly used Australia Post dataset (AP) for...
performing the experiments. This dataset consists of 200 data points, each describing a postal district, presented by the coordinates. Furthermore, it also provides the demand between each pair of points (matrix $W$).

Applying KGPMH in practice, particularly to the AP dataset, requires a proper parametrization of the model to determine all the parameters introduced in Section 3.2.1. For doing this, extensive experiments were performed, meaning that, in a repeatable procedure, we initialized the parameters based on the approach that will be explained below and checked the results in terms of the objective functions’ values. In the end, we picked a set of parameters that led us to obtain the best achievement. For the initial characterization, a uniform distribution is employed to feed the matrices $F_1$, $F_2$, $C_1$, $C_2$ for each of which the intervals $[200, 600]$, $[10, 15]$, $[10, 20]$ and $[20, 40]$ is considered, respectively. In addition, we initialize $\Delta = [2h, 2.5h, 3h]$, $\sigma = [24h, 24h, 24h]$, $\alpha = 0.2$, $\gamma = [0.7, 0.85, 1]$, and $I = [0.8m, m, 1.2m]$, where $m$ is average value of AP data. We also take three capacity and congestion levels into the account. In this fashion, $C = \{S, M, L\}$ symbolizes three levels of capacity, small, medium, and large, and besides, $G = \{1, 2, 3\}$ represents three congestion levels. We also let $\tau = 0.3$ and $\gamma_j = (1 + \tau)^{j-1}$ such that $g \in G$. According to the availability of the coordinates, we are able to obtain the Euclidean distance between each pair of data points. We define the traveling time between each two data points by: $t_{i,j} = t_{ij}$, where, $d_{ij}$ is the Euclidean distance between $i$ and $j$ and $t$ is a factor that depends on the numbers of data. It guarantees that the service time limit, which is considered around 24 hours for each period, is satisfied.

In addition, we consider the value of 5 for the parameter $\alpha$. Please be noted that from now on, we refer to OFV1 and OFV2 as the first and second objective function’s values, respectively.

We also point out that we have implemented our algorithm with C++ programming language and used a computer system that contained the processor Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz, 2712 Mhz, 2 Core(s), 4 Logical Processor(s) and 8 GB of RAM.

5.1 Estimating the VRCE

As our first attempt to assess our algorithm, this part intends to investigate how well the proposed evaluation technique, VRCE, can impact the results. To do so, we have solved the model with and without considering VRCE. To be more precise, in the case of not considering VRCE, we just used the domination definition and also an elementary function $g$ to evaluate solutions. Function $g$ is assumed in a way that for any given solution $s$, $f(s)$ returns the number of solutions that $s$ dominates. Since, in this case, neither a vector representation is being considered, nor a clustering algorithm is being applied, the genetic algorithm would have only relied on choosing parent solutions based on their ranks, obtained by the naive function $f$. Fig. 6a and Fig. 6b respectively illustrate the first and the second objective function’s values obtained when i) the proposed vector representation evaluation method is used (blue curve) and 2) when just the simple domination definition is applied (red curve). As can be seen, the results associated with the blue curves significantly outperform those presented by the red curve.

5.2 Comparison with other clustering-based techniques

As already mentioned, since the functionality of our algorithms heavily relies on the clustering approach, which is applied to the VRCE, we would have also liked to compare our results with those obtained by the two other clustering-based techniques: the CA-MOEA and CRMO. CA-MOEA’s idea is to replace the distance-based truncation style in NSGA-II with a hierarchical clustering technique (HCM) to cope with MOPs and MaOPs, respectively. This is while CRMO utilizes a series of predefined reference lines to bunch individuals into sub-regions so that each of the individuals nearest to the Pareto front will be chosen. Please note that these two algorithms can both perfectly be adapted to solve PMHP.

Therefore, same as what we did in the previous section, we again correspond two figures (Fig. 6a and Fig. 6b), which respectively present the first and the second objective functions’ values obtained by KGPMH (blue curve), CA-MOEA (green curve), and CRMO (gray curve). Observe that except in just one occasion regarding the second objective function, KGPMH results in obtaining less objective function values.

5.3 Comparison with NSGA-II and MOPSO algorithms

NSGA-II and MOPSO are two of the most reliable and widely-used algorithms for solving facility location problems; See Their importance, as well as their extensive use, encouraged us to see how our algorithm can perform against them. In this regard, let us first briefly explain these two heuristic algorithms. We then

\footnote{In this paper, we coined the term CRMO for the algorithm proposed in [12].}
assess their results against the ones obtained by our proposed technique.

