European Journal of Operational Research 314 (2024) 79–93

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

# Efficient solution approaches for the bi-criteria $p$-hub median and dispersion problem

Prasanna Ramamoorthy [a,*], Navneet Vidyarthi [b], Manish Verma [c]

[a] Department of Management Studies, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, India
[b] John Molson School of Business and CIRRELT, Concordia University, Montreal, QC, H3G 1M8, Canada
[c] Degroote School of Business, McMaster University, Hamilton, Ontario, L8S 4M4, Canada

ARTICLE INFO

*Keywords:*
Location
$p$-hub median problem
Hub dispersion
Cutting plane method
Benders decomposition

ABSTRACT

In this paper, we study the bi-criteria $p$-hub median and dispersion problem, that arises in the design of hub networks where the dispersion of hubs is desired to mitigate the risk of disruptions. The problem is formulated as a bi-objective mixed integer program, where the first objective is to minimize the total cost of routing the flows through $p$ hubs and the second objective is to maximize the minimum distance (or dispersion) among the selected $p$ hub locations themselves. We present two exact solution approaches that guaranteed to obtain the entire non-dominated Pareto frontier. The first is a cutting plane method in which a $p$-hub median problem with a particular dispersion distance is solved at each iteration. Three formulations of the problem, based on the different type of cuts and preprocessing, are presented. We study the dominance relationship among the three formulations. Through computational experiments, we show that the proposed cutting plane method is efficient in solving medium size instances of the problem and our strongest formulation is at least 40% computationally faster than the others. For solving large instances of the problem, we present a decomposition method where the $p$-hub median problem with dispersion distance is solved using an accelerated Benders decomposition approach. We present several problem specific enhancements to the algorithm such as starting with a better solution, efficient ways of solving decomposed subproblem and adding Pareto optimal Benders cuts to the master problem. The computational results on the Turkish network (TR81), US423, and Australian Post (AP) dataset show that the cutting plane algorithm with the proposed decomposition procedure is three to four times faster than the commercial solver.

## 1. Introduction

Hub networks have wide applications ranging from telecommunications, airline transportation, less-than-truckload freight transportation, rail freight transportation, supply chains to power distribution systems. These networks use transshipment, consolidation, or sorting points for commodities, called hub facilities, to connect a large number of origin–destination (O–D) pairs using a small number of links. Commodities with same origin but different destinations are consolidated at the hubs and are then combined with other commodities having different origins but the same destination. The use of hub facilities helps centralize commodity handling and sorting operations, reduce set-up costs, and achieve economies of scale on routing costs through the consolidation of flows. Another advantage is that a hub network typically contains fewer links than a network with direct connection between all nodes. This is particularly useful in applications like telecommunication and energy distribution, where link construction cost is significant. For details on hub networks, the readers are referred to the review papers by Alumur and Kara (2008), Campbell and O'Kelly (2012), Farahani et al. (2013), Contreras (2015), Alumur et al. (2021), and Contreras (2021).

Hub networks have been well studied in the literature since the seminal papers by O'Kelly (1986a, 1986b, 1987). Campbell (1992) classified the problem based on different objectives and presented the first mathematical formulation for the variants of the problem: $p$-hub median, $p$-hub maximal covering, and $p$-hub center problems. The corresponding objectives of these problems are: (i) minimize the cost of satisfying the demand, (ii) maximize the demand covered, and (iii) minimize the maximum cost between any O–D pair. These problems are further classified based on the assignment of non-hub nodes to hub nodes as either single allocation or multiple allocation. In single allocation, a non-hub node is allocated to only one hub whereas in multiple allocation, a non-hub node can be allocated to more than one hub. These problems can also be classified into uncapacitated and capacitated hub location problems.

* Corresponding author.
*E-mail addresses:* prasanna.r@dms.iitd.ac.in (P. Ramamoorthy), n.vidyarthi@concordia.ca (N. Vidyarthi), mverma@mcmaster.ca (M. Verma).

https://doi.org/10.1016/j.ejor.2023.09.032
Received 10 March 2022; Accepted 25 September 2023
Available online 27 September 2023

Hub networks typically have asymmetric degree distribution among its nodes with hub nodes having the most and spoke nodes having the least degree. This asymmetric degree distribution ensures that the impact of a disruption on a hub node in the network is far greater than the impact of a similar disruption at a spoke node. Researchers have classified these disruptions into two types: (i) intentional disruptions, where the disruptor is a rational adversary, e.g., a terrorist whose objective is to cause the maximum disruption in the network, and (ii) natural disruptions, e.g. events like tsunami, earthquake etc., where the maximal damage cannot be ascribed to. For the former case, researchers have taken a multi-level approach motivated by Stackelberg games to address these problems (Ghaffarinasab & Atayi, 2017; Ghaffarinasab & Motallebzadeh, 2017; Lei, 2013; Ramamoorthy et al., 2018). The main classes of problems studied in this case is the (i) hub interdiction problem (Ghaffarinasab & Atayi, 2017; Ghaffarinasab & Motallebzadeh, 2017; Lei, 2013; Ramamoorthy et al., 2018), (ii) hub protection problem (Lei, 2013; Ramamoorthy et al., 2018), and (iii) hub network design problem under the risk of interdiction (Parvaresh et al., 2013, 2014). In the hub interdiction problem, the objective is to identify critical hubs in a hub network. This problem is typically modeled as a bilevel mixed-integer program where in the first level an attacker (leader) identifies hubs to interdict so as to maximize the minimum post-interdiction routing cost of the defender (follower) in the second level. In the second level, the defender satisfies the demand at the minimum cost by routing the flows through the surviving hubs. Hub protection problem is an extension of the hub interdiction problem where the defender is the leader who protects a set of hubs so as to minimize the maximum hub interdiction cost of the attacker (follower) at the second level. The second and third level problems in the hub protection problem are the same as that of the hub interdiction problem. In a hub network design problem under the risk of interdiction, at the first level, the defender locates hubs so as to minimize hub location (pre-interdiction hub location and routing) and hub interdiction cost. The second and third level problem constitute the hub interdiction problem.

To protect the operations of a hub network in the event of any disruptions (natural or intentional), researchers have studied network design problems that consider improving the reliability of the overall network. Kim and O'Kelly (2009) studied the reliable $p$-hub location problems in telecommunication networks, where the objective is to maximize the reliability of the hub-and-spoke network through locating $p$ hubs. The authors studied several variants addressed to improve the reliability of the hub network in both single and multiple allocation settings. To improve reliability, the authors also studied maximizing dispersion between the located hubs. An et al. (2015) studied the reliable hub location problem where each hub is assigned a probability of disruption. The model improves reliability of the network by assigning backup hubs and alternative routes in case of disruption in any of the hubs in the hub network. Azizi et al. (2016) studied the impact of hub failures on the design of hub network and proposed the use of backup hubs. The objective of the problem is to minimize the weighted sum of transportation cost before hub failure and expected transportation cost post hub failure. The authors also develop an evolutionary algorithm to solve large instances of the problem.

Another approach to minimize the impact of disruptions in hub networks is to disperse the location of hub facilities. For instance, hub locations in the earliest designs of ARPANet were geographically dispersed to protect against probable attacks or failures. O'Kelly and Kim (2007) showed the potential advantage of dispersed hubs in improving network reliability while analyzing the cascading failure of internet services in Korea. In their study, the authors show that a failure of just a single hub can cause extensive damage to the network when hub facilities are highly clustered geographically. Kim and O'Kelly (2009) cite another example of the disruption of internet service due to the 2006 earthquake in Taiwan to highlight the importance of dispersing components to maintain the reliability of telecommunication systems. Note that such hub network where hubs are dispersed can result in high cost of operations since the facilities are now located far away from the demand centers.

Another related class of problems are *facility dispersion problems*, where the objective is to locate dispersed facilities. In one of early papers in this area, Kuby (1987) presented several integer programming models for $p$-dispersion problems where $p$ facilities are to be located such that the minimum distance between all the facilities is maximized. In addition to that, a maxisum problem is studied in which the objective is to maximize the distance between all located facilities. Finally, the author presents a multi-objective version combining both objectives. Erkut (1990) presented an alternate formulation for the $p$-dispersion problem and a branch and bound based solution algorithm. Erkut et al. (1994) presented several heuristics to solve the $p$-dispersion problem. Pisinger (2006) developed several upper bounding techniques and exact algorithms for maxisum dispersion problem. Recently, Sayyady et al. (2015) presented a pure integer program for the $p$-dispersion problem. Sayah and Irnich (2017) presented a new compact formulation for the $p$-dispersion problem. The authors also present some preprocessing techniques and valid inequalities to improve the computational performance of the formulation. Landete et al. (2023) studied the capacitated dispersion problem in which capacity restrictions are imposed on the amount of demand that can be handled by the facilities. The authors present several formulations of the problem in different spaces using variables associated with nodes, edges, and costs. These formulations are further strengthened using valid inequalities and variable fixing procedures.

Although the dispersion of facilities is preferred from a reliability perspective, decision makers are also interested in minimizing cost, maximizing coverage or service to achieve better operational performance. Accordingly, decision makers have taken a bi-criteria approach where one objective is to improve reliability while minimizing total operational cost. Sayyady and Fathi (2016) studied a bi-criteria $p$-median facility dispersion problem, where the first objective is to minimize the cost of operations and the second one is to maximize the dispersion between located facilities. Similarly, Tutunchi and Fathi (2019) studied a bi-criteria $p$-center facility dispersion problem in which the objectives are to minimize the maximum distance between facilities and maximize the minimum distance between facilities. The authors present exact solution algorithms with valid inequalities.

In the context of hub location, the literature on hub dispersion is scarce. To the best of our knowledge, Kim and O'Kelly (2009) is the only paper to study hub dispersion. The authors presented the *$p$-hub mandatory dispersion problem*, which determines the optimal hub locations to improve network reliability while retaining the mandatory dispersion of hubs requiring them to be farther apart than a certain minimum separation. The goal of the proposed model is to avoid concentrating strategic assets (hubs) in one area to reduce the growing vulnerability that may arise from congestion or disruptions in a network. They also proposed a bi-objective hub location model to study the trade-off between the maximization of network reliability and the maximization of geographical separation among hubs.

Table 1 provides a summary of literature on bi-objective hub location problems. Bi-objective hub location problems have been studied by Costa et al. (2008), Fallah-Tafti et al. (2022), Ghaffarinasab (2020), Khodemani-Yazdi et al. (2019), Kim and O'Kelly (2009), Köksalan and Soylu (2010), Madani et al. (2018), Mohammadi et al. (2016), Monemi et al. (2021), Shang et al. (2021) among others. In most of these papers, the first objective is to minimize the demand weighted transportation cost (and fixed cost) whereas the second objective is to minimize service time (at hubs), travel/delivery time (origin–destination), route length, or congestion (at hubs). In this paper, we introduce a new class of hub location problems aimed at improving the reliability of the network by dispersing hub facilities while routing flows from origins to destinations via hubs economically. The problem is termed as *bi-criteria or bi-objective (uncapacitated, multiple allocation) $p$-hub median and dispersion problem* (B$p$-HMDP). The first objective is to minimize the demand weighted transportation cost of flows through the hubs (median

**Table 1**
Summary of literature on bi-objective hub location problems.

| Reference | Objective 1 | Objective 2 | Solution method |
|---|---|---|---|
| Costa et al. (2008) | Min. trans. cost | Min. service time | Iterative algorithm |
| Kim and O'Kelly (2009) | Max. network reliability | Max. hub dispersion | Branch and Bound |
| Köksalan and Soylu (2010) | Min. trans. cost | Min. traveling cost | Evolutionary algorithm |
| Mohammadi et al. (2016) | Min. trans. + fixed cost | Min.-max. travel time | VNS, Metaheuristic |
| Madani et al. (2018) | Min. flow covered | Min. congestion | Non-dominated sorting GA-II |
| Khodemani-Yazdi et al. (2019) | Min. trans. + fixed cost | Min.-max. route length | VNS |
| Ghaffarinasab (2020) | Min. trans. cost | Max. route length | Tabu search algorithm |
| Monemi et al. (2021) | Min. trans. cost | Max. flow balancing | Tabu search algorithm |
| Shang et al. (2021) | Min. trans. cost | Min. delivery time | VNS, NSGA-II |
| Fallah-Tafti et al. (2022) | Max. profit | Min. travel time | Adaptive LNS, Metaheuristic |
| This paper | Min. trans. cost | Max–min. hub dispersion | Cutting plane algorithm, Decomposition method |

**Table 2**
Pareto optimal solutions of an instance of CAB dataset, 25 nodes and 5 hubs.

| Solution no. | Hubs located | Median Obj. (MO) | Disper. Obj. (DO) | % diff MO | %diff DO |
|---|---|---|---|---|---|
| 1 | 3, 6, 11, 13, 16 | 634,659 | 720.47 | – | – |
| 2 | 7, 11, 13, 16, 20 | 682,992 | 780.95 | 7.6 | 8.4 |
| 3 | 11, 13, 16, 20, 22 | 688,891 | 880.55 | 0.9 | 12.8 |
| 4 | 10, 11, 13, 17, 22 | 719,537 | 986.82 | 4.4 | 12.1 |
| 5 | 10, 13, 17, 18, 22 | 750,588 | 1021.61 | 4.3 | 3.5 |
| 6 | 10, 13, 16, 18, 22 | 752,081 | 1048.54 | 0.2 | 2.6 |
| 7 | 2, 14, 18, 22, 23 | 903,029 | 1124.78 | 20.1 | 7.3 |

objective), whereas the second objective is to maximize the minimum dispersion between the hub facilities while locating $p$ uncapacitated hubs and allocating the non-hub nodes to multiple hubs.

