CIRRELT

CIRRELT-2025-40

The *p*-median problem with externalities

Guilherme O. Chagas
Leandro C. Coelho
Gilbert Laporte

November 2025

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

The *p*-median problem with externalities

Guilherme O. Chagas[1*], Leandro C. Coelho[2], Gilbert Laporte[3,4]

1. CIRRELT and Faculté des Sciences de l'Administration, Université Laval, Québec, Canada
2. Canada Research Chair in Integrated Logistics, Université Laval, Québec, Canada
3. CIRRELT and HEC Montréal, Montréal, Canada
4. University of Bath, Bath, United Kingdom

Abstract: We introduce, model, and solve the p-median problem with externalities (p-MPE), a generalization of the classic p-median problem (p-MP) that accounts for externalities such as congestion and pollution. Unlike the traditional p-MP, which assigns users to facilities via shortest paths and ignores how many users share each arc, in the p-MPE, both facility locations and user-to-facility paths are selected at the same time while penalizing repeated arc usage. To handle this problem, we introduce and compare five mixed-integer linear programs. We also propose valid inequalities and develop two algorithmic enhancements: a MIP start procedure that combines a p-MP heuristic with a min-cost flow method on an augmented graph, and a solution-improvement algorithm based on the same core idea. Computational experiments on 135 classic and modified p-MP instances involving up to 900 vertices show that the best of our models consistently yields good quality solutions within reasonable computing times, proving optimality in nearly half of the cases within one hour. These results demonstrate that explicitly accounting for user-facility shortest paths and externalities is computationally viable, and solutions in this context can differ from p-MP solutions.

Keywords: Facility location, p-median problem, congestion and pollution externalities, path-dependent costs, mixed-integer linear programming.

Acknowledgements: This work was partly supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) [grant number 2025-03964]. We thank the Digital Research Alliance of Canada for providing high-performance computing resources.

---

* Corresponding author: guilherme.oliveira-chagas.1@ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2025

1. Introduction

We introduce the *p-median problem with externalities* (p-MPE) which extends the classic *p-median problem* (p-MP) by accounting for externalities such as congestion and pollution. We first describe the p-MP introduced by Hakimi (1964, 1965), and which aims to select, from a set of potential sites, the locations of p facilities in such a way that the total travel cost (or distance) between users and their nearest facility is minimized. This problem is known to be $\mathcal{NP}$-hard (Kariv and Hakimi, 1979).

The p-MP can be defined on a connected, undirected, and weighted graph $G = (V, E)$, where V is the vertex set, $|V| = n$, and E is the set of edges $[i, j]$, with $i, j \in V$, $i < j$. By convention, $[i, j]$ must be interpreted as $[j, i]$ if $i > j$. Each vertex $i \in V$ has an associated demand d_i, and each edge $[i, j] \in E$ has an associated cost c_{ij}. For modeling purposes, we also define the *directed arc set* corresponding to E as $A = \{(i, j) \mid [i, j] \in E\} \cup \{(j, i) \mid [i, j] \in E\}$ with $c_{ij} = c_{ji}$. Accordingly, we define the *directed graph* $D = (V, A)$, where $|A| = m = 2|E|$.

In the classic integer linear program (ILP) (ReVelle and Swain, 1970; Marín and Pelegrín, 2019) for the p-MP, the set of vertices V contains two subsets I and J, where I represents the users and J is the set of potential facility locations. Here, we assume that $I = J = V$, as is the case of the OR-Library p-MP instances of Beasley (1990). Let $\mathcal{A}_{ij} \subseteq A$ be the set of arcs composing a least-cost path from user $i \in V$ to location $j \in V$ and let $c'_{ij} = \sum_{(k,l) \in \mathcal{A}_{ij}} c_{kl}$ be the cost of this path.

Hakimi (1964) showed that there always exists an optimal p-MP solution in which all facilities are located at vertices of G. This is commonly known as the Hakimi property. To model the p-MP, we define the following decision variables:

- y_j: a binary variable equal to one if and only if a facility is opened at vertex $j \in V$;
- x_{ij}: a binary variable equal to one if and only if user $i \in V$ is assigned to facility $j \in V$.

The classic ILP for the p-MP is then as follows:

$$\text{minimize} \sum_{i \in V} \sum_{j \in V} d_i c'_{ij} x_{ij} \tag{1a}$$

subject to

$$\sum_{j \in V} x_{ij} = 1 \qquad i \in V \tag{1b}$$

$$x_{ij} \leq y_j \qquad i, j \in V \tag{1c}$$

$$\sum_{j \in V} y_j = p \tag{1d}$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V \tag{1e}$$

$$y_j \in \{0,1\} \qquad j \in V. \tag{1f}$$

The objective function (1a) minimizes the total weighted cost. Constraints (1b) ensure that each user i is assigned to one facility, and user i can only be assigned to a facility $j \in V$ if this facility is open, as per constraints (1c). Constraint (1d) ensures that p facilities are selected. Variables x_{ij} and y_j are binary as in (1e) and (1f), respectively.

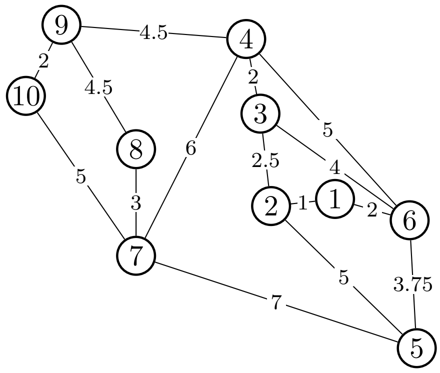
A key assumption in the p-MP is that users always take a least-cost path to their assigned facility, disregarding externalities, such as network congestion and air pollution. We extend the p-MP by accounting for such externalities. Hence, we propose the p-MPE, which aims to minimize the total combined cost of (i) user travel distances and (ii) externalities associated with traffic along the paths connecting users to facilities. This means that the p-MP is a special case of the p-MPE which is therefore $\mathcal{NP}$-hard.

The p-MPE arises, for example, in the location of primary schools in contexts where children are driven by their parents. This creates traffic on the network, particularly near the schools where many driving paths converge. As a result, while it may be desirable to live near a school in terms of travel distance, traffic congestion and pollution may be heavy around the school and render it undesirable.

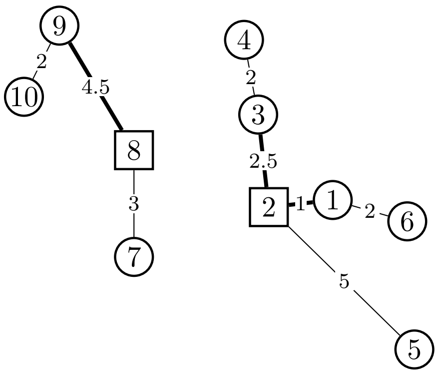
Figure 1 illustrates the difference between the p-MP and the p-MPE. Consider the sparse, undirected, weighted graph G with 10 vertices of Figure 1a. An example of a p-MP solution on G is shown in Figure 1b, where only the edges used by users appear. In this solution, vertices 2 and 8 are selected as facilities, and users are assigned to their nearest facility via shortest paths. Note that edges $[8,9]$, $[2,3]$, and $[1,2]$ are each used twice by users to reach facilities, and are highlighted in bold. For instance, users 9 and 10 both traverse edge $[8,9]$ to reach facility 8. The cost of this p-MP solution is 30, which is the total cost of shortest paths connecting users to facilities.

If we impose a penalty for externalities on the edges that are used, we may obtain the p-MPE solution illustrated in Figure 1c. In this example, we consider that the penalty is given by the function $f_{kl}(r) = c_{kl} r^2$, where r is the number of times edge $[k,l]$ is used. In the solution depicted in Figure 1c, vertex 7 is selected to locate a facility instead of vertex 8 (as in Figure 1b). The reason is that user 4 can now be assigned to facility 7, then reducing the load on edge $[2,3]$, hence reducing the externalities. In the p-MPE solution, only the edges $[7,8]$ and $[1,2]$ are used twice (in bold). The solution favors using edge $[7,8]$ twice

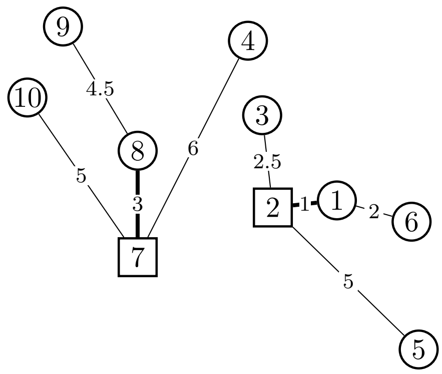
Figure 1: Illustration of different solutions for the p-MP and the p-MPE.



(a) Undirected graph G with $n = 10$.



(b) p-MP solution example ($p = 2$), with cost 30.



(c) p-MPE solution example ($p = 2$), with cost 74, of which 33 is the non-optimal p-MP cost.

instead of $[8, 9]$, because $[7, 8]$ is shorter and therefore incurs a smaller penalty. The cost of this p-MPE solution is 74, where 33 is related to the paths users take to reach their facilities (equivalent to a non-optimal p-MP solution) and 41 is related to penalties incurred by the externalities. On the other hand, if we take the p-MP solution of Figure 1b and compute its equivalent p-MPE cost, it is $30 + 46 = 76$.

The remainder of this paper is organized as follows. We present a literature review in Section 2. In Section 3 we show that the Hakimi property holds for the p-MPE, we model this problem by means of five compact mixed-integer linear programming (MILP) formulations, and we propose valid inequalities. In Section 4 we describe an initialization procedure and an improvement heuristic. In Section 5, we empirically compare these models through extensive computational experiments on 135 instances involving up to 900 vertices, and derived from classic OR-Library p-MP instances (Beasley, 1990). Conclusions follow in Section 6.

2. Literature review

The p-MP has been extensively studied since its introduction by Hakimi (1964, 1965), and numerous exact and heuristics algorithms have been proposed for this problem. For a recent survey, we refer the reader to Marín and Pelegrín (2019). Benchmark datasets introduced by Beasley (1990) have become a common basis for comparing p-MP algorithms.

Researchers have long recognized that real-world facility location often involves *congestion effects* or other externalities, which are ignored in the basic p-MP. A well-studied extension is the *congested facility location problem*, in which facilities have limited service capacity and customers may experience queuing delays at busy facilities (Desrochers et al., 1995; Marianov and ReVelle, 1996). From a different angle on congestion in facility location problems, Melkote and Daskin (2001) introduced the *capacitated facility location/network design* problem. Instead of relocating facilities, their model allows the creation of new connections to redirect demand away from congested facilities. A comprehensive review by Berman and Krass (2019) summarizes many of these models, showing how stochastic demand and congestion lead to more realistic but more extensive location-allocation models. In all these cases, however, congestion is associated with the facilities themselves rather than with the paths traversed by the users.

A related research stream considers *congestion in transportation networks* integrated with facility location. In many applications, especially in urban logistics, travel time depends on traffic levels and network capacity. Among some relevant problems in this context, we can highlight time-dependent routing problems, in which the time taken by a vehicle to reach a user depends on the period of the time horizon (Malandraki and Daskin, 1992). This variation on the travel time is often related to congestion and, in some models, to random disruptions such as incidents or weather, which can be dealt with in different ways (Gendreau et al., 2015). Schmidt et al. (2019) provide another example of congestion in routing problems integrated with facility location. These authors studied a time-dependent location-routing problem that selects a depot location and routes vehicles from this depot while minimizing travel times over a time horizon that captures congestion effects.

Yet another problem is the *multiple shortest path problem with path deconfliction* (Hughes et al., 2021), whose goal is to compute a path for each origin-destination pair on the same directed graph while minimizing both travel distance and penalties for reusing arcs. The key difference between this problem and the p-MPE is that, in the deconfliction problem, the destination nodes are fixed and known in advance, whereas in our case they are decision variables. Consequently, p-MPE generalizes the deconfliction setting problem and is therefore more challenging.

Closer to our work are models that incorporate *link congestion or flow-dependent costs* in location-allocation decisions. In transportation science, this is akin to moving from a user-optimal to a system-optimal assignment, where total travel time is minimized by internalizing congestion externalities on each path (Sheffi, 1985). For example, Berman and Larson (1982) dealt with the median problem with congestion. In their context, congestion occurs within facilities because they have a limited number of servers and requests arrive randomly. Since the nearest facility may be busy, users can then be dispatched from the next closest available facility. These authors showed that ignoring congestion may yield poor quality solutions.

Another example is the work of Bayram et al. (2023) who introduced the *hub network design problem with congestion, capacity, and uncertainty in demands*. In this problem, the goal is to minimize the cost related to capacity acquisition, hub congestion, and transportation. The authors showed that accounting for congestion and uncertainty changes the optimal hub location, capacity decisions, and routing compared with a congestion-free setting. Still in the context hub location, Hu and Hu (2025) studied a capacitated single-hub location problem with congestion on the hub but not on the paths, as in the p-MPE. Other studies introduced multi-objective models balancing efficiency and congestion, e.g., minimizing total distance while limiting maximum arc utilization (Yang and Bell, 1998).