NSGA-II is one of the most well-known algorithms for solving multi-objective problems proposed by Srinivas and Deb [60] based on adding two operators: non-dominated sorting and crowding distance to the single-objective GA. On the other hand, in MOPSO, which was developed by Coello et al. [17], a swarm is made of particles in which each solution acts as a particle. This algorithm works based on the group’s intelligence, which means every particle knows its prior best position and best position of the entire swarm. Each particle moves at a certain velocity in the feasible space of the problem to produce new solutions. Algorithm 2 and algorithm 3 respectively show the pseudocodes of NSGA-II and MOPSO algorithms. It is worth pointing out that both of these methodologies can be nicely generalized for applying to our problem, PMHM; See [11] and [39].

Table 2 provides a detailed comparison among KGPMH, NSGA-II, and MOPSO algorithms in terms of each objective function’s value and the time elapsed for executing the algorithm. As can be clearly observed, the KGPMH algorithm outperforms both NSGA-II and MOPSO algorithms in all cases in terms of detecting better solutions (lower objective functions’ values) as well as spending less running time. Please be also advised that

![Graph](image_url)

**Algorithm 2:** NSGA-II pseudocode.

```plaintext
1 Create a random initial population of size and calculate fitness of each person;
2 Apply non-dominated sorting on the population;
3 while iteration < desired number do
4 Generate crossover and mutation children and calculate fitness of each child;
5 Combine parents and children to make a union population;
6 Apply non-dominated sorting on the new population and select population size persons and remove the rest;
7 Return the best solutions;
```
optimization problems where the objective to be optimized can be expressed as a linear function or a convex quadratic function. For computing a Pareto optimal solution to mixed-integer programming, CPLEX uses a very general and robust algorithm based on branch and cut, equipped with the node heuristic that employs techniques to construct a feasible solution from the current (fractional) branch-and-cut node.

Hence, due to the promising efficiency of this solver, we decided to evaluate our results against the ones obtained by it. However, because the CPLEX optimizer applies the branch and cut method, it usually has difficulty solving large-scale problems due to the very high execution time. Therefore, we had to pick a small scale of PMHM so that CPLEX is able to handle it. In consequence, we considered the AP dataset with 100 data points. Contemplating this number of nodes enables us to obtain the Pareto optimal solution using CPLEX solver, version 12.9.0.

Table 3 presents the results obtained by applying KGPMH in this experiment is equal to 15.

Table 1: MOPSO, NSGA-II and KGPMH parameters.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOPSO</td>
<td>Repository size</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Inertia weight</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>Personal learning coefficient</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>Global learning coefficient</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Iteration</td>
<td>70</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>Crossover coefficient</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>Mutation coefficient</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Iteration</td>
<td>70</td>
</tr>
<tr>
<td>KGPMH</td>
<td>Learning rate</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Time constant</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Number of iteration</td>
<td>70</td>
</tr>
</tbody>
</table>

5.4 Comparison with (near) optimal Pareto solutions discovered by CPLEX solver

Although measuring the quality of our algorithm against two other triumphant and famous heuristic approaches raised promising results, it would still be satisfying to see how it performs versus techniques that rely more on analytic approaches, which are likely expected to perform more accurately. One of the fundamental theoretical perspectives to follow is the branch and cut manner, which is a combinatorial optimization method for solving (mixed) integer linear programs \(^{(55-56)}\). Roughly speaking, branch and cut concerns running a branch and bound algorithm \(^{(10)}\) and utilizing cutting planes to tighten the linear programming relaxations. We refer the interested reader to \(^{(41,49)}\) for grasping a detailed vision of this approach. In addition, it can be noted that one of the best ways to solve mixed-integer problems based on this technique is using a potent solver named CPLEX ILOG, designed and commercialized by IBM. CPLEX presents libraries for different programming languages such as C, C++, Java, .NET, and Python, that solve linear programming (LP) and corresponding problems. To be more specific, it is capable of solving linearly or quadratically constrained optimization problems where the objective to be optimized can be expressed as a linear function or a convex quadratic function. For computing a Pareto optimal solution to mixed-integer programming, CPLEX uses a very general and robust algorithm based on branch and cut, equipped with the node heuristic that employs techniques to construct a feasible solution from the current (fractional) branch-and-cut node.

In this section, we are going to examine how allowing hub mobility can affect the total cost and, in particular, helps decrease it. For doing this, we turn our focus toward the most simplistic way for exploring this issue: performing a sensitivity analyses over different multiple of the parameters \(C_2\) and \(F_2\) and compare the computed total cost with the one obtained when mobility condition is ignored (i.e., parameters \(F_2\) and \(C_2\) and so variables \(Z_2\) and \(R\) are omitted form the model). This procedure helps us better understand how increasing the cost of constructing routes between hubs

and bound algorithm \(^{(10)}\) and utilizing cutting planes to tighten the linear programming relaxations. We refer the interested reader to \(^{(41,49)}\) for grasping a detailed vision of this approach. In addition, it can be noted that one of the best ways to solve mixed-integer problems based on this technique is using a potent solver named CPLEX ILOG, designed and commercialized by IBM. CPLEX presents libraries for different programming languages such as C, C++, Java, .NET, and Python, that solve linear programming (LP) and corresponding problems. To be more specific, it is capable of solving linearly or quadratically constrained optimization problems where the objective to be optimized can be expressed as a linear function or a convex quadratic function. For computing a Pareto optimal solution to mixed-integer programming, CPLEX uses a very general and robust algorithm based on branch and cut, equipped with the node heuristic that employs techniques to construct a feasible solution from the current (fractional) branch-and-cut node.