Although multi-objective problems are studied in facility and hub location literature, (see Fallah-Tafti et al. 2022, Farahani et al. 2010, Ghaffarinasab 2020 and references therein), this paper is amongst the few to study bi-objective hub location problem with conflicting objectives that maximize reliability through the dispersion of hubs while minimizing the operational cost. The bi-objective model generates a Pareto frontier of optimal solutions that provides the decision maker with multiple alternatives based on the desired level of reliability and cost, which differs from other reliability based models that have predominantly single objective function. Solution methods for bi-objective hub location problems are predominantly based on heuristic and metaheuristic approaches such as evolutionary algorithms (EA), non-dominated sorting genetic algorithm — II (NSGA-II), variable neighborhood search (VNS), large neighborhood search (LNS), and tabu search among others.

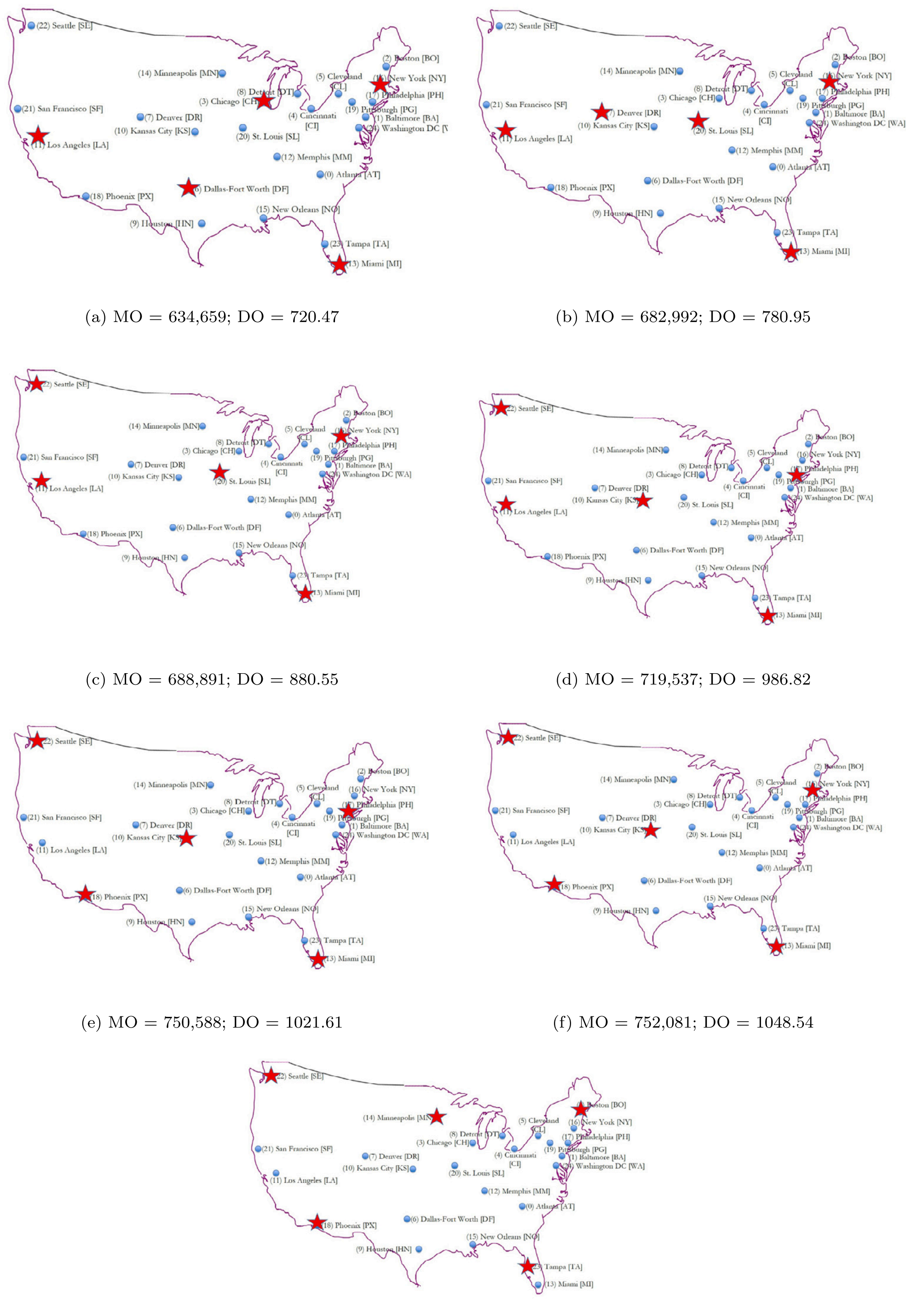
To illustrate the trade-offs between the total transportation cost and the hub dispersion, consider the Civil Aeronautics Board (CAB)[1] dataset with 25 nodes. The optimal solution of the $p$-hub median problem with 5 hubs is shown in Fig. 1(a); this solution entails a transportation cost of \$634,659 and a minimum dispersion distance of 720.47 units (refer to Solution no. 1 in Table 2). On the other hand, the optimal solution of the max–min $p$-hub dispersion problem with 5 hubs is shown in Fig. 1(g); this solution entails a transportation cost of \$903,029 and a minimum dispersion distance of 1124.78 units (refer to Solution no. 7 in Table 2). As it can be seen, the optimal hub locations with the minimal median objective (e.g., 3, 6, 11, 13, and 16) are completely different from the one with max–min dispersion (e.g., 2, 14, 18, 22, and 23). There are seven solutions in the optimal Pareto frontier in total. The hub locations for each of these solutions are shown in Figs. 1(a) to 1(g) and listed in Table 2. In Table 2, MO denotes the median objective function value and DO denotes the dispersion objective function value of the corresponding solution. The columns "% diff MO" and "% diff DO" present the percentage difference in median and dispersion objective function values of the current and the previous solution in the Pareto frontier. In at least two solutions in the Pareto frontier (#3 and #6), with a marginal loss (increase) in median objective, there can be substantial gain in dispersion objective. For example, a decision maker comparing solutions 2 and 3 would prefer solution 3 over 2, as the former provides better dispersion for almost the same cost as the latter. Therefore, the bi-objective model generates a Pareto frontier of optimal solutions and provides the decision maker with multiple options based on the desired level of cost and dispersion.

### 1.1. Contribution

The contribution of this paper is two-fold. First, motivated by the importance of designing a resilient and reliable hub networks, this is one of the few papers on bi-objective hub location problems to address (max–min) hub dispersion along with the minimization of transportation cost. Secondly, hub location problems are a difficult class of NP-hard combinatorial optimization problems. Solution methods for bi-objective hub location problems are predominantly based on heuristic and metaheuristic approaches (refer to Table 1). To the best of our knowledge, this is the first paper to present exact solution approaches for the bi-objective hub location problem. We present an iterative cutting plane algorithm that is guaranteed to generate the entire *non-dominated (i.e., all efficient) Pareto solutions.* At each iteration of the algorithm, a $p$-hub median problem with dispersion constraint ($p$-HMPD$_d$) is solved. We exploit the structure of the problem to generate different cutting planes and preprocessing procedures, resulting in three different formulations of $p$-HMPD$_d$. We study the dominance relationship among the three formulations to identify the tightest one. Through computational results, we validate our theoretical findings, and show that the proposed cutting plane method is very effective for solving the moderate instances of the problem. For solving large instances of the problem, we present a Benders decomposition based cutting plane method. The decomposition procedure is enhanced with several improvements such as starting with a better solution and efficient ways of solving decomposed subproblem and adding Pareto optimal Benders cuts to the master problem. We present extensive computational results to show the efficiency of our methods.

The remainder of the paper is organized as follows. In Section 2, we present the mixed-integer programming model of the bi-criteria $p$-hub median and dispersion problem. In Section 3, we present the cutting

[1] Available at the OR library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html).



(a) MO = 634,659; DO = 720.47

(b) MO = 682,992; DO = 780.95

(c) MO = 688,891; DO = 880.55

(d) MO = 719,537; DO = 986.82

(e) MO = 750,588; DO = 1021.61

(f) MO = 752,081; DO = 1048.54

(g) MO = 903,029; DO = 1124.78

**Fig. 1.** Hub locations for the Pareto optimal solution of CAB dataset, 25 nodes and 5 hubs.

plane algorithm. We also present three reformulations of B$p$-HMDP as a single objective $p$-hub median problem with dispersion distance constraints. These reformulations are amenable to cutting plane algorithm. In Section 4, we present theoretical comparisons of the three reformulations to identify their strengths. In Section 5, we present an alternate solution approach using an accelerated Benders decomposition procedure with several improvements to solve large instances of the problem. In Section 6, we present extensive computational results using the proposed solution approaches. Conclusions follow in Section 7.

## 2. Model formulation

Let $N$ be the set of all nodes representing the origins and destinations of the commodities. Also, let $H \subseteq N$ be a set of candidate hubs, and $p$ denote the number of hubs to be located. Let $W_{ij}$ denote the amount of flow from origin node $i \in N$ to destination node $j \in N$ through one or at most two of the hubs from the set $H$. We use $k$ and $m$ as indices to denote a hub which is connected to origin nodes, $k \in H$ and destination nodes, $m \in H$ respectively. Let $d_{ijkm}$ represent the per unit transportation cost from origin $i$ to destination $j$, through hubs $k$ and $m$, in that order. Then, $d_{ijkm} = \alpha d_{ik} + \delta d_{km} + \gamma d_{mj}$, where $\alpha$, $\delta$, and $\gamma$ are non-negative discount factors on collection, transshipment, and distribution links, respectively and $d_{ik}$, $d_{km}$, and $d_{mj}$ represent the cost of traversing from node $i$ to $k$, $k$ to $m$, and $m$ to $j$, respectively. Typically, $\delta < \alpha$ and $\delta < \gamma$ due to economies of scale arising from consolidation of flows on inter-hub links. The problem is to decide the optimal location of hubs and path(s) between all origin and destination pairs $(i, j)$ such that every path traverses one or more hubs to benefit from the consolidation at hubs. The objectives are (1) to minimize the total demand weighted transportation cost for all origin–destination pairs $(i, j)$ through located hub and (2) to maximize the minimum distance between hubs. We use the binary decision variable $y_k = 1$ to represent if a hub is located at node $k$; 0 otherwise. Let the variable $X_{ijkm} \geq 0$ represent the fraction of total traffic from node $i$ to node $j$ routed via hubs located at nodes $k$ and $m$, in that order. The decision variable pertaining to the max–min dispersion objective is defined as follows: let $D$ be the minimum distance between any pair of open hubs; $D = \{\min\{d_{km}\} \mid \forall\, k, m \in H, k \neq m, \forall\ y_k = 1, y_m = 1\}$. With these notations, we build on one of the strongest known four-index formulations of the uncapacitated multiple allocation hub location problem (UMAHLP), proposed by Hamacher et al. (2004).

The mixed-integer linear formulation of the bi-criteria $p$-hub median and dispersion problem is stated as follows:

$$[\text{B}p\text{-HMDP}]: \min_X \sum_{i\in N}\sum_{j\in N}\sum_{k\in H}\sum_{m\in H} W_{ij} d_{ijkm} X_{ijkm} \tag{1}$$

$$\max_y \quad D \tag{2}$$

$$\text{s.t.} \sum_{k\in H} y_k = p \tag{3}$$

$$\sum_{k\in H}\sum_{m\in H} X_{ijkm} = 1 \qquad \forall\, i, j \in N \tag{4}$$

$$\sum_{m\in H} X_{ijkm} + \sum_{m\in H, m\neq k} X_{ijmk} \leq y_k \qquad \forall i, j \in N; k \in H \tag{5}$$

$$D \leq d_{km} + M(2 - y_k - y_m) \qquad \forall k, m \in H \tag{6}$$

$$X_{ijkm} \geq 0 \qquad \forall i, j \in N; k, m \in H \tag{7}$$

$$y_k \in \{0, 1\} \qquad \forall k \in H \tag{8}$$

where $M$ is a very large number and can be set to $\max_{k,m\in H}\{d_{km}\}$.

