Efficient flow models for the uncapacitated multiple allocation p-hub median problem on non-triangular networks

Jack Brimberg, Raca Todosijević, Dragan Urošević, Nenad Mladenovic

Version of Record: https://www.sciencedirect.com/science/article/pii/S0360835221006276
Manuscript_a3c10f1161299ff4364756f5a98df329

Efficient Flow Models for the Uncapacitated Multiple Allocation p-Hub Median Problem on Non-Triangular Networks

Jack Brimberg[a], Raca Todosijević[b,c], Dragan Urošević[d], Nenad Mladenović[e]

[a]*Royal Military College of Canada, Kingston, Ontario, Canada*
[b]*Université Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201, F-59313 Valenciennes, France*
[c]*INSA Hauts-de-France, F-59313 Valenciennes, France*
[d]*Mathematical Institute SANU, 11000 Belgrade, Serbia*
[e]*Khalifa University, PO Box 127788, Abu Dhabi, United Arab Emirates*

Abstract

This paper examines the uncapacitated multiple allocation p-hub median problem (UMApHMP) in a general setting where a given network may violate the triangle inequality, thus leading to flow paths with more than two hubs connecting origin-to-destination pairs. We present two improved flow formulations using a new "augmented graph" that allows a substantial reduction in the number of constraints, and significant improvement in performance of a standard off-the-shelf MILP solver compared to the recent 4-index flow model proposed in Brimberg et al. (2019). Results presented even surpass the performance of specialized algorithms developed for solving the standard "triangular" case. The linear programming relaxations of the presented models are also investigated.

Keywords: p-hub; hub location; multiple allocation; triangle inequality; flow formulations

Email addresses: Jack.Brimberg@rmc.ca (Jack Brimberg), racatodosijevic@gmail.com (Raca Todosijević), draganu@mi.sanu.ac.rs (Dragan Urošević), nenadmladenovic12@gmail.com (Nenad Mladenović)

Efficient Flow Models for the Uncapacitated Multiple Allocation p-Hub Median Problem on Non-Triangular Networks

Abstract

This paper examines the uncapacitated multiple allocation p-hub median problem (UMApHMP) in a general setting where a given network may violate the triangle inequality, thus leading to flow paths with more than two hubs connecting origin-to-destination pairs. We present two improved flow formulations using a new "augmented graph" that allows a substantial reduction in the number of constraints, and significant improvement in performance of a standard off-the-shelf MILP solver compared to the recent 4-index flow model proposed in Brimberg et al. (2019). Results presented even surpass the performance of specialized algorithms developed for solving the standard "triangular" case. The linear programming relaxations of the presented models are also investigated.

Keywords: p-hub; hub location; multiple allocation; triangle inequality; flow formulations

1. Introduction

Hub location models may be classified along several important features. These features include the type of objective function such as median or center, for example, whether or not the hubs and/or edges have capacity constraints, and the selected allocation scheme used for assigning non-hub nodes to hub nodes. For more detailed classification, solution approaches and applications of hub location problems we refer the reader to [1, 4, 5, 10, 13, 14, 17, 19, 21–29, 32, 35–37].

In this paper we examine a particular model known as the uncapacitated multiple allocation p-hub median problem (UMApHMP). As implied in the problem name, the objective (*median*) aims to locate p hubs in order to minimize the total cost (or equivalently the average cost) to send given demands of a commodity between all pairs of source and destination nodes, when there are no capacity constraints on hubs or edges (*uncapacitated*), and each node has access to all p hubs (*multiple allocation*).

The UMApHMP was first proposed by [7] who formulated the problem using variables with up to 4 indices (4-index formulation). This 4-index formulation quickly explodes in size for larger instances. Hence in order to solve larger problem instances, researchers have resorted to reformulating the problem and

proposing models where variables use fewer indices. For example, [3, 16, 20, 31] formulate tighter versions of the problem. A 3-index formulation of UMApHMP is given in [16], where the authors succeeded to solve instances with up to $n = 50$ nodes. More recently, Garcia et al. [18] proposed a formulation based on a preprocessing step which reduces the size of the problem from $O(n^4)$ to $O(n^2)$ variables. Interestingly, their branch-and-cut algorithm is successful only for large values of p, (e.g., $n = 200$ and $p \geq 145$), since the number of constraints for smaller p becomes unmanageable. For this reason, their formulation is not included for comparison in this paper, since networks with such large values of p are not that realistic.

A related problem in the literature is provided by the uncapacitated multiple allocation hub location problem (UMAHLP) where fixed costs are imposed at the nodes selected to become hubs and the number of hubs to open (p) is not given. As noted in de Camargo et al. [11] the computational effort is very sensitive to the coefficient of variation of the fixed costs. In this paper the authors apply a Benders decomposition to tackle the problem. See also , Cánovas et al. [8] who construct a dual-ascent algorithm.

The "standard" models above assume that the triangle inequality holds; that is, shortest paths in the network are always used, including hub-to-hub connections. This in turn implies that all source-to-destination paths will require at most two intermediate hubs. As pointed out recently in Brimberg et al. (2019) [6], this restriction is unrealistic in many applications where the path between a source node and destination node must use for various reasons three or more hubs, e.g., when direct connections between hubs are too long, or when shortest paths may not be suitable for the type of transport used between nodes. For example, depending on the types of aircraft available for use by an air-transport company, there may be limitations on the distance that can be flown directly from one hub (airport) to another. This would make direct connections between certain pairs of nodes impossible, which could be modelled by imposing a very large cost (or distance) for transfer flows on the connecting arc or by setting transfer flows to zero on such arcs. In this case, a transfer flow between two distant hub nodes would have to first go through one or more "additional" intermediate hubs, resulting in some source and destination nodes requiring three or more hubs on the path connecting them. Consider, as another example, relay towers in a communication network that can communicate directly only if the Euclidian distance between them is within a certain range. (Otherwise, the radio signal is too weak.) Again, certain paths connecting origins and destinations would require three or more intermediate hubs (relay towers), which would likely result in a non-triangular network. Since we do not know the hub locations beforehand, the shortest usable paths that apply to the hub sub-network are not known. Recognizing this deficiency, [6] provides two new formulations, referred to as "path" and "flow"

that generalize the UMApHMP model to the "*non-triangular*" case.

In this paper, we further examine the flow model in [6], which is more amenable to exact solution than the path formulation. Two new formulations for the flow model are presented here. Both use a new concept we refer to as an augmented graph. Candidate nodes for hub location are duplicated on the network, so that, for example node i and its duplicate, node $-i$, act respectively as a non-hub node and candidate hub node. Thus, the original node i can act either as a non-hub node alone, or both a non-hub node and hub node (which becomes node $-i$). Although this increases the size of the network, the flow pattern on the network is simplified by separating hub-to-hub flows from the other types of flows (source-to-hub and hub-to-destination). For example, our first formulation uses 4-index variables as in the flow model in [6], but the augmented graph allows a substantial reduction in the number of constraints. The second formulation uses flow variables aggregated by source node for the hub-to-hub flows, which further simplifies the model formulation. In addition, we propose an extension of the 3-index formulation in [16] which allows this model to also solve the non-triangular case. Computational results demonstrate that the models presented here are more amenable to exact solution by off-the shelf software such as CPLEX than the standard (triangular) models in the literature, and the flow model of [6]. Problem instances with up to 100 nodes are solved exactly in less than one hour with CPLEX over a wide range of values for p. Since practitioners most often rely on commercially available software such as CPLEX we believe this is an important improvement over previous results. Moreover, the paper does not only provide empirical analysis of different models, but also provides theoretical insights on the strength of the bounds obtained solving linear programming relaxations of the various formulations.

The rest of the paper is organized as follows. The next section provides a detailed description of the UMApHMP. Sections 3 and 4 present mathematical models used to formulate UMApHMP with triangle inequality imposed and UMApHMP without triangle inequality imposed. Section 5 presents a theoretical study of the linear programming relaxations of various models presented, showing that some formulations provide stronger lower bounds than others. Section 6 provides an extensive computational study on the performance of all models using CPLEX. The last section summarizes our conclusions and final recommendations.

2. Problem description

The uncapacitated multiple allocation p-hub median problem (UMApHMP) is defined on a complete symmetric graph $G = (N, E)$, where $N = \{1, 2, ..., n\}$ represents the set of nodes, and $E = \{(i, j) : i, j \in N\}$ the set of arcs. The graph G should not be confused with the underlying network, which is assumed to be connected, but will not normally have direct links between each pair of nodes. Arcs (i, j) are added to G, generally, to represent the shortest "useable" path from i to j on the network where physical arcs are missing. For each O–D (origin-destination) pair $i - j,\ i, j \in N$, the demand t_{ij} that has to be transferred from node i to node j is given. The direct transfer between nodes is not allowed but must be accomplished via hub nodes. So, the UMApHMP consists in choosing exactly p nodes from the set N to be hubs, where p is given in advance, so that the total transportation cost is minimized assuming that any non-hub node may use any hub node to communicate with other nodes and the flows to or from a non-hub node can be received or sent through more than one hub (multiple allocation scheme). If we assume that the transfer from node i to node j is accomplished via the path $i - h_1^{ij} - h_2^{ij} - \cdots - h_k^{ij} - j$ where each $h_l^{ij},\ l \in \{1, 2, \ldots, k\}$ stands for a selected hub, then the transportation cost per unit flow along such a path is calculated as:

$$d_{ij} = \gamma c_{ih_1^{ij}} + \alpha \sum_{l=1}^{k-1} c_{h_l^{ij} h_{l+1}^{ij}} + \delta c_{h_k^{ij} j},$$

where c_{lm} denotes the length of each arc $(l, m) \in E$, and parameters γ, α and δ are unit rates for collection (origin-hub), transfer (hub-hub) and distribution (hub-destination), respectively. In general, parameter α is used as a discount factor to provide reduced unit costs on arcs between hubs, so $\alpha < \gamma$ and $\alpha < \delta$. Note that in the UMApHMP there are no imposed capacity restrictions on nodes $i \in N$ or arcs $(i, j) \in E$. In addition, we assume that $c_{ij} = c_{ji}$, but not necessarily $t_{ij} = t_{ji}$, for each $(i, j) \in E$. The total flow originating at node i is denoted as $O_i = \sum_{j \in N} t_{ij}$, while the total flow to be received by node j is denoted as $D_j = \sum_{i \in N} t_{ij}$.

In the case where the lengths (or distances) c_{ij} satisfy the triangle inequality, each path from an origin to a destination will contain at most two hub nodes (see [7]). On the other hand, if the triangle inequality is violated, a hub path $h_i - h_j - h_k$ may be shorter than the direct path $h_i - h_k$. Therefore, it may be beneficial to allow the path between an O–D pair to contain more than two hubs (see [6]). In order to differentiate these two cases, the UMApHMP where distances satisfy the triangle inequality will be referred as (Standard) UMApHMP, while the UMApHMP where distances do not satisfy the triangle inequality will be referred as the Generalized UMApHMP (G-UMApHMP).

In the next section we present the mathematical formulations of Standard and Generalized UMApHMP. Some formulations are presented for the first time. Complete details of existing formulations are also given to facilitate comparison of the various models.

3. Mathematical formulations of Standard UMApHMP

In this section we review the so called 4-index and 3-index formulations of Standard UMApHMP, that are widely used in the literature.

3.1. 4-index formulation

The four index formulation [7] uses two sets of variables: binary variables $z_k,\ k \in N$ that take the value 1, if and only if, a node k is used as a hub (i.e., a hub is opened at node k), and 0 otherwise; and continuous variables y_{ijkl} which determine the portion of flow from i to j that is transferred via hubs k and l. Recall that when the triangle inequality applies throughout the network, each O-D path will traverse at most two hub nodes. The complete 4-index formulation, which is based on this insight, is provided in the Appendix. This formulation will be referred to as **4-index UMApHMP**.

3.2. 3-index formulation

The three index formulation [16] employs the following sets of variables in addition to the set of z_k variables previously used in the 4-index formulation :

- u_{ik} := direct flow from node i to hub node k,
- y_{ikl} := flow from node i that is transferred via the arc connecting hub nodes k and l,
- v_{ilj} := flow from origin i which is delivered to destination j via arc (l, j).