To our knowledge, no prior work explicitly integrates path-specific congestion costs into the p-MP framework. The p-MPE proposed here fills this gap by generalizing the p-MP to include externality costs on arcs used by multiple users. The p-MPE penalizes overused road segments, considering, for example, congestion and pollution impacts of shared travel paths. We couple facility location with multi-commodity flow assignment and introduce new computational challenges. The next section presents five compact mixed-integer programming models for this problem.

3. Mathematical models

We now present five compact models for the p-MPE. The aim in each model is to select p facilities and minimize the total cost of assigning users to facilities, which consists of the path length plus penalties for using arcs multiple times. To capture the penalty incurred when an arc $(k, l) \in A$ is traversed by one user or more, we introduce a function $f_{kl}(r)$, where r denotes the number of users whose paths include arc (k, l). This function is monotonic and superlinear, i.e, it grows at least as fast as a linear function (e.g., $f_{kl}(r) = c_{kl}r^2$) and accounts for the additional cost of arc usage, associated with, for example, congestion or pollution. For consistency, we also account for the penalty when $r = 1$, since even a single user may generate externalities.

For a vertex $k \in V$, let $\delta^-(k) = \{(l,k) \in A\}$ and $\delta^+(k) = \{(k,l) \in A\}$ be the incoming and outgoing sets of arcs of k, respectively. We assume that the number p of facilities ranges between 1 and n. Also, let $q = n - p$ and let the directed graph density be denoted by $\mathcal{D} = m/(n^2 - n)$. Before presenting the models, we show that the Hakimi property holds for the p-MPE. Subsequently, we introduce a four-index model for the p-MPE in Section 3.2. We then present four three-index models in Sections 3.3, 3.4, 3.5, and 3.6. A brief discussion on the number of variables and constraints of the models is provided in Section 3.7.

At this point, it is appropriate to comment on the meaning of d_i in the p-MPE. The standard interpretation of d_i in the p-MP is a demand, i.e., the number of users located at vertex i, and it is assumed that all these users follow the same path to reach their assigned facility. However, this does not work for the p-MPE because the goal of this problem is to spread traffic across the arcs. In the context of the p-MPE, it makes more sense to interpret vertex demands as vehicles and to assume that vertices are replicated as often as necessary to achieve a unit demand at each vertex. In the context of the p-MPE, the coefficient d_i is best interpreted as a weight associated with the level of nuisance created by the vehicle located at vertex i. However, the interpretation of d_i does not affect the structure of the models.

3.1. Hakimi property for the p-MPE

Theorem 1. *The p-MPE has an optimal solution with all facilities located at vertices.*

Proof. Consider an undirected and weighted graph $G = (V, E)$. Let $k, l \in V$ such that edge $[k, l] \in E$ with $c_{kl} > 0$. Additionally, let $f_{kl}(r) = c_{kl} f(r)$ where $f(r)$ is any non-decreasing, monotonic, and superlinear function of r. Assume that in an optimal solution there is a facility located at k and there are $r + r'$ users assigned to k, where $r > 0$ and $r' \geq 0$. Of these users, r reach k by traversing edge $[k, l]$ (equivalently, arc (l, k)), while the remaining r' reach k via edges other than $[k, l]$, i.e., along arcs in $\delta(k)^- \setminus \{(l, k)\}$. The contribution to the p-MPE objective function from edge $[k, l]$ under this solution is

$$c_{kl} r + f_{kl}(r) = c_{kl} r + c_{kl} f(r).$$

Let us now add a new vertex x between vertices k and l, thus splitting the edge $[k, l]$ into $[k, x]$ and $[x, l]$ with $c_{kx}, c_{xl} > 0$ such that $c_{kl} = c_{kx} + c_{xl}$. Consider moving the facility from k to x. Then

- the r users who previously traversed edge $[k, l]$ (arc (l, k)) to reach facility k will now traverse edge $[x, l]$ (arc (l, x)) to reach the new facility at x. This new path utilizes the edge $[x, l]$, and its contribution to the objective function is $c_{xl}r + c_{xl}f(r)$;
- the r' users who previously arrived at k via $\delta(k)^- \setminus \{(l, k)\}$, plus the user located at k itself, now traverse $[k, x]$ to reach the facility at x. Thus $[k, x]$ carries $r' + 1$ units, generating a cost $c_{kx}(r' + 1) + c_{kx}f(r' + 1)$.

Therefore, the new total contribution to the p-MPE objective function over the split edges is

$$c_{xl}r + c_{xl}f(r) + c_{kx}(r' + 1) + c_{kx}f(r' + 1).$$

For the move to x to strictly improve the objective function compared to having the facility at k, we would need

$$c_{xl}r + c_{xl}f(r) + c_{kx}(r' + 1) + c_{kx}f(r' + 1) < c_{kl}r + c_{kl}f(r). \tag{2}$$

Using $c_{kl} = c_{kx} + c_{xl}$, inequality (2) simplifies to

$$c_{kx}(r' + 1 + f(r' + 1) - r - f(r)) < 0. \tag{3}$$

Then, either

$$c_{kx} < 0 \quad \text{or} \quad r' + 1 + f(r' + 1) < r + f(r).$$

However, $c_{kx} > 0$ by definition. Then, from (3) we have

$$r' + 1 < r, \tag{4}$$

since $f(\cdot)$ is monotonic and superlinear. Then, a facility located at x anywhere between k and l can only be in an optimal solution if the inequality (4) is respected. However, we will show that, when the facility lies on x it cannot be optimally located. Note that for a constant pair $(r, r' + 1)$ with $r \geq 0$, $r' + 1 \geq 0$, the expression

$$c_{xl}r + c_{xl}f(r) + c_{kx}(r' + 1) + c_{kx}f(r' + 1)$$

is affine in (c_{kx}, c_{xl}) with $c_{kx} + c_{xl} = c_{kl}$ and $c_{kx}, c_{xl} > 0$. That is, it is a linear equation in the variables c_{kx} and c_{xl}. If $r > r' + 1$, which is the only case possible as shown previously, then $f(r) > f(r' + 1)$ and the expression is minimized by taking $c_{xl} = 0$ (that is, locating the facility at k). If $r < r' + 1$, this expression is minimized by taking $c_{kx} = 0$ (locating the facility at l). If $r = r' + 1$, its value is constant over the edge, so no point between k and l is better than locating the facility at the vertices k or l. Then, in all these cases, the optimal cost is obtained at the vertices k or l. Therefore, the Hakimi property holds for the p-MPE. □

3.2. p-MPE four-index model

A straightforward way of formulating the p-MPE is to emulate that of the p-MP, by assigning user i to facility j while at the same time also determining the path linking them. The idea is to have a copy of graph D for each pair of user-facility $i, j \in V$. This way, it is possible to check first whether user i is assigned to facility j and, if so, which path is used. We then count the number of times each arc $(k, l) \in A$ is used and compute the total penalty. Our four-index model uses the following binary decision variables:

- y_j: binary variable equal to one if and only if vertex $j \in V$ is selected to locate a facility;
- w^r_{kl}: binary variable equal to one if and only if edge $[k, l] \in E$ is traversed r times, where $r = 0, \dots, q$;
- x^{ij}_{kl}: binary variable equal to one if and only if user i is assigned to facility j and, if so, whether arc $(k, l) \in A \setminus \delta^-(i)$ is used in the path connecting vertices i and j. Variables x^{ij}_{kl} are not activated when $i = l$, i.e., for the incoming arcs of $i \in V$, as a user only leaves its initial position once and never returns to it.

Note that the index r of the variables w^r_{kl} ranges from 0 to q. This is because, as in the p-MP, in any optimal solution of p-MPE each of the p facilities serves itself. Then, at most $q = n - p$ users may traverse a given arc. The p-MPE four-index model `M0` is as follows:

$$(\texttt{M0}) \text{ minimize} \sum_{i \in V} \sum_{j \in V} \sum_{(k,l) \in A} d_i c_{kl} x^{ij}_{kl} + \sum_{r=1}^{q} \sum_{\substack{(k,l) \in A, \\ l > k}} f_{kl}(r) w^r_{kl} \tag{5a}$$

subject to

$$\sum_{j \in V} \sum_{(k,j) \in \delta^-(j)} x^{ij}_{kj} = 1 \qquad i \in V \tag{5b}$$

$$x^{ij}_{kl} \le y_j \qquad i, j \in V, (k,l) \in A \tag{5c}$$

$$\sum_{j \in V} y_j = p \tag{5d}$$

$$\sum_{(i,l) \in \delta^+(i)} x^{ij}_{il} = \sum_{(l,j) \in \delta^-(j)} x^{ij}_{lj} \qquad i, j \in V \tag{5e}$$

$$\sum_{(l,k)\in\delta^-(k)} x_{lk}^{ij} - \sum_{(k,l)\in\delta^+(k)} x_{kl}^{ij} = \begin{cases} \sum_{(l,j)\in\delta^-(j)} x_{lj}^{ij}, & \text{if } k = j, \\ 0, & \text{otherwise.} \end{cases} \quad i,j,k \in V \quad (5f)$$

$$\sum_{r=0}^{q} w_{kl}^r \leq 1 \quad (k,l) \in A, l > k \quad (5g)$$

$$\sum_{i\in V}\sum_{j\in V} x_{kl}^{ij} + x_{lk}^{ij} \leq \sum_{r=0}^{q} r w_{kl}^r \quad (k,l) \in A, l > k \quad (5h)$$

$$w_{kl}^r \in \{0,1\} \quad r = 0,\ldots,q, \; (k,l) \in A, l > k \quad (5i)$$

$$y_j \in \{0,1\} \quad j \in V \quad (5j)$$

$$x_{kl}^{ij} \in \{0,1\} \quad i,j \in V, \; (k,l) \in A \setminus \delta^-(i). \quad (5k)$$

Objective function (5a) minimizes the total cost associated with the path connecting each user to its assigned facility, weighted by $f_{kl}(r)$, which is a function of the number of times r arc (k,l) is traversed. Constraints (5b) ensure that each user $i \in V$ is assigned to a facility $j \in V$, and this assignment is done by means of an arc $(k,j) \in \delta^-(j)$ of the graph of the pair user-facility i and j. A user can only be assigned to an open facility, which is guaranteed by constraints (5c). Constraint (5d) ensures that p facilities are selected. Constraints (5e) and (5f) define the flow conservation conditions and are used to find the path connecting user i and facility j. Constraints (5g) mean that at most one variable w_{kl}^r, with $r = 0,\ldots,q$, is used to count how many times arc (k,l) (or (l,k)) is traversed. Constraints (5h) count the number of times each arc is used. In these constraints, both arcs (k,l) and (l,k) appear on the left-hand side because, in an optimal solution, only one of them can be used. Constraints (5i), (5j), and (5k) define the domains of the variables.

Clearly, many of the variables of model (5a)–(5k) are redundant and can be eliminated. This motivates the alternative models presented in the following sections.

3.3. p-MPE three-index model

We now present the first of four three-index models, which we refer to as `M1`. This model is directly derived from `M0`. Instead of having flow variables for each pair of users and facilities candidates, we have only for each user vertex. So, we have a copy of graph D for each $i \in V$, where the facility $j \in V$ to which $i \in V$ is assigned is determined by the last vertex on the path rooted at i. In `M1`, variables y_i and w_{kl}^r are defined as in `M0`, and we also use the following decision variables:

- x^i_{kl}: binary variable equal to one if and only if arc $(k,l) \in A \setminus \delta^-(i)$ is contained in the path connecting user $i \in V$ to its facility.