The objective function (1) minimizes the demand weighted transportation cost for all O–D pairs $(i, j)$ through located hub whereas the objective function (2) maximizes the minimum distance between located hubs. Constraint (3) ensures that $p$ hubs are located. Constraint set (4) states that demand flow between origin $i$ and destination $j$ should be routed completely through the located hubs. Constraint set (5) states that demand can be routed only through a located hub. Constraint set (6) is the (linear) dispersion constraint which assigns the minimum dispersion between located hubs to the variable $D$. Constraint set (6) was proposed by Kuby (1987) to maximize the minimum dispersion between facilities. Constraints (7) and (8) are the standard non-negativity and integrality restrictions.

B$p$-HMDP is NP-hard as the problem without the second objective is the classical multiple allocation $p$-hub median problem (MA$p$HMP), which is known to be NP-hard. B$p$-HMDP contains solutions that lie on a Pareto frontier. The predominant concept in defining an optimal point is that of Pareto non-dominated points or efficient points, which are in turn either strictly efficient or weakly efficient. For the sake of completeness, their definitions are given below:

Let $\mathbf{Y}$ be the feasible solution set for location decision vectors to the problem MA$p$HMP. By definition, $\mathbf{Y} = \{Y^1, Y^2, Y^3, \dots, Y^{\binom{|N|}{p}}\}$. We define $f^1(Y^x)$ and $f^2(Y^x)$ be the objective function values of median and dispersion objective respectively.

**Definition 1.** Strictly Efficient Solutions: Solution, $Y^b \in \mathbf{Y}$ is a Strictly efficient or Strictly non-dominating Pareto solution if there does not exist any other solution $Y^{b_1} \in \mathbf{Y}$ such that $f^1(Y^{b_1}) \leq f^1(Y^b)$ and $f^2(Y^{b_1}) \geq f^2(Y^b)$.

**Definition 2.** Weakly Efficient Solutions: Solution, $Y^b \in \mathbf{Y}$ is a Weakly efficient or Weakly non-dominating Pareto solution if there does not exist any other solution $Y^{b_1} \in \mathbf{Y}$ such that $f^1(Y^{b_1}) < f^1(Y^b)$ and $f^2(Y^{b_1}) > f^2(Y^b)$.

The set of all non-dominating Pareto solutions constitute the optimal Pareto frontier. In the following sections, we present a cutting plane method and a decomposition approach to generate Pareto frontier for B$p$-HMDP.

## 3. Cutting plane method

We present an exact method for solving B$p$-HMDP. The method is based on the observation that on a Pareto optimal point corresponding to minimal dispersion distance $d$ and hub locations $Y_d$, the next immediate unvisited non-dominating Pareto optimal solution will contain hubs that have minimal dispersion distance strictly greater than $d$ amongst them. The above condition can be implemented by adding a cut that removes such hub pairs with dispersion distance lesser than or equal to $d$. The cutting plane method is outlined in Algorithm 1. At each iteration of the algorithm, we solve a uncapacitated, multiple allocation, $p$-median hub location problem with additional set of cuts known as dispersion cuts (DCs). These cuts separate those hub pairs that have dispersion distance lesser than or equal to the minimal dispersion distance obtained from the previous iteration. The algorithm terminates when no such $p$ hubs are left to locate in the solution space of the hub location problem. Note that at every iteration, we also reduce the solution space of the hub location problem. We present three formulations based on different DCs in Sections Section 3.1, 3.2, and 3.3 respectively.

Let $\text{DC}_d$ denote the set of dispersion constraints corresponding to the dispersion distance $d$. The $p$-hub median location problem with dispersion constraints $\text{DC}_d$ for a given dispersion distance $d$ is as follows:

$$[p\text{-HMPD}_d]: \min_{y,X} \sum_{i\in N}\sum_{j\in N}\sum_{k\in H}\sum_{m\in H} W_{ij} d_{ijkm} X_{ijkm}$$

$$\text{s.t. (3)–(5), (7), (8)}$$

$$\text{DC}_d \tag{9}$$

**Algorithm 1** Cutting Plane Algorithm

```
procedure CUTTING PLANE ALGORITHM
    iter ← 0
    d ← 0
    Generate p-HMPD_d
    while p-HMPD_d ≠ infeasible do
        Solve p-HMPD_d
        Y_d ← {k | y_k = 1}
        Obtain d̂ ← min_{k,m}{d_km}; ∀ k, m ∈ Y_d
        Update d ← d̂
        Update p-HMPD_d
    end while
    iter ← iter + 1
end procedure
```

In the following proposition, we present the correctness of the cutting plane algorithm.

**Proposition 1.** *The cutting plane algorithm generates all the non-dominated Pareto optimal solutions of Bp-HMDP.*

**Proof.** Let $\mathbf{Y_p} = \{Y_p^1, Y_p^2, Y_p^3, \ldots, Y_p^n\}$ be the set of all Pareto optimal hub locations generated for a problem instance by the cutting plane algorithm in $n$ iterations. Suppose by contradiction there exists a Pareto optimal solution $Y^x$ such that $f^1(Y_p^k) < f^1(Y^x) < f^1(Y_p^{k+1})$ and $f^2(Y_p^k) < f^2(Y^x) < f^2(Y_p^{k+1})$ where $Y_p^k$ and $Y_p^{k+1}$ are Pareto optimal hub locations obtained at consecutive iterations $k$ and $k+1$ respectively. This implies that at $k+1$th iteration the minimization problem $p$-HMDP$_{f^2(Y_p^k)}$ has a candidate solution $Y^x$ to choose from since $f^1(Y^x) < f^1(Y_p^{k+1})$. The preceding statement is a contradiction since $Y_p^{k+1}$ is the solution of the algorithm at iteration $k+1$. Hence, the cutting plane algorithm generates all non-dominated Pareto optimal solutions of B$p$-HMDP. □

Next, we present three alternate formulations of $p$-HMPD$_d$ and compare their strengths. The alternate formulations are obtained through different DCs that aim at reducing the size of the formulation. Through theoretical comparisons, we identify the strongest formulation amongst them and validate our findings through computational experiments.

### 3.1. Formulation 1

We present a set of dispersion constraints that eliminates the pair of hub locations which are less than or equal to the dispersion distance, $d$:

$$\{y_k + y_m \leq 1;\ (k,m) \mid d_{km} \leq d \quad \forall\, k,m \in H\}. \tag{DC1}$$

Similar dispersion constraint has been used by Sayyady et al. (2015) for facility dispersion problems.

The upper bound on the number of added DC1 to $p$-HMPD$_d$ is $\binom{|N|}{2}$, which can significantly impact the computational time of the formulation. We present a reduced constraint set, rDC1, which eliminates the redundant DC1s by symmetries between the distances. The reduced DC can be written as:

$$\{y_k + y_m \leq 1;\ (k,m) \mid d_{km} \leq d\ \forall\, k,m \in H;\ k < m\}. \tag{rDC1}$$

Let the corresponding formulation of $p$-HMPD$_d$ with rDC1 be denoted by $p$-HMPD$_d^1$.

### 3.2. Formulation 2

In this formulation, we define the dispersion constraint for every node $i \in N$. For that, let us define a set, $N_i = \{j \mid d_{ij} \leq d\}$. The dispersion constraint (DC2) is then defined as follows:

$$M_i y_i + \sum_{j \in N_i} y_j \leq M_i \quad \forall i \in N \tag{DC2}$$

where $M_i$ refers to a large number indexed for a particular $i$ (commonly referred to as Big-$M$). A very large value of $M_i$ can result in a poor LP relaxation while a smaller value can render the model infeasible. In the following proposition, we present a valid and tightest value of $M_i$ for the above constraint so that the formulation remains valid. We define the tightest value as the minimal largest value of $M_i$ that is valid for the formulation.

**Proposition 2.** *Let $M_i' = min(|N_i|, p)$. Then $M_i'$ is a valid value of $M_i$ for the dispersion constraint $DC2$.*

**Proof.** To prove that $M_i'$ is a valid value of $M_i$, we consider the following two cases based on the possible values of $M_i$:

- Case 1: $min(|N_i|, p) = |N_i|$. Let DC2$_i$ be the constraint corresponding to node $i$ which is obtained by adding rDC1 constraints where $i = k$ or $m$. The resulting constraint $DC2_i$ is as follows:

  $$|N_i| y_i + \sum_{j \in N_i} y_j \leq |N_i|$$

  Let $x$ be a value such that $x < |N_i|$. Substituting $M_i = x$ and $y_i = 0$, we get $\sum_{j \in N_i} y_j \leq x$ which results in an invalid condition. Therefore, the value of $M_i$ should be at least equal to $|N_i|$. Hence, $M_i'$ is a valid value of $M_i$ when $|N_i| < p$.
- Case 2: $min(|N_i|, p) = p$. In the previous case, we proved that the value of $M_i$ should be at least equal to $|N_i|$. However, the cardinality constraint (3) bounds the number of hubs to be located, i.e., the total number of $y$ variables that can take value 1, at $p$. Therefore, for the case where $N_i > p$, $p$ is a valid value of $M_i$. □

We denote the corresponding formulation of $p$-HMPD$_d$ with DC2 as $p$-HMPD$_d^2$.

### 3.3. Formulation 3

In this formulation, we use preprocessing to remove flow variables ($X_{ijkm}$) that take the value of zero in the optimal solution of the problem at each iteration of the cutting plane algorithm. This formulation makes use of the fact that if the distance $d_{km}$ between a particular hub pair $(k,m)$ is less than or equal to the dispersion distance $d$ for the iteration, then such a hub pair will not be used to route flow from any origin–destination pair $(i,j)$, in other words, the corresponding $x_{ijkm}$ variable equals zero. The main advantage of this formulation lies in the reduction of the number of decision variables at each iteration of the cutting plane method resulting in better LP relaxation and also reduced problem size. For this, we define a set $F_d = \{(k,m) \mid d_{km} > d;\ k \neq m\}$. Let rDC1$_d$ denotes the set of dispersion constraints pertaining to the dispersion distance $d$. The formulation is presented below:

$$[p\text{-HMPD}_d^3]: \tag{10}$$

$$\min_{y,X} \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} W_{ij} d_{ijkk} X_{ijkk} + \sum_{i \in N} \sum_{j \in N} \sum_{(k,m) \in F_d} W_{ij} d_{ijkm} X_{ijkm} \tag{11}$$

$$\text{s.t.} \sum_{k \in H} y_k = p$$

$$X_{ijkk} + \sum_{(k,m) \in F_d} X_{ijkm} = 1 \qquad \forall\, i,j \in N \tag{12}$$

$$X_{ijkk} + \sum_{m \mid (k,m) \in F_d} X_{ijkm} + \sum_{m \mid (m,k) \in F_d} X_{ijmk} \leq y_k \qquad \forall i,j,k \in H \tag{13}$$

$$X_{ijkk} \geq 0 \qquad \forall i,j \in N, k \in H \tag{14}$$

$$X_{ijkm} \geq 0; \qquad \forall i,j \in N; (k,m) \in F_d \tag{15}$$

$$y_k \in \{0,1\} \qquad \forall k \in H \tag{16}$$

$$\text{rDC1}_d$$

In the rest of the paper, we denote the bi-objective problem instance using formulations $p$-HMPD$_d^1$, $p$-HMPD$_d^2$, and $p$-HMPD$_d^3$ as $p$-HMPD$^1$, $p$-HMPD$^2$, and $p$-HMPD$^3$ respectively.

## 4. Dominance relationship

In this section, we study the theoretical properties of the three formulations of $p$-HMPD$_d$ to identify their strengths by comparing their linear programming (LP) relaxations. Let $LP^1$ and $LP^2$ denote the LP relaxations of two mixed integer programming (MIP) formulations of an optimization problem (with minimization objective). Also we use $MIP^1$ and $MIP^2$ to denote the two MIP formulations. Let $c(LP^1)$ and $c(LP^2)$ denote the feasible regions of the two LPs. $MIP^1$ dominates $MIP^2$ iff $c(LP^1) \subset c(LP^2)$. Alternatively, if the two formulations have the same LP feasible region then neither dominates the other and those formulations are known to be equivalent. The formulation which dominates the other is known to have better LP relaxation objective function value than the other, which in turn may result in lesser number of nodes visited in the branch-and-bound tree. We use this concept to identify the tighter formulation among the formulations presented in the previous section.

**Proposition 3.** *$p$-HMPD$_d^1$ dominates $p$-HMPD$_d^2$.*

**Proof.** Let $c(LP_d^1)$ and $c(LP_d^2)$ represent the LP feasible regions of $p$-HMPD$_d^1$ and $p$-HMPD$_d^2$ respectively. Let us consider the node $i$ and the associated set $N_i = \{j \mid d_{ij} \le d\}$. Let $j \in N_i$ be another node such that $d_{ij} \le d$. First let us construct a point $y_i = c_y$ such that $0 < c_y < 1$. The point $(y_i, y_j) = (c_y, 1 - c_y) \in c(LP_1^d), c(LP_2^d)$. For the case when $|N_i| > 1$, let us construct a new point $(y_i, y_j) = (c_y, 1)$ which lies in $c(LP - d^2)$ but not $c(LP_d^2)$. Therefore, $c(LP_d^1) \subset c(LP_d^2)$, but $c(LP_d^2) \not\subset c(LP_d^1)$. Thus, it follows that $p$-HMPD$_d^1$ dominates $p$-HMPD$_d^2$. □

To compare formulations $p$-HMPD$_d^3$ and $p$-HMPD$_d^1$ we directly compare their LP relaxation objectives as the dimensions of both the formulations are not the same. We use $O(LP_d^1)$ and $O(LP_d^3)$ to denote the objective function values of $c(LP_d^1)$ and $c(LP_d^3)$. We present the following proposition.