Using these variables we have the following formulation denoted as **3-index UMApHMP**:

$$\min \sum_{i,k \in N} \gamma c_{ik} u_{ik} + \sum_{i,m,l \in N} \alpha c_{ml} y_{iml} + \sum_{i,m,j \in N} \delta c_{mj} v_{imj} \tag{1}$$

subject to:

$$\sum_{k \in N} z_k = p, \tag{2}$$

$$\sum_{k \in N} u_{ik} = O_i, \ \forall i \in N, \tag{3}$$

$$\sum_{l \in N} v_{ilj} = t_{ij}, \ \forall i, j \in N, \tag{4}$$

$$u_{ik} + \sum_{m \in N} y_{imk} = \sum_{j \in N} v_{ikj} + \sum_{m \in N} y_{ikm}, \ \forall i, k \in N, \tag{5}$$

$$u_{ik} \leq O_i z_k, \ \forall i, k \in N, \tag{6}$$

$$\sum_{i \in N} v_{ilj} \leq D_j z_l, \ \forall j, l \in N, \tag{7}$$

$$u_{ik}, y_{ikl}, v_{ilj} \geq 0, \ \forall i, j, k, l \in N, \tag{8}$$

$$z_k \in \{0, 1\} \ \forall k \in N. \tag{9}$$

The objective function (1) minimizes the total transportation cost, while the constraint (2) ensures that exactly p hubs are opened. Constraints (3)-(5) are flow conservation constraints that guarantee the flow between each O-D pair is routed. Constraints (6) do not allow flow on the arc $i - k$ if node k is not a hub. Similarly, constraints (7) forbid that the flow going from i to j is transferred along an arc $l - j$ if node l is not a hub. Finally, the type and range of variables are provided in the two last sets of constraints.

This model has $n + n^2 + 2n^3$ variables and $1 + n + 4n^2$ constraints (without variable-type constraints (8) and (9)). By pre-setting the variables $y_{ikk} = 0$, the number of variables may be reduced by n^2.

Property 1. *The 3-index UMApHMP is not valid for use when the triangle inequality is violated.*

Proof. In order to prove this statement, we consider the following counterexample (Figure 1). This example is a modification of the one given in Brimberg et al. (2019) [6]. In this example the number of sought hubs is 3, the distances c_{ij} are provided next to the arcs, and cost parameters are set as follows $\gamma = \delta = 1$ and $\alpha = 0.5$ The flow from a node to itself is set to 0, as well as flow between nodes 1 and 3, while for all other pairs, flow equals 1. The optimal hub locations are denoted by a red color at nodes 1,2, and 3. The objective function value of this solution is 27. On the other hand, the above 3-index formulation provides nodes

1, 4, and 5 as optimal hubs with the objective function value of 26 (Figure 2). However, such a solution obviously yields a much greater objective function value. Namely, the cheapest cost to transfer flow from node 1 to node 4 is c_{14} =200. On the other hand, if we take a look at the values of the variables in the "optimal" solution (see Figure 3) found by 3-index UMApHMP, we see that this "optimal" solution uses all five nodes as intermediate nodes, although there are just three hub nodes. For example, the unit flow from node 1 to node 4, traverses non-hub node 2 ($y_{112} = y_{124} = 1$), which contradicts the use of hub nodes only as intermediate nodes. □

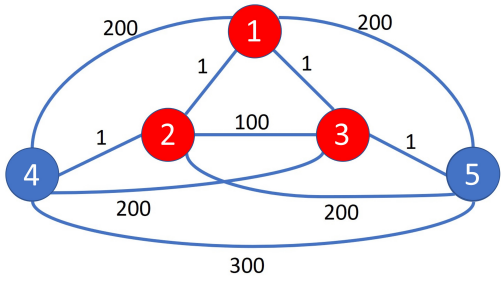

Figure 1: Small network with 5 nodes and 3 hubs optimally located at nodes 1,2 and 3

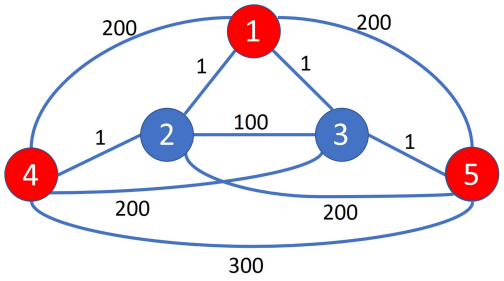

Figure 2: "Optimal" solution of the 3-index UMApHMP (hubs at 1, 4, and 5)

u_ik		value
i	k	
1	1	3
2	1	3
2	4	1
3	1	2
3	5	1
4	4	4
5	5	4

y_ikl			value
i	k	l	
1	1	2	1
1	1	3	1
1	2	4	1
1	3	5	1
2	1	3	1
2	3	5	1
3	1	2	1
3	2	4	1
4	1	3	1
4	2	1	3
4	3	5	1
4	4	2	3
5	1	2	1
5	2	4	1
5	3	1	3
5	5	3	3

v_ilj			value
i	l	j	
1	1	2	1
1	4	4	1
1	5	5	1
2	1	1	1
2	1	3	1
2	4	4	1
2	5	5	1
3	1	2	1
3	4	4	1
3	5	5	1
4	1	1	1
4	4	2	1
4	1	3	1
4	5	5	1
5	1	1	1
5	1	2	1
5	5	3	1
5	4	4	1

Figure 3: Non zero values of variables in optimal solution provided by 3-index UMApHMP

4. Mathematical formulations for Generalized UMApHMP

In this section we present five mathematical formulations for the Generalized UMApHMP. All models are new except for the flow model recently appearing in [6], and presented here for completeness. Brimberg et al. [6] also proposed a path formulation, but since that model exhibited poor performance, we decided not to include it here.

4.1. Modifications to 3-index UMApHMP

The 3-index UMApHMP formulation becomes valid for non-triangular networks if the following constraints are added:

$$\sum_{l \in N, l \neq k} y_{ilk} \leq O_i z_k, \ \forall i, k \in N. \tag{10}$$

The above constraint set forbids an intermediate node to be a non-hub node. Note that when the triangle inequality holds, these constraints are redundant. Constraints (6) ensure that each flow is first transferred to a hub, and similarly constraints (7) impose that each flow is delivered to a node directly from a hub. Since each flow path requires at most two hubs when the triangle inequality holds, the constraints (6) and (7) are enough to ensure that intermediate nodes on each path are hubs only. We will refer to 3-index UMApHMP + (10) as 3-index G-UMApHMP1 or G-UMApHMP1 for short.

We may also combine constraints (6) and (10) into:

$$u_{ik} + \sum_{l \in N, l \neq k} y_{ilk} \leq O_i z_k, \ \forall i, k \in N. \tag{11}$$

This tightening of constraints is not restrictive since multiple paths from a source node to an intermediate node can be reduced to a single path without affecting optimality. Note that 3-index UMApHMP and 3-index UMApHMP -(6) + (11) have the same number of constraints. This second modification will be referred to as G-UMApHMP2 for short.

4.2. Flow formulation of Brimberg et al. 2019 [6]

The flow formulation (FF) of [6] uses the following variables. Binary variables z_k, $k \in N$, as before, are used to indicate if node k is chosen as a hub or not. Flow variables w_{ijlm}, $i, j, l, m \in N$, which are binary, are used to determine if the flow from node i to node j is transferred along the arc (l, m), or not. The remaining two sets of variables are also binary, and used to identify hubs to which origin and destination nodes are allocated. So, we have variables u_{ijm}, $i, j, m \in N$, such that variable u_{ijm} receives value 1, if

and only if, origin node i is directly allocated to hub node m in order to transfer flow to destination node j. Similarly, variables v_{ijm} $i, j, m \in N$ receive value 1, if and only if, node j is directly allocated to hub node m in order to receive flow from node i. The complete formulation referred to as **FF** is provided in the Appendix.

4.3. Formulations on an augmented graph

4.3.1. Improved flow formulation

In this section we propose a new flow formulation, which simplifies the preceding model ([6]) by reducing the number of constraints significantly and relaxing some integrality requirements.

We first construct a "representative graph" $G' = (N', E')$, $N' = N \cup H$, where $H \subseteq \{-1, -2, \ldots, -n\}$ is used to double those nodes in N which are candidate hub locations; and $E' = E_A \cup E_H$, where $E_A = \{(i,j) | i \in N, j \in H \veebar i \in H, j \in N\}$ denotes the set of arcs used for collection (origin-hub) and distribution (hub-destination); and $E_H = \{(i,j) | i, j \in H, i \neq j\}$ denotes the set of arcs used for transfer (hub-hub).

Note that the new graph effectively doubles the number of nodes and the number of edges. We assume here for simplicity $c_{ij} = c_{ij'} = c_{j'i} = c_{ji}$ and $c_{ij} = c_{i'j'} = c_{j'i'} = c_{ji}$, where $j' = -j$ and $i' = -i$. In addition, $c_{ii'} = c_{i'i} = 0$. However, in general, the arc costs (or distances) $c_{ij'}$ and $c_{j'i}$ for collection and distribution may be set to different values than those used for transfer ($c_{i'j'}$ and $c_{j'i'}$) to enable the augmented graph to be more versatile to decision-makers. For example, hub-to-hub arcs with distances below a specified threshold can be penalized by increasing the associated costs $c_{i'j'}$. This could induce in a beneficial way economies of scale by shifting flows to longer arcs, and larger and cheaper modes of transport. Hub-to-hub arcs that are too long could be penalized in a similar way to ensure they are not used. Thus, the augmented graph provides a useful tool to set useable shortest paths for collection and distribution flows (using arcs in E_A) to different values than useable shortest paths for transfer flows (using arcs in E_H) to better reflect reality. The custom design of hub networks that better utilize economies of scale is a subject of growing interest (e.g., see [2]).

As seen below, the augmented graph also leads to a much simplified flow model.

The new flow formulation, apart from variables z_m, $m \in H$ previously defined, uses variables $w_{ijim}, w_{ijmj}, w_{ijlm}$ i, j $N, l, m \in H$. The variables w_{ijim}, w_{ijlm}, and w_{ijmj} represent, respectively, the proportions of flow from i to j that are transferred along arcs (i,m) (origin to hub), (l,m) (hub to hub) and (m,j) (hub to destination).

Hence, the new flow model, denoted as **improved FF**, is as follows:

$$\min \sum_{i,j\in N} t_{ij}[\gamma \sum_{m\in H} c_{im}w_{ijim} + \delta \sum_{l\in H} c_{lj}w_{ijlj} + \alpha \sum_{l,m\in H, l\neq m} c_{lm}w_{ijlm}] \tag{12}$$

subject to:

$$\sum_{m\in H} z_m = p \tag{13}$$

$$\sum_{m\in H} w_{ijim} = 1,\ \forall i,j \in N \tag{14}$$

$$\sum_{l\in H} w_{ijlj} = 1,\ \forall i,j \in N \tag{15}$$

$$w_{ijim} + \sum_{l\in H, l\neq m} w_{ijlm} = \sum_{l\in H, l\neq m} w_{ijml} + w_{ijmj},\ \forall i,j \in N, m \in H, \tag{16}$$

$$w_{ijim} + \sum_{l\in H, l\neq m} w_{ijlm} \leq z_m,\ \forall i,j \in N, m \in H, \tag{17}$$

$$z_m \in \{0,1\}, w_{ijlm}, w_{ijlj}, w_{ijim} \geq 0,\ \forall i,j \in N,\ l,m \in H. \tag{18}$$

The objective function (12) reflects the total transportation cost, while constraint (13), as before, imposes the opening of exactly p hubs. Constraints (14) impose that for each O-D pair (i,j), the flow going from i to j is completely dispatched to hubs, while constraints (15) ensure that this flow arrives via hubs at destination j. Flow conservation in the hub sub-network is guaranteed by constraints (16). Constraints (17) impose that the flow into any node $m \in H$ can be accomplished only if a hub is opened at node m. Constraints (16) and (17) together guarantee that all flows are balanced and any flow path from a source node i to a destination node j can only use intermediate nodes in H where hubs are opened.

The last constraints (18) describe the variables.

The above model has $n + n^3 + n^4$ variables and $1 + 2n^2 + 2n^3$ constraints (without constraints (18)). This is a significant improvement in comparison to the preceding flow formulation FF.