The p-MPE three-index MILP model is presented next:

$$\texttt{(M1)}\ \text{minimize} \sum_{i \in V} \sum_{(k,l) \in A} d_i c_{kl} x^i_{kl} + \sum_{r=1}^{q} \sum_{\substack{(k,l) \in A, \\ l > k}} f_{kl}(r) w^r_{kl} \tag{6a}$$

subject to

$$\sum_{(i,l) \in \delta^+(i)} x^i_{il} = 1 \qquad i \in V \tag{6b}$$

$$\sum_{(l,k) \in \delta^-(k)} x^i_{lk} - \sum_{(k,l) \in \delta^+(k)} x^i_{kl} = 0 \qquad i,k \in V, i \neq k \tag{6c}$$

$$\sum_{(l,j) \in \delta^-(j)} x^i_{lj} - \sum_{(j,l) \in \delta^+(j)} x^i_{jl} \leq y_j \qquad i,j \in V \tag{6d}$$

$$\sum_{j \in V} y_j = p \tag{6e}$$

$$\sum_{r=0}^{q} w^r_{kl} \leq 1 \qquad (k,l) \in A, l > k \tag{6f}$$

$$\sum_{i \in V} x^i_{kl} + x^i_{lk} \leq \sum_{r=0}^{q} r w^r_{kl} \qquad (k,l) \in A, l > k \tag{6g}$$

$$w^r_{kl} \in \{0,1\} \qquad r = 0,\dots,q, (k,l) \in A, l > k \tag{6h}$$

$$y_j \in \{0,1\} \qquad j \in V \tag{6i}$$

$$x^i_{kl} \in \{0,1\} \qquad i \in V, (k,l) \in A \setminus \delta^-(i). \tag{6j}$$

Objective function (6a) is similar to (5a), but in (6a) we make use of the three-index variables x^i_{kl} and z^i_{kj}. The flow conservation constraints are (6b), (6c), and (6d). The first set requires the outgoing flow from a user $i \in V$ to be one; that is, every user i is assigned a facility. The second set guarantees that if the flow reaches a user vertex $k \in V$, it must leave it. If vertex j is selected as a facility, then the difference between its incoming and outgoing flows is at most one. Otherwise, the difference between its incoming and outgoing flows must be zero. These requirements are imposed by constraints (6d) for each pair of vertices i and j. Constraint (6e) ensures that p facilities are selected. Constraints (6g) correspond to (5h). Finally, the domains of variables w^r_{kl}, y_j, and x^i_{kl} are defined by (6h), (6i), and (6j), respectively.

3.4. p-MPE three-index model with constraints on arc penalties

In the M2 model, we replace the three-index variables w^r_{kl} of the previous two models with the two-index variables w_{kl}. These variables represent the penalty cost of using arc $(k,l) \in A$, which is achieved by linking its cost function c_{kl} and arc usage x^i_{kl} as explained below. In addition, we remove constraints (6f) and (6g) (which amount to $m-n$ constraints) and add $(nm-n^2)/2$ new constraints. Unlike the previous models, where only the objective function depends on the penalty function, in M2 some constraints also depend on the penalty function, as we will present. The M2 model is as follows:

$$
\begin{aligned}
(\texttt{M2})\ \text{minimize} & \sum_{i\in V}\sum_{(k,l)\in A} d_i c_{kl} x^i_{kl} + \sum_{(k,l)\in A} w_{kl} && && \text{(7a)}\\
\text{subject to} & && && \\
& \sum_{(i,l)\in\delta^+(i)} x^i_{il} = 1 && i \in V && \text{(7b)}\\
& \sum_{(l,k)\in\delta^-(k)} x^i_{lk} - \sum_{(k,l)\in\delta^+(k)} x^i_{kl} = 0 && i,k \in V, i \neq k && \text{(7c)}\\
& \sum_{(l,j)\in\delta^-(j)} x^i_{lj} - \sum_{(j,l)\in\delta^+(j)} x^i_{jl} \le y_j && i,j \in V && \text{(7d)}\\
& \sum_{j\in V} y_j = p && && \text{(7e)}\\
& w_{kl} \ge L_{kl}(r) && r = 1,\dots,q, (k,l)\in A, l > k && \text{(7f)}\\
& 0 \le w_{kl} \le U_{kl} && (k,l)\in A, l>k && \text{(7g)}\\
& y_j \in \{0,1\} && j \in V && \text{(7h)}\\
& x^i_{kl} \in \{0,1\} && i\in V, (k,l)\in A\setminus\delta^-(i). && \text{(7i)}
\end{aligned}
$$

Objective function (7a) is similar to (6a), but in (7a) we make use of the two-index variables w_{kl}. Constraints (7b), (7c), (7d), and (7e) are the same as (6b), (6c), and (6d), (6e), respectively. Constraints (7f) set the lower bounds for the w_{kl} variables, ensuring that if arc (k,l) is used r times, then $w_{kl} \ge L_{kl}(r)$, where $L_{kl}(r)$ is a lower-bound function that depends on the penalty function. Similarly, constraints (7g) set the upper bounds for w_{kl} to U_{kl}, which is also penalty-function dependent Variables x^i_{kl} and y_j are binaries as in (7h) and (7i), respectively.

We can find a lower bound $L_{kl}(r)$ of constraints (7f) function based on the penalty function $f_{kl}(r)$. For example, consider $f_{kl}(r) = c_{kl}r^2$ and $q = 3$. We can obtain:

$$L_{kl}(1) = c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk}\Big)$$
$$L_{kl}(2) = c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk}\Big) + 2c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk} - 1\Big)$$
$$L_{kl}(3) = c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk}\Big) + 2c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk} - 1\Big) + 2c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk} - 2\Big).$$

In general, for any $1 \le r \le q$ we have:

$$w_{kl} \ge L_{kl}(r) = c_{kl}\Big(\sum_{i\in V} x^i_{kl} + x^i_{lk}\Big) + \sum_{h=2}^{r} 2c_{kl}\Big(\sum_{i\in V}(x^i_{kl} + x^i_{lk}) - h + 1\Big).$$

By simplifying the above expression, we obtain constraints (8).

$$w_{kl} \ge c_{kl}(r - r^2) + c_{kl}(2r-1)\sum_{i\in V} x^i_{kl} + x^i_{lk} \quad r = 1,\ldots,q, (k,l) \in A, l > k. \tag{8}$$

For the upper bound values of variables w_{kl}, we set $U_{kl} = c_{kl}q^2$. Additionally, these upper bounds can be further tightened as

$$w_{kl} \le (\min\{|V_{kl}|, n\})^2 c_{kl}y_k + \Bigg(\min\Bigg\{\sum_{\substack{(k,l')\in\delta^+(k)\\ l'\neq l}} |V_{kl'}|, n\Bigg\}\Bigg)^2 c_{kl}(1-y_k), \tag{9}$$

where $V_{kl} \subseteq V \setminus \{k\}$ is the set of vertices that are reachable from vertex l after removing vertex k from graph D.

Following the same idea, we can also set lower and upper bounds for the variables w_{kl}. For example, if we use a cubic penalty function $f_{kl}(r) = c_{kl}r^3$, we can impose the following constraints to define lower bounds on w_{kl}:

$$w_{kl} \ge c_{kl}\sum_{i\in V}\big(x^i_{kl} + x^i_{lk}\big) + c_{kl}\sum_{h=2}^{r} 6(h-1)\sum_{i\in V}\big(x^i_{kl} + x^i_{lk} - h + 1\big),\ r = 1,\ldots,q,$$
$$(k,l) \in A, l > k. \tag{10}$$

For the upper bounds in this cubic case, we set $U_{kl} = c_{kl}q^3$.

3.5. p-MPE three-index model with two-index flow variables

In this model, instead of creating a copy of the graph for each user vertex $i \in V$, we use a single graph in which the arcs can support up to q flow units. To model this condition, the three-index binary flow variables x^i_{kl} defined in (6j) and (7i) are replaced by two-index integer flow variables, defined as follows:

- z_{kl}: integer variable representing the number of users traversing arc (k, l).

To use these integer variables, we assume that the demands d_i are identical, which is the case of the instances used in this work. In addition, we reintroduce the assignment variables x_{ij}, defined in (1e), and use the variables y_i and w^r_{kl} defined in (1f) and (5i), respectively. The model `M3` is as follows:

$$\text{(M3) minimize} \sum_{(k,l)\in A} c_{kl} z_{kl} + \sum_{r=1}^{q} \sum_{(k,l)\in A} f_{kl}(r) w^r_{kl} \tag{11a}$$

subject to

$$\sum_{j\in V} x_{ij} = 1 \qquad i \in V \tag{11b}$$

$$x_{ij} \leq y_j \qquad i, j \in V \tag{11c}$$

$$\sum_{j\in V} y_j = p \tag{11d}$$

$$\sum_{r=0}^{q} w^r_{kl} \leq 1 \qquad (k, l) \in A, l > k \tag{11e}$$

$$z_{kl} + z_{lk} \leq \sum_{r=0}^{q} r w^r_{kl} \qquad (k, l) \in A, l > k \tag{11f}$$

$$\sum_{(j,l)\in\delta^+(j)} z_{jl} \leq q(1 - y_j) \qquad j \in V \tag{11g}$$

$$\sum_{(l,j)\in\delta^-(j)} z_{lj} + (1 - y_j) = \sum_{(j,l)\in\delta^+(j)} z_{jl} + \sum_{i\in V} x_{ij} \qquad j \in V \tag{11h}$$

$$y_j \in \{0, 1\} \qquad j \in V \tag{11i}$$

$$x_{ij} \in \{0, 1\} \qquad i, j \in V \tag{11j}$$

$$z_{kl} \in \{0, \ldots, q\} \qquad (k, l) \in A \tag{11k}$$

$$w^r_{kl} \in \{0, 1\} \qquad r = 0, \ldots, q, \quad (k, l) \in A, l > k. \tag{11l}$$

The objective function (11a) minimizes the total cost associated with the number of times each arc is traversed and the penalty costs. Constraints (11b)–(11d) are the same as

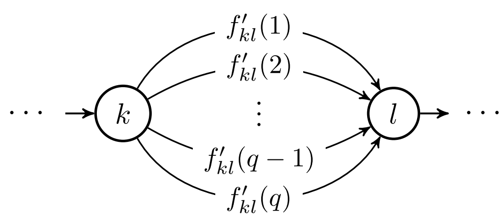
(1b)–(1d), and (11e) as (5g). Moreover, constraints (11f) are similar to (5g), but here we only use the variables z_{kl} and z_{lk} on the left-hand side as z_{kl} and z_{lk} give us the number of times arc (k, l) ((l, k)) is used. Constraints (11g) ensure that if facility j is chosen to be opened, then none of its outgoing arcs can be used. Constraints (11h) are modified degree constraints, which ensure that, if j is not a facility, then the incoming flow (number of users using j incoming arcs) plus the flow of user j must be equal to its outgoing flow (number of outgoing arcs of j). Note that, in this case, $\sum_{i\in V} x_{ij} = 0$ because of constraints (11c) since $y_j = 0$. On the other hand, if j is a facility, then we ensure that the incoming flow must be equal to the number of its assigned users, as $\sum_{(j,l)\in\delta^+(j)} z_{jl} = 0$ because of constraints (11g). The domain of the variables are defined by (11i)–(11l).

3.6. p-MPE three-index improved model

Based on the idea of model M3, we now present an improved compact model M4. In this model, we use binary variables y_i as defined in (1f), and we eliminate the assignment variables x_{ij} and the integer flow variables z_{kl}. As in the M3, in model M4 we also assume that demands d_i are identical. In addition, we introduce new decision variables x^r_{kl} for each arc $(k, l) \in A$ and $r = 1, \ldots, q$, which simultaneously represent both the flow carried from vertex k to vertex l through (k, l) (as in the variables x^i_{kl} of previous models) and the number of times the arc (k, l) is traversed (as in the variables w^r_{kl} of previous models).

Unlike the variables w^r_{kl} from the previous models, which are only defined for $(k, l) \in A$ with $l > k$, here the new variables x^r_{kl} are defined for all arcs in A, as they also represent the flow from k to l. These variables can be interpreted as representing q parallel arcs $(k, l)^r$, $r = 1, \ldots, q$, for each $(k, l) \in A$, each with unit capacity, where the cost of using arc $(k, l)^r$ is strictly increasing in r. Figure 2 illustrates these q parallel arcs between two vertices $k, l \in V$, along with their associated costs.

Figure 2: Parallel arcs from vertex k to vertex l.



The new decision variables are then defined as

- x^r_{kl}: binary variable equal to one if and only if parallel arc $(k, l)^r$ is used and zero otherwise.

To incorporate the idea of parallel arcs into model M4, we define marginal (incremental) costs $f'_{kl}(r)$ such that

$$c_{kl}r + f_{kl}(r) = \sum_{h=1}^{r} f'_{kl}(h), \;\; f'_{kl}(1) < f'_{kl}(2) < \cdots < f'_{kl}(r).$$

Since the p-MPE is a minimization problem, the parallel arc $(k,l)^r$ with $r > 1$ is only used if $(k,l)^{r-1}$ is already in use. For example, suppose $f_{kl}(r) = c_{kl}r^2$. When $r = 3$, we obtain

$$f'_{kl}(1) = 2c_{kl}, \qquad f'_{kl}(2) = 4c_{kl}, \qquad f'_{kl}(3) = 6c_{kl}.$$

In general, for every $1 \leq r \leq q$ and, for example, $f_{kl}(r) = c_{kl}r^2$, we define (similarly to constraints (8)):

$$f'_{kl}(r) = c_{kl}2r. \tag{12}$$

The model M4 is then

$$\text{(M4) minimize} \sum_{r=1}^{q} \sum_{(k,l)\in A} f'_{kl}(r)x^r_{kl} \tag{13a}$$

subject to

$$\sum_{r=1}^{q} \sum_{(k,i)\in\delta^-(i)} x^r_{ki} + (1 - y_i) \leq \sum_{r=1}^{q} \sum_{(i,l)\in\delta^+(i)} x^r_{il} + qy_i \qquad i \in V \tag{13b}$$

$$x^r_{il} \leq 1 - y_i \qquad r = 1,\ldots,q, i \in V, (i,l) \in \delta^+(i) \tag{13c}$$

$$\sum_{i\in V} y_i = p \tag{13d}$$

$$y_i \in \{0,1\} \qquad i \in V \tag{13e}$$

$$x^r_{kl} \in \{0,1\} \qquad r = 1,\ldots,q, (k,l) \in A. \tag{13f}$$

Similarly to the previous models, the objective function (13a) minimizes the total cost associated with the number of times each arc is traversed considering the arcs weights and penalty costs. Constraints (13b) ensure that the flow leaving a vertex $i \in V$ is the entry flow plus one unit if that node is not a facility ($y_i = 0$). If $y_i = 1$, then the in-flow is up to q since constraints (13b) impose that no out-flow is possible from a facility vertex. Exactly p facilities are selected by (13d) and variables y_i and x^r_{kl} are binary as per (13e) and (13f), respectively.