In this section, we are going to examine how allowing hub mobility can affect the total cost and, in particular, helps decrease it. For doing this, we turn our focus toward the most simplistic way for exploring this issue: performing a sensitivity analyses over different multiple of the parameters \(C_2\) and \(F_2\) and compare the computed total cost with the one obtained when mobility condition is ignored (i.e., parameters \(F_2\) and \(C_2\) and so variables \(Z_2\) and \(R\) are omitted form the model). This procedure helps us better understand how increasing the cost of constructing routes between hubs

5.5 Effect of considering the mobility for hubs on total cost

In this section, we are going to examine how allowing hub mobility can affect the total cost and, in particular, helps decrease it. For doing this, we turn our focus toward the most simplistic way for exploring this issue: performing a sensitivity analyses over different multiple of the parameters \(C_2\) and \(F_2\) and compare the computed total cost with the one obtained when mobility condition is ignored (i.e., parameters \(F_2\) and \(C_2\) and so variables \(Z_2\) and \(R\) are omitted form the model). This procedure helps us better understand how increasing the cost of constructing routes between hubs

Table 3 presents the results obtained by applying KGPMH in this experiment is equal to 15.

Table 1: MOPSO, NSGA-II and KGPMH parameters.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOPSO</td>
<td>Repository size</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Inertia weight</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>Personal learning coefficient</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>Global learning coefficient</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Iteration</td>
<td>70</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>Crossover coefficient</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>Mutation coefficient</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Iteration</td>
<td>70</td>
</tr>
<tr>
<td>KGPMH</td>
<td>Learning rate</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Time constant</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Number of iteration</td>
<td>70</td>
</tr>
</tbody>
</table>
The parameter service time is directly related to the variable ready time, which is basically the time spent at hubs for processing the flows. On the other hand, ready time considerably impacts the total cost because, intuitively, it is natural to think that forcing hubs to process the flows and transportation faster requires spending more money. Consequently, service time can severely affect the total cost. Table 4 and Fig. 9 clearly show this issue. As can be seen, once service time starts to increase, the total cost triggers to decrease. However, more interestingly, reducing the service time from a certain limit no longer has an impact on the total cost.

Therefore, one can think of considering enough large value for service time can make us decrease the costs. Nevertheless, the critical argument, on the other hand, is that in the real world, there is usually a time limit for sending each shipment on each transit network, and (C2) and the cost of transferring mobile hubs (F2) increase the total cost to the extent that it gets equal to the value obtained when mobility is not considered at all. To gain an initial insight, consider 20 data points of the AP dataset and assumed that \( p = 5 \). Now, let us apply the KGPMH algorithm to the problem. Point out that the final obtained value of the first and second objective functions are respectively 5972, 47.49. Fig. 7 illustrates the network scheme derived from the final solution. Now, if the hubs’ mobility is ignored, the total cost will be 5592, leading us to conclude that taking mobility into consideration can significantly help us decrease the total costs. What is really important to note is that, in real world, the cost of constructing routes between hubs (C2) and the cost of transferring hubs through these routes (F2) are usually much lower than the cost of constructing new hubs (F1).

To grasp a more detailed understanding, we draw your attention to Fig. 8 that illustrates the results corresponding to the implemented sensitivity analysis. It express how much change will occur in total cost value provided that different real value coefficient of C2 and F2 are considered in the case that hubs have mobility. Paying attention to the results, one can note that what is really promising and crucial regarding the model is that whenever the cost of constructing routes between hubs (C2) and the cost of transferring a mobile hub (F2) are both lower than the cost of establishing a hub, the proposed model decides (tends) to move the hubs, while when the cost of establishing route and the cost of transferring a mobile hub are sufficiently high, the proposed model acts as a non-mobile model and establishes a hub instead of relocating hubs.

5.6 The impact of considering service time

Recall that the parameter service time is directly related to the variable ready time, which is basically the time spent at hubs for processing the flows. On the other hand, ready time considerably impacts the total cost because, intuitively, it is natural to think that forcing hubs to process the flows and transportation faster requires spending more money. Consequently, service time can severely affect the total cost. Table 4 and Fig. 9 clearly show this issue. As can be seen, once service time starts to increase, the total cost triggers to decrease. However, more interestingly, reducing the service time from a certain limit no longer has an impact on the total cost.