4.3.2. Improved 3-index formulation for G-UMApHMP

The next 3-index formulation is similar to G-UMApHMP1 and 2 with an exception that the augmented graph is used here. The variables $u_{im}, i \in N, m \in H$ quantify the total flow from origin i that is transferred along arc (i, m); the variables $v_{ilj}, i, j \in N, l \in H$ denote the flow from origin i to destination j delivered on arc (l, j); and $y_{ilm}, i \in N, l, m \in H, l \neq m$ gives the total amount of flow from origin i transferred along arc (l, m). In this model, binary variables z_m are used as well, with the same meaning as earlier. Hence, the new model, referred to as **3-index G-UMApHMP3** (G-UMApHMP3 for short), is as follows:

$$\min \ [\gamma \sum_{i \in N, m \in H} c_{im} u_{im} + \delta \sum_{i,j \in N, l \in H} c_{lj} v_{ilj} + \alpha \sum_{i \in N, l, m \in H, l \neq m} c_{lm} y_{ilm}] \tag{19}$$

subject to:

$$\sum_{m \in H} z_m = p \tag{20}$$

$$\sum_{m \in H} u_{im} = O_i, \ \forall i \in N, \tag{21}$$

$$\sum_{l \in H} v_{ilj} = t_{ij}, \ \forall i, j \in N, \tag{22}$$

$$u_{im} + \sum_{l \in H, l \neq m} y_{ilm} = \sum_{j \in N} v_{imj} + \sum_{l \in H, , l \neq m} y_{iml}, \ \forall i \in N, m \in H, \tag{23}$$

$$u_{im} + \sum_{l \in H, l \neq m} y_{ilm} \leq O_i z_m, \ \forall i \in N, m \in H, \tag{24}$$

$$z_m \in \{0, 1\}, y_{ilm}, v_{ilj}, u_{im} \geq 0, \ \forall i, j \in N, l, m \in H. \tag{25}$$

The objective function (19) and constraint (20) are the same as before. Constraints (21) ensure that all flow originating at node i is dispatched, while constraints (22) guarantee that the demand of each node j is respected. Flow conservation at each intermediate hub node is achieved by constraints (23). Constraints (24) ensure that an intermediate node may be only a chosen hub node. The last set of constraints (25) designate the variables of the model.

Note that an analogous set of constraints as in (7) is omitted here, since these constraints are automatically satisfied. (A similar conclusion may be applied to G-UMApHMP2).

This model has $n + 2n^3$ variables (similarly as 3-index UMApHMP), and $1 + n + 3n^2$ constraints. This represents a significant reduction in the number of constraints compared to the two preceding 3-index formulations of G-UMApHMP.

Note that under the standard assumption given above where cost parameters along corresponding arcs for collection, distribution, and transfer have the same value ($c_{ij'} = c_{j'i} = c_{i'j'} = c_{j'i'}$), the nodes in H may be conveniently re-labelled as $\{1, 2, \ldots, n\}(= N)$ without loss of generality in both improved FF and G-UMApHMP3. In effect we return to the original graph. The duplication of the set of network nodes N by a new set H in an augmented graph allows us to better conceptualize the flows, and to simplify the formulations. As discussed above, the augmented graph may also be used as an experimental tool to derive different hub configurations under different scenarios that better reflect reality, without increasing the complexity of the model (and, actually, possibly reducing it).

5. Theoretical comparison of LP relaxations

In this section we derive properties comparing the strength of the Linear Programming (LP) relaxations of the models presented above. By LP relaxation and LP solution, we mean that the integer requirements on all binary variables are relaxed.

For the sake of simplicity, we set $H = N$ in the proofs below; that is, $k \in H$ is the duplicate node of $k \in N, \forall k = 1, 2, \ldots, n$.

Theorem 2. *Any feasible LP solution of G-UMApHMP2 is also a feasible LP solution of G-UMApHMP1 and 3.*

Proof. First note that the y_{ikk}'s can all be set to 0 without changing actual flows in the network. Thus trivial solutions with any y_{ikk}'s > 0 are not considered. For any feasible LP solution of G-UMApHMP2, it is clear that since (11) is satisfied, then (6) and (10) of G-UMApHMP1 must also be satisfied. All other constraints of these two models are the same, and hence, the theorem must be true for these two. Furthermore, transferring the values of all variables in any feasible LP solution of G-UMApHMP2 directly onto the corresponding variables of G-UMApHMP3 clearly must be feasible in the LP relaxation of G-UMApHMP3. (The converse is not true since (7) may be violated). Hence, the theorem is also true for G-UMApHMP2 and 3. □

Corollary 3. *The optimal value of the LP relaxation of G-UMApHMP2 is greater than or equal to the optimal value of the LP relaxation of both G-UMApHMP1 and G-UMApHMP3.*

Theorem 4. *The optimal value of the LP relaxation of improved FF is equal to or greater than that of the LP relaxation of G-UMApHMP3.*

Proof. Note that summing up constraints (14), (16) and (17), multiplied by t_{ij}, over $j \in N$, setting $u_{im} = \sum_{j \in N} t_{ij} w_{ijim}$, $y_{ilm} = \sum_{j \in N} t_{ij} w_{ijlm}$ and replacing $v_{ilj} = t_{ij} w_{ijlj}$, the improved FF formulation transforms to the G-UMApHMP3. For example, (14) converts to (21) as follows:

$$\sum_{m \in H} w_{ijim} = 1, \; \forall i, j \in N$$

$$\Longrightarrow \; t_{ij} \sum_{m \in H} w_{ijim} = t_{ij}, \; \forall i, j \in N$$

$$\Longrightarrow \; \sum_{m \in H} \sum_{j \in N} t_{ij} w_{ijim} = \sum_{m \in H} u_{im} = \sum_{j \in N} t_{ij} = O_i, \; \forall i \in N.$$

Similarly, (16) converts to (23), and (17) converts to (24). Since, G-UMApHMP3 uses aggregated constraints from improved FF, the LP relaxation of improved FF is not weaker than the LP relaxation of G-UMApHMP3.□

Theorem 5. *A feasible solution of the LP relaxation of improved FF is a feasible solution of the LP relaxations of G-UMApHMP1 and G-UMApHMP2.*

Proof. Similarly as in the preceding proof, we set $u_{im} = \sum_{j \in N} t_{ij} w_{ijim}$, $y_{ilm} = \sum_{j \in N} t_{ij} w_{ijlm}$ and $v_{ilj} = t_{ij} w_{ijlj}$, and we pre-set redundant variables $y_{ikk} = 0, \forall i, k$. Then, summing up constraints (14) and (16) multiplied by t_{ij} over $j \in N$, we obtain the constraints (3) and (5). Constraints (4) follow directly from constraints (15) by applying the imposed relation between variables v_{ilj} and w_{ijlj}. Summing (17), multiplied by t_{ij}, over $j \in N$, we have the constraint $u_{ik} + \sum_{l \in N, l \neq k} y_{ilk} \leq O_i z_k, \;\; i, k \in N$, which gives (11) and induces constraints (6) and (10). Note that constraints (16) and (17) impose that $w_{ijkj} + \sum_{l \in H, l \neq k} w_{ijkl} \leq z_k, \;\; i, j, k \in N$, and hence, $w_{ijkj} \leq z_k$. Consequently, after multiplication of $w_{ijkj} \leq z_k$ by t_{ij}, and summing over i, we have $\sum_{i \in N} t_{ij} w_{ijkj} = \sum_{i \in N} v_{ikj} \leq \sum_{i \in N} t_{ij} z_k = D_j z_k, \forall j, k \in N$, which duplicates the constraints (7). □

The next corollary follows from the fact that if an LP solution is feasible for 3-index UMApHMP with added constraints (10) (i.e., G-UMApHMP1), it is also feasible for the 3-index formulation without (10).

Corollary 6. *On instances satisfying the triangle inequality, a feasible solution of the LP relaxation of improved FF is a feasible solution of the LP relaxation of 3-index UMApHMP.*

All these findings lead to the following corollary.

Corollary 7. *The LP relaxation of improved FF is not weaker than that of the standard 3-index UMApHMP model and all presented 3-index G-UMApHMP models.*

The following results refer exclusively to networks that satisfy the triangle inequality. As noted above, an optimal solution may always be found in such cases having flow paths that all contain at most two hub nodes. This property also applies to the LP relaxation, where hubs can be partially opened ($0 < z_k < 1$). Consider the Figure 4, showing a partially opened hub at node k, and an origin node i and destination node j. Thanks to the triangle inequality, the flow from origin i to destination j that traverses arc (k, j) can be restricted to paths $i \to k \to j$, $i \to m_1 \to k \to j$,..., $i \to m_s \to k \to j$. Similarly, the flow on arc (i, k) is restricted to paths $i \to k \to j$, $i \to k \to l_1 \to j$,..., $i \to k \to l_r \to j$. This observation will be used to prove the next result.

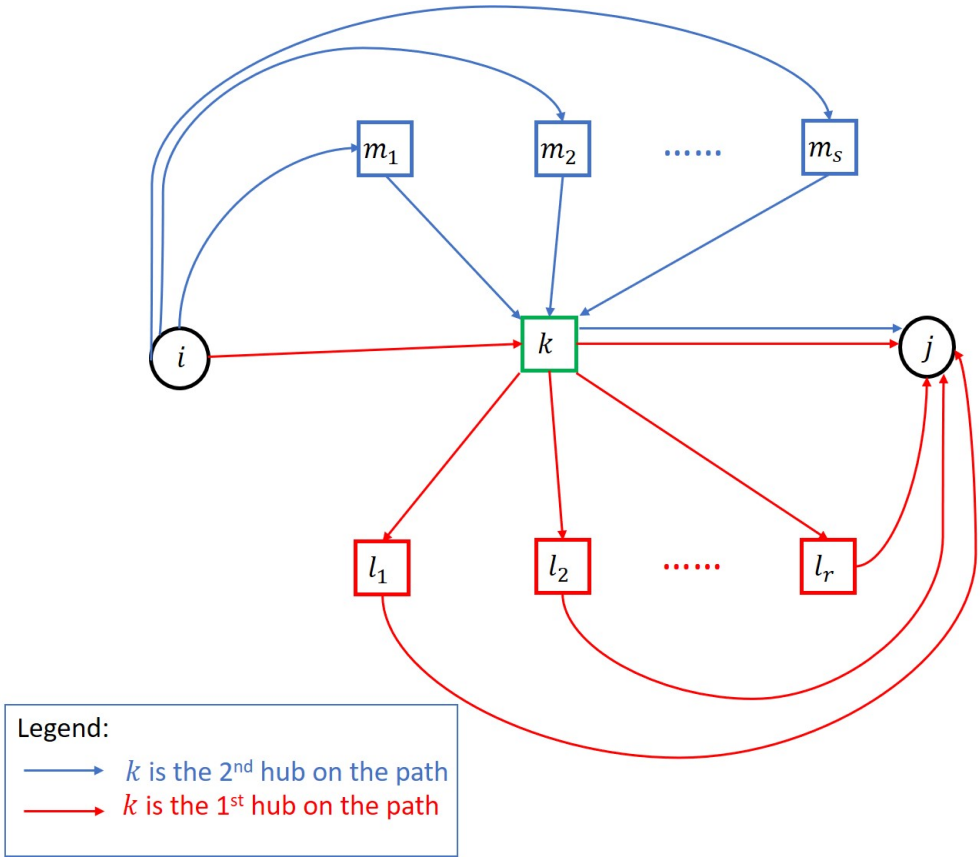

Figure 4: Flow paths from origin node i to destination node j containing partially-open hub k (triangle inequality applies).

Theorem 8. *On instances satisfying the triangle inequality, the optimal value of the LP relaxation of im-*

proved FF is equal to the optimal value of the LP relaxation of 4-index UMApHMP.

Proof. Let $\{z_k^*,\ k \in N\} \cup \{y_{ijkl}^*,\ i,j,k,l \in N\}$ denote an optimal LP solution of 4-index UMApHMP. Referring to the objective function (28), each $y_{ijkl}^* > 0$ sends $t_{ij}y_{ijkl}^*$ units from origin i to destination j along path $i \to k \to l \to j$ (where l can equal k) at a cost, $(\gamma c_{ik} + \alpha c_{kl} + \delta c_{lj})t_{ij}y_{ijkl}^*$. To obtain exactly the same network flows and objective value in the improved FF model, we use the following transformation (see Figure 4):

$$z_k = z_k^* \text{ (or } z_k = z_{(-k)}^*, \text{ depending on the labelling used)},\ \forall k \in H;$$

$$w_{ijkl} = y_{ijkl}^*,\ \forall i,j \in N, k,l \in H, k \neq l;$$

$$w_{ijik} = \sum_{l \in N, l \neq k} y_{ijkl}^* + y_{ijkk}^*,\ \forall i,j \in N, k \in H;$$

$$w_{ijkj} = \sum_{m \in N, m \neq k} y_{ijmk}^* + y_{ijkk}^*,\ \forall i,j \in N, k \in H.$$

Verifying the constraints in the improved FF formulation, we see that constraint (13) is immediately satisfied. For (14), the transformation gives:

$\sum_{k \in H} w_{ijik} = \sum_{k \in N} \sum_{l \in N} y_{ijkl}^* = 1$ (from (30));

similarly, (15) is satisfied; for (16) we get:

$w_{ijik} + \sum_{l \in H, l \neq k} w_{ijlk} = \sum_{l \in N, l \neq k} y_{ijkl}^* + y_{ijkk}^* + \sum_{l \in N, l \neq k} y_{ijlk}^* = \sum_{l \in H, l \neq k} w_{ijkl} + w_{ijkj}$;

and finally (17): $w_{ijik} + \sum_{l \in H, l \neq k} w_{ijlk} = \sum_{l \in N} y_{ijkl}^* \leq z_k^*$ (from (31)) $= z_k$.