The following constraints are valid inequalities for the `M4` model and can help tighten its linear relaxation:

$$x_{kl}^{r} \leq x_{kl}^{r-1}, \ r = 2, \ldots, q, (k,l) \in A. \tag{14}$$

Constraints (14) are symmetry-breaking valid inequalities. They impose that the r-th parallel arc $(k,l)^r$ can be used only if for $r > 1$ the $(r-1)$-th one is already used.

$$\sum_{(k,i) \in \delta^-(i)} x_{ki}^{q} \leq y_i, \ i \in V. \tag{15}$$

If a vertex $i \in V$ is a facility, then at most q users can be assigned to it. However, if i is a user vertex, then at most $q-1$ units of flow can enter this vertex, since its own demand also contributes to the outflow through its outgoing arcs (as defined by constraints (13b)). Therefore, the valid inequalities (15) ensure that the variable x_{ki}^{q} can only take value 1 if i is a facility ($y_i = 1$).

$$\sum_{(k,i) \in \delta^-(i)} x_{ki}^{r} \leq \left\lfloor \frac{q}{r} \right\rfloor, \ r = \bar{r}, \ldots, q, \bar{r} = \left\lfloor \frac{q}{|\delta^-(i)|} \right\rfloor + 1, i \in V, |\delta^-(i)| > 1. \tag{16}$$

Following the idea of constraints (15), we know that the total flow across all parallel arcs of the ingoing arcs of a node i ($\delta^-(i)$), must be at most q. Moreover, the r-th parallel arc, with $r > 1$, can only be used if the corresponding $(r-1)$-th arc is already used. So, for a given r, there can be at most $\lfloor q/r \rfloor$ parallel arcs of order r active simultaneously.

For example, suppose $q = 6$ and $|\delta^-(i)| = 3$. Then at most one parallel arc with $r = 4$ can be used, since $\lfloor q/r \rfloor = 1$. In other words, no combination uses more than one "4-th" parallel arc among the three ingoing arcs of $\delta^-(i)$ because the maximum inflow is $q = 6$. In this case, if one "4-th" parallel arc is used, then either one "2-nd" parallel arc is also used (as $4 + 2 = 6$), or two "1-st" arcs are used (as $4 + 1 + 1 = 6$). Constraints (16) help to break this symmetry by imposing that at most $\lfloor q/r \rfloor$ variables x_{ki}^{r} from the ingoing arcs $\delta^-(i)$ may take value one.

$$\sum_{(l,i) \in \delta^-(i) \setminus (k,i)} x_{li}^{q-r+1} \leq (|\delta^-(i)| - 1)(1 - x_{ki}^{r}), \qquad r = 1, \ldots, q, i \in V, (k,i) \in \delta^-(i), \ |\delta^-(i)| > 1. \tag{17}$$

Following the idea of the previous valid inequalities, we can further improve symmetry breaking with constraints (17). These valid inequalities ensure that if the parallel arc $(k,i)^r$, with $(k,i) \in \delta^-(i)$, is used, then it is not possible to simultaneously use the $(q-r+1)$-st parallel arc of any other $(k,l) \in \delta^-(i) \setminus \{(k,i)\}$. For example, let $x^1_{ki} = 1$, meaning that one unit of flow enters vertex i through arc $(k,i) \in \delta^-(i)$. In this case, there is no feasible solution in which the q-th parallel arc of some $(k,l) \in \delta^-(i) \setminus \{(k,i)\}$ is also used.

3.7. Numbers of variables and constraints

Table 1 reports the number of variables and constraints for each model presented in the previous sections. In this table, we provide the asymptotic complexity of these quantities under two contrasting settings: a complete directed graph, i.e., $m = n^2 - n$, and a sparse graph, i.e., when $m \ll n^2$.

Table 1: Number of variables and constraints of the proposed models.

Model	Number of variables	Graph: Dense	Graph: Sparse
M0	$(n^2m - n^3 - nm - n^2 - pm + pn + m + n)/2$	$\Theta(n^4)$	$\Theta(n^2m)$
M1	$(2nm - 2n^2 - pm + pn + m + n)/2$	$\Theta(n^3)$	$\Theta(nm)$
M2	$(nm - n^2 + m + n)/2$	$\Theta(n^3)$	$\Theta(nm)$
M3	$(nm + n^2 - pm + pn + 3m + n)/2$	$\Theta(n^3)$	$\Theta(nm)$
M4	$nm - pm + n$	$\Theta(n^3)$	$\Theta(nm)$
	Number of constraints	**Dense**	**Sparse**
M0	$n^2m + n^3 + n^2 + m + 1$	$\Theta(n^4)$	$\Theta(n^2m)$
M1	$2n^2 + m - n + 1$	$\Theta(n^2)$	$\Theta(n^2)$
M2	$(nm + 3n^2 - pm + pn + 2m - 2n + 2)/2$	$\Theta(n^3)$	$\Theta(nm)$
M3	$n^2 + m + 2n + 1$	$\Theta(n^2)$	$\Theta(n^2)$
M4	$(nm - n^2 - pm + pn + 2n + 2)/2$	$\Theta(n^3)$	$\Theta(nm)$

As detailed in Section 5, we used the OR-Library (Beasley, 1990) instances in our experiments. These well-known instances consist of connected sparse graphs, with a graph density coefficient of $\mathcal{D} = 0.04$. Therefore, in these instances $m \ll n^2$.

4. Algorithmic details

In this section, we describe some implementation features of our solution methodology. In Section 4.1, we detail the procedure for finding an initial solution for our models. Section 4.2 presents a solution-improvement heuristic.

4.1. Initial solution

To generate an initial solution for all tested models, we first apply a parallel variable neighborhood search (VNS) heuristic (Chagas et al., 2024) to obtain a p-MP solution. Next, we take the set of p facilities identified by this heuristic and solve a minimum-cost network flow from a super-source representing all vertices of V to a super-sink representing these facilities. This problem can be solved by using the successive shortest path algorithm in $O(q(m + n \log n))$ time (Ahuja et al., 1993). In Section 4.2, we detail this procedure using a graph with parallel arcs, which is equivalent to solving our p-MPE for a given set of p facilities. Then, we use this solution to initialize the MILP associated with each model.

4.2. Improvement heuristic

Building on the graph-with-parallel-arcs idea of model `M4`, described in Section 3.6, we apply a solution-improvement heuristic. This heuristic is invoked whenever a new integer solution is found within the branch-and-bound framework of the commercial solver used in this work. We first take the p facilities from the incumbent integer solution. Because the solver may return a feasible but suboptimal solution, this incumbent is not necessarily an optimal p-MPE solution. We then attempt to improve it by computing an optimal p-MPE solution solving the problem described below.

Let $\mathcal{J}_p$ denote the set of all subsets of V of cardinality p. Given a set of p facilities $J^* \in \mathcal{J}_p$, we can compute its p-MPE cost by solving a min-cost network flow in an augmented graph D' of D. Let $D' = (V \cup \{s, t\}, A \cup A')$, where s and t are the source and sink vertices, respectively, and

$$A' = \{(s, i) \mid \forall i \in I\} \cup \{(j, t) \mid \forall j \in J^*\} \cup \{(k, l)^r \mid \forall (k, l) \in A, r = 2, \ldots, n\}.$$

In other words, A' consists of the arcs from the source vertex s to each user vertex $i \in V$, the arcs from each facility vertex $j \in J^*$ to the sink t, and q parallel arcs $(k, l)^r$ for each arc $(k, l) \in A$, as shown in Figure 2. Every arc connecting s to a vertex in I has a zero cost and a unit flow capacity, while every arc connecting a facility vertex in J^* to the sink t also has a zero cost and a flow capacity of q units. Moreover, each parallel arc $(k, l)^r \in A'$ has a flow capacity of one unit and a cost of $f'_{kl}(r)$. This cost function is detailed in Section 3.6.

With the graph D', we can use a minimum-cost network flow algorithm to compute in polynomial time the cost of sending q units of flow from s to t. This cost is equivalent to assigning users in V to facilities in J^* and thus yields the p-MPE solution cost for any set in $\mathcal{J}_p$. This solution-improvement heuristic is applied to every set of p facilities in each new integer solution found. We use a hash table to keep track of previously visited facility sets in $\mathcal{J}_p$, since the commercial solver may produce a solution with the same $J^* \subseteq \mathcal{J}_p$. After computing the optimal p-MPE solution, we pass it back to the solver by setting the new solution. To retrieve the incumbent integer solution within the branch-and-bound framework, and to submit the improved solution (if any), we use the solver's *callback* interface.

5. Computational experiments

In this section, we evaluate the performance of our models using the 40 p-MP instances from the OR-Library (Beasley, 1990). These instances are sparse, undirected, weighted, and connected graphs. In addition, $d_i = 1$ for all users $i \in V$. In the p-MP setting, researchers typically compute shortest paths between every pair of vertices to obtain a complete graph whose arc weights correspond to the lengths of the shortest paths between the respective vertex pairs. In our work, we consider both the 40 original sparse instances from this set, without altering the graphs, and modified variants. In total, we tested our models on 135 instances, comprising the following:

- 40 original p-MP instances from the OR-Library, with $n = 100, 200, \ldots, 900$ and $\mathcal{D} = 0.04$. The sizes of these instances and their corresponding p values are reported in Table A.9;
- 15 additional small instances with $n = 25$, 50, and 75 (also with $\mathcal{D} = 0.04$), derived from the first five OR-Library instances (`pmed1`–`pmed5`). Details of these instances are provided in Table A.8;
- 40 modified instances with $\mathcal{D} = 0.08$, derived from the original OR-Library instances without changing their n or p values;
- 40 modified instances with $\mathcal{D} = 0.4$, also derived from the original OR-Library instances without altering their n or p values.

All algorithms were implemented in *C*++ and compiled with the *g*++ compiler, version 12.3.1. We used Gurobi's *C*++ API v.12.0.0 to solve the linear programs and the Boost *C*++ Libraries implementation of the successive shortest path algorithm to solve the minimum-cost network flow problems. All tests were executed on a computer cluster equipped with AMD EPYC™ 7532 processors (32 threads at 2.4 GHz) and up to 512 GB of RAM. We ran all tests with a one-hour time limit. All instances derived from the original 40 OR-Library instances, along with the best solutions obtained by our models across all experiments, are available at https://github.com/Pigzaum/p-MPE.

In the tests presented in this section, we omit the results for the `M0` model described in Section 3.2 because it failed to find solutions in most instances, either due to insufficient memory or because the time limit was exceeded. For the other models M1, M2, M3, and M4, we evaluate performance by reporting the upper bound (UB), the lower bound (LB), the percentage optimality gap (%) defined as $100(UB - LB)/UB$, and the execution time in seconds (t (s)).

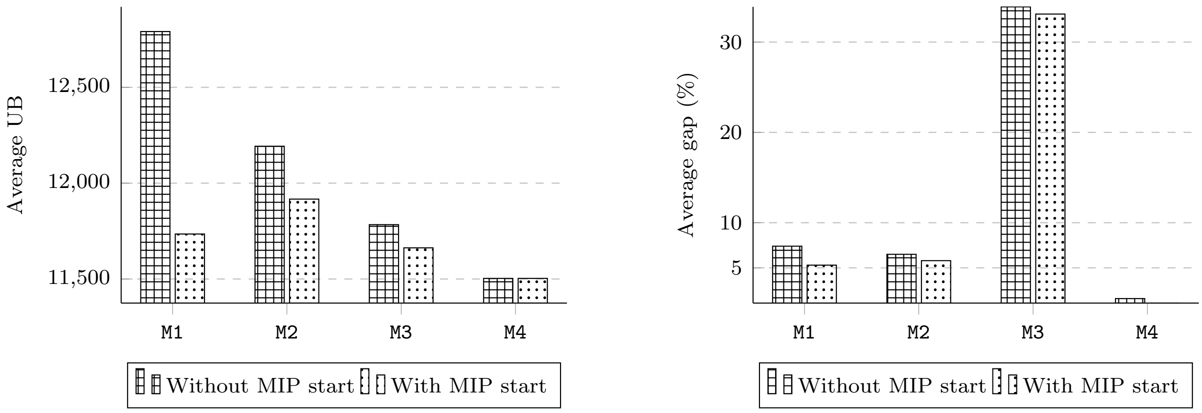
The remainder of this section is organized as follows. Section 5.1 presents preliminary tests performed to assess the models and the procedures described in Section 4. The results for the original and the derived OR-Library instances are shown in Section 5.2.