**Proposition 4.** $O(LP_d^3) \ge O(LP_d^1)$.

**Proof.** Let $X_d^1$ and $X_d^3$ denote the set of $X_{ijkm}$ variables in the formulations $p$-HMPD$_d^1$ and $p$-HMPD$_d^3$. By definition of these two sets, $X_d^3 \subseteq X_d^1$. At a particular iteration of the cutting plane algorithm with dispersion distance $d^*$, let us partition the set $X_d^1$ as $X_d^{1a}$ and $X_d^{1b}$. The set $X_d^{1a}$ contains all $X_{ijkm}$ variables such that $d_{km} \le d^*$. The set $X_d^{1b} = X_d^1 - X_d^{1a}$. Note that $X_d^{1b} \subset X_d^1$ and $X_d^3$. Solving $LP_d^1$, if all $X_{ijkm} \in X_d^{1b} = 0$ in the optimal solution, then $O(LP_d^1) = O(LP_d^3)$ else then $O(LP_d^1) < O(LP_d^3)$. Therefore, $O(LP_d^3) \ge O(LP_d^1)$. □

**Corollary.** $p$-HMPD$_d^3$ dominates $p$-HMPD$_d^2$.

**Proof.** $p$-HMPD$_d^1$ dominates $p$-HMPD$_d^2$. Since $p$-HMPD$_d^3$ is a reduced version $p$-HMPD$_d^1$, $p$-HMPD$_d^3$ dominates $p$-HMPD$_d^1$. Therefore, the proposition follows. □

### 4.1. Validation of dominance relationship

We verify the dominance relationship among the three formulations by comparing the objective function values of the LP relaxations for a particular instance of the problem. We use the first 50 nodes of USA423 dataset for this purpose. The number of hubs to be located ($p$) is set to 10. The coefficient of transshipment, $\delta$, is set to 0.9 whereas the coefficients of collection, $\alpha$, and distribution, $\gamma$, are set to 1. The results of the experiment are presented in Table 3. In total, twenty Pareto optimal solutions are generated by the cutting plane algorithm. Under the column "Objective", the sub-column "DO" displays the dispersion objective of the Pareto optimal solution, while the sub-column "MO" displays the median objective function values. Under the columns, $p$-HMPD$_d^1$, $p$-HMPD$_d^2$, and $p$-HMPD$_d^3$, we list the number of constraints, variables, and the LP integrality gap under the sub-columns, "#Cons"., "#Variables", and "Gap (%)". The LP integrality gap is the gap between the objective function value of optimal LP solution and the optimal integral solution. It is computed as a percentage difference between the MIP objective value and LP objective value.

Comparing the number of variables, we see that $p$-HMPD$_d^3$ has significantly lesser number of variables as we move along the Pareto optimal frontier. Comparing the number of constraints, we see that the formulation $p$-HMPD$_d^2$ has less number of constraints compared to the other two formulations since it uses aggregated form of constraint. Upon comparison of the LP integrality gap of formulations $p$-HMPD$_d^1$ and $p$-HMPD$_d^2$, we see that the formulation $p$-HMPD$_d^1$ provides a good LP lower bound to the IP problem than $p$-HMPD$_d^2$, thus validating our dominance results. Between formulations $p$-HMPD$_d^1$ and $p$-HMPD$_d^3$, we see that the latter provides a better integrality gap than the former in almost all instances. This is also in line with our theoretical findings. In Section 6 we present a detailed computational comparison among the three formulations using instances generated from the Australian Post (AP), Turkish network (TR81), and USA423 datasets. In the following section, we present a decomposition algorithm to solve $p$-HMPD$_d$.

## 5. Decomposition method

We present a decomposition method to solve large instances of B$p$-HMDP. The decomposition method is based on the observation from the literature that similar approaches have been found to be efficient in solving standalone large-scale hub location problems (Contreras et al., 2011; de Camargo et al., 2011, 2008, 2009; de Sá et al., 2018). The procedure is implemented within the cutting plane algorithm where at each iteration, we solve the $p$-hub median problem with dispersion distance $d$ ($p$-HMPD$_d$) using an accelerated Benders decomposition method instead of directly solving the MIP using a commercial solver.

Benders decomposition is a well-known partitioning procedure for solving mixed integer programs (Benders, 1962). Standard implementation of the procedure involves decomposing the MIP into a master problem with integer variables and a subproblem containing continuous variables. It is an iterative procedure where at each iteration a cut which is obtained from solving the subproblem is added to the master problem. The original algorithm suffers from slow convergence to optimality. Since then, multiple enhancement techniques to the procedure have been proposed to speed up the algorithm. In the context of hub location, Contreras et al. (2011) proposed several enhancements to the Benders decomposition procedure to solve large instances of the problem. For recent developments in the Benders decomposition method, we refer the readers to the review paper by Rahmaniani et al. (2017) which lists major variants and improvements.

In this paper, we propose an accelerated Benders decomposition method using some of the algorithmic enhancements pertaining to the hub location problem. We use the branch-and-cut version of Benders decomposition algorithm to solve the problem. The steps are outlined in Algorithm 2 and Algorithm 3. The approach was proposed by Fortz and Poss (2009) in which a single master problem is solved throughout the procedure. At various nodes of the branch and bound tree, subproblems are solved and Benders cuts are added to the master problem using a *callback* function. The decomposition method starts by solving a classical $p$-hub median problem without any dispersion cuts. The $p$-hub median problem is decomposed into a master problem containing integer variables and a subproblem containing continuous variables in a Benders decomposition framework. In our implementation, at each node of the master problem, the corresponding subproblem is called to generate Benders cuts to be added to the branch and cut tree. To start off with a better solution at the root node, we solve a $p$-median facility location problem to identify $p$ candidate hubs. We also perform root node enhancement for Benders decomposition outlined in McDaniel and Devine (1977) to generate cuts for the root node. From the subproblem, we generate Pareto optimal Benders cuts to be added to master problem for faster convergence. This approach was first developed by Magnanti and Wong (1981). Generating Pareto optimal Benders cuts is usually achieved through solving two subproblems (a standalone subproblem and subproblem with interior point on the binary variables). However, in our approach, we generate a single equivalent subproblem instead of two. In the following subsection, we present the decomposition method followed by the branch-and-cut implementation of the procedure. Finally, we present algorithmic enhancements to the branch-and-cut implementation of Benders decomposition method.

### 5.1. Benders decomposition

In our Benders decomposition procedure, we use the strongest formulation namely, $p$-HMPD$_d^3$. For the sake of completeness, the formulation is restated below:

[$p$-HMPD$_d^3$] :

$$\min_{y,X} \sum_{i\in N}\sum_{j\in N}\sum_{k\in H} W_{ij} d_{ijkk} X_{ijkk} + \sum_{i\in N}\sum_{j\in N}\sum_{(k,m)\in F_d} W_{ij} d_{ijkm} X_{ijkm}$$

$$\text{s.t.} \sum_{k\in H} y_k = p$$

**Table 3**
Comparison of LP relaxation objectives for the formulations (USA423 dataset, Instance with $|N| = 50$, $p = 10$, $\alpha, \delta, \gamma = 1, 0.9, 1$).

| S. no. | Objective | | Formulation: $p$-HMPD$_d^1$ | | | Formulation: $p$-HMPD$_d^2$ | | | Formulation: $p$-HMPD$_d^3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | DO | MO | #Cons. | #Variables | Gap (%) | #Cons. | #Variables | Gap (%) | #Cons. | #Variables | Gap (%) |
| 1 | 133 | 1,396,180 | 127,501 | 6,250,050 | 0 | 127,551 | 6,250,050 | 0 | 127,501 | 6,250,050 | 0 |
| 2 | 162 | 1,398,120 | 127,525 | 6,250,050 | 0 | 127,551 | 6,250,050 | 0 | 127,525 | 6,130,050 | 0 |
| 3 | 168 | 1,398,930 | 127,543 | 6,250,050 | 0.04 | 127,551 | 6,250,050 | 0.07 | 127,543 | 6,040,050 | 0.04 |
| 4 | 170 | 1,399,790 | 127,546 | 6,250,050 | 0.10 | 127,551 | 6,250,050 | 0.12 | 127,546 | 6,025,050 | 0.09 |
| 5 | 214 | 1,399,880 | 127,548 | 6,250,050 | 0.10 | 127,551 | 6,250,050 | 0.13 | 127,548 | 6,015,050 | 0.10 |
| 6 | 262 | 1,420,930 | 127,574 | 6,250,050 | 0.56 | 127,551 | 6,250,050 | 0.82 | 127,574 | 5,885,050 | 0.55 |
| 7 | 277 | 1,422,300 | 127,600 | 6,250,050 | 0.65 | 127,551 | 6,250,050 | 1.19 | 127,600 | 5,755,050 | 0.64 |
| 8 | 288 | 1,427,070 | 127,611 | 6,250,050 | 0.89 | 127,551 | 6,250,050 | 1.46 | 127,611 | 5,700,050 | 0.89 |
| 9 | 314 | 1,428,430 | 127,621 | 6,250,050 | 0.99 | 127,551 | 6,250,050 | 1.56 | 127,621 | 5,650,050 | 0.99 |
| 10 | 341 | 1,431,780 | 127,634 | 6,250,050 | 1.22 | 127,551 | 6,250,050 | 1.79 | 127,634 | 5,585,050 | 1.21 |
| 11 | 378 | 1,435,490 | 127,645 | 6,250,050 | 1.56 | 127,551 | 6,250,050 | 2.16 | 127,645 | 5,530,050 | 1.43 |
| 12 | 441 | 1,435,550 | 127,669 | 6,250,050 | 1.47 | 127,551 | 6,250,050 | 2.20 | 127,669 | 5,410,050 | 1.42 |
| 13 | 445 | 1,436,340 | 127,696 | 6,250,050 | 1.52 | 127,551 | 6,250,050 | 2.28 | 127,696 | 5,275,050 | 1.48 |
| 14 | 472 | 1,436,790 | 127,699 | 6,250,050 | 1.55 | 127,551 | 6,250,050 | 2.31 | 127,699 | 5,260,050 | 1.51 |
| 15 | 488 | 1,436,940 | 127,711 | 6,250,050 | 1.56 | 127,551 | 6,250,050 | 2.32 | 127,711 | 5,200,050 | 1.52 |
| 16 | 516 | 1,437,090 | 127,717 | 6,250,050 | 1.57 | 127,551 | 6,250,050 | 2.30 | 127,717 | 5,170,050 | 1.53 |
| 17 | 534 | 1,437,940 | 127,733 | 6,250,050 | 1.63 | 127,551 | 6,250,050 | 2.43 | 127,733 | 5,090,050 | 1.56 |
| 18 | 535 | 1,442,440 | 127,746 | 6,250,050 | 1.90 | 127,551 | 6,250,050 | 2.70 | 127,746 | 5,025,050 | 1.70 |
| 19 | 570 | 1,449,260 | 127,748 | 6,250,050 | 2.40 | 127,551 | 6,250,050 | 3.20 | 127,748 | 5,015,050 | 2.20 |
| 20 | 585 | 1,453,670 | 127,763 | 6,250,050 | 2.70 | 127,551 | 6,250,050 | 3.30 | 127,763 | 4,940,050 | 2.50 |

$$X_{ijkk} + \sum_{(k,m)\in F_d} X_{ijkm} = 1 \qquad \forall\, i,j \in N$$

$$X_{ijkk} + \sum_{m|\in(k,m)\in F} X_{ijkm} + \sum_{m\neq k|\in(k,m)\in F_d} X_{ijmk} \le y_k \qquad \forall i,j,k \in H$$

$$X_{ijkk} \ge 0 \qquad \forall i,j,k \in H$$

$$X_{ijkm} \ge 0; \qquad \forall i,j \in N; (k,m) \in F_d$$

$$y_k \in \{0,1\}$$

rDC1

As a standard step in Benders decomposition algorithm, we split the $p$-HMPD into a Master problem (MP$_d$) containing only integer variables and a subproblem (SP$_d$) containing only continuous variables.