We conclude that the given optimal LP solution of 4-index UMApHMP is a feasible solution of the LP relaxation of improved FF.

The reverse transformation that converts an optimal LP solution of improved FF to the same network flows and objective value in the 4-index UMApHMP model is given by:

$$y_{ijkl} = w_{ijkl}^*,\ \forall i,j,k,l \in N, k \neq l;$$

$$y_{ijkk} = w_{ijik}^* - \sum_{l \in H, l \neq k} w_{ijkl}^* = w_{ijkj}^* - \sum_{m \in H, m \neq k} w_{ijmk}^*,\ \forall i,j,k \in N.$$

In similar fashion, we may verify that all constraints of 4-index UMApHMP are satisfied, and hence, the given optimal LP solution of improved FF is a feasible solution of the LP relaxation of 4-index UMApHMP. □.

Now that the equivalence of the LP relaxations of improved FF and 4-index UMApHMP is proven, we may deduce the following.

Corollary 9. *On instances obeying the triangle inequality, the LP relaxation of 4-index UMApHMP is not weaker than that of 3-index UMApHMP or all presented 3-index G-UMApHMP formulations.*

The following proposition provides more insights on whether adding constraints (10) affects the LP relaxation of 3-index UMApHMP when the triangle inequality holds.

Property 10. *If the triangle inequality holds, an optimal solution of the LP relaxation of 3-index UMApHMP is not necessarily a feasible solution of the LP relaxation of 3-index UMApHMP with added constraints* (10)*, (i.e.,G-UMApHMP1).*

Proof. In order to show this we compare the LP relaxation values of both models on the entire set of test instances satisfying the triangle inequality (see Section 6 for more information). The average value of a solution of the LP relaxation of 3-index UMApHMP is 106520.96. On the other hand, when we add constraints (10) the average value of a solution of the LP relaxation of 3-index UMApHMP becomes 106527.90. This means the optimal solution of the LP relaxation of 3-index UMApHMP does not necessarily satisfy constraints (10), and therefore it may be infeasible for the LP relaxation of G-UMApHMP1. □

6. Computational results

In this section we present a comparison of all models presented here. Each model is solved using the CPLEX MIP 12.8 solver with time limit of 3600 seconds imposed on each instance. For each model, we report the solution value found by CPLEX (Columns '`value`') and CPU time consumed (Columns '`time`'). A time reported for a certain model of less than 3600 seconds indicates that CPLEX found an optimal solution; otherwise CPLEX only found a feasible solution. On each test instance not solved to optimality by a model, we calculate the gap (Columns '`gap`') as the percentage deviation of the objective value from the corresponding best lower bound found by CPLEX. For each model M the gap is calculated as:

$$gap_M = \frac{value_M - LB_M}{LB_M} \times 100.$$

Note that if an instance is solved to optimality, a gap is not provided ($gap = 0$). Hence, the corresponding entries contain the sign '$-$'. In addition, on the largest instances we report the number of nodes enumerated at the branch and bound search tree by CPLEX (Columns '`# Nodes`').

For testing purposes, we used two data sets: one containing instances where the triangle inequality is satisfied, and another where the triangled inequality is violated. The first set is actually the AP (Australian

Post) data set, which is widely used in the hub location literature [15]. In each instance of this set with up to 50 nodes, the cost parameters α, γ and δ are set to 0.75, 3, and 2, respectively. For instances with 100 nodes, two sets of cost parameters are considered: $\alpha = 0.75$, $\gamma = 3$ and $\delta = 2$; and $\alpha = 0.75$, $\gamma = \delta = 1$. The second data set is generated by the authors (see [6]), and denoted as C_AP. Each C_AP instance is derived from an AP instance by multiplying the original edge distances (c_{ij}) by randomly generated numbers from the interval $[\frac{2}{3}, \frac{3}{2}]$. As a result, the triangle inequality is not satisfied throughout the network as in the original instance. In addition, the cost parameters in the C_AP instances are set to 0.75, 1, and 1, respectively, for instances with fewer than 100 nodes. For instances with 100 nodes, again, two sets of cost parameters are considered: $\alpha = 0.75$, $\gamma = 3$ and $\delta = 2$; and $\alpha = 0.75$, $\gamma = \delta = 1$.

In the presented tables, the first columns provide the number of nodes in the instance and the number of hubs sought. More precisely, each entry is in the form AP$n.p$ or C_AP$n.p$.

6.1. Comparison on AP data set: triangle inequality satisfied

In this section we compare all models presented in the paper, except G-UMApHMP1 and G-UMApHMP2, on the AP data set, where each test instance satisfies the triangle inequality. G-UMApHMP1 and G-UMApHMP2 are not included in the comparison, because they are derived from 3-index UMApHMP by adding constraints that are redundant in the case when the triangle inequality holds. From reported results, Tables 1 - 3, we infer that only the 3-index formulations are capable to handle instances with up to 100 nodes, the flow formulation (FF) from Brimberg et al. [6] may solve instances up to 40 nodes, while the two remaining formulations (4-index UMApHMP and improved FF) solve instances with up to 50 nodes. The flow formulation of [6] turns out to be the slowest. On instances with up to 40 nodes, it consumes much more CPU time to optimally solve an instance than any other formulation. We also observe that improved FF is significantly faster than the previous FF (e.g., compare CPU times on instance AP40.5 where improved FF is about 5 times faster). This observation confirms our expectations that reducing the number of constraints by defining the problem on an augmented graph may enable FF to solve larger instances in less CPU time. However, if we compare results of the 4-index formulation and improved FF, we observe that the 4-index formulation is almost two times faster although both formulations use 4-index variables. This may be explained by the fact that 4-index UMApHMP is based on the assumption that each path may have at most two hubs, while improved FF considers the more general case where a larger number of intermediate hubs is allowed. In other words, the solution space of improved FF is much larger than that of 4-index UMApHMP. If we compare the results of formulations that use 3-index variables (i.e., 3-index UMApHMP

and G-UMApHMP3), we observe that 3-index UMApHMP is slightly faster. Both formulations are able to optimally solve instances with up to 50 nodes and instances with 100 nodes and the number of requested hubs ranging from 50 to 95. In addition, they both succeeded to optimally solve the instance AP100.40 for cost parameters $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$. On the other hand both fail to do so on instances with 100 nodes and the number of requested hubs ranging from 5 up to 30. Besides these instances, the instance AP100.40 for cost parameters $\alpha = 0.75$, $\gamma = \delta = 1$ remained elusive for both models. On instances AP100.5 and AP100.10, for any choice of cost parameters, G-UMApHMP3 found much better solutions than the ones obtained by solving 3-index UMApHMP. On the remaining instances not solved optimally (AP100.20, and AP100.30 instances), solutions of 3-index UMApHMP and G-UMApHMP3 are equal for cost parameters $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$, while for cost parameters $\alpha = 0.75$, $\gamma = \delta = 1$, solutions provided by G-UMApHMP3 are slightly better. In addition, it may be observed that lower bounds obtained solving G-UMApHMP3 for one hour are much better than those obtained solving 3-index UMApHMP. This leads to the conclusion that G-UMApHMP3 is the best choice for solving instances satisfying the triangle inequality although not specifically designed for such instances. It should be also emphasized that on instances not solved optimally, a solution value provided by G-UMApHMP3 is in the range $[0.09\%, 4.80\%]$ above the corresponding lower bound value. These are very encouraging results compared to the state-of-the-art where even specialized algorithms are unable to solve test instances over the range of values for parameters n and p considered here. For example, the branch-and-cut algorithm developed in [18] is not even able to find a feasible solution within a CPU time limit of 10 hours for AP instances with $n = 100$ and $p \leq 45$. Comparing number of nodes enumerated at the branch and bound search tree by CPLEX, we observe that G-UMApHMP3 is usually able to enumerate more nodes than 3-index UMApHMP. In addition, we observe that the on instances with the cost parameters $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$, the average number of enumerated nodes is smaller than the average number of nodes enumerated solving instances with the cost parameters $\alpha = 0.75$, $\gamma = \delta = 1$. The similar observation may be derived comparing the average time consumed to solve an instance. Hence, we may conclude that the instances with the cost parameters $\alpha = 0.75$, $\gamma = \delta = 1$ are a bit harder than the other one. This may be explained by the fact that in the case $\gamma = \delta$ the collection and distribution costs are equally important in the objective function (unlike the case $\gamma \neq \delta$). Therefore the solution space contains much more non-dominated solutions that cannot be easily discarded while exploring the branch and bound search tree.

Table 1: Comparison on AP data set - instances with up to 50 nodes

	Optimal	4-index UMApHMP	improved FF	FF	3-index UMApHMP	G-UMApHMP3
instance	value	time	time	time	time	time
AP10.2	163603.94	0.25	0.26	0.60	0.21	0.37
AP10.3	131581.79	0.28	0.26	0.70	0.48	0.40
AP10.4	107354.73	0.25	0.37	0.51	0.43	0.39
AP10.5	86028.88	0.25	0.23	0.49	0.29	0.35
AP20.2	168599.79	4.07	3.20	7.55	0.74	1.21
AP20.3	148048.30	4.29	3.69	10.36	1.16	2.21
AP20.4	131665.43	4.90	5.54	8.32	0.97	3.00
AP20.5	118934.97	4.54	5.01	8.28	1.00	4.52
AP20.10	80287.81	3.40	2.57	6.98	0.83	2.96
AP25.2	171298.10	14.55	18.63	29.35	1.63	2.73
AP25.3	151080.66	15.30	21.64	28.29	2.75	6.26
AP25.4	135638.58	12.89	19.55	34.77	2.66	10.35
AP25.5	120581.99	9.53	14.78	23.34	1.78	7.76
AP25.10	86754.96	10.84	13.73	19.40	1.43	7.30
AP40.2	173415.96	208.75	207.95	668.15	13.59	28.93
AP40.3	155458.61	297.02	305.00	1069.36	60.62	66.73
AP40.4	140682.74	150.73	246.96	1071.49	60.06	82.34
AP40.5	130384.74	154.76	231.25	1056.82	41.65	63.14
AP40.10	99452.67	145.91	162.25	537.27	24.97	42.28
AP50.2	174390.03	749.54	640.42	-	65.72	105.73
AP50.3	156014.73	642.92	1055.61	-	302.72	309.06
AP50.4	141153.38	559.72	692.08	-	241.47	274.60
AP50.5	129412.60	528.65	1984.03	-	193.74	186.47
AP50.10	100508.95	447.60	1432.75	-	219.60	143.22
AP50.15	85032.89	350.01	530.37	-	161.56	83.51
AP50.20	73490.33	321.84	400.78	-	70.59	56.99
AP50.25	65127.00	201.02	240.75	-	46.57	29.98
AP50.30	58868.00	224.63	259.89	-	29.36	23.57
AP50.35	65127.00	195.23	237.93	-	44.82	29.57
Average	122413.09	181.51	301.29	-	54.94	54.34

Table 2: Comparison of 3-index formulations on AP data set: instances with 100 nodes $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$

	3-index UMApHMP				G-UMApHMP3			
instance	value	gap	#Nodes	time	value	gap	#Nodes	time
AP100.5	141671.46	10.58	0	3600.79	134683.47	3.19	0	3600.16
AP100.10	106111.56	3.86	0	3600.82	105310.31	2.01	0	3600.51
AP100.20	79191.02	0.83	0	3600.86	79191.02	0.38	0	3600.39
AP100.30	67177.97	0.32	0	3601.46	67177.97	0.09	61	3600.89
AP100.40	59563.04	-	39	1644.42	59563.04	-	447	927.13
AP100.50	54461.94	-	198	419.09	54461.94	-	176	286.71
AP100.80	47213.08	-	24	100.84	47213.08	-	30	152.61
AP100.90	46000.79	-	0	70.05	46000.79	-	11	113.63
AP100.95	45533.51	-	0	27.21	45533.51	-	5	43.60
Average	71880.49	3.90	29.00	1851.73	71015.01	1.42	81.11	1769.51

Table 3: Comparison of 3-index formulations on AP data set: instances with 100 nodes, $\alpha = 0.75$, $\gamma = \delta = 1$

	3-index UMApHMP				G-UMApHMP3			
instance	value	gap	#Nodes	time	value	gap	#Nodes	time
AP100.5	70582.50	15.02	0	3600.56	65954.30	4.80	0	3600.53
AP100.10	60633.10	6.80	0	3600.87	59538.52	3.18	0	3600.67
AP100.20	53547.20	1.93	0	3600.44	53463.75	1.36	0	3600.86
AP100.30	50554.59	0.89	0	3601.89	50393.03	0.51	9	3601.53
AP100.40	48636.01	0.36	23	3601.07	48634.90	0.26	416	3601.47
AP100.50	47370.42	-	573	3190.90	47370.42	-	878	2234.84
AP100.80	45703.26	-	0	116.03	45703.26	-	5	219.32
AP100.90	45442.16	-	0	46.26	45442.16	-	0	107.16
AP100.95	45533.51	-	0	24.31	45533.51	-	0	46.58
Average	52000.30	5.00	66.22	2375.81	51337.09	2.02	145.33	2290.33

6.2. Comparison on C_AP data set: triangle inequality violated

In this section we compare models on our C_AP data set containing instances with violated triangle inequality. The 4-index UMApHMP model is excluded from the comparison since it is not applicable when the triangle inequality is violated. On the other hand, 3-index UMApHMP is included in the comparison, since we have added constraints (10) in G-UMApHMP1, and added (11) and subtracted (6) in G-UMApHMP2, in order to make the model suitable for instances with violated triangle inequality. The results are provided in Tables 4 - 7.