5.1. *Preliminary tests*

In this section, we present the results of preliminary tests to assess the initial solution algorithm (presented in Section 4.1), the solution improvement heuristic (presented in Section 4.2), and the valid inequalities proposed for M4. For these tests, we used the 10 first instances from the OR-Library (pmed1–pmed10), which range from 100 to 200 vertices. In all tests presented in this section, we consider $f_{kl}(r) = c_{kl}r^2$ as the penalty function for using r times arc (k, l) and, thus, we used constraints (8) in model M2.

Figure 3 summarizes the experiments evaluating the effect of the initial MIP start algorithm (Section 4.1) on the solution quality of all models. The left chart compares the average UB with and without the initial algorithm, and the right chart compares the average optimality gaps.

Figure 3: Impact of the MIP start algorithm on solution quality on the four models.



As shown in Figure 3, the initial solution provided by the MIP start algorithm improves the final solution quality across all models. For the UB, the only exception is M4, which finds an optimal solution on all 10 instances even without the initial solution. However, the initialization algorithm of Section 4.1 was able to slightly reduce the optimality gap for M4, since starting from a stronger incumbent helps the solver's branch-and-bound procedure converge faster. Based on these results, we then used the initialization algorithm for all models and all tests.

Figure 4 depicts the average UB (left chart) and the average gap (%) (right chart). These charts compare the execution of all models with and without the improvement heuristic of Section 4.2. Similar to the tests conducted for the initial solution algorithm, the improvement heuristic enhances the solution quality of all models. However, as seen by comparing Figures 3 and 4, the gains in terms of average UB from the heuristic of Section 4.2 are not as significant as those produced by the initial solution algorithm. This is because the improvement heuristic cannot always improve the solver's incumbent. On the other hand, the MIP start provided by the initial solution algorithm is consistently better than the solver's initial incumbent, helping it converge faster. Based on these results, we apply the improvement procedure to all models and all experiments.

Figure 4: Impact of the improvement heuristic on solution quality on the four models.

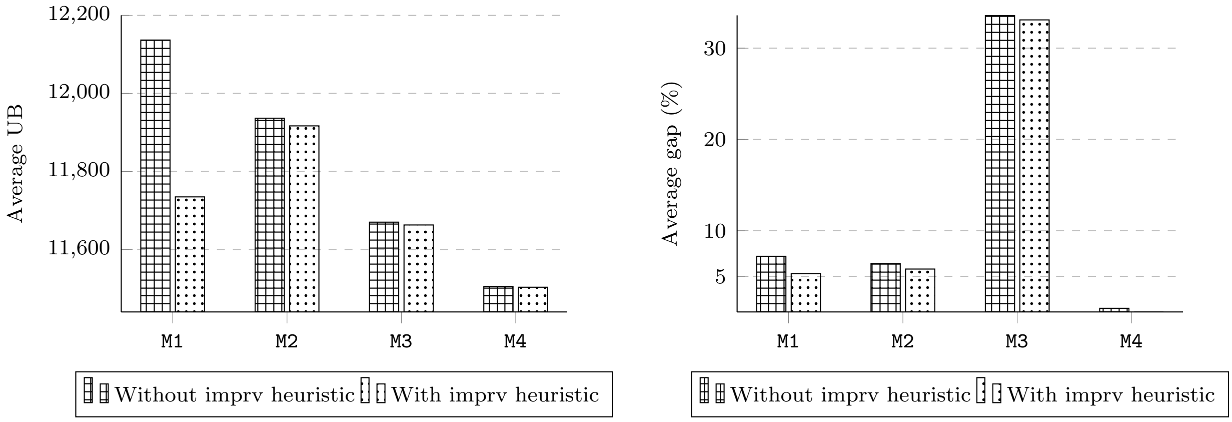
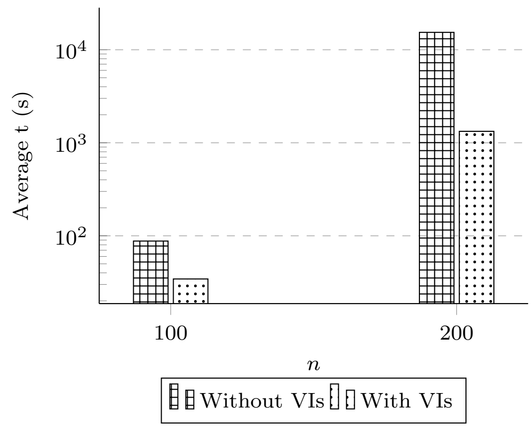


Figure 5 reports the average execution time (in seconds, logarithmic scale) for the M4 model with and without valid inequalities (14)–(17) (VIs). The 10 instances are partitioned by size, and average times are reported separately for the groups with 100 and 200 vertices. In addition to strengthening the linear relaxation bound, these constraints help Gurobi's presolve remove more variables. This leads to a faster convergence of the branch-and-bound algorithm, as observed in Figure 5, where the use of valid inequalities yields a substantial reduction in the average execution time.

5.2. Results

In what follows, Section 5.2.1 reports the results obtained on the sparse instances, while Section 5.2.2 presents those for denser instances. In both sections, we consider the penalty function $f_{kl}(r) = c_{kl}r^2$ and thus use constraints (8) in M2. Then, in Section 5.2.3, we

Figure 5: Comparison of the average execution time to solve `M4` with and without valid inequalities (14)–(17).



evaluate the models under an alternative arc-penalty function, $f_{kl}(r) = c_{kl}r^3$, and thus use constraints (10) in `M2`. Detailed results for all tests are provided in Appendix A.

5.2.1. Tests on the original instances

In this section, we present results on 55 sparse instances, partitioned into two sets. The first set comprises 15 small-size instances derived from the five first p-MP OR-Library instances (`pmed1`, `pmed2`, `pmed3`, `pmed4`, and `pmed5`) each originally having 100 vertices. For each of these five instances, we generated three variants with $n = 25, 50$, and 75, yielding 15 instances in total. The parameter p for each generated instance was scaled down proportionally to the original p and to the new number of vertices, whose values are presented in Table A.8. The second set consists of the original 40 p-MP OR-Library instances. All these instances have a graph density $\mathcal{D} = 0.04$.

Table 2 summarizes the results on the 15 small instances. It reports, for each model, the number of optimal solutions obtained (# opt), as well as the average upper bound (UB), lower bound (LB), percentage gap, and runtime (seconds). In addition, it shows, for each metric, the number of best results obtained. The column "Number of best t" gives the number of instances for which the solver proved optimality and did so with the lowest runtime (in seconds), strictly below the time limit.

In Table 2, all models obtained the optimal solution for every instance. In addition, all models except `M2` proved optimality for all instances. Model `M2` could not prove optimality for three of the 15 instances, with an average gap of 5.4%. Regarding execution times, model `M1` was, on average, slightly faster than `M3` and `M4`.

Table 3 shows a summary of the results on the 40 original OR-Library instances. This table follows the same layout as Table 2. The average results for `M2` are omitted from

Table 2: Summary of results on the small instances.

Model	# opt	Average results				Number of best			
		UB	LB	gap (%)	t (s)	UB	LB	gap	t
M1	15	10506.1	10506.1	0.0	7.8	15	15	15	12
M2	15	10506.1	9721.5	5.4	898.8	15	12	12	0
M3	15	10506.1	10506.1	0.0	9.2	15	15	15	2
M4	15	10506.1	10506.1	0.0	8.3	15	15	15	1

Table 3 because this model failed on some instances due to insufficient memory. So, we do not include them, as comparing its averages with those of the other models would not be fair.

Table 3: Summary of results on the OR-Library instances.

Model	# opt	Average results				Number of best			
		UB	LB	gap (%)	t (s)	UB	LB	gap	t
M1	9	32462.2	7102.3	36.2	2870.1	10	9	9	1
M2	8	-	-	-	-	8	8	8	0
M3	2	32015.3	2364.8	76.8	3468.3	5	2	2	0
M4	19	22006.9	12558.3	18.9	2056.9	40	40	40	18

As one can note from Table 3, model `M4` clearly outperformed the other models, achieving the smallest average UB, LB, optimality gap, and execution time. It found an optimal solution for 19 out of the 40 instances and obtained the best results in terms of UB, LB, and optimality gap on all instances tested.

Although all four models have, asymptotically, the same number of variables and constraints (as shown in Table 1), the hidden terms in the asymptotic order of `M4` are smaller than those of the other models, especially on sparse graphs, which is the setting of this section's experiments. In addition, the valid inequalities proposed for `M4` strengthen its linear relaxation and enhance the solver's presolve, which helps explain why `M4` dominated the other models in these tests.

5.2.2. Tests on denser instances

In this section, we present results on modified OR-Library p-MP instances. The 40 original instances were made denser by randomly adding arcs between vertices. We considered two density scenarios: (i) doubling the original density from $\mathcal{D} = 0.04$ to $\mathcal{D} = 0.08$ of all 40 instances; and (ii) increasing the density from $\mathcal{D} = 0.04$ to $\mathcal{D} = 0.40$ on all 40 instances. We then ran the four models on these two new sets of instances.

Table 4 presents the results for the modified instances with a graph density of 0.08. This table follows the same structure as those in Section 5.2.1. The average results for M2 are omitted from Table 4 because it failed on some instances due to insufficient memory.

Table 4: Summary of results on the modified instances with $\mathcal{D} = 0.08$.

Model	# opt	Average results				Number of best			
		UB	LB	gap (%)	t (s)	UB	LB	gap	t
M1	11	17881.9	3886.1	37.5	2734.8	11	15	12	4
M2	6	-	-	-	-	7	6	6	0
M3	2	17753.7	1108.6	78.3	3528.6	4	2	2	0
M4	17	12116.8	5431.7	25.5	2392.2	40	36	39	13

Similarly to the results presented in Section 5.2.1, model M4 outperformed the other models, achieving the best average results and the highest number of best results on every metric. Because the instances in Table 4 contain more arcs than those of the previous section, the models are larger, with more variables and constraints. This explains why fewer optimal solutions were obtained overall and why the execution times are higher compared with the results in Table 3.

Table 5 shows the results of tests with the modified instances of density 0.4. The average results for M1 and M2 are omitted because these models failed on some instances due to insufficient memory. As can be seen from Table 5, model M4 also outperformed the other models. Since these instances are denser than those from the previous tests, the models become larger. This is reflected in the increased average computational times for models M3 and M4, and in the fact that M1 failed to solve some instances due to insufficient memory. Moreover, the number of instances for which M4 proved optimality decreased from 17 to 15, as shown in Table 4. These results are consistent with the asymptotic growth in the number of variables and constraints shown in Table 1. The denser the instances, the more variables and constraints are required across all models.

Table 5: Summary of results on the modified instances with $\mathcal{D} = 0.4$.

Model	# opt	Average results				Number of best			
		UB	LB	gap (%)	t (s)	UB	LB	gap	t
M1	11	-	-	-	-	17	19	20	4
M2	4	-	-	-	-	4	4	4	0
M3	2	3900.2	216.1	80.8	3526.3	9	10	11	0
M4	15	3827.1	957.4	40.7	2537.6	40	37	37	11

Table 6 compares results by instance size across the three densities tested: the original instances with $\mathcal{D} = 0.04$ and the modified ones with $\mathcal{D} = 0.08$ and $\mathcal{D} = 0.40$. We report only the results for model M4, as it performed best among the models tested. In Table 6, the first column indicates the instance size, and the second column (denoted by #) gives the number of instances of that size. The UB and execution time values shown in this table are the average of the results of the respective instance size.

Table 6: Comparison of M4 results on large instances across three densities.

		$\mathcal{D} = 0.04$			$\mathcal{D} = 0.08$			$\mathcal{D} = 0.4$		
n	#	# opt	UB	t (s)	# opt	UB	t (s)	# opt	UB	t (s)
100	5	5	10102.0	34.4	5	5162.8	59.2	5	1183.6	69.7
200	5	4	12904.4	1389.0	3	6739.2	1577.2	3	1698.0	1713.8
300	5	3	15267.2	1666.1	3	7901.2	2033.5	2	2395.2	2515.2
400	5	2	17092.0	2207.7	2	9422.4	2445.4	2	2739.6	2386.3
500	5	2	20170.0	2240.6	2	11272.4	2900.2	2	3462.8	2586.2
600	5	2	22876.4	2289.2	2	12809.6	2772.2	1	4224.0	3212.5
700	4	1	30963.5	2880.9	0	16825.5	3660.5	0	5744.5	3642.6
800	3	0	41566.0	3602.8	0	23930.7	3629.7	0	8067.3	3957.7
900	3	0	46554.7	3603.6	0	26346.7	3739.2	0	9128.7	4214.7

As one can note from Table 6, the higher the density and instance size, the fewer optimal solutions are obtained. This trend is also reflected in the average execution time, which increases with density. This is aligned with the discussion on the previous tables: denser instances introduce more variables and constraints to the models. On the other hand, the average UB decreases as density increases. This is because more arcs are added to the graphs. So, it is easier to find solutions where arcs are reused less frequently when compared with solutions in sparser graphs.