Let $Y = \{y_k \in \{0,1\} | \sum_{k\in H} y_k = p\}$. For any fixed binary solution $\bar{\mathbf{y}} \in Y$, the resulting problem in the space of $\mathbf{X}$ variables, which we refer to as the *primal subproblem* (PS$_d$), can be stated as:

$$[PS_d]: \min_{\mathbf{X}} \sum_{i\in N}\sum_{j\in N}\sum_{k\in H} W_{ij} d_{ijkk} X_{ijkk} + \sum_{i\in N}\sum_{j\in N}\sum_{(k,m)\in F} W_{ij} d_{ijkm} X_{ijkm} \tag{17}$$

$$\text{s.t.} \sum_{k\in H} X_{ijkk} + \sum_{(k,m)\in F_d} X_{ijkm} = 1 \qquad \forall\, i,j \in N \tag{18}$$

$$X_{ijkk} + \sum_{m|\in(k,m)\in F_d} X_{ijkm} + \sum_{m|\in(m,k)\in F_d} X_{ijmk} \le \bar{y}_k \qquad \forall\, i,j \in N, k \in H \tag{19}$$

$$X_{ijkk} \ge 0 \qquad \forall\, i,j,k \in H, \tag{20}$$

$$X_{ijkm} \ge 0 \qquad \forall\, i,j \in N, (k,m) \in F_d \tag{21}$$

Associating dual variables $\phi_{ij}$ and $\delta_{ijk}$ for the constraints (18) and (19), respectively, we get the following dual subproblem:

$$[DS_d]: \max_{\phi,\delta} \sum_{i\in N}\sum_{j\in N} \phi_{ij} - \sum_{i\in N}\sum_{j\in N}\sum_{k\in H} \delta_{ijk} \bar{y}_k$$

$$\text{s.t.}\, \phi_{ij} - \delta_{ijk} \le W_{ij} d_{ijkm} \qquad \forall\, i,j \in N, k,m \in H, k = m \tag{22}$$

$$\phi_{ij} - \delta_{ijk} - \delta_{ijm} \le W_{ij} d_{ijkm} \qquad \forall\, i,j \in N, (k,m) \in F \tag{23}$$

$$-\infty \le \phi_{ij} \le \infty, \delta_{ijk} \ge 0 \qquad \forall\, i,j \in N, k \in H \tag{24}$$

The above dual maximization problem is bounded since the variables $\delta_{ijk}$ is non-negative, and the constraints (22) and (23) ensure that the free variable $\phi_{ij}$ has a finite lower bound. The master problem is as follows:

$$[MP_d]: \max_{\mathbf{y},\theta} \theta \tag{25}$$

$$\text{s.t.} \sum_{k\in H} y_k = p \tag{26}$$

$$\theta \le \sum_{i\in N}\sum_{j\in N} \bar{\phi}_{ij} - \sum_{i\in N}\sum_{j\in N}\sum_{k\in H} \bar{\delta}_{ijk} y_k \tag{27}$$

$$\text{rDC1} \tag{28}$$

$$\theta \ge 0, y_k \in \{0,1\} \qquad \forall k \in H \tag{29}$$

Solving the MP$_d$ with Benders cut (27) gives the upper bound, while the subproblem gives a lower bound to the original problem. The algorithm terminates when the difference between upper and the best lower bound is within a pre-specified tolerance $\epsilon$.

### 5.2. Branch-and-cut implementation of Benders decomposition

Standard implementation of Benders decomposition is an iterative procedure where $MP_d$ and $SP_d$ are solved consecutively. This approach generally results in slow convergence as the problem size becomes large. Modern implementation of Benders decomposition proposed by Fortz and Poss (2009) overcomes this difficulty by generating Benders cut for each solution in the branch and bound tree of $MP_d$ instead of adding it only at optimality. This approach entails solving only one MIP for each dispersion distance thereby providing a great advantage in terms of computational time. We use the branch-and-cut based Benders decomposition procedure to overcome slow convergence. In addition, in the next subsection, we also add algorithmic enhancements to the procedure to improve the computational performance of the algorithm.

### 5.3. Algorithmic enhancements to the Benders decomposition

To speed up the convergence of the Benders decomposition algorithm, various algorithmic enhancements have been proposed in the literature. The enhancements range from starting from a better solution, managing the memory issues of solving large scale subproblems through decomposition and solve a large number of individual subproblems and producing efficient cuts to speed up the overall convergence of the algorithm. In this paper, we employ several of these techniques as listed below.

#### 5.3.1. Root node enhancement of the branch-and-cut tree

We enhance the root node of the Benders branch-and-cut tree by adding cuts using the procedure outlined in McDaniel and Devine

(1977). Under this approach, the integer variables ($y$) are relaxed as continuous variables and Benders decomposition approach is performed to obtain the solution to the relaxed version of the original problem. Throughout this approach, the generated Benders cuts are added to the master problem. We also enhance the root node through reducing the $MP_d$ solution space by using preprocessing based variable fixing procedures. We employ the preprocessing based approaches presented in Contreras et al. (2010) and Zetina et al. (2021) for hub location problems. In addition to that, we also present a dispersion based preprocessing procedure to fix variables in the subproblem. The preprocessing based procedures are listed below:

- *Preprocessing based on reduced costs:* In our problem, the subproblem provides an upper bound while the master problem yields a lower bound to the original problem. For any nonbasic variable ($y_i$) in the master problem let its corresponding reduced cost be $RC(y_i)$. Then, if $RC(y_i) + LB > UB$, then the variable $y_i$ can be set to zero for that particular Pareto solution.
- *Preprocessing through variable fixing:* In this method, a specific set of location variables are fixed $(y_i = (1,0) \mid i \in Q)$ and the master problem is solved to get $LB(Q)$, if $LB(Q) > UB$ then such variables in the set $Q$ will no longer be part of the optimal solution and henceforth be set to zero (for $y_i = 1$) or one (for $y_i = 0$) for the rest of the iterations. In our decomposition algorithm, once the optimization problem is solved using the approach proposed by McDaniel and Devine (1977), we apply this preprocessing approach on two sets of location variables. If a location variable ($y$) takes value less than 0.2, then the corresponding variable is fixed to zero. Such variables are set to zero to check whether they can be fixed to one in the master problem. The other set of $y$ variables takes values greater than 0.7 in the optimal solution. Such variables are set to 1 to check whether they can be fixed to 0 in the master problem. The threshold values (0.2 and 0.7) are set based on our preliminary computational experiments.

### 5.3.2. Starting with a better solution

To start off with a better solution for the subproblem, we first propose solving a $p$- median facility location problem for a given dispersion distance ($FCLP_d$). The objective of the standard facility location problem is modified in our case to efficiently generate solutions for the hub location problem. In our case, the amount of demand originating and destined to a particular node $i$ is weighed along with transportation cost in the objective function. The formulation is presented below:

$$[FCLP_d]: \quad \min_{\mathbf{X}} \sum_{i\in N}\sum_{j\in N} d_{ij}X_{ij} \tag{30}$$

$$\text{s.t.} \sum_{j\in N} X_{ij} = 1 \qquad \forall i \in N \tag{31}$$

$$X_{ij} \le y_j \qquad \forall\, i, j \in N \tag{32}$$

$$\sum_{j\in N} y_j = p \tag{33}$$

$$\text{rDC1}_d \tag{34}$$

$$X_{ij} \ge 0, y_j \in \{0,1\} \qquad \forall\, i, j \in N \tag{35}$$

The variable $y_j$ is the binary variable denoting the presence of a facility at node $j$ and $X_{ij}$ denotes the assignment of customer $i$ to facility $j$. The solution to the above optimization problem is then used for solving the sub problem. In the following sub sections we propose efficient ways of solving the subproblem.

### 5.3.3. Efficient solution of the subproblem

For a given set of located hubs, the subproblem ($\text{DS}_d$) can be decomposed for each origin–destination pair $(i,j)$ as a problem of finding the shortest path between the origin $i$ and sink $j$ through a pair of hubs (de Camargo et al., 2008). The problem $PS_d^{ij}$ is presented below:

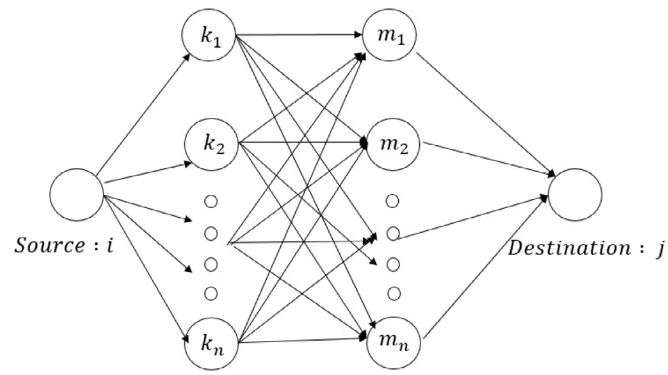



**Fig. 2.** Each problem instance of $\text{DS}_d$.

$$[PS_d^{ij}]: \min_{\mathbf{X}} \sum_{k\in H} W_{ij}d_{ijkk}X_{kk} + \sum_{(k,m)\in F_d} W_{ij}d_{ijkm}X_{km} \tag{36}$$

$$\text{s.t.} \sum_{k\in H} X_{kk} + \sum_{(k,m)\in F_d} X_{km} = 1 \tag{37}$$

$$X_{kk} + \sum_{m|\in(k,m)\in F_d} X_{km} + \sum_{m|\in(m,k)\in F_d} X_{mk} \le \bar{y}_k \qquad \forall\, k \in H \tag{38}$$

$$X_{kk}, X_{km} \ge 0 \qquad \forall\, (k,m) \in F_d\ k \in H \tag{39}$$

Accordingly, the dual subproblem $DS_d^{ij}$ is presented below:

$$[DS_d^{ij}]: \max_{\phi,\delta} \phi_{ij} - \sum_{k\in H} \delta_{ijk}\bar{y}_k$$

$$\text{s.t.}\phi_{ij} - \delta_{ijk} \le W_{ij}d_{ijkm} \qquad \forall\, k, m \in H, k = m \tag{40}$$

$$\phi_{ij} - \delta_{ijk} - \delta_{ijm} \le W_{ij}d_{ijkm} \qquad \forall\, (k,m) \in F_d \tag{41}$$

$$\infty \le \phi_{ij} \le \infty, \delta_{ijk} \ge 0 \qquad \forall\, k \in H \tag{42}$$

Instead of solving the dual problem $DS_d^{ij}$, we solve the primal problem $PS_d^{ij}$ to exploit its network structure for solving efficiently. The primal problem $PS_d^{ij}$ can be interpreted as the problem of allocation of hubs to a particular origin–destination pair $(i,j)$. This is technically equivalent to a transportation problem with a origin node sending one unit of flow to a destination node with multiple intermediate nodes to choose from. The schematic is given in Fig. 2.

### 5.3.4. Pareto optimal Benders cuts

In order to improve the convergence of the Benders decomposition procedure, Magnanti and Wong (1981) proposed a way of generating efficient Benders cuts that dominates all other possible cuts for the subproblem. For $(\phi'_{ij}, \delta'_{ijk}), (\phi''_{ij}, \delta''_{ijk}) \in DS_{ij}$, a cut $(\theta_{ij} \ge \phi'_{ij} + \delta'_{ijk}y_k)$ dominates another cut $(\theta_{ij} \ge \phi''_{ij} + \delta''_{ijk}y_k)$ if $(\phi'_{ij} + \delta'_{ijk}y_k) \ge (\phi''_{ij} + \delta''_{ijk}y_k)$ for all $y_k \in \mathbb{Y}$ with strict inequality for at least $y_k \in \mathbb{Y}$. The method employs a core-point which is defined as an interior point in the convex hull of $\mathbb{Y}$. Given a core-point, $\bar{y}_k^p \in ri(\mathbb{Y}^c)$ , where $ri(\mathbb{Y}^c)$ denotes the relative interior of the convex hull of $\mathbb{Y}^c$, the Pareto optimal Benders cuts can be generated by solving the following optimization problem ($\bar{DS}_d^{ij}$):

$$[\bar{DS}_d^{ij}]: \max_{\phi,\delta} \phi_{ij} - \sum_{k\in H} \delta_{ijk}\bar{y}_k^p \tag{43}$$

$$\text{s.t. } \phi_{ij} - \delta_{ijk} \le W_{ij}d_{ijkk} \qquad \forall\, k \in H \tag{44}$$

$$\phi_{ij} - \delta_{ijk} - \delta_{ijm} \le W_{ij}d_{ijkm} \qquad \forall\, (k,m) \in F_d \tag{45}$$

$$\phi_{ij} - \sum_{k\in H} \delta_{ijk}\bar{y}_k = Obj_{ij} \tag{46}$$

$$-\infty \le \phi_{ij} \le \infty, \delta_{ijk} \ge 0 \qquad \forall\, k \in H \tag{47}$$

Here, $Obj_{ij}$ is the objective function value of the original problem $DS_d^{ij}$. The Pareto optimal cut generated from $D\bar{S}_d^{ij}$ is as follows:

$$\theta_{ij} \leq \bar{\phi}_{ij}^{p} - \sum_{k \in H} \bar{\delta}_{ijk}^{p} y_k$$

Constraint (46) destroys the network structure of the dual of $D\bar{S}_d^{ij}$. However, we reformulate the problem as a parametric transportation problem using the procedure that follows (Magnanti et al., 1986; Zetina et al., 2021). Associating dual variable $X_{km}$ with constraints (44) and (45) and $\eta_{ij}$ with constraints (46) respectively, the dual of the problem can be written as:

$$[P\bar{S}_d^{ij}] : \min \sum_{k \in H} W_{ij} d_{ijkk} X_{kk} + \sum_{(k,m) \in F_d} W_{ij} d_{ijkm} X_{km} - Obj_{ij} \eta_{ij}$$

$$\text{s.t.} X_{kk} + \sum_{(k,m) \in F_d} X_{km} = 1 + \eta_{ij} \tag{48}$$

$$X_{kk} + \sum_{m \mid \in (k,m) \in F_d} X_{km} + \sum_{m \mid \in (m,k) \in F_d} X_{km} \leq y_k^p + \eta_{ij} \bar{y}_k \qquad \forall\, k \in H, (k,m) \in F_d \tag{49}$$

$$\eta_{ij}, X_{km} \geq 0; \qquad \forall\, (k,m) \in F_d \; X_{kk} \in H \tag{50}$$

This above formulation is of a parametric transportation problem in the parameter $\eta_{ij}$ which induces a variable demand $(1+\eta_{ij})$ units, a variable arc capacity of $y_k^p + \eta_{ij}$ for arcs in the current design ($\bar{y}_k = 1$) and an arc capacity of $y_k^p$ for arcs absent in the current design ($\bar{y}_k = 0$). To solve the problem efficiently using network simplex procedure, we first fix the value of the variables $\eta_{ij}$ so that the subtrahend term in the objective is constant and therefore can be removed from the formulation. Hence we will be solving only $P\bar{S}_d^{ij}$ instead of solving both $PS_d^{ij}$ and $P\bar{S}_d^{ij}$.