From the reported results we may infer that all models are capable to optimally solve instances with up to 40 nodes. The two fastest formulations turn out to be (3-index) G-UMApHMP3 and (4-index) improved

FF, followed by G-UMApHMP2 and G-UMApHMP1, while the flow formulation from [6] turns out to be the slowest. In addition, the flow formulation from [6] is unable to handle any larger instance, while improved FF succeeds to optimally solve instances with 50 nodes. This once again confirms the benefits of using an augmented graph to formulate the problem. Note that G-UMApHMP1, G-UMApHMP2, G-UMApHMP3 failed to optimally solve or provide proof of optimality on 5, 6 and 3 instances with 50 nodes, respectively. However, on three instances G-UMApHMP3 found optimal solutions but did not prove their optimality, G-UMApHMP2 on four instances, while G-UMApHMP1 did the same on two instances. On larger instances with 100 nodes, G-UMApHMP3 exhibits better performance than the other two 3-index G-UMApHMP formulations: it consumes less time and very often provides a better solution. In addition, in all instances with 100 nodes the LB found solving G-UMApHMP3 is better than that of G-UMApHMP1 and G-UMApHMP2.

All these observations lead to the conclusion that improved FF and (3-index) G-UMApHMP3 are the best options for solving instances with up to 50 nodes. Moreover, improved FF is the only one formulation capable to optimally solve all instances with up to 50 nodes. On larger instances with 100 nodes we may identify (3-index) G-UMApHMP3 as the best option.

Comparing the number of nodes enumerated at the branch and bound search tree (Tables 6 and 7), we observe that the best performing G-UMApHMP3 model is usually able to explore more nodes than the two other models. In addition, we may again conclude that the instances with cost parameters $\alpha = 0.75$, $\gamma = \delta = 1$ are harder to solve than the instances with cost parameters $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$, since the branch and bound trees enumerate more nodes, and therefore, consume more CPU time.

Table 4: Comparison on C_AP data set: instances with up to 40 nodes

	Optimal	improved FF	FF	G-UMApHMP1	G-UMApHMP2	G-UMApHMP3
instance	value	time	time	time	time	time
C_AP10.2	74571.32	0.20	0.59	0.29	0.23	0.2
C_AP10.3	65503.33	0.34	0.70	0.53	0.25	0.21
C_AP10.4	57810.46	0.22	0.62	0.49	0.28	0.22
C_AP10.5	51747.21	0.23	0.47	0.37	0.27	0.17
C_AP20.2	76151.88	4.16	10.43	4.62	5.11	1.24
C_AP20.3	68413.68	7.59	13.27	4.11	4.91	3.04
C_AP20.4	62341.22	11.65	16.08	5.75	5.76	6.91
C_AP20.5	57562.75	3.22	8.72	8.23	4.42	7.13
C_AP20.10	47422.26	4.18	7.99	3.93	3.22	4.16
C_AP25.2	78706.76	10.44	29.47	25.63	19.5	4.02
C_AP25.3	69864.40	10.10	25.11	13.29	15.18	9.09
C_AP25.4	64296.61	9.02	26.43	12.88	11.66	14.69
C_AP25.5	60669.25	10.80	27.37	14.31	14.55	16.41
C_AP25.10	50716.65	6.96	17.09	19.37	10.73	9.79
C_AP40.2	79702.48	221.71	1629.38	473.21	533.84	75.62
C_AP40.3	71227.88	455.60	2785.10	685.52	627.85	183.93
C_AP40.4	66246.22	329.71	1568.72	446.96	550.84	175.98
C_AP40.5	62888.63	369.47	1594.90	349.03	440.15	144.08
C_AP40.10	53616.93	400.09	1174.02	863.09	337.87	272.24
Average	64182.10	97.67	470.34	154.30	136.14	48.90

Table 5: Comparison on C_AP data set: instances with 50 nodes

	improved FF		G-UMApHMP1				G-UMApHMP2				G-UMApHMP3			
instance	value	time	value	gap	#Nodes	time	value	gap	#Nodes	time	value	gap	#Nodes	time
C_AP50.2	78411.75	516.18	78411.75	-	0	1640.50	78411.75	-	0	2532.70	78411.75	-	0	235.47
C_AP50.3	70447.52	1210.91	70447.52	2.79	6	3600.38	70447.52	3.53	5	3600.35	70447.52	-	43	750.18
C_AP50.4	65597.10	1911.62	65597.10	1.88	156	3600.27	65597.10	0.01	176	3422.05	65597.10	-	276	1006.61
C_AP50.5	61995.40	1759.66	61995.40	-	422	3413.26	61995.40	1.11	493	3600.31	61995.40	-	455	1229.87
C_AP50.10	53395.73	3390.48	53401.50	1.67	1603	3600.34	53435.33	1.87	1933	3600.28	53395.73	1.09	2688	3600.41
C_AP50.15	48752.69	3246.70	48830.17	1.85	2555	3600.26	48752.69	1.56	2664	3600.35	48752.69	1.25	3885	3600.72
C_AP50.20	45553.04	1331.52	45599.95	1.12	3955	3600.26	45606.51	1.10	4354	3600.28	45553.04	0.42	5594	3600.69
C_AP50.25	43085.91	484.25	43085.91	-	5254	1558.42	43085.91	-	2665	926.09	43085.91	-	2489	558.61
C_AP50.30	41326.46	371.52	41326.46	-	4019	634.70	41326.46	-	2424	353.65	41326.46	-	2721	287.41
C_AP50.35	39891.91	241.36	39891.91	-	1074	183.06	39891.91	-	588	107.37	39891.91	-	622	87.09
Average	54845.75	1446.42	54858.77	1.86	1904.40	2543.15	54855.06	1.53	1530.20	2534.34	54845.75	0.92	1877.30	1495.71

Table 6: Comparison of 3-index formulations on C_AP data set: instances with 100 nodes , $\alpha = 0.75$, $\gamma = \delta = 1$

	G-UMApHMP1				G-UMApHMP2				G-UMApHMP3			
instance	value	gap	#Nodes	time	value	gap	#Nodes	time	value	gap	#Nodes	time
C_AP100.5	68944.56	32.14	0	3600.77	68944.56	32.12	0	3600.77	65371.56	17.25	0	3600.97
C_AP100.10	56752.48	17.66	0	3600.90	56752.48	17.40	0	3601.65	55685.58	12.01	0	3601.16
C_AP100.20	47674.81	8.20	0	3600.98	48483.11	10.08	0	3600.93	48023.72	7.92	0	3601.22
C_AP100.30	43829.19	5.67	0	3601.20	43944.72	5.81	0	3600.78	43973.36	5.65	0	3601.34
C_AP100.40	41495.17	4.74	0	3601.02	41077.21	3.55	0	3601.19	40966.21	3.08	45	3601.72
C_AP100.50	39112.62	1.95	70	3601.69	39083.28	1.78	120	3601.36	39051.98	1.51	251	3602.09
C_AP100.80	36290.16	0.13	1232	3601.70	36290.16	0.09	2412	3600.81	36290.16	0.08	2415	3601.21
C_AP100.90	35795.85	-	1524	781.21	35795.85	-	972	634.18	35795.85	-	656	463.02
C_AP100.95	35627.62	-	46	212.94	35627.62	-	27	392.52	35627.62	-	24	204.89
Average	45058.05	10.07	319.11	2911.38	45111.00	10.12	392.33	2914.91	44531.78	6.79	376.78	2875.29

Table 7: Comparison of 3-index formulations on C_AP data set: instances with 100 nodes, $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$

	G-UMApHMP1				G-UMApHMP2				G-UMApHMP3			
instance	value	gap	#Nodes	time	value	gap	#Nodes	time	value	gap	#Nodes	time
C_AP100.5	146053.31	22.57	0	3600.45	153821.26	29.07	0	3600.54	139835.04	17.56	0	3600.59
C_AP100.10	110821.07	20.26	0	3600.39	110821.07	19.96	0	3600.41	104458.69	11.68	0	3601.06
C_AP100.20	77996.23	11.83	0	3600.73	78608.93	12.36	0	3600.64	76780.86	8.51	0	3600.46
C_AP100.30	61413.44	5.44	0	3600.49	61681.89	6.06	0	3601.39	61388.03	4.55	4	3601.55
C_AP100.40	52453.95	2.58	0	3601.52	52453.95	2.51	0	3600.55	52446.00	2.23	53	3601.41
C_AP100.50	46621.41	1.17	137	3601.09	46693.75	1.21	55	3601.63	46608.94	0.79	634	3601.43
C_AP100.80	37902.79	-	1896	1873.81	37902.79	-	1979	2544.07	37902.79	-	1764	1434.92
C_AP100.90	36330.40	-	64	129.89	36330.40	-	74	140.71	36330.40	-	99	164.93
C_AP100.95	35847.91	-	37	160.97	35847.91	-	30	195.00	35847.91	-	29	104.91
Average	67271.17	10.64	237.11	2641.04	68240.22	11.86	237.56	2720.55	65733.18	7.55	287.00	2590.14

6.2.1. Breaking symmetry in the 3-index G-UMApHMP3 model: the case $\delta = \gamma$

In the case $\delta = \gamma$, sending flow of t_{ji} units from node j to node i along a path P will have exactly the same cost as sending t_{ji} units along path P but in the reverse direction from i to j. Hence, in this case the problem may be reformulated as sending $(t_{ij} + t_{ji})$ units of flow from node i to node j, for each pair (i, j), $i < j$ and zero flow between pairs (i, j), $i > j$. Taking this observation into account the preceding 3-index

model may be simplified by replacing constraints (22) by:

$$\sum_{l \in H} v_{ilj} = \begin{cases} 0 & \text{if } i > j \\ t_{ii} & \text{if } i = j \\ t_{ij} + t_{ji} & \text{if } i < j \end{cases} \tag{26}$$

Note also that the O_i (rhs of (21)) must be re-calibrated:

$$O_i = t_{ii} + \sum_{j \in N, j > i} (t_{ij} + t_{ji}) \tag{27}$$

The 3-index G-UMApHMP3 formulation which exploits symmetry by using the above constraints will be referred to as 3-index G-UMApHMP symmetry. The comparison of the G-UMApHMP3 formulation and 3-index G-UMApHMP symmetry on the largest instances is presented in Table 8. As we can see, on all instances but one, G-UMApHMP symmetry provides better or equal solutions consuming less CPU time on average. In addition, it proves optimality of one more solution. Regarding the quality of the lower bounds, it provides lower bounds with greater values on all instances.