5.2.3. Tests with a different penalty cost

In this section, we evaluate different penalty cost functions and how they impact the solutions. For these tests, we use the original 40 instances from the OR-Library. In addition to the quadratic function ($f_{kl}(r) = c_{kl}r^2$) used in all tests so far, we use a cubic function ($f_{kl}(r) = c_{kl}r^3$).

To incorporate this new penalty cost function, we update $f_{kl}(r)$ in models M1 and M3. Model M4 can be adapted in a straightforward way: we set the penalty cost function to $f'_{kl}(r) = 2c_{kl} + \sum_{h=1}^{r-1} 6hc_{kl}$ for the cubic case. The arc weights in the solution-improvement procedure of Section 4.2 are modified in the same way. For model M2, we use constraints (10) for the cubic case to set variables w_{kl} lower bound.

Table 7 presents the results with the cubic penalty function. The average results for `M2` are omitted because it failed on some instances due to insufficient memory.

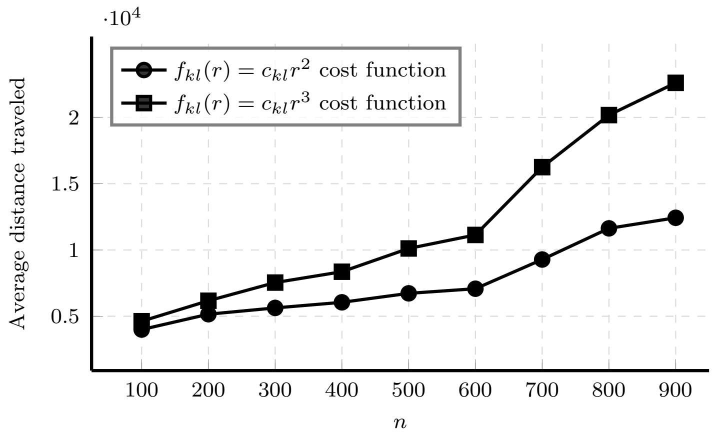
Table 7: Summary of results with penalty function $f_{kl}(r) = r^3 c_{kl}$.

Model	# opt	Average results				Number of best			
		UB	LB	gap (%)	t (s)	UB	LB	gap	t
`M1`	8	63947.6	8657.8	42.0	2970.2	8	7	10	2
`M2`	4	-	-	-	-	4	4	4	0
`M3`	3	64036.2	2994.0	77.0	3481.1	3	3	4	0
`M4`	20	53744.4	19417.4	24.3	2216.8	40	39	39	17

Since the cost function $f_{kl}(r)$ is the only difference between the tests related to Table 7 and those shown in Table 3, the behavior of the results in both tables is similar. Indeed, model `M4` obtained nearly the same number of optimal solutions as in the tests with $f_{kl}(r) = c_{kl}r^2$, and it outperformed the other models on every metric.

However, when we compare the distance traveled by users under the quadratic and cubic cost functions, a difference can be observed. As one might expect, the higher the penalty for multiple traversals of the same arc, the more the model will avoid using any arc more than once, and thus the total distance traveled by the user will tend to increase. This behavior is illustrated in Figure 6, which shows, for model `M4`, the average total distance that users travel to their assigned facility and compares the different penalty costs. In this figure, we partition the instances by number of users and show the average results for each size. As can be seen in Figure 6, the total distance traveled by users is consistently higher under the cubic cost function across all instance sizes, as users tend to avoid reusing arcs.

Figure 6: Comparison of total user travel distance under quadratic ($f_{kl}(r) = c_{kl}r^2$) and cubic ($f_{kl}(r) = c_{kl}r^3$) cost functions.



6. Conclusions

We have introduced the p-MPE, which generalizes the p-MP by incorporating externalities (e.g., congestion and pollution) on the arcs connecting users to facilities. We have modeled these effects using a cost function that penalizes the reuse of arcs, so that solutions tend to avoid using the same arc more than once, hence lessening negative externalities.

We have cast the p-MPE into five MILP models. Starting from a direct four-index model that simultaneously assigns users and selects paths, we progressively simplified the modeling using three-index models. In addition, we have introduced an initial solution algorithm and a solution-improvement procedure. The `M4` model uses a parallel-arc construction with increasing costs that models superlinear arc penalties and eliminates the need of extra variables as in the other models. For this model, we proposed valid inequalities that both strengthen the linear relaxation and reduce symmetry.

Extensive computational experiments indicate that the best model is `M4`. This model consistently obtained the best results both in terms of average solution values and number of optimal solutions, number of best UBs, number of best LBs, number of best optimality gaps, and number of best execution times. Moreover, across all scenarios, it proved optimality for 100% of the instances with fewer than 200 vertices, 66.7% of the instances with 200 vertices, 53.3% of the instances with 300 vertices, 40% of the instances with 400 vertices, and 16% of the instances with more than 500 vertices. Overall, the model proved optimality for 49.1% of all instances across all scenarios and instance sizes. When congestion penalties rise faster than linearly with arc usage, the p-MPE tends to spread the flow across the graph, thus reducing the overuse of arcs and relocating facilities to reduce local bottlenecks.

Acknowledgments

This work was partly supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) [grant number 2025-03964]. We thank the Digital Research Alliance of Canada for providing high-performance computing resources.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Hoboken, New Jersey.

Bayram, V., Yıldız, B., Farham, M.S., 2023. Hub network design problem with capacity, congestion, and stochastic demand considerations. Transportation Science 57, 1276–1295.

Beasley, J.E., 1990. OR-Library: distributing test problems by electronic mail. Journal of the Operational Research Society 41, 1069–1072.

Berman, O., Krass, D., 2019. Stochastic location models with congestion, in: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. 2nd ed.. Springer, Cham, Switzerland. chapter 17, pp. 477–535.

Berman, O., Larson, R.C., 1982. The median problem with congestion. Computers & Operations Research 9, 119–126.

Chagas, G.O., Lorena, L.A.N., dos Santos, R.D.C., Renaud, J., Coelho, L.C., 2024. A parallel variable neighborhood search for α-neighbor facility location problems. Computers & Operations Research 165, 106589.

Desrochers, M., Marcotte, P., Stan, M., 1995. The congested facility location problem. Location Science 3, 9–23.

Gendreau, M., Ghiani, G., Guerriero, E., 2015. Time-dependent routing problems: A review. Computers & Operations Research 64, 189–197.

Hakimi, S.L., 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research 12, 450–459.

Hakimi, S.L., 1965. Optimum distribution of switching centers and some graph-related theoretic. Operations Research 12, 462–475.

Hu, Q.M., Hu, S., 2025. Single allocation hub location problems with congestion: Mixed-integer second-order cone programming and Benders decomposition. Transportation Research Part E: Logistics and Transportation Review 201, 104285.

Hughes, M.S., Lunday, B.J., Weir, J.D., Hopkinson, K.M., 2021. The multiple shortest path problem with path deconfliction. European Journal of Operational Research 292, 818–829.

Kariv, O., Hakimi, S.L., 1979. An algorithmic approach to network location problems II: The p-medians. SIAM Journal on Applied Mathematics 37, 539–560.

Malandraki, C., Daskin, M.S., 1992. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. Transportation Science 26, 185–200.

Marianov, V., ReVelle, C.S., 1996. The queueing maximal availability location problem: A model for the siting of emergency vehicles. European Journal of Operational Research 93, 110–120.

Marín, A., Pelegrín, M., 2019. p-median problems, in: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. 2nd ed.. Springer, Cham, Switzerland. chapter 2, pp. 25–50.

Melkote, S., Daskin, M.S., 2001. Capacitated facility location/network design problems. European Journal of Operational Research 129, 481–495.

ReVelle, C.S., Swain, R., 1970. Central facilities location. Geographical Analysis 2, 20–42.

Schmidt, C.E., Silva, A.C.L., Darvish, M., Coelho, L.C., 2019. The time-dependent location-routing problem. Transportation Research Part E: Logistics and Transportation Review 128, 293–315.

Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice Hall, Englewood Cliffs, NJ.

Yang, H., Bell, M.G.H., 1998. Models and algorithms for road network design: A review and some new developments. Transport Reviews 18, 257–278.

Appendix A. Detailed results

In this appendix, we present the detailed results of the tests discussed in Section 5.2. All tables report the upper bound (UB), lower bound (LB), percentage optimality gap, and total execution time in seconds (t (s)) for the `M1`, `M2`, `M3`, and `M4` models, described in Sections 3.3, 3.4, 3.5, and 3.6, respectively.

Tables A.8 and A.9 present the results related to Section 5.2.1. Table A.8 shows the results of the tests performed on the 15 instances derived from those in the OR-Library, and Table A.9 presents the results of the tests performed on the original 40 p-MP instances from the OR-Library. Tables A.10 and A.11 report the detailed results related to Section 5.2.2, whereas Table A.12 report the detailed results related to Section 5.2.3.

Table A.8: Results of the tests performed on the small instances, with $f_{kl}(r) = c_{kl}r^2$.

			M1				M2				M3				M4			
Instance	n	p	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)
pmed1_25	25	2	15534	15534.0	0.0	0.3	15534	15534.0	0.0	1.2	15534	15534.0	0.0	0.2	15534	15534.0	0.0	0.5
pmed1_50	50	3	32376	32376.0	0.0	1.2	32376	32376.0	0.0	52.6	32376	32376.0	0.0	1.9	32376	32376.0	0.0	9.3
pmed1_75	75	4	22250	22250.0	0.0	17.1	22250	18773.9	15.6	3600.2	22250	22250.0	0.0	20.7	22250	22250.0	0.0	36.0
pmed2_25	25	4	8282	8282.0	0.0	0.2	8282	8282.0	0.0	2.2	8282	8282.0	0.0	0.3	8282	8282.0	0.0	0.8
pmed2_50	50	5	11752	11752.0	0.0	0.7	11752	11752.0	0.0	512.7	11752	11752.0	0.0	1.3	11752	11752.0	0.0	6.4
pmed2_75	75	8	12750	12750.0	0.0	29.5	12750	8739.5	31.5	3600.1	12750	12750.0	0.0	36.4	12750	12750.0	0.0	29.9
pmed3_25	25	3	6182	6182.0	0.0	0.2	6182	6182.0	0.0	2.8	6182	6182.0	0.0	0.1	6182	6182.0	0.0	1.2
pmed3_50	50	5	14078	14078.0	0.0	2.1	14078	14078.0	0.0	533.2	14078	14078.0	0.0	2.1	14078	14078.0	0.0	9.2
pmed3_75	75	8	12542	12542.0	0.0	62.4	12542	8258.5	34.2	3600.1	12542	12542.0	0.0	64.0	12542	12542.0	0.0	24.8
pmed4_25	25	5	4118	4118.0	0.0	0.1	4118	4118.0	0.0	6.7	4118	4118.0	0.0	0.2	4118	4118.0	0.0	0.5
pmed4_50	50	10	5176	5176.0	0.0	0.6	5176	5176.0	0.0	59.4	5176	5176.0	0.0	0.8	5176	5176.0	0.0	1.4
pmed4_75	75	15	6258	6258.0	0.0	2.2	6258	6258.0	0.0	1405.7	6258	6258.0	0.0	7.0	6258	6258.0	0.0	3.2
pmed5_25	25	9	1100	1100.0	0.0	0.0	1100	1100.0	0.0	1.5	1100	1100.0	0.0	0.1	1100	1100.0	0.0	0.2
pmed5_50	50	17	2372	2372.0	0.0	0.1	2372	2372.0	0.0	15.3	2372	2372.0	0.0	0.7	2372	2372.0	0.0	0.3
pmed5_75	75	25	2822	2822.0	0.0	0.4	2822	2822.0	0.0	88.7	2822	2822.0	0.0	2.8	2822	2822.0	0.0	1.0
Average			10506.1	10506.1	0.0	7.8	10506.1	9721.5	5.4	898.8	10506.1	10506.1	0.0	9.2	10506.1	10506.1	0.0	8.3
# best			15	15	15	12	15	5										

Table A.9: Results of the tests performed on the p-MP OR-Library instances, with $f_{kl}(r) = c_{kl} r^2$.