**Proposition 5.** *$\eta_{ij} \geq 0$ is an optimal value to $PS_d^{ij}$.*

**Proof.** The variable $\eta_{ij} = 1$ corresponds to one unit of additional demand/capacity that provides a discount of $Obj_{ij}$ in addition to increasing the cost by $W_{ij} d_{ijkm}$. By constraint (48), we must send at least one unit of fixed demand in the network. Once this is sent, the variable $\eta_{ij}$ will increase in value as long as the marginal cost of sending becomes equal to $Obj_{ij}$. Note that the total fixed capacity used in the network is 1, i.e., $\sum_k y_k^p = 1$ (from (48)). Hence, any additional flow in the network must use arcs in the current design, for which the marginal cost will be $Obj_{ij}$ and it will be completely compensated. Therefore, $\eta_{ij} \geq 0$ is an optimal value to $PS_d^{ij}$. □

As discussed above, by substituting the value of $\eta_{ij} = 1$, the problem $P\bar{S}_d^{ij}$ reduces to a standard transportation problem. Since, the subtrahend term in the objective function becomes a constant by substituting the value of $\eta_{ij}$, it obviates the need for solving $PS_d^{ij}$. Thus, in our implementation, we solve only $P\bar{S}_d^{ij}$ instead of both $PS_d^{ij}$ and $P\bar{S}_d^{ij}$ as in the standard implementation. This reduction in number of problems to solve provides us significant computational advantage as shown in Section 6.2.

### 5.4. Decomposition algorithm

We present the details of the proposed decomposition algorithm. At the start of the algorithm ($c = 0$), we set the dispersion distance $d$ to zero. The problem $p$-HMDP$_d^3$ is decomposed into a master problem ($MP_d$) and a subproblem ($SP_d$). We start off with a better solution for $SP_d$ using initial solution obtained by solving $FLP_d$. Then using the hub locations obtained from solving $FLP_d$, the subproblem $SP_d$ is solved and initial Benders cuts are added to $MP_d$. Upon relaxing the location variables ($y_k$) in $MP_d$ and solving it, we obtain fractional vector of $y$ values ($Y_i^f$) which is again passed to $SP_d$. We solve $SP_d$ and generate Benders cuts that are added to $MP_d$. $MP_d$ is further enhanced through preprocessing techniques based on variable fixing and reduced costs. Integralising the location variables ($y_k$), $MP_d$ is solved through Benders branch-and-cut approach. From the resulting location solution ($Y_c$), the dispersion distance $d$ is updated and the iteration continues. The algorithm terminates when $MP_d$ becomes infeasible. The procedure is outlined in Algorithm 2. Note than in the proposed decomposition algorithm, the subproblem $SP_d$ is decomposed and solved for each O–D pair $(i,j)$ using $PS_d^{ij}$.

**Algorithm 2** Decomposition Algorithm

**procedure** DECOMPOSITION ALGORITHM
  Set $c \leftarrow 0$
  **if** $c = 0$ **then**
    $d \leftarrow 0$
  **else** Generate $d$ from $Y_{i-1}$
    Generate $p$-HMDP$_d^3$
    Generate $MP_d$ and $SP_d$
    **while** $MP_d \neq$infeasible **do**
      Obtain $Y_c^{flp}$ using $FLP_d$
      $Y_c \leftarrow Y_c^{flp}$
      Pass $Y_i$ to $SP_d$
      **for** $i \in N$ **do**
        **for** $j \in N$ **do**
          Add benders cuts to $MP_d$ by through solving $PS_d^{ij}$
        **end for**
      **end for**
      Relax $y$ variables in $MP_d$
      Solve $MP_d$ to obtain $Y_c^{frac}$
      Pass $Y_c^{frac}$ to $SP_d$
      **for** $i \in N$ **do**
        **for** $j \in N$ **do**
          Add benders cuts to $MP_d$ by through solving $PS_d^{ij}$
        **end for**
      **end for**
      Fix $y_k$ in $MP_d$ using preprocessing
      Integralise $y$ variables in $MP_d$
      Solve $MP_d$ using branch-and-cut benders
      Obtain $Y_i$
    **end while**
  **end if**
  Update $c \leftarrow c + 1$
**end procedure**

**Algorithm 3** Branch-and-Cut Benders

**procedure** BRANCH-AND-CUT BENDERS
  **while** $MP_d \neq$ optimal **do**
    for every incumbent node in the branch-and-cut tree, solve $P\bar{S}_d^{ij}$ to add Benders cuts that violate current incumbent solution to $MP_d$
  **end while**
**end procedure**

## 6. Computational results

We conducted extensive computational experiments to assess the performance of the proposed exact solution methods for the bi-objective $p$-hub median and dispersion problem. The formulations and the solution algorithms were coded in C++ and run on workstation with 3.6 GHz Intel Xeon processor with 64 GB RAM and six cores. All the instances are solved using CPLEX 12.8. The numerical tests were performed on instances derived from the Australian Post (AP) dataset, the USA423 dataset, and the Turkish network (TR81) obtained from the OR library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html).

In the first set of experiments, we test the performance of the cutting plane algorithm with different formulations of $p$-HMDP$_d$, where the MIP model of $p$-HMDP$_d$ is solved using CPLEX 12.8 directly. For the Turkish (TR81) dataset, we generate problem instances with first 30, 40, 50 and 60 nodes of the dataset. The discount factors for collection ($\alpha$) and distribution ($\gamma$) are set to 1. The discount factor for transshipment ($\delta$) is varied from 0.1, 0.5, to 0.9. The number of hubs to locate ($p$) in all our experiments is varied in the set $\{10, 15, 20\}$. For the USA423 dataset, we generate problem instances with first 30, 40, 50 and 60 nodes of the dataset. The discount factors for collection ($\alpha$) and distribution ($\gamma$) are set to 1. The discount factor for transshipment ($\delta$) is varied from 0.1, 0.5, to 0.9. The number of hubs to locate ($p$) in all our experiments is varied in the set $\{10, 15, 20\}$. For the AP dataset, the instances are generated using first 50, 60, and 70 nodes of the dataset. The discount factors for collection ($\alpha$) and distribution ($\gamma$) are set to 3 and 2 respectively. The discount factor for transshipment ($\delta$) is varied from 0.25, 0.5, 0.75. The number of hubs to locate ($p$) in all our experiments is varied in the set $\{10, 15, 20\}$.

In the second set of experiments, we test the performance of the cutting plane algorithm with Benders decomposition method, where the $p$-HMDP$_d$ is solved using the branch-and-cut algorithm. For this, we use TR81 instances with $|N| \in \{60, 70, 81\}$ and AP and USA423 instances with $|N| \in \{70, 80, 90, 100\}$ respectively. The number of hubs to locate ($p$) is varied in the set $\{10, 15, 20\}$. We vary the co-efficient of transshipment, $\delta$ in the set $\{0.25, 0.5, 0.75\}$ for AP dataset instances, while $\delta$ is varied in the set $\{0.1, 0.5, 0.9\}$ for USAS423 and TR81 datasets. The factors $\alpha$ and $\gamma$ are set to 3 and 2 for AP dataset and 1 for USA423 and TR81 dataset. In all the experiments, we set a time limit to 86,400 s (i.e. 24 h) of CPU time. Instances that could not be solved to optimality within this time limit are marked with the label "time". Similarly, instances that could not be solved due to memory restrictions are marked with the label "memory".

### 6.1. Performance of the cutting plane algorithm with alternative formulations

In this subsection, we compare the computational performance of the three formulations of $p$-HMPD$_d$ using the proposed cutting plane approach. In the first part of computational experiments, we compare the strength of the three formulations over the 27 problem instances derived from the TR81 dataset. In Table 4, we present the detailed results for these instances. The first three columns, list the problem parameters — the number of nodes ($|N|$), the number of hubs to locate ($p$) and discount factors. The discount factor for transshipment ($\delta$) is varied from 0.1, 0.5, to 0.9. The discount factors for collection ($\alpha$) and distribution ($\gamma$) are set to 1. The next three columns report the computational time (in seconds) for the three formulations. The columns, "1/2", "3/2", "3/1" report the percentage reduction in computational time achieved using $p$-HMPD$^1$ over $p$-HMPD$^2$, $p$-HMPD$^3$ over $p$-HMPD$^2$, and $p$-HMPD$^3$ over $p$-HMPD$^1$ respectively. The column, "Tot". denote the total number of solutions visited by the three formulations. Note that all the three formulations yield same solutions when implemented in the cutting plane procedure. The last three columns report the average time to obtain one solution in the optimal Pareto frontier through $p$-HMPD$^1$, $p$-HMPD$^2$, and $p$-HMPD$^3$ respectively. From our experiments, we observe that all the three formulations generate all Strictly efficient Pareto solutions of the problem.

The results in Table 4 clearly show the computational efficiency of $p$-HMPD$^3$, as highlighted by its lower CPU times compared to the other formulations. This is in line with our theoretical findings which showed that $p$-HMPD$^3$ is the strongest amongst the three formulations, mainly due to a strong LP relaxation. Comparison of the results between $p$-HMPD$^1$ and $p$-HMPD$^3$ shows that the latter is faster than the former on 21 (out of 27) instances. This observation also is in line with our theoretical observation that $p$-HMPD$^3$ yields a better upper bound than $p$-HMPD$^1$ with fewer variables. The average CPU time required to solve one instance for $|N| = 30$ is 107 s for $p$-HMPD$^1$, 117 s for $p$-HMPD$^2$, and 103 s for $p$-HMPD$^3$. Similarly, the figures for $|N| = 50$ instances are: 4,967 s for $p$-HMPD$^1$, 5,819 s for $p$-HMPD$^2$, 4404 s for $p$-HMPD$^3$. For instances with $|N| = 30$, $p$-HMPD$^3$ provides a gain of 12% over $p$-HMPD$^2$ and 4% over $p$-HMPD$^1$. Similarly, the figures for $|N| = 40$, and $|N| = 50$ are: 12% and 5%, and 19% and 11% respectively.

In Table S1, presented as the supplementary material along with this paper, we report the efficiency of our proposed formulations in solving problem instances derived from the USA423 dataset. For this, we generate problem instances with first 30, 40, 50 and 60 nodes of the dataset. The discount factors for collection ($\alpha$) and distribution ($\gamma$) are set to 1. The discount factor for transshipment ($\delta$) is varied from 0.1, 0.5, to 0.9. The number of hubs to locate for each of the instances is varied from 10, 15, to 20. For instances with $|N| = 60$, we do not report results pertaining to $p = 15$ and 20 hubs as none of the formulations could solve owing to memory restrictions. The computational results are similar to that on the TR81 network. In all the instances, the computational time for $p$-HMPD$^2$ is higher than either $p$-HMPD$^1$ and $p$-HMPD$^3$. Out of the thirty instances, $p$-HMPD$^3$ is faster on 17 instances while $p$-HMPD$^1$ is faster on 13 instances. For instances with $|N| = 30$, 40, and 60, $p$-HMPD$^3$ gives better average computational performance than $p$-HMPD$^1$. For instances with $|N| =$ 50, $p$-HMPD$^1$ is computationally faster than $p$-HMPD$^3$ on average.