Therefore, one can think of considering enough large value for service time can make us decrease the costs. Nevertheless, the critical argument, on the other hand, is that in the real world, there is usually a time limit for sending each shipment on each transit network, and
Fig. 7: (1), (2) and (3) show the network scheme of the AP dataset with considering 20 data points, in period 1, 2 and 3, respectively. Red nodes are hubs.

Fig. 8: Comparison of the network total cost caused by various coefficients of $F_2$ and $C_2$.

Therefore, we must consider (obtain) a reasonable service time to feed into the model. As a result, the considerably crucial point is to make a balance, or more precisely, determine an appropriate value for the service time at the very beginning. In other words, it is required to provide a proper service time that is feasible and maintain a transit network's efficiency.

Table 4: Comparison of total costs with and without the service time on five different problem instances.

<table>
<thead>
<tr>
<th># nodes</th>
<th>PMHM with service time</th>
<th>PMHM without service time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OFV1</td>
<td>CT (s)</td>
</tr>
<tr>
<td>10</td>
<td>3763</td>
<td>5.3</td>
</tr>
<tr>
<td>15</td>
<td>4912</td>
<td>7.9</td>
</tr>
<tr>
<td>20</td>
<td>6982</td>
<td>16.5</td>
</tr>
<tr>
<td>25</td>
<td>8212</td>
<td>27.9</td>
</tr>
<tr>
<td>40</td>
<td>18388</td>
<td>155.03</td>
</tr>
</tbody>
</table>

Furthermore, we uncovered a direct relationship between the effect of considering service time and the size of the problem, meaning that for a larger-scale problem, we encounter a more intense change in the total cost rather than when the service time is not considered. Fig. 10 proves this assertion.

We conclude this section by noting that our proposed method achieved promising solutions in terms of objective functions' values and computational time, compared to the ones obtained by two state-of-the-art heuristic approaches, CPLEX solver, and two other clustering approaches. On the other hand, taking the mobility of hubs into account allowed us to decrease the total cost wisely. Furthermore, the effect of service time on the model and, therefore, on the total cost (the value of the first objective function) was investigated.
A heuristic approach for solving a new model of facility location problem

6 Conclusion

This paper introduced a multi-period single-allocation $p$-mobile hub location problem with a service time limit. There are many real-world cases, such as telecommunication, chain stores in the supply chain, systems, massage delivery systems, and post-delivery, which this problem can model. In the proposed model, hubs were equipped with specific levels of capacity and congestion. The aim was to design a transportation network for each period to locate $p$ hub vertices for satisfying all the demands with respect to minimizing the total cost, hubs’ congestion, and ready time at hubs. In this fashion, the total cost can be divided into transportation cost, construction of mobile infrastructure, establishing of hubs, and transferring mobile hubs. On the other hand, service time refers to the sum of time spent gathering all the flows into the hub and the time spent at that hub to process the flows. Moreover, ready time in each hub is defined as the last arrival time of all flows from non-hub nodes to that hub vertex.

Besides modeling the problem, we also proposed a hybrid meta-heuristic method for solving the problem. We devised a novel function for evaluating solutions and provided a vector representation for each solution based on that. We then employed the K-means clustering algorithm and a genetic approach to evaluate and generate solutions. We finally measured the quality of our algorithm against several other state-of-the-art facility location algorithms. The results verified that our proposed method is a promising alternative for all the comparable algorithms. We also discussed the impress of considering hubs’ mobility and selecting the right service time to achieve a reasonable total cost and maintain the maximum quality and speed of meeting demands.

Future work, on the other side, can be devoted to the question of whether our method is robust enough for other multi-objective optimization instances. In this fashion, one direction to follow is to generalize the approach so that it can be fed with any multi-objective optimization problems. Another avenue to pursue is applying other clustering algorithms, such as using the hierarchical clustering approach, particularly employing techniques that can automatically detect the number of clusters. In addition, we will look out for alternative
evaluation functions to seek more accurate solutions. Even more, applying several methods based on sophisticated optimization techniques is recommended.

7 Conflict of interest

The authors declare that there is no any potential conflict of interest regarding the publication of this paper.

8 Ethical approval

The authors certify that they have no any affiliation with or involvement with human participants or animals performed by any of the authors in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this paper.

References

50. Mohitkarzadeh, M., Tavakkoli-Moghaddam, R., Tríki, C., Rahimi, Y.: A hybrid of clustering and meta-heuristic algorithms to solve a p-mobile hub location-allocation problem with the depreciation cost of hub facilities. Engineering Applications of Artificial Intelligence 98, 104121 (2021)