			M1				M2				M3				M4			
Instance	n	p	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)
pmed1	100	5	**18656**	18656.0	0.0	533	**18656**	18656.0	0.0	506	**18656**	8900.6	52.3	3600	**18656**	18656	0.0	71
pmed2	100	10	**10878**	10878.0	0.0	68	**10878**	10878.0	0.0	101	**10878**	6995.8	35.7	3600	**10878**	10878	0.0	51
pmed3	100	10	**11218**	11218.0	0.0	148	**11218**	11218.0	0.0	166	11300	7336.3	35.1	3600	**11218**	11218	0.0	43
pmed4	100	20	**6834**	6834.0	0.0	5	**6834**	6834.0	0.0	20	**6834**	6834.0	0.0	1446	**6834**	6834	0.0	5
pmed5	100	33	**2924**	2924.0	0.0	3	**2924**	2924.0	0.0	13	**2924**	2924.0	0.0	472	**2924**	2924	0.0	2
pmed6	200	5	29046	19721.7	32.1	3600	30588	19408.0	36.6	3601	27252	7324.6	73.1	3600	**27012**	23918	11.5	3600
pmed7	200	10	17478	14576.7	16.6	3600	17762	14441.4	18.7	3601	17910	6805.3	62.0	3600	**17218**	17218	0.0	3026
pmed8	200	20	11352	10859.0	4.3	3600	11346	11078.6	2.4	3601	11736	6668.1	43.2	3600	**11330**	11330	0.0	282
pmed9	200	40	**6200**	6200.0	0.0	160	**6200**	6200.0	0.0	383	6376	4525.4	29.0	3600	**6200**	6200	0.0	27
pmed10	200	67	**2762**	2762.0	0.0	164	**2762**	2762.0	0.0	315	**2762**	2745.4	0.6	3600	**2762**	2762	0.0	10
pmed11	300	5	53908	21231.2	60.6	3600	56068	21140.5	62.3	3602	40626	5848.5	85.6	3600	**32350**	23550	27.2	3600
pmed12	300	10	23424	16766.1	28.4	3600	29980	16799.8	44.0	3601	25164	5483.9	78.2	3600	**22660**	20026	11.6	3600
pmed13	300	30	11041	10105.6	8.5	3600	11634	10161.6	12.7	3603	12066	5053.3	58.1	3600	**10882**	10882	0.0	969
pmed14	300	60	6786	6591.2	2.9	3601	6792	6674.8	1.7	3602	7290	3526.9	51.6	3601	**6774**	6774	0.0	133
pmed15	300	100	**3670**	3670.0	0.0	137	**3670**	3670.0	0.0	2105	3746	2559.8	31.7	3600	**3670**	3670	0.0	27
pmed16	400	5	64686	0.0	100.0	3600				*	63072	5565.8	91.2	3600	**38520**	25992	32.5	3600
pmed17	400	10	33978	17893.7	47.3	3600	34444	0.0	100.0	3641	34286	0.0	100.0	3600	**24628**	20964	14.9	3600
pmed18	400	40	12846	10871.7	15.4	3601	14128	0.0	100.0	3608	14016	4685.0	66.6	3600	**12090**	12048	0.3	3601
pmed19	400	80	6462	6277.4	2.9	3602	6978	0.0	100.0	3606	7052	0.0	100.0	3601	**6440**	6440	0.0	172
pmed20	400	133	**3782**	3782.0	0.0	1900	3920	0.0	100.0	3615	3922	810.1	79.3	3600	**3782**	3782	0.0	64
pmed21	500	5	69982	0.0	100.0	3601				*	69524	0.0	100.0	3600	**44322**	29184	34.2	3602
pmed22	500	10	47002	0.0	100.0	3600				*	45686	0.0	100.0	3600	**34230**	23610	31.0	3600
pmed23	500	50	13186	10657.2	19.2	3602				*	13574	0.0	100.0	3600	**11690**	11498	1.6	3601
pmed24	500	100	6692	6495.2	2.9	3601				*	7236	0.0	100.0	3600	**6668**	6668	0.0	272
pmed25	500	167	**3940**	3914.3	0.7	3602				*	4216	0.0	100.0	3600	**3940**	3940	0.0	129
pmed26	600	5	91646	0.0	100.0	3613				*	91008	0.0	100.0	3600	**54488**	11828	78.3	3602
pmed27	600	10	56220	0.0	100.0	3605				*	55814	0.0	100.0	3601	**37366**	23570	36.9	3601
pmed28	600	60	13208	10209.6	22.7	3603				*	13626	0.0	100.0	3600	**11420**	10968	4.0	3601
pmed29	600	120	7092	6605.7	6.9	3609				*	7556	0.0	100.0	3600	**6874**	6874	0.0	453
pmed30	600	200	4236	4193.6	1.0	3603				*	4528	0.0	100.0	3600	**4234**	4234	0.0	188
pmed31	700	5	121040	0.0	100.0	3612				*	118534	0.0	100.0	3600	**63002**	12518	80.1	3601
pmed32	700	10	66592	0.0	100.0	3610				*	66554	0.0	100.0	3600	**41906**	19882	52.6	3602
pmed33	700	70	14144	10659.8	24.6	3604				*	14244	0.0	100.0	3600	**12174**	11202	8.0	3603
pmed34	700	140	7038	6586.0	6.4	3605				*	7488	0.0	100.0	3600	**6772**	6772	0.0	717
pmed35	800	5	113090	0.0	100.0	3602				*	111410	0.0	100.0	3601	**65004**	12832	80.3	3604
pmed36	800	10	72082	0.0	100.0	3602				*	70294	0.0	100.0	3601	**46668**	12926	72.3	3602
pmed37	800	80	14618	11441.2	21.7	3601				*	14578	0.0	100.0	3601	**13026**	11856	9.0	3602
pmed38	900	5	153488	0.0	100.0	3602				*	151460	0.0	100.0	3601	**79738**	12294	84.6	3603
pmed39	900	10	69970	0.0	100.0	3603				*	70052	0.0	100.0	3601	**46738**	11570	75.2	3603
pmed40	900	90	15292	11512.8	24.7	3603				*	15364	0.0	100.0	3601	**13188**	12040	8.7	3605
Average			32462.2	7102.3	36.2	2870.1	-	-	-	-	32015.3	2364.8	76.8	3468.3	22006.9	12558.3	18.9	2056.9
# best			10	9	9	1	8	8	8	0	5	2	2	0	40	40	40	18

* Out of memory (exceeded 512 GB RAM limit).

Table A.10: Results of the tests performed on the p-MP OR-Library instances with graph density of 0.08 and with $f_{kl}(r) = c_{kl}r^2$.

			M1				M2				M3				M4			
Instance	n	p	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)
pmed1	100	5	**9586**	9586.0	0.0	1573	**9586**	9066.4	5.4	3601	**9586**	4891.8	49.0	3600	**9586**	9586.0	0.0	212
pmed2	100	10	**5110**	5110.0	0.0	26	**5110**	5110.0	0.0	62	5204	3653.1	29.8	3600	**5110**	5110.0	0.0	36
pmed3	100	10	**5512**	5512.0	0.0	77	**5512**	5512.0	0.0	151	**5512**	4024.4	27.0	3600	**5512**	5512.0	0.0	37
pmed4	100	20	**3884**	3884.0	0.0	10	**3884**	3884.0	0.0	34	**3934**	3683.0	6.4	3600	**3884**	3884.0	0.0	7
pmed5	100	33	**1722**	1722.0	0.0	4	**1722**	1722.0	0.0	30	**1722**	1722.0	0.0	754	**1722**	1722.0	0.0	5
pmed6	200	5	14368	9959.6	30.7	3600	14408	9768.5	32.2	3601	14238	3921.7	72.5	3600	**13652**	12030.0	11.9	3600
pmed7	200	10	9622	7737.6	19.6	3601	9818	7717.2	21.4	3601	9864	3823.3	61.2	3601	**9512**	9118.0	4.1	3602
pmed8	200	20	5842	5420.5	7.2	3600	5766	5531.1	4.1	3601	5988	3193.4	46.7	3600	**5734**	5734.0	0.0	595
pmed9	200	40	**3220**	3220.0	0.0	560	**3220**	3220.0	0.0	1047	3296	2276.7	30.9	3600	**3220**	3220.0	0.0	56
pmed10	200	67	**1578**	1578.0	0.0	28	**1578**	1578.0	0.0	786	**1578**	1578.0	0.0	3551	**1578**	1578.0	0.0	33
pmed11	300	5	26224	11190.2	57.3	3600	26232	0.0	100.0	3648	18696	3002.5	83.9	3601	**16916**	12026.0	28.9	3600
pmed12	300	10	17092	8651.6	49.4	3601	17026	0.0	100.0	3608	14254	2831.7	80.1	3600	**11366**	10156.0	10.6	3600
pmed13	300	30	5884	5158.4	12.3	3601	6764	0.0	100.0	3623	6518	2262.5	65.3	3601	**5660**	5660.0	0.0	2628
pmed14	300	60	3488	3453.9	1.0	3601	3730	0.0	100.0	3624	3726	2078.3	44.2	3600	**3482**	3482.0	0.0	190
pmed15	300	100	**2082**	2082.0	0.0	305	2208	0.0	100.0	3677	2160	1403.1	35.0	3601	**2082**	2082.0	0.0	148
pmed16	400	5	33752	0.0	100.0	3600				*	29400	0.0	100.0	3600	**20710**	8028.0	61.2	3601
pmed17	400	10	18828	9957.6	47.1	3600				*	17692	0.0	100.0	3600	**13904**	9890.0	28.9	3601
pmed18	400	40	7148	6037.3	15.5	3601				*	7646	0.0	100.0	3600	**6694**	6532.0	2.4	3631
pmed19	400	80	3690	3561.4	3.5	3602				*	4016	0.0	100.0	3600	**3664**	3664.0	0.0	1177
pmed20	400	133	**2140**	2140.0	0.0	565				*	2232	0.0	100.0	3600	**2140**	2140.0	0.0	218
pmed21	500	5	45234	0.0	100.0	3607				*	39350	0.0	100.0	3600	**26060**	6398.0	75.4	3601
pmed22	500	10	25688	11112.1	56.7	3601				*	24138	0.0	100.0	3600	**18416**	7200.0	60.9	3601
pmed23	500	50	6612	5528.4	16.4	3603				*	7178	0.0	100.0	3619	**6208**	5960.0	4.0	3636
pmed24	500	100	3590	3511.5	2.2	3603				*	3902	0.0	100.0	3600	**3538**	3538.0	0.0	2747
pmed25	500	167	**2140**	2140	0.0	663				*	2294	0.0	100.0	3600	**2140**	2140.0	0.0	915
pmed26	600	5	55424	0.0	100.0	3608				*	56470	0.0	100.0	3600	**32230**	6704.0	79.2	3602
pmed27	600	10	28648	0.0	100.0	3609				*	29642	0.0	100.0	3600	**19376**	6472.0	66.6	3602
pmed28	600	60	7628	5716.4	25.1	3609				*	7494	0.0	100.0	3600	**6630**	5952.0	10.2	3663
pmed29	600	120	3666	3464.0	5.5	3632				*	3820	0.0	100.0	3600	**3510**	3510.0	0.0	2104
pmed30	600	200	**2302**	2302.0	0.0	1053				*	2450	0.0	100.0	3600	**2302**	2302.0	0.0	889
pmed31	700	5	56642	0.0	100.0	3603				*	57904	0.0	100.0	3601	**34488**	6720.0	80.5	3606
pmed32	700	10	33734	0.0	100.0	3603				*	32794	0.0	100.0	3601	**22050**	6372.0	71.1	3603
pmed33	700	70	7726	5875.8	23.9	3602				*	7750	0.0	100.0	3601	**6866**	6158.0	10.3	3831
pmed34	700	140	4066	3847.5	5.4	3607				*	4184	0.0	100.0	3601	**3898**	3742.0	4.0	3603
pmed35	800	5	72248	0.0	100.0	3606				*	74516	0.0	100.0	3601	**37656**	2780.0	92.6	3604
pmed36	800	10	41950	0.0	100.0	3603				*	42938	0.0	100.0	3601	**27258**	5344.0	80.4	3619
pmed37	800	80	7746	5986.0	22.7	3611				*	7792	0.0	100.0	3601	**6878**	5684.0	17.4	3666
pmed38	900	5	76418	0.0	100.0	3606				*	85402	0.0	100.0	3602	**44732**	2862.0	93.6	3606
pmed39	900	10	44944	0.0	100.0	3605				*	44724	0.0	100.0	3602	**26938**	1258.0	95.3	4005
pmed40	900	90	8098	0.0	100.0	3604				*	8144	0.0	100.0	3601	**7370**	5018.0	31.9	3606
Average			17881.9	3886.1	37.5	2734.8	-	-	-	-	17753.7	1108.6	78.3	3528.6	12116.8	5431.7	25.5	2392.2
# best			11	15	12	4	7	6	6	0	4	2	2	0	40	36	39	13

* Out of memory (exceeded 512 GB RAM limit).