In the next set computational experiments, we compare the computational performance of the two best formulations namely, $p$-HMPD$^1$ and $p$-HMPD$^3$ using the AP dataset. The results are provided in Table S2 that appears in the supplementary material attached along with the paper. The instances are generated using first 50, 60, and 70 nodes in the AP dataset. Note that we do not report results pertaining to the instances with $|N| = 70$ and $p = 20$ as both formulations could not solve the instances due to memory restrictions. The average CPU time for 50-node instances using $p$-HMPD$^3$ is 4,017 s compared to 4,335 s using $p$-HMPD$^1$. The average computational gain in this case is 7%. The average CPU time for 60-node instances using $p$-HMPD$^3$ is 11,988 s compared to 14,922 s using $p$-HMPD$^1$ and the average computational gain in this case is 20%. Also, for all these instances, $p$-HMPD$^3$ is faster than $p$-HMPD$^1$ on 11 of the 18 instances. For the instances where $p$-HMPD$^3$ is faster than $p$-HMPD$^1$, the reduction in CPU time ranges from 5% to 40%. Note that $p$-HMPD$^1$ could not solve any of the 70-node instances either within the time limit or due to memory requirement, whereas $p$-HMPD$^3$ could solve 6 of them. Hence, the results show that the formulation $p$-HMPD$^3$ is computationally superior as problem size increases.

In summary, our computational experiments validate our theoretical findings and show the advantage of $p$-HMPD$^1$ and $p$-HMPD$^3$ over $p$-HMPD$^2$. Between $p$-HMPD$^1$ and $p$-HMPD$^3$, though the former gives better performance than the latter in some instances, $p$-HMPD$^3$ provides superior performance in handling larger data instances as evidenced by our computational experiments. In the following subsection, we show the computational performance of our decomposition approach using our best formulation namely, $p$-HMPD$^3$.

#### 6.1.1. Statistical analysis of the results

We present a detailed statistical analysis of the results of the three formulations. We perform pairwise Wilcoxon rank test on the three formulations to identify the formulation with the best statistical guarantee of computational performance. If a particular experimental instance could not be solved either due to memory or time limit using at least one of the three formulations, then the instance is not counted in the statistical analysis. The null hypothesis is as follows: *There is no significant improvement in computational performance using the two formulations*. We set the level of significance ($\alpha$) to be 0.1. Comparing $p$-HMPD$^2$ with $p$-HMPD$^1$ and $p$-HMPD$^3$ pairwise, we see that the null hypothesis is rejected with a very high level of significance. Between $p$-HMPD$^1$ and $p$-HMPD$^3$, we see that we are able to reject the null hypothesis when $\alpha = 0.1$ for both TR81 and AP instances. The null

**Table 4**
Computational results of the cutting plane algorithm on the Turkish network.

| Parameters | | | CPU time | | | % Reduction | | | Tot. | Average time | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N | $p$ | $\alpha$, $\delta$, $\gamma$ | $p$-HMPD$^1$ | $p$-HMPD$^2$ | $p$-HMPD$^3$ | 1/2' | 3/2' | 3/1' | # | $p$-HMPD$^1$ | $p$-HMPD$^2$ | $p$-HMPD$^3$ |
| 30 | 10 | 1,0.1,1 | 115 | 132 | **106** | 13 | 20 | 8 | 11 | 10 | 12 | 9 |
| | | 1,0.5,1 | 115 | 140 | **109** | 18 | 22 | 5 | 11 | 11 | 13 | 10 |
| | | 1,0.9,1 | 176 | 182 | **154** | 3 | 15 | 16 | 13 | 13 | 14 | 12 |
| | 15 | 1,0.1,1 | **137** | 141 | **137** | 3 | 3 | 0 | 14 | 10 | 10 | 10 |
| | | 1,0.5,1 | **121** | 135 | **121** | 10 | 10 | 0 | 13 | 9 | 10 | 9 |
| | | 1,0.9,1 | **129** | 138 | **129** | 7 | 7 | 0 | 13 | 10 | 11 | 10 |
| | 20 | 1,0.1,1 | **58** | 60 | **58** | 3 | 3 | 0 | 6 | 10 | 10 | 10 |
| | | 1,0.5,1 | **57** | 61 | **57** | 7 | 7 | 0 | 6 | 10 | 10 | 10 |
| | | 1,0.9,1 | **55** | 60 | **55** | 8 | 8 | 0 | 6 | 9 | 10 | 9 |
| Average | | | 107 | 117 | **103** | 9 | 12 | 4 | | 10 | 11 | 10 |
| 40 | 10 | 1,0.1,1 | 1276 | 1383 | **1124** | 8 | 19 | 12 | 18 | 71 | 77 | 62 |
| | | 1,0.5,1 | 1175 | 1221 | **1043** | 4 | 15 | 11 | 15 | 78 | 81 | 69 |
| | | 1,0.9,1 | 966 | 1129 | **816** | 14 | 28 | 16 | 13 | 74 | 87 | 63 |
| | 15 | 1,0.1,1 | **806** | 921 | 874 | 12 | 5 | – | 17 | 47 | 54 | 51 |
| | | 1,0.5,1 | **816** | 856 | 825 | 5 | 4 | – | 16 | 51 | 54 | 51 |
| | | 1,0.9,1 | **956** | 1079 | 975 | 11 | 10 | – | 16 | 60 | 67 | 61 |
| | 20 | 1,0.1,1 | 626 | 632 | **602** | 1 | 5 | 4 | 17 | 37 | 37 | 35 |
| | | 1,0.5,1 | **560** | 570 | 566 | 2 | 1 | – | 14 | 40 | 40 | 40 |
| | | 1,0.9,1 | 624 | 726 | **600** | 14 | 17 | 4 | 14 | 45 | 43 | 43 |
| Average | | | 867 | 946 | **825** | 8 | 12 | 5 | | 56 | 60 | 53 |
| 50 | 10 | 1,0.1,1 | 6301 | 6987 | **4736** | 10 | 37 | 25 | 21 | 300 | 333 | 225 |
| | | 1,0.5,1 | 9273 | 11,019 | **6990** | 16 | 37 | 31 | 26 | 357 | 424 | 269 |
| | | 1,0.9,1 | 9296 | 12,483 | **8195** | 26 | 34 | 12 | 24 | 387 | 520 | 341 |
| | 15 | 1,0.1,1 | **4057** | 4426 | 4309 | 8 | 3 | – | 18 | 225 | 246 | 239 |
| | | 1,0.5,1 | **3286** | 4242 | 3998 | 23 | 6 | – | 15 | 219 | 283 | 266 |
| | | 1,0.9,1 | 3864 | 3915 | **3669** | 1 | 6 | 5 | 16 | 241 | 241 | 229 |
| | 20 | 1,0.1,1 | 2624 | 2804 | **2474** | 6 | 12 | 6 | 18 | 146 | 156 | 137 |
| | | 1,0.5,1 | 2475 | 2691 | **2328** | 8 | 13 | 6 | 17 | 146 | 158 | 137 |
| | | 1,0.9,1 | 3530 | 3803 | **2933** | 7 | 23 | 17 | 19 | 186 | 200 | 154 |
| Average | | | 4967 | 5819 | **4404** | 12 | 19 | 11 | | 245 | 285 | 222 |

hypothesis is not rejected for USA423 dataset. However, considering the performance of $p$-HMPD$^3$ in solving large instances and overall superior performance of $p$-HMPD$^3$ over $p$-HMPD$^1$, we conclude that $p$-HMPD$^3$ is computationally better than $p$-HMPD$^1$.

#### 6.1.2. Performance profile of the three formulations

In this subsection, we present performance profile of the three formulations over the three datasets. We consider only those instances that are solved to optimality within the time limit. We also exclude instances that could not be solved due to memory limitations. Performance profile for a particular formulation is the cumulative distribution function of its computational time (Dolan & Moré, 2002). Let $t^i_f$ be the computational time required to solve a particular problem instance $i$ using the formulation $f$, and $n$ be the number of instances solved, we define the performance ratio $r_f$ as: $r_f = \dfrac{t^i_f}{min(t^i_f)\ \ \forall f}$. Now, we define a function $p_f(\tau)$ as the probability that the performance ratio $r_f$ is within a factor $\tau$ of the best possible ratio as: $p_f(\tau) = \dfrac{\text{size}\{r_f \le \tau\}}{n}$. The function $p_f$ is the cumulative distribution function of the performance ratio $r_f$. We compare the performance profile for the three formulations using the three datasets (TN, USA423, AP). The performance profile for TN is given in Fig. 3, while for USA423 and AP datasets the performance profiles are presented in the supplementary material as figures S1, and S2 respectively. In the figures, the factor ($\tau$) is plotted along the horizontal axis, while the distribution function ($p_f$) is plotted along the vertical axis. From the performance profiles of the three datasets, we deduce that both $p$-HMPD$^1$ and $p$-HMPD$^3$ outperforms $p$-HMPD$^2$ for any value of $\tau$ across all three datasets. Between $p$-HMPD$^1$ and $p$-HMPD$^3$ we see that $p$-HMPD$^3$ gives better performance guarantee than $p$-HMPD$^1$ across all datasets. Considering those instances that could not be solved either due to memory or time limit, we conclude that $p$-HMPD$^3$ dominates both $p$-HMPD$^1$ and $p$-HMPD$^2$ in terms of its computational performance.

### 6.2. Performance of decomposition method for large instances

We present the results of the cutting plane algorithm with decomposition method for solving large instances of the problem. Table 5 reports the results based on the first 60, 70, and all 81 nodes of the Turkish (TR81) dataset. Similarly, Tables S3 and S4 report the results of the decomposition method over instances with $N \in \{70, 80, 90, 100\}$ based on the first 70, 80, 90, and 100 nodes in the USA423 and AP dataset. These tables appear in the supplementary material attached along with the paper. In these tables, the column marked, "Instance Parameters" represents the parameters set for the instance. The column "CPU Time" presents the computational time for the problem instance. We set a time limit of 86,400 s (i.e. 24 h) in this experiment. The problem instances which could not be solved within the time limit are marked with "*". Under the column "# Pareto Solutions" two sub-columns are presented namely, "Total" and "DM". The sub-column, "Total" reports the total number of Pareto optimal solutions for a particular problem instance, while the sub-column "DM" reports the number of Pareto optimal solutions obtained within the time limit. For obtaining the total number of Pareto optimal solutions, we run the decomposition algorithm till the problem becomes infeasible. The last column, "Average Time" report the average time to generate one solution in the Pareto frontier.

For the TR81 dataset, we see that the decomposition approach is able to solve all problem instances for $|N| = 60$ and 70, and all but one instance for $|N| = 81$ nodes. The average time to solve an instance is 9319 s for $|N| = 60$ and 16,801 s for $|N| = 70$ respectively. For instances with $|N| = 81$, the average time to solve is 33,587 s excluding the instance that could not be solved within the time limit. The average

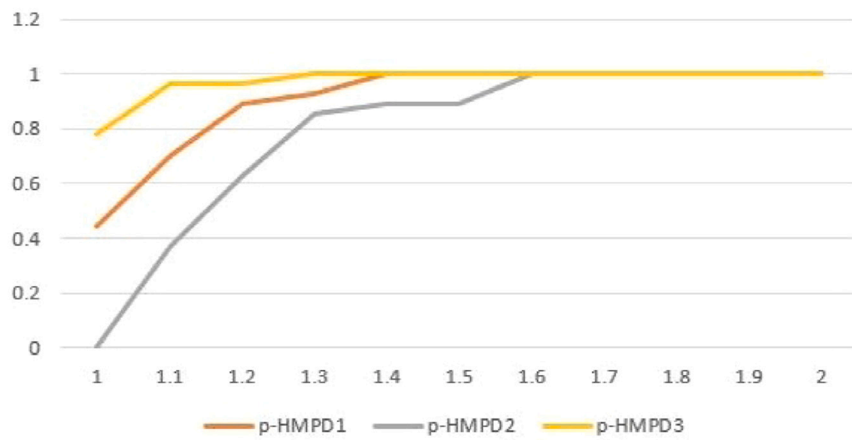



**Fig. 3.** Performance profile for Turkish Network dataset.

**Table 5**
Computational results for decomposition method on Turkish network.

| Instance parameters | | | CPU time (s) | # Pareto solutions | | Avg. time (s) |
|---|---|---|---|---|---|---|
| $\|N\|$ | $p$ | $\alpha, \delta, \gamma$ | | Total | DM | |
| 60 | 10 | 1, 0.9, 1 | 8.420 | 18 | 18 | 468 |
| | 15 | | 11,136 | 25 | 25 | 445 |
| | 20 | | 9672 | 21 | 21 | 461 |
| | 10 | 1, 0.5, 1 | 8319 | 21 | 21 | 398 |
| | 15 | | 9434 | 22 | 22 | 429 |
| | 20 | | 7636 | 19 | 19 | 402 |
| | 10 | 1, 0.1, 1 | 7864 | 17 | 17 | 462 |
| | 15 | | 11,892 | 23 | 23 | 517 |
| | 20 | | 9501 | 22 | 22 | 432 |
| Average | | | 9319 | | | 446 |
| 70 | 10 | 1, 0.9, 1 | 16,573 | 21 | 21 | 789 |
| | 15 | | 24,560 | 28 | 28 | 878 |
| | 20 | | 15,118 | 16 | 16 | 945 |
| | 10 | 1, 0.5, 1 | 17,307 | 23 | 23 | 752 |
| | 15 | | 17,323 | 23 | 23 | 753 |
| | 20 | | 10,281 | 14 | 14 | 734 |
| | 10 | 1, 0.1, 1 | 18,936 | 24 | 24 | 789 |
| | 15 | | 18,012 | 28 | 28 | 643 |
| | 20 | | 13,100 | 18 | 18 | 728 |
| Average | | | 16,801 | | | 779 |
| 81 | 10 | 1, 0.9, 1 | 38,630 | 19 | 19 | 2033 |
| | 15 | | 86,400[a] | 35 | 31 | 2767 |
| | 20 | | 42,106 | 23 | 23 | 1830 |
| | 10 | 1, 0.5, 1 | 34,029 | 22 | 22 | 1546 |
| | 15 | | 31,139 | 25 | 25 | 1256 |
| | 20 | | 27,334 | 20 | 20 | 1367 |
| | 10 | 1, 0.1, 1 | 34,694 | 23 | 23 | 1508 |
| | 15 | | 33,043 | 25 | 25 | 1322 |
| | 20 | | 27,727 | 20 | 20 | 1386 |
| Average | | | | | | 1668 |

[a] Time limit exceeded.

time to generate one Pareto instance is 446 s for $|N| = 60$, 779 s for $|N| = 70$, and 1,668 s for $|N| = 81$.