Table 8: Symmetry breaking: Comparison of 3-index formulations on C_AP instances with 100 nodes

	G-UMApHMP3				3-index G-UMApHMP symmetry			
instance	value	LB	gap	time	value	LB	gap	time
C_AP100.5	65371.56	55754.70	17.25	3600.97	65285.92	57818.88	12.91	3600.64
C_AP100.10	55685.58	49716.01	12.01	3601.16	56728.10	50921.78	11.40	3600.64
C_AP100.20	48023.72	44497.72	7.92	3601.22	48019.48	44638.81	7.57	3601.20
C_AP100.30	43973.36	41619.85	5.65	3601.34	43494.05	41848.15	3.93	3601.97
C_AP100.40	40966.21	39743.75	3.08	3601.72	40883.08	39838.64	2.62	3601.50
C_AP100.50	39051.98	38471.36	1.51	3602.09	38987.19	38558.37	1.11	3600.98
C_AP100.80	36290.16	36260.07	0.08	3601.21	36290.16	-	-	2133.76
C_AP100.90	35795.85	-	-	463.02	35795.85	-	-	309.91
C_AP100.95	35627.62	-	-	204.89	35627.62	-	-	262.90
Average	44531.78	43723.35	6.79	2875.29	44567.94	45604.11	6.59	2701.50

6.3. Managerial insights

In this section, we aim to provide some managerial insights. For this purpose we took the C_AP40.5 test instance, a medium size instance among those tested, and solved it by applying the formulation allowing at most two hubs on each path (i.e., 4-index UMApHMP) and a formulation allowing more than two hubs on a path (i.e., Improved FF). The aim of such a test is to verify if the sets of selected hubs in these two

cases differ or not, as well as to check the amount of savings one may gain allowing more than two hubs on a path. The summary results are presented in the Tables 9 and 10. In these two tables we consider different parameter settings for α, δ and γ, and report the optimal objective function values and optimal set of hubs for each case (more than two hubs allowed and at most two hubs on each path). The column '`(%) deviation`' reports the percentage gain of allowing more than two hubs on a path, which is calculated as:

$$deviation = \frac{value_{UMApHMP} - value_{G-UMApHMP}}{value_{UMApHMP}} \times 100,$$

where $value_{UMApHMP}$ corresponds to the optimal solution value of the 4-index UMApHMP formulation, which allows not more than two hubs on a path, while $value_{G-UMApHMP}$ corresponds to the optimal solution value of Improved FF, which allows more than two hubs on a path. Finally, the last column '`same hubs`' indicates if the optimal solutions in the two considered cases are the same or not.

From the reported results we infer that even on a relatively small instance with 40 nodes, savings may be up to 1.33%. On instances with $\gamma = 3$ and $\delta = 2$, the saving increases as α increases, while on instances with $\gamma = \delta = 1$, this is not the case. An interesting issue may be observed for the parameter setting $\alpha = \gamma = \delta = 1$, where the sets of optimal hubs differ in one hub, but the saving is only 0.02%. Apart from that case, for all other parameter settings, the savings are in the range [0.50%, 1.33%]. Comparing the optimal sets of selected hubs, we observe that only on 2 out of 8 cases, they coincide. However, even in these two cases with the same sets of optimal hubs, the savings are not negligible, i.e., 0.50%, and 1.33%. Finally, we should observe that changing the value of parameter α may affect the optimal set of chosen hubs either when at most two hubs are allowed (UMApHMP case) or more than two hubs are allowed (G-UMApHMP case). For parameter setting $\gamma = 3$ and $\delta = 2$, we have 2 different solutions for G-UMApHMP and 3 different solutions for UMApHMP. The same conclusion holds when $\gamma = \delta = 1$.

Table 9: Comparison of solutions on C_AP40.5 instance with different α values: $\gamma = 3$, $\delta = 2$

	more than 2 hubs allowed on a path		at most two hubs allowed		(%)	same
α	value	hubs	value	hubs	deviation	hubs
0.25	113399.59	4 12 23 27 28	113971.29	4 12 23 27 28	0.50	yes
0.50	123308.59	4 12 23 27 28	124304.57	4 12 23 27 30	0.81	no
0.75	132070.43	12 15 27 28 30	133350.46	4 12 23 27 30	0.97	no
1	137787.35	12 15 27 28 30	139624.24	12 15 27 28 30	1.33	yes

Table 10: Comparison of solutions on C_AP40.5 instance with different α values: $\gamma = \delta = 1$

	more than 2 hubs allowed on a path		at most two hubs allowed		(%)	same
α	value	hubs	value	hubs	deviation	hubs
0.25	52056.87	4 12 23 27 28	52472.65	4 12 23 27 30	0.80	no
0.50	58571.95	12 15 27 28 30	59243.37	12 13 27 28 31	1.15	no
0.75	62888.63	12 15 27 28 30	63252.19	12 13 27 28 31	0.58	no
1	65608.89	12 15 27 28 30	65624.14	12 22 27 28 30	0.02	no

In the next series of experiments, we aim to count the number of paths that use exactly k intermediate hubs, where $k \in \{1, 2, \dots p\}$, and also to count the number of nodes that are allocated to exactly k hubs. Again, for testing purposes we choose the C_AP40.5 instance, and apply the Improved FF to solve it for any choice of parameters α, γ and δ. The summary results are provided in Tables 11 and 12. The tables report the number of paths using exactly k hubs for the k value specified in the column heading. Similarly we report the number of nodes allocated to exactly k hubs, where k is given in the column heading. Finally, we report the total collection and distribution costs in the optimal solution as well as the total transfer cost divided by α. Note that the total collection cost is specified by the first term, the total distribution cost is expressed by the second term, while the total transfer cost is given by the third term in the objective function (12).

From the reported results we infer that as α increases, the total transfer cost decreases while the collection and distribution costs increase. Also, we observe that as α increases, the number of paths that use only one hub increases. This may be explained by the fact that as α increases, the transfer cost becomes more important in the objective function, and therefore, the paths with zero transfer cost are favored. In addition, we observe that for any choice of α value, we have paths that use up to 4 intermediate hubs, and nodes may be allocated to up to 5 hubs. Interestingly, for $\gamma = 3$, $\delta = 2$, the number of nodes allocated to 3 or more hubs is not greater than 8, while for $\gamma = \delta = 1$, this number goes up to 26. Also, it should be noted that in the cases when optimal solutions of G-UMApHMP for different α values coincide, the structure of solutions in terms of the number of used hubs on a path and the number of hubs allocated to a node may be very different (e.g., compare the solutions in Table 11 for $\alpha = 0.50$ and $\alpha = 1$).

There is another indirect but important benefit of allowing O-D paths with more than two intermediate hubs. For example, compare the path $i \to h_1^{(1)} \to h_2^{(1)} \to j$ with two intermediate hubs, and $i \to h_1^{(2)} \to h_2^{(2)} \to h_3^{(2)} \to j$ with three. The two-hub path provides a dedicated flow from node i to node j, while the 3-hub path allows some flow to disembark or embark at an additional point (the middle hub). The end

result is that smaller but more frequent shipments are possible between O-D pairs, and hence, better quality service.

Table 11: Structure of solutions on C_AP40.5 instances with different α: $\gamma = \delta = 1$

	# paths using 1-5 hubs					# nodes assigned 1-5 hubs					costs		
α	1	2	3	4	5	1	2	3	4	5	collection	transfer/α	distribution
0.25	602	812	166	20	0	30	8	0	2	0	19245.00	38480.67	23191.70
0.50	820	626	142	12	0	19	13	6	1	1	21009.45	20696.11	27214.45
0.75	1060	470	62	8	0	10	16	11	2	1	22672.74	14545.49	29306.77
1	1376	213	10	1	0	1	13	15	7	4	26707.13	3713.97	35187.79

Table 12: Structure of solutions on C_AP40.5 instances with different α: $\gamma = 3$, $\delta = 2$

	# paths using 1-5 hubs					# nodes assigned 1-5 hubs					costs		
α	1	2	3	4	5	1	2	3	4	5	collection	transfer/α	distribution
0.25	507	855	210	28	0	32	6	2	0	0	56965.96	40872.57	46215.48
0.50	593	819	168	20	0	30	8	0	2	0	57496.81	38850.50	46386.52
0.75	657	744	169	30	0	24	10	4	2	0	60586.62	24046.76	53448.75
1	756	668	157	19	0	19	13	6	1	1	61234.43	21997.89	54555.03

6.4. Comparison of LP relaxations

In this section we compare the linear programming (LP) relaxations of the various models. The LP relaxations are derived from their original formulations by relaxing integrality requirements of all variables (i.e., all variables are considered as continuous). The results are presented in Tables 13-18. For each formulation, we report the value of the LP relaxation and the time needed to solve it. In addition, the second column in each table reports optimal solution values (if known) of the original problem formulation. In the case an optimal value is not known, we report the best solution value found so far (BKS) and emphasize that value using italics.

From the reported results we infer that the G-UMApHMP3 formulation has the weakest LP relaxation, but also requires the least CPU time. Solving the LP relaxation of the G-UMApHMP3 formulation requires not more than 200 seconds on the largest instances. On the other hand solving the LP relaxation of the 3-index UMApHMP, G-UMApHMP1 and G-UMApHMP2 formulations may consume a significant portion of time. For example, on instance C_AP100.5, G-UMApHMP1 and G-UMApHMP2 consume more than 2000 seconds. However, the quality of an LP solution of G-UMApHMP1 and G-UMApHMP2 is much

better than that of the G-UMApHMP3. As noted in Corollary 3, G-UMApHMP2 has the strongest LP relaxation. Empirical results show that it is slightly better than the LP relaxation of G-UMApHMP1, but solution times are generally larger.

On AP instances with up to 50 nodes, LP relaxations of 4-index UMApHMP and Improved FF formulations exhibit an interesting performance. The LP relaxation values, of both formulations, coincide with the optimal ones an all test cases. In addition, the same phenomenon arises for the FF model on 5 out of 29 instances (see the boldfaced values in Table 13). Regarding average CPU time consumptions, solving LP relaxations of 4-index UMApHMP and Improved FF require similar time. Note that the LP relaxation of Improved FF yields better solution values in shorter time than that of FF.

On C_AP instances with up to 50 nodes we also remark that LP relaxation values of the Improved FF and FF models may be the same as the optimal ones on several instances. In the case of the Improved FF model, this happens on 11 out of 29 instances, while in the case of FF, matching occurs on 2 out of 29 instances (see the boldfaced values in Table 16). In addition, we may again conclude that the LP relaxation of Improved FF provides better solution values in shorter time than that of FF.

Overall, we may conclude that the 4-index UMApHMP and Improved FF formulations have the strongest LP relaxations, while the weakest LP relaxation is that of 3-index G-UMApHMP3. In addition, we may say that the LP relaxation of 3-index UMApHMP provides the best trade-off between solution quality and CPU time consumption. Comparing LP relaxations of Improved FF and FF, empirical evidence shows that the LP relaxation of Improved FF is better than the LP relaxation of FF in terms of both solution quality and CPU time consumption. This observation may justify the superiority of the Improved FF model over the FF model, and confirm the usefulness of modelling on the augmented graph.