Table A.11: Results of the tests performed on the p-MP OR-Library instances with graph density of 0.4 nd with $f_{kl}(r) = c_{kl}r^2$.

			M1				M2				M3				M4			
Instance	n	p	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)
pmed1	100	5	**1926**	1895.4	1.6	3600	1974	1741.8	11.8	3600	2088	853.3	59.1	3601	**1926**	1926.0	0.0	187
pmed2	100	10	**1264**	1264.0	0.0	426	**1264**	1264.0	0.0	964	1268	802.5	36.7	3600	**1264**	1264.0	0.0	73
pmed3	100	10	**1216**	1216.0	0.0	257	**1216**	1216.0	0.0	523	1226	847.3	30.9	3600	**1216**	1216.0	0.0	53
pmed4	100	20	**1088**	1088.0	0.0	16	**1088**	1088.0	0.0	192	**1088**	1088.0	0.0	2183	**1088**	1088.0	0.0	21
pmed5	100	33	**424**	424.0	0.0	8	**424**	424.0	0.0	173	**424**	424.0	0.0	384	**424**	424.0	0.0	15
pmed6	200	5	3392	2297.7	32.3	3600	3392	0.0	100.0	3603	3392	715.3	78.9	3601	**3362**	3038.0	9.6	3602
pmed7	200	10	2460	2016.0	18.0	3601	2460	0.0	100.0	3603	2498	863.8	65.4	3601	**2396**	2346.0	2.1	3602
pmed8	200	20	1498	1351.7	9.8	3602	1502	0.0	100.0	3603	1510	627.7	58.4	3601	**1438**	1438.0	0.0	778
pmed9	200	40	**830**	830.0	0.0	868	882	0.0	100.0	3651	854	527.7	38.2	3601	**830**	830.0	0.0	330
pmed10	200	67	**464**	464.0	0.0	93	490	0.0	100.0	3659	476	383.9	19.3	3600	**464**	464.0	0.0	258
pmed11	300	5	5420	3054.7	43.6	3601				*	5420	0.0	100.0	3600	**5328**	1582.0	70.3	3607
pmed12	300	10	3562	2367.6	33.5	3601				*	3598	0.0	100.0	3600	**3464**	1524.0	56.0	3607
pmed13	300	30	1696	1488.1	12.3	3627	1702	0.0	100.0	3739	1696	645.5	61.9	3600	**1660**	1614.0	2.8	3608
pmed14	300	60	**942**	941.9	0.0	1439				*	974	420.1	56.9	3601	**942**	942.0	0.0	1548
pmed15	300	100	**582**	582.0	0.0	443				*	608	444.8	26.8	3601	**582**	582.0	0.0	206
pmed16	400	5	6448	0.0	100.0	3627				*	6418	0.0	100.0	3602	**5966**	1914.0	67.9	3609
pmed17	400	10	4292	0.0	100.0	3609				*	4292	0.0	100.0	3602	**4230**	1760.0	58.4	3609
pmed18	400	40	1780	1532.5	13.9	3611				*	1782	0.0	100.0	3601	**1766**	1668.0	5.5	3608
pmed19	400	80	1132	1055.5	6.8	3605				*	1138	0.0	100.0	3601	**1078**	1078.0	0.0	700
pmed20	400	133	**658**	658.0	0.0	1024				*	680	0.0	100.0	3601	**658**	658.0	0.0	405
pmed21	500	5	8432	0.0	100.0	3619				*	8432	0.0	100.0	3603	**7934**	1780.0	77.6	3618
pmed22	500	10	5664	0.0	100.0	3621				*	5664	0.0	100.0	3603	**5456**	1136.0	79.2	3617
pmed23	500	50	2022	1689.1	16.5	3610				*	2022	0.0	100.0	3604	**1966**	1382.0	29.7	3615
pmed24	500	100	1220	1159.9	4.9	3623				*	1220	0.0	100.0	3642	**1186**	1186.0	0.0	1273
pmed25	500	167	**772**	772.0	0.0	1863				*	792	0.0	100.0	3602	**772**	772.0	0.0	808
pmed26	600	5	11020	0.0	100.0	3692				*	11020	0.0	100.0	3606	**10484**	58.0	99.4	3631
pmed27	600	10	6386	0.0	100.0	3689				*	6386	0.0	100.0	3604	**6318**	122.0	98.1	3627
pmed28	600	60	2232	0.0	100.0	3696				*	2230	0.0	100.0	3604	**2194**	1446.0	34.1	3629
pmed29	600	120	1326	0.0	100.0	3662				*	1312	0.0	100.0	3603	**1262**	1042.0	17.4	3623
pmed30	600	200	**862**	862.0	0.0	2728				*	872	0.0	100.0	3603	**862**	862.0	0.0	1552
pmed31	700	5	**11586**	0.0	100.0	3642				*	**11586**	0.0	100.0	3610	**11586**	0.0	100.0	3645
pmed32	700	10	**7602**	0.0	100.0	3630				*	**7602**	0.0	100.0	3608	**7602**	0.0	100.0	3645
pmed33	700	70	2402	0.0	100.0	3633				*	2404	0.0	100.0	3607	**2372**	0.0	100.0	3640
pmed34	700	140	1450	0.0	100.0	3631				*	1436	0.0	100.0	3606	**1418**	1154.0	18.6	3640
pmed35	800	5	**13016**	0.0	100.0	3759.8				*	**13016**	0.0	100.0	3617	**13016**	0.0	100.0	3859
pmed36	800	10	**8628**	0.0	100.0	3656.6				*	**8628**	0.0	100.0	3609	**8628**	0.0	100.0	3669
pmed37	800	80				*				*	2572	0.0	100.0	3611	**2558**	0.0	100.0	4345
pmed38	900	5	**14944**	0.0	100.0	6721				*	**14944**	0.0	100.0	4136	**14944**	0.0	100.0	4123
pmed39	900	10				*				*	**9712**	0.0	100.0	4043	**9712**	0.0	100.0	[illegible]
pmed40	900	90				*				*	**2730**	0.0	100.0	4150	**2730**	0.0	100.0	4534
Average			-	-	-	-	-	-	-	-	3900.2	216.1	80.8	3526.3	3827.1	957.4	40.7	2537.6
# best			17	19	20	4	4	4	4	0	9	10	11	0	40	37	37	11

* Out of memory (exceeded 512 GB RAM limit).

Table A.12: Results of the tests performed on the p-MP OR-Library instances with $f_{kl}(r) = c_{kl}r^3$.

			M1				M2				M3				M4			
Instance	n	p	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)	UB	LB	gap (%)	t (s)
pmed1	100	5	36646	20583.2	43.8	3600	36646	18911.3	48.4	3600	35768	12033.9	66.4	3600	**35594**	35594.0	0.0	18
pmed2	100	10	15502	13649.0	12.0	3600	15490	12372.0	20.1	3601	15205	9475.6	37.7	3600	**15200**	15200.0	0.0	27
pmed3	100	10	15984	13581.2	15.0	3600	15544	12258.8	21.1	3601	16450	9806.0	40.4	3600	**15542**	15542.0	0.0	14
pmed4	100	20	**7398**	7398.0	0.0	18	**7398**	7398.0	0.0	42	**7398**	7398.0	0.0	2932	**7398**	7398.0	0.0	4
pmed5	100	33	**3042**	3042.0	0.0	2	**3042**	3042.0	0.0	7	**3042**	3041.8	0.0	122	**3042**	3042.0	0.0	2
pmed6	200	5	62900	23233.9	63.1	3600	62900	20984.4	66.6	3601	62900	10531.3	83.3	3600	**55672**	55672.0	0.0	467
pmed7	200	10	29246	17424.8	40.4	3600	29246	15506.7	47.0	3600	29246	9648.9	67.0	3600	**24776**	24776.0	0.0	309
pmed8	200	20	14222	12124.0	14.8	3600	14570	11468.6	21.3	3601	14654	9523.0	35.0	3600	**13306**	13306.0	0.0	103
pmed9	200	40	**6772**	6772.0	0.0	551	**6772**	6772.0	0.0	3518	7066	5881.9	16.8	3600	**6772**	6772.0	0.0	57
pmed10	200	67	**2914**	2914.0	0.0	33	**2914**	2914.0	0.0	238	**2914**	2914.0	0.0	2978	**2914**	2914.0	0.0	30
pmed11	300	5	124470	25317.4	79.7	3600	124470	0.0	100.0	3606	124470	6109.4	95.1	3600	**80138**	49634.0	38.1	3600
pmed12	300	10	42820	20066.0	53.1	3600	42820	17722.6	58.6	3605	42811	5984.2	86.0	3600	**36146**	36146.0	0.0	2315
pmed13	300	30	14644	11546.4	21.2	3601	14570	10555.5	27.6	3606	14644	5491.8	62.5	3600	**13092**	13092.0	0.0	1298
pmed14	300	60	7460	7196.2	3.5	3600	8396	0.0	100.0	3622	8282	4123.3	50.2	3600	**7312**	7312.0	0.0	387
pmed15	300	100	**3780**	3780.0	0.0	124	4178	0.0	100.0	3626	4064	3439.1	15.4	3600	**3780**	3780.0	0.0	102
pmed16	400	5	133408	25511.7	80.9	3600				*	133408	0.0	100.0	3600	**98904**	53606.0	45.8	3601
pmed17	400	10	49012	21010.9	57.1	3600				*	49012	5677.0	88.4	3600	**41904**	40050.0	4.4	3601
pmed18	400	40	15324	12360.3	19.3	3600				*	15486	0.0	100.0	3600	**14204**	14204.0	0.0	3600
pmed19	400	80	6954	6718.1	3.4	3608				*	7684	0.0	100.0	3600	**6880**	6880.0	0.0	1065
pmed20	400	133	**3928**	3927.9	0.0	396				*	4166	2309.8	44.6	3600	**3928**	3928.0	0.0	383
pmed21	500	5	145624	0.0	100.0	3601				*	145624	0.0	100.0	3600	**138804**	46206.0	66.7	3601
pmed22	500	10	69252	0.0	100.0	3601				*	69252	6370.0	90.8	3600	**57628**	43928.0	23.6	3601
pmed23	500	50	15272	11989.4	21.5	3600				*	15272	0.0	100.0	3600	**13960**	13760.0	1.4	3601
pmed24	500	100	7912	6960.8	12.0	3601				*	7934	0.0	100.0	3600	**7120**	7120.0	0.0	2184
pmed25	500	167	**4076**	4075.7	0.0	799				*	4626	0.0	100.0	3600	**4076**	4076.0	0.0	948
pmed26	600	5	239576	0.0	100.0	3601				*	239576	0.0	100.0	3600	**197442**	40168.0	79.7	3602
pmed27	600	10	89754	0.0	100.0	3606				*	89754	0.0	100.0	3600	**69564**	35220.0	49.4	3601
pmed28	600	60	14774	11133.7	24.6	3601				*	15014	0.0	100.0	3600	**14126**	11170.0	20.9	3633
pmed29	600	120	8348	7154.2	14.3	3601				*	8166	0.0	100.0	3600	**7380**	7364.0	0.2	3601
pmed30	600	200	**4338**	4337.7	0.0	1644				*	4812	0.0	100.0	3600	**4338**	4338.0	0.0	1377
pmed31	700	5	272492	0.0	100.0	3601				*	272492	0.0	100.0	3600	**227420**	39964.0	82.4	3604
pmed32	700	10	97926	0.0	100.0	3602				*	97926	0.0	100.0	3601	**85536**	25550.0	70.1	3616
pmed33	700	70	16092	11563.6	28.1	3602				*	16034	0.0	100.0	3600	**15430**	11984.0	22.3	4425
pmed34	700	140	8330	7120.9	14.5	3601				*	8294	0.0	100.0	3600	**7398**	2878.0	61.1	3753
pmed35	800	5	273626	0.0	100.0	3602				*	273626	0.0	100.0	3601	**244336**	28486.0	88.3	3731
pmed36	800	10	107228	0.0	100.0	3601				*	107228	0.0	100.0	3601	**98864**	4168.0	95.8	3613
pmed37	800	80	16390	11843.0	27.7	3601				*	16464	0.0	100.0	3601	**15866**	14068.0	11.3	3606
pmed38	900	5	433996	0.0	100.0	3603				*	433996	0.0	100.0	3601	**324474**	13454.4	95.9	3986
pmed39	900	10	119766	0.0	100.0	3603				*	119766	0.0	100.0	3601	**112924**	0.0	100.0	3999
pmed40	900	90	16708	11974.5	28.3	3603				*	16924	0.0	100.0	3601	**16586**	13946.0	15.9	3607
Average			63947.6	8657.8	42.0	2970.2	-	-	-	-	64036.2	2994.0	77.0	3481.1	53744.4	19417.4	24.3	2216.8
# best			8	7	10	2	4	4	4	0	3	3	4	0	40	39	39	17

* Out of memory (exceeded 512 GB RAM limit).