We see similar computational performance on instances from USA423 and AP datasets in Tables S3 and S4. In Table S3, we observe that the decomposition approach is able to solve all USA423 problem instances for $|N| = 70$, 80, and 90 nodes. The average CPU time to solve is 14,307 s for 70 nodes, 27,592 s for 80 and 53,655 s for 90 nodes respectively. For instances with $|N| = 100$, the algorithm is able to solve only two out of nine instances within the time limit. For instances that could not be solved within the time limit, the results show that the algorithm is able to generate at least 85% of the total number of Pareto optimal solutions. The average time to generate a Pareto optimal solution are 523, 880, 1404 and 2208 s for $|N| = 70$, 80, and 90, and 100 respectively.

In Table S4, we present the results of the AP instances using our decomposition algorithm. The table shows that for 70-node instances, which could not be solved using our cutting plane algorithm (that relies on CPLEX for solving MIP) due to memory restrictions, the decomposition method is able to generate all solutions in the Pareto optimal frontier within an average of 16,814 s. The average time to generate one Pareto optimal solution using the decomposition method is 619 s. For instances with $|N| = 80$ and 90 nodes, the decomposition method is able to solve the problem within an average time of 32,857 s (9.12 h) and 63,742 s (17.70 h) respectively. For problem instances with $|N| = 100$, the decomposition method is able to generate 70 to 80% of the total number of Pareto optimal solutions within the time limit (86,400 s). The results demonstrate the efficiency of our decomposition method for solving large instances of the bi-objective hub median and dispersion problem.

## 7. Conclusion

We studied the bi-criteria $p$-hub median and dispersion problem that arises in the strategic design of hub networks where the dispersion of hubs is desired to mitigate disruption risks. We presented a cutting plane algorithm for solving the problem and propose preprocessing to improve its convergence. The preprocessing step eliminates redundant variables of the original problem thereby reducing problem size and computational time. Our computational results on test instances from Turkish network, USA432 dataset and AP dataset, show that medium sized problem instances up to 60 nodes and 20 hubs can be solved efficiently using our strongest formulation and the cutting plane approach. For solving large problem instances, we presented a branch-and-cut based Benders decomposition with added improvements such as starting with a better solution and efficient ways of solving decomposed sub problem and adding Pareto optimal Benders cuts to the master problem. Computational results confirm that large sized problem instances up to 100 nodes and 20 hubs can be solved using the decomposition method efficiently. The research opens up lots of interesting avenues for future research. One such direction is the incorporation of dispersion as an objective in other hub location problems namely, $p$-hub center, $p$-hub covering and fixed charge hub location problems. Design of efficient solution approaches would clearly depend on the specific problem and its properties. Another promising direction is to explore the use of multi-objective evolutionary approaches (MOEAs) such as non-dominated sorting genetic algorithm II (NSGA-II) and strength Pareto evolutionary algorithm II (SPEA-II).

## Acknowledgments

The authors thank the editor and three anonymous reviewers for their valuable comments on a previous version of this paper. This research was supported by the National Science and Engineering Research Council of Canada (NSERC) grant to the second author.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2023.09.032.

## References

Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., & O'Kelly, M. E. (2021). Perspectives on modeling hub location problems. *European Journal of Operational Research*, *291*(1), 1–17.

Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, *190*(1), 1–21.

An, Y., Zhang, Y., & Zeng, B. (2015). The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research, Part B (Methodological)*, *77*, 103–122.

Azizi, N., Chauhan, S., Salhi, S., & Vidyarthi, N. (2016). The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. *Computers & Operations Research*, *65*, 174–188.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*(1), 238–252.

Campbell, J. F. (1992). Location and allocation for distribution systems with transshipments and transportion economies of scale. *Annals of Operations Research*, *40*(1), 77–99.

Campbell, J. F., & O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, *46*(2), 153–169.

Contreras, I. (2015). Hub location problems. In *Location science* (pp. 311–344). Springer International Publishing.

Contreras, I. (2021). Hub network design. In T. G. Crainic, M. Gendreau, & B. Gendron (Eds.), *Network design with applications to transportation and logistics* (pp. 567–598). Springer International Publishing.

Contreras, I., Cordeau, J.-F., & Laporte, G. (2011). Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, *59*(6), 1477–1490.

Contreras, I., Fernández, E., & Marín, A. (2010). The tree of hubs location problem. *European Journal of Operational Research*, *202*(2), 390–400.

Costa, M. G., Captivo, M. E., & Clímaco, J. (2008). Capacitated single allocation hub location problem—A bi-criteria approach. *Computers & Operations Research*, *35*(11), 3671–3695.

de Camargo, R. S., de Miranda, G., Jr., & Ferreira, R. P. (2011). A hybrid outer-approximation/benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, *39*(5), 329–337.

de Camargo, R. S., Miranda, G., Jr., & Luna, H. P. (2008). Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research*, *35*(4), 1047–1064.

de Camargo, R. S., de Miranda, G., Jr., & Luna, H. P. L. (2009). Benders decomposition for hub location problems with economies of scale. *Transportation Science*, *43*(1), 86–97.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, *91*, 201–213.

Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, *46*(1), 48–60.

Erkut, E., Ülküsal, Y., & Yenicerioğlu, O. (1994). A comparison of p-dispersion heuristics. *Computers & Operations Research*, *21*(10), 1103–1113.

Fallah-Tafti, M., Honarvar, M., Tavakkoli-Moghaddam, R., & Sadegheih, A. (2022). Mathematical modeling of a bi-objective hub location-routing problem for rapid transit networks. *RAIRO-Operations Research*, *56*(5), 3733–3763.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, *64*(4), 1096–1109.

Farahani, R. Z., SteadieSeifi, M., & Asgari, N. (2010). Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, *34*(7), 1689–1709.

Fortz, B., & Poss, M. (2009). An improved benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, *37*(5), 359–364.

Ghaffarinasab, N. (2020). A tabu search heuristic for the bi-objective star hub location problem. *International Journal of Management Science and Engineering Management*, *15*(3), 213–225.

Ghaffarinasab, N., & Atayi, R. (2017). An implicit enumeration algorithm for the hub interdiction median problem with fortification. *European Journal of Operational Research*, *267*(1), 23–39.

Ghaffarinasab, N., & Motallebzadeh, A. (2017). Hub interdiction problem variants: Models and metaheuristic solution algorithms. *European Journal of Operational Research*, *267*(2), 496–512.

Hamacher, H. W., Labbé, M., Nickel, S., & Sonneborn, T. (2004). Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics*, *145*(1), 104–116.

Khodemani-Yazdi, M., Tavakkoli-Moghaddam, R., Bashiri, M., & Rahimi, Y. (2019). Solving a new bi-objective hierarchical hub location problem with an m/ m/ c queuing framework. *Engineering Applications of Artificial Intelligence*, *78*, 53–70.

Kim, H., & O'Kelly, M. E. (2009). Reliable p-hub location problems in telecommunication networks. *Geographical Analysis*, *41*(3), 283–306.

Köksalan, M., & Soylu, B. (2010). Bicriteria p-hub location problems and evolutionary algorithms. *INFORMS Journal on Computing*, *22*(4), 528–542.

Kuby, M. J. (1987). Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geographical Analysis*, *19*(4), 315–329.

Landete, M., Peiró, J., & Yaman, H. (2023). Formulations and valid inequalities for the capacitated dispersion problem. *Networks*, *81*(2), 294–315.

Lei, T. L. (2013). Identifying critical facilities in hub-and-spoke networks: A hub interdiction median problem. *Geographical Analysis*, *45*(2), 105–122.

Madani, S. R., Nookabadi, A. S., & Hejazi, S. R. (2018). A bi-objective, reliable single allocation p-hub maximal covering location problem: Mathematical formulation and solution approach. *Journal of Air Transport Management*, *68*, 118–136.

Magnanti, T. L., Mireault, P., & Wong, R. T. (1986). *Tailoring Benders decomposition for uncapacitated network design*. Springer.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, *29*(3), 464–484.

McDaniel, D., & Devine, M. (1977). A modified benders' partitioning algorithm for mixed integer programming. *Management Science*, *24*(3), 312–319.

Mohammadi, M., Tavakkoli-Moghaddam, R., Siadat, A., & Rahimi, Y. (2016). A game-based meta-heuristic for a fuzzy bi-objective reliable hub location problem. *Engineering Applications of Artificial Intelligence*, *50*, 1–19.

Monemi, R., Gelareh, S., Nagih, A., & Jones, D. (2021). Bi-objective load balancing multiple allocation hub location: A compromise programming approach. *Annals of Operations Research*, *296*, 363–406.

O'Kelly, M. E. (1986). Activity levels at hub facilities in interacting networks. *Geographical Analysis*, *18*(4), 343–356.

O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, *20*(2), 92–106.

O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, *32*(3), 393–404.

O'Kelly, M. E., & Kim, H. (2007). Survivability of commercial backbones with peering: A case study of Korean networks. In A. T. Murray, & T. H. Grubesic (Eds.), *Critical infrastructure: Reliability and vulnerability* (pp. 107–128). Berlin, Heidelberg: Springer Berlin Heidelberg.

Parvaresh, F., Golpayegany, S. H., Husseini, S. M., & Karimi, B. (2013). Solving the p-hub median problem under intentional disruptions using simulated annealing. *Networks and Spatial Economics*, *13*(4), 445–470.

Parvaresh, F., Husseini, S. M., Golpayegany, S. H., & Karimi, B. (2014). Hub network design problem in the presence of disruptions. *Journal of Intelligent Manufacturing*, *25*(4), 755–774.

Pisinger, D. (2006). Upper bounds and exact algorithms for p-dispersion problems. *Computers & Operations Research*, *33*(5), 1380–1398.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817.

Ramamoorthy, P., Jayaswal, S., Sinha, A., & Vidyarthi, N. (2018). Multiple allocation hub interdiction and protection problems: Model formulations and solution approaches. *European Journal of Operational Research*, *270*(1), 230–245.

de Sá, E. M., Morabito, R., & de Camargo, R. S. (2018). Benders decomposition applied to a robust multiple allocation incomplete hub location problem. *Computers & Operations Research*, *89*, 31–50.

Sayah, D., & Irnich, S. (2017). A new compact formulation for the discrete p-dispersion problem. *European Journal of Operational Research*, *256*(1), 62–67.

Sayyady, F., & Fathi, Y. (2016). An integer programming approach for solving the p-dispersion problem. *European Journal of Operational Research*, *253*(1), 216–225.

Sayyady, F., Tutunchi, G. K., & Fathi, Y. (2015). P-median and p-dispersion problems: A bi-criteria analysis. *Computers & Operations Research*, *61*, 46–55.

Shang, X., Yang, K., Jia, B., Gao, Z., & Ji, H. (2021). Heuristic algorithms for the bi-objective hierarchical multimodal hub location problem in cargo delivery systems. *Applied Mathematical Modelling*, *91*, 412–437.

Tutunchi, G. K., & Fathi, Y. (2019). Effective methods for solving the bi-criteria p-center and p-dispersion problem. *Computers & Operations Research*, *101*, 43–54.

Zetina, C. A., Contreras, I., & Jayaswal, S. (2021). An exact algorithm for large-scale non-convex quadratic facility location. arXiv preprint arXiv:2107.09746.