Table 13: Comparison of LP relaxations on AP data set: instances up to 50 nodes

		4-index UMApHMP		Improved FF		FF		3-index UMApHMP		G-UMApHMP3	
instance	OPT	value	time	value	time	value	time	value	time	value	time
AP10.2	163603.94	163603.94	0.14	163603.94	0.21	160019.25	0.27	159987.33	0.25	115652.34	0.14
AP10.3	131581.79	131581.79	0.13	131581.79	0.17	123300.27	0.21	124069.04	0.08	92697.34	0.13
AP10.4	107354.73	107354.73	0.11	107354.73	0.19	100660.75	0.19	102371.78	0.08	78367.63	0.14
AP10.5	86028.88	86028.88	0.10	86028.88	0.15	84207.16	0.18	84171.68	0.08	66707.41	0.18
AP20.2	168599.79	168599.79	1.88	168599.79	2.34	**168599.79**	4.32	166044.99	0.54	125322.27	0.33
AP20.3	148048.30	148048.30	2.18	148048.30	2.30	145680.02	4.29	140918.18	0.27	103265.80	0.22
AP20.4	131665.43	131665.43	1.93	131665.43	2.28	130729.31	4.44	125155.95	0.22	92914.10	0.24
AP20.5	118934.97	118934.97	2.11	118934.97	1.93	117826.94	4.96	113197.15	0.23	85084.73	0.23
AP20.10	80287.81	80287.81	0.82	80287.81	1.27	79344.65	1.96	77681.88	0.22	62580.47	0.15
AP25.2	171298.10	171298.10	5.45	171298.10	6.51	171188.00	12.00	169042.94	0.67	127846.36	0.59
AP25.3	151080.66	151080.66	6.56	151080.66	7.25	150524.40	14.10	145170.18	0.74	105168.61	0.39
AP25.4	135638.58	135638.58	6.79	135638.58	8.10	134000.79	12.43	129069.78	0.49	94760.67	0.34
AP25.5	120581.99	120581.99	4.20	120581.99	5.64	120147.10	12.61	116369.56	0.46	86948.79	0.33
AP25.10	86754.96	86754.96	4.07	86754.96	4.25	86188.18	6.10	83408.06	0.27	66302.22	0.27
AP40.2	173415.96	173415.96	64.72	173415.96	108.46	**173415.96**	161.44	171713.44	9.52	130249.07	2.44
AP40.3	155458.61	155458.61	82.37	155458.61	114.92	**155458.61**	192.67	148518.59	4.86	108638.25	1.90
AP40.4	140682.74	140682.74	89.63	140682.74	96.44	**140682.74**	165.20	134222.11	3.68	98611.62	1.82
AP40.5	130384.74	130384.74	86.89	130384.74	97.61	130301.26	228.85	124450.58	2.36	92131.17	1.37
AP40.10	99452.67	99452.67	59.87	99452.67	85.19	98918.23	154.27	95581.67	1.14	74528.96	0.95
AP50.2	174390.03	174390.03	277.23	174390.03	419.93	174330.42	732.17	171902.25	23.12	130859.20	7.37
AP50.3	156014.73	156014.73	432.81	156014.73	471.07	**156014.73**	676.60	148879.41	21.53	108674.75	5.78
AP50.4	141153.38	141153.38	324.21	141153.38	354.33	141134.83	852.77	134431.57	11.45	98876.74	5.69
AP50.5	129412.60	129412.60	525.24	129412.60	366.01	129294.39	672.95	124367.12	8.37	92056.79	4.57
AP50.10	100508.95	100508.95	329.01	100508.95	360.48	100261.55	764.25	96350.28	3.10	73903.95	2.88
AP50.15	85032.89	85032.89	357.39	85032.89	335.70	84736.27	584.19	82061.48	2.90	65557.64	1.62
AP50.20	73490.33	73490.33	307.87	73490.33	343.32	73466.98	425.59	71784.14	2.45	59734.08	1.47
AP50.25	65127.00	65127.00	251.38	65127.00	207.63	65115.52	253.99	64083.00	2.08	55485.90	1.18
AP50.30	58868.00	58868.00	161.24	58868.00	154.24	58800.77	176.63	58119.73	1.93	52210.91	1.07
AP50.35	65127.00	65127.00	248.81	65127.00	198.23	65115.52	261.52	64083.00	2.08	55485.90	1.12
Average	122413.09	122413.09	125.35	122413.09	129.52	121360.84	220.04	118179.55	3.63	89676.68	1.55

Table 14: Comparison of LP relaxations on AP data set: instances with 100 nodes, $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$

		3-index UMApHMP		G-UMApHMP3	
instance	OPT/BKS	value	time	value	time
AP100.5	*134683.47*	127022.54	832.69	93920.41	449.39
AP100.10	*105310.31*	99822.58	100.37	76512.34	190.58
AP100.20	*79191.02*	76639.61	32.40	62749.74	53.98
AP100.30	*67177.97*	65688.85	24.58	56518.28	24.31
AP100.40	59563.04	58695.33	18.79	52594.01	17.27
AP100.50	54461.93	54079.88	17.20	49934.15	12.00
AP100.80	47213.08	47145.70	13.48	46311.42	10.60
AP100.90	46000.79	45962.68	12.55	45649.66	9.63
AP100.95	45533.51	45532.51	10.34	45418.85	10.69
Average	71113.07	68954.41	118.04	58845.43	86.49

Table 15: Comparison of LP relaxations on AP data set: instances with 100 nodes, $\alpha = 0.75$, $\gamma = \delta = 1$

		3-index UMApHMP		G-UMApHMP3	
instance	OPT/BKS	value	time	value	time
AP100.5	*65954.30*	59061.76	1038.65	51244.51	199.98
AP100.10	*59538.52*	54629.37	172.19	49330.29	93.29
AP100.20	*53463.75*	50990.10	41.69	47712.74	47.30
AP100.30	*50393.03*	49096.80	24.85	46885.08	20.33
AP100.40	*48634.90*	47841.12	17.10	46353.55	12.96
AP100.50	47370.42	46999.23	16.68	45983.83	10.27
AP100.80	45703.26	45641.19	16.53	45436.06	8.47
AP100.90	45442.16	45416.60	11.69	45341.53	8.92
AP100.95	45533.51	45334.91	10.93	45306.82	8.51
Average	51337.09	49445.67	150.03	47066.05	45.56

Table 16: Comparison of LP relaxations on C_AP data set: instances up to 50 nodes

		Improved FF		FF		G-UMApHMP1		G-UMApHMP2		G-UMApHMP3	
instance	OPT	value	time	value	time	value	time	value	time	value	time
C_AP10.2	74571.32	**74571.32**	0.10	73885.08	0.26	67547.22	0.12	67547.22	0.07	49875.66	0.03
C_AP10.3	65503.33	65373.79	0.10	61566.77	0.22	56304.45	0.10	56304.45	0.04	46866.77	0.02
C_AP10.4	57810.46	**57810.46**	0.10	54565.84	0.23	51085.20	0.09	51085.20	0.06	44915.34	0.10
C_AP10.5	51747.21	**51747.21**	0.07	50340.42	0.19	47859.24	0.07	47859.24	0.05	43497.04	0.09
C_AP20.2	76151.88	**76151.88**	2.33	75867.21	4.11	69951.04	0.56	69951.04	0.66	50474.22	0.26
C_AP20.3	68413.68	67684.31	2.27	66313.84	4.14	59213.17	0.34	59213.17	0.37	47215.31	0.20
C_AP20.4	62341.22	61893.66	2.40	60442.15	3.42	54235.77	0.24	54235.77	0.22	45612.04	0.20
C_AP20.5	57562.75	**57562.75**	1.83	56808.15	3.44	51391.88	0.21	51392.05	0.22	44610.16	0.20
C_AP20.10	47422.26	47399.94	1.34	47001.88	2.25	44848.85	0.14	44848.85	0.18	41719.18	0.17
C_AP25.2	78706.76	**78706.76**	7.98	78646.34	12.61	71150.16	1.61	71150.16	1.87	52518.44	0.28
C_AP25.3	69864.40	**69864.40**	7.04	69723.29	13.54	61535.72	0.85	61539.38	1.03	48736.49	0.25
C_AP25.4	64296.61	**64296.61**	5.62	**64296.61**	13.18	56317.66	0.39	56319.19	0.47	46909.00	0.23
C_AP25.5	60669.25	60664.07	7.06	60075.24	11.13	53427.36	0.30	53428.58	0.36	45793.57	0.24
C_AP25.10	50716.65	**50716.65**	3.94	50436.82	6.50	47093.68	0.21	47105.30	0.20	43148.94	0.19
C_AP40.2	79702.48	79160.16	131.10	78862.54	278.92	70721.81	23.51	70722.61	26.55	51994.37	1.54
C_AP40.3	71227.88	70400.70	149.70	69939.45	400.21	61514.37	10.73	61517.90	13.71	47795.52	1.07
C_AP40.4	66246.22	65491.55	135.78	65045.87	217.00	56629.37	4.90	56633.83	6.20	46169.75	0.84
C_AP40.5	62888.63	62187.27	190.35	61771.38	190.63	53820.38	2.38	53822.40	2.91	45218.78	0.68
C_AP40.10	53616.93	53037.71	151.80	52540.36	210.98	47765.82	0.82	47796.87	1.01	42801.70	0.46
C_AP50.2	78411.75	**78411.75**	599.84	**78411.75**	871.85	70272.84	80.31	70273.99	102.93	51036.81	5.31
C_AP50.3	70447.52	69666.20	530.64	69346.61	959.35	60227.63	52.83	60235.14	136.22	46554.06	4.05
C_AP50.4	65597.10	64569.35	650.35	64193.25	1148.95	55395.78	19.38	55398.60	42.79	44724.26	2.71
C_AP50.5	61995.40	61003.69	539.65	60637.06	846.31	52505.70	13.18	52515.24	25.86	43594.83	2.47
C_AP50.10	53395.73	52285.50	767.31	51936.27	747.57	46558.00	1.92	46588.95	5.46	40051.30	1.06
C_AP50.15	48752.69	47920.57	434.42	47552.59	541.45	43879.58	1.49	43899.81	3.19	41205.47	1.36
C_AP50.20	45553.04	45155.57	310.30	44886.42	393.60	42031.01	1.48	42044.29	3.18	39257.33	1.03
C_AP50.25	43085.91	43065.71	279.04	42928.54	395.34	40677.32	1.51	40682.37	3.27	38679.18	0.99
C_AP50.30	41326.46	41311.13	224.71	41244.84	319.21	39617.64	1.46	39617.76	3.28	38226.95	0.93
C_AP50.35	39891.91	**39891.91**	171.16	39868.77	281.13	38775.74	1.43	38777.71	3.23	37874.60	0.85
Average	60962.67	60620.78	183.05	59970.18	271.65	54219.12	7.67	54224.38	13.30	45071.62	0.96

Table 17: Comparison of LP relaxations on C_AP data set: , $\alpha = 0.75$, $\gamma = \delta = 1$

		G-UMApHMP1		G-UMApHMP2		G-UMApHMP3	
instance	OPT/BKS	value	time	value	time	value	time
C_AP100.5	*65285.92*	52177.34	2055.86	52183.12	2465.57	42540.48	197.79
C_AP100.10	*55685.58*	45840.76	366.85	45864.76	371.82	39902.67	103.86
C_AP100.20	*47674.81*	41641.14	68.18	41654.80	68.64	38086.72	31.90
C_AP100.30	*43829.19*	39483.91	28.32	39494.82	23.93	37183.61	17.67
C_AP100.40	*40966.21*	38137.99	14.39	38140.49	14.26	36610.10	12.38
C_AP100.50	*39051.98*	37281.08	14.06	37283.82	13.57	36233.94	10.87
C_AP100.80	*36290.16*	35846.47	14.04	35847.61	13.29	35654.03	10.20
C_AP100.90	35795.85	35625.36	9.98	35625.89	13.04	35559.63	11.69
C_AP100.95	35627.62	35553.16	8.52	35553.16	10.13	35529.64	10.85
Average	45259.12	40178.17	286.69	40183.16	332.69	37477.87	45.25

Table 18: Comparison of LP relaxations on C_AP data set: , $\alpha = 0.75$, $\gamma = 3$, $\delta = 2$

		G-UMApHMP1		G-UMApHMP2		G-UMApHMP3	
instance	OPT/BKS	value	time	value	time	value	time
C_AP100.5	*139835.04*	119154.92	2393.69	119178.54	2468.50	84904.34	316.35
C_AP100.10	*104458.69*	90622.32	853.62	90700.23	839.87	67080.52	160.88
C_AP100.20	*76780.86*	67581.76	181.78	67654.84	207.75	53427.59	50.80
C_AP100.30	*61388.03*	56085.00	50.21	56130.20	50.50	46989.82	18.67
C_AP100.40	*52446.00*	49285.14	20.46	49318.83	24.00	43025.95	13.81
C_AP100.50	*46608.94*	44675.19	13.85	44689.80	13.78	40333.54	9.04
C_AP100.80	37902.79	37324.17	13.54	37327.08	12.64	36514.96	8.22
C_AP100.90	36330.40	36141.00	9.00	36142.70	12.68	35856.54	7.95
C_AP100.95	35847.91	35751.01	8.70	35752.47	8.40	35635.64	8.28
Average	67271.17	59624.50	393.87	59654.97	404.24	49307.66	66.00

7. Conclusions

This paper proposes several new mathematical models for the Uncapacitated Multiple Allocation p-Hub Median Problem (UMApHMP). Two of these models are improved flow formulations on a new "augmented graph", which allows a substantial reduction in the number of constraints. The performance of the newly proposed models is assessed using a generic off-the-shelf software (CPLEX) on a wide set of instances, and compared with performances of existing UMApHMP models. The main benefit of the new models is that they can be used when the triangle inequality is satisfied in the problem data or, more generally when it is not. In addition, they enable us to solve in reasonable computing time (less than one hour) test instances of

both types having up to 100 nodes ($n = 100$), and a wide range of values of p. As expected, the results for triangular networks are better than non-triangular. Note that the current best model from the literature that is capable of handling instances with violated triangle inequality (see [6]) can only solve instances with up to 40 nodes on CPLEX. Meanwhile, for the standard triangular case, the specialized algorithm given in [18] is not even able to find a feasible solution for $n = 100$, $p \leq 45$, and a CPU time limit of 10 hours. Thus, the computational results obtained here are very encouraging.

Moreover, we show that the 3-index model for UMApHMP designed for instances with the triangle inequality cannot be directly used for solving instances where the triangle inequality is violated. However, we provide a set of constraints to be added to the model in order to make it capable to handle instances with and without the triangle inequality.

The paper provides also a theoretical and empirical study of the linear programming relaxations of all models presented in the paper. These observations may be used in the future for developing efficient matheuristics capable of solving larger problem instances, and/or finding tighter lower bounds on the optimal solution. We also present a parametric analysis on the unit transfer cost (α), which compares some aggregate measures affecting solution quality when flow paths with more than two intermediate hubs are allowed. This type of analysis may help managers better understand the nature of hub networks, and hence, better design and manage them.

The concept of an "augmented graph" introduced here not only allows more efficient formulations (as shown in the new models), but also increases flexibility in modelling. As a future research direction, we intend to study the usefulness of the augmented graph as an experimental tool for designing hub networks. Some hub related problems that can be studied in this context are those with path-based formulations, which also work with graphs that do not satisfy the triangle inequality and/or allow more than two hubs on a path. Some recent examples include hub location problems with multiple capacity levels [9], service network design and hub location problem [30], modular hub location problem with single assignments [34], multiple allocation incomplete hub location problem [12], and the profit maximizing capacitated hub location problem with multiple demand classes [33].

References

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

[2] Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., OKelly, M. E., 2020. Perspectives on modeling hub location problems. European Journal of Operational Research, DOI: doi.org/10.1016/j.ejor.2020.09.039.

[3] Boland, N., Krishnamoorthy, M., Ernst, A. T., Ebery, J., 2004. Preprocessing and cutting for multiple allocation hub location problems. European Journal of Operational Research 155 (3), 638–653.

[4] Brimberg, J., Mladenović, N., Todosijević, R., Urošević, D., 2015. A basic variable neighborhood search heuristic for the uncapacitated multiple allocation p-hub center problem. Optimization Letters, DOI: 10.1007/s11590-015-0973-5.

[5] Brimberg, J., Mladenović, N., Todosijević, R., Urošević, D., 2015. General variable neighborhood search for the uncapacitated single allocation p-hub center problem. Optimization Letters, DOI: 10.1007/s11590-016-1004-x.

[6] Brimberg, J., Mladenović, N., Todosijević, R., Urošević, D., 2019. A non-triangular hub location problem. Optimization Letters, doi.org/10.1007/s11590-019-01392-2.

[7] Campbell, J. F., 1994. Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72 (2), 387–405.

[8] Cánovas, L., García, S., Marín, A., 2007. Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. European Journal of Operational Research 179 (3), 990–1007.

[9] Contreras, I., Cordeau, J.-F., Laporte, G., 2012. Exact solution of large-scale hub location problems with multiple capacity levels. Transportation Science 46 (4), 439–459.

[10] Correia, I., Nickel, S., Saldanha-da Gama, F., 2018. A stochastic multi-period capacitated multiple allocation hub location problem: Formulation and inequalities. Omega 74, 122–134.

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

[12] 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.

[13] Ernst, A. T., Hamacher, H., Jiang, H., Krishnamoorthy, M., Woeginger, G., 2009. Uncapacitated single and multiple allocation p-hub center problems. Computers & Operations Research 36 (7), 2230–2241.

[14] Ernst, A. T., Jiang, H., Krishnamoorthy, M., Baatar, H., 2011. Reformulations and computational results for uncapacitated single and multiple allocation hub covering problems. Working Paper Series 1, 1–18.

[15] Ernst, A. T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location science 4 (3), 139–154.

[16] Ernst, A. T., Krishnamoorthy, M., 1998. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. European Journal of Operational Research 104 (1), 100–112.

[17] 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.

[18] García, S., Landete, M., Marín, A., 2012. New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem. European Journal of Operational Research 220 (1), 48–57.

[19] Gavriliouk, E. O., 2009. Aggregation in hub location problems. Computers & Operations Research 36 (12), 3136–3142.

[20] 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.

[21] Hoff, A., Peiró, J., Corberán, Á., Martí, R., 2017. Heuristics for the capacitated modular hub location problem. Computers &

Operations Research 86, 94–109.

[22] Hwang, Y. H., Lee, Y. H., 2012. Uncapacitated single allocation p-hub maximal covering problem. Computers & Industrial Engineering 63 (2), 382–389.

[23] Ilić, A., Urošević, D., Brimberg, J., Mladenović, N., 2010. A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research 206 (2), 289–300.

[24] Janković, O., Mišković, S., Stanimirović, Z., Todosijević, R., 2017. Novel formulations and vns-based heuristics for single and multiple allocation p-hub maximal covering problems. Annals of Operations Research 259 (1-2), 191–216.

[25] Martí, R., Corberán, Á., Peiró, J., 2015. Scatter search for an uncapacitated p-hub median problem. Computers & Operations Research 58, 53–66.

[26] Meyer, T., Ernst, A. T., Krishnamoorthy, M., 2009. A 2-phase algorithm for solving the single allocation p-hub center problem. Computers & Operations Research 36 (12), 3143–3151.

[27] Mikić, M., Todosijević, R., Urošević, D., 2019. Less is more: General variable neighborhood search for the capacitated modular hub location problem. Computers & Operations Research 110, 101–115.

[28] Peiró, J., Corberán, A., Martí, R., 2014. Grasp for the uncapacitated r-allocation p-hub median problem. Computers & Operations Research 43 (1), 50–60.

[29] Peker, M., Kara, B. Y., 2015. The p-hub maximal covering problem and extensions for gradual decay functions. Omega 54, 158–172.

[30] Rothenbächer, A.-K., Drexl, M., Irnich, S., 2016. Branch-and-price-and-cut for a service network design and hub location problem. European Journal of Operational Research 255 (3), 935–947.

[31] Skorin-Kapov, D., Skorin-Kapov, J., O'Kelly, M., 1996. Tight linear programming relaxations of uncapacitated p-hub median problems. European Journal of Operational Research 94 (3), 582–593.

[32] Taherkhani, G., Alumur, S. A., 2019. Profit maximizing hub location problems. Omega 86, 1–15.

[33] Taherkhani, G., Alumur, S. A., Hosseini, M., 2020. Benders decomposition for the profit maximizing capacitated hub location problem with multiple demand classes. Transportation Science 54 (6), 1446–1470.

[34] Tanash, M., Contreras, I., Vidyarthi, N., 2017. An exact algorithm for the modular hub location problem with single assignments. Computers & Operations Research 85, 32–44.

[35] Todosijević, R., Urošević, D., Mladenović, N., Hanafi, S., 2015. A general variable neighborhood search for solving the uncapacitated r-allocation p-hub median problem. Optimization Letters, DOI: 10.1007/s11590-015-0867-6.

[36] Weng, K., Yang, C., Ma, Y., 2006. Two artificial intelligence heuristics in solving multiple allocation hub maximal covering problem. Intelligent Computing, 737–744.

[37] Yaman, H., 2011. Allocation strategies in hub networks. European Journal of Operational Research 211 (3), 442–451.

Appendix

4-index formulation

The four index formulation [7] uses two sets of variables: binary variables z_k, $k \in N$ that take the value 1, if and only if, a node k is used as a hub (i.e., a hub is opened at node k), and 0 otherwise; and continuous

variables y_{ijkl} which determine the portion of flow from i to j that is transferred via hubs k and l. Recall that when the triangle inequality applies throughout the network, each O-D path will traverse at most two hub nodes. The standard 4-index model (denoted as **4-index UMApHMP**) is given by (28)-(33):

$$\min \sum_{i,j,k,l \in N} t_{ij}(\gamma c_{ik} + \alpha c_{kl} + \delta c_{lj}) y_{ijkl} \tag{28}$$

subject to:

$$\sum_{k \in N} z_k = p, \tag{29}$$

$$\sum_{k,l \in N} y_{ijkl} = 1, \ \forall i, j \in N, \tag{30}$$

$$\sum_{l \in N} y_{ijkl} \leq z_k, \ \forall i, j, k \in N, \tag{31}$$

$$\sum_{k \in N} y_{ijkl} \leq z_l, \ \forall i, j, l \in N, \tag{32}$$

$$y_{ijkl} \geq 0, z_k \in \{0, 1\}, \ \forall i, j, k, l \in N, \tag{33}$$

The objective function (28) minimizes the total transportation cost. Constraint (29) ensures the establishment of exactly p hub nodes, while constraints (30) ensure that flow between each O-D pair is entirely routed. Constraints (31) and (32) guarantee that the flow between each O-D pair is routed only via hub nodes. More precisely, if $z_k = 0$, then all variables y_{ijkl} and y_{ijlk} are zero, and therefore any transfer of flow via an intermediate non-hub node k is forbidden. Finally, the last set of constraints describes the type and range of the variables used in the model.

The above model contains $n^4 + n$ variables and $1 + n^2 + 2n^3$ constraints (not including variable type in (33)).

Flow formulation of Brimberg et al. 2019 [6]

The flow formulation (FF) of [6] uses the following variables. Binary variables z_k, $k \in N$, as before, are used to indicate if node k is chosen as a hub or not. Flow variables w_{ijlm}, $i, j, l, m \in N$, which are binary, are used to determine if the flow from node i to node j is transferred along the arc (l, m), or not. The

remaining two sets of variables are also binary, and used to identify hubs to which origin and destination nodes are allocated. So, we have variables u_{ijm}, $i, j, m \in N$, such that variable u_{ijm} receives value 1, if and only if, origin node i is directly allocated to hub node m in order to transfer flow to destination node j. Similarly, variables v_{ijm} $i, j, m \in N$ receive value 1, if and only if, node j is directly allocated to hub node m in order to receive flow from node i. Hence, we have the following formulation, denoted as **FF**:

$$\min \sum_{i,j \in N} t_{ij}[(\gamma - \alpha) \sum_{m \in N} c_{im} u_{ijm} + (\delta - \alpha) \sum_{l \in N} c_{lj} v_{ijl} + \alpha \sum_{l,m \in N} c_{lm} w_{ijlm}] \tag{34}$$

subject to:

$$\sum_{m \in N} z_m = p, \tag{35}$$

$$\sum_{m \in N} w_{ijlm} - \sum_{m \in N} w_{ijml} = \begin{cases} 1, & \text{if } l = i \\ 0, & \text{if } l \neq i, j, \\ -1, & \text{if } l = j \end{cases} \quad , \ i, j, l \in N, i \neq j \tag{36}$$

$$\sum_{m \in N} w_{iilm} - \sum_{m \in N} w_{iiml} = 0, \ \forall i, l \in N \tag{37}$$

$$\sum_{m \in N} w_{iiim} = 1, \ \forall i \in N \tag{38}$$

$$\sum_{l \in N} w_{ijlm} \leq z_m, \ \forall i, j, m \in N, m \neq j \tag{39}$$

$$\sum_{l \in N} w_{ijlm} \leq z_m + 1, \ \forall i, j, m \in N, m = j \tag{40}$$

$$w_{ijij} \leq z_i + z_j, \ \forall i, j \in N \tag{41}$$

$$u_{ijm} \leq w_{ijim}, \ \forall i, j, m \in N \tag{42}$$

$$\sum_{m \in N} u_{ijm} = 1, \ \forall i, j \in N \tag{43}$$

$$v_{ijm} \leq w_{ijmj}, \ \forall i, j, m \in N \tag{44}$$

$$\sum_{m \in N} v_{ijm} = 1, \ \forall i, j \in N \tag{45}$$

$$z_m, w_{ijlm}, u_{ijm}, v_{ijm} \in \{0, 1\}, \ \forall i, j, l, m \in N. \tag{46}$$

The objective function and constraint (35) have the same meaning as in the previous models. Constraints (36) are the flow conservation constraints for each O-D pair (i, j), $i \neq j$. On the other hand, constraints (37) and (38) ensure the flow conservation on the cycle that appears in the case $i = j$, and guarantee that the transfer of flow is accomplished along a cycle. Constraints (39) - (41) imply that solely hub nodes are used as intermediate nodes on a path (cycle) established between an O-D pair. In addition, constraints (41) allow direct transportation between an O-D pair only if at least one of them is a hub. Constraints (42)-(45) determine direct node-hub assignments. Namely, for each O-D pair (i, j) on the established path from i to j, there is exactly one hub assigned directly to i and exactly one hub assigned directly to j. Constraints (46) explicitly define the variables of the model.

This model has $n + 2n^3 + n^4$ variables and $1 + n + 3n^2 + 4n^3$ constraints (not including variable type constraints (46)).