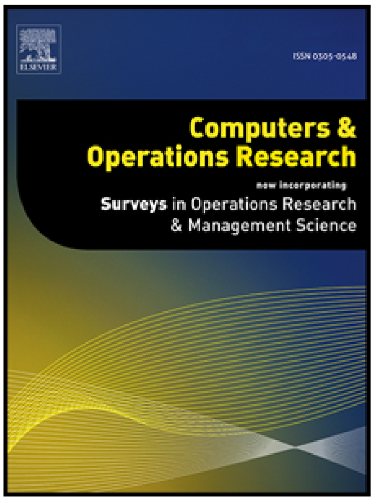
Accepted Manuscript



The 2-Allocation p-Hub Median Problem and a Modified Benders Decomposition Method for Solving Hub Location Problems

Hamid Mokhtar, Mohan Krishnamoorthy, Andreas T. Ernst

PII: S0305-0548(18)30247-8
DOI: https://doi.org/10.1016/j.cor.2018.09.006
Reference: CAOR 4554

To appear in: *Computers and Operations Research*

Received date: 10 October 2017
Revised date: 29 August 2018
Accepted date: 16 September 2018

Please cite this article as: Hamid Mokhtar, Mohan Krishnamoorthy, Andreas T. Ernst, The 2-Allocation p-Hub Median Problem and a Modified Benders Decomposition Method for Solving Hub Location Problems, *Computers and Operations Research* (2018), doi: https://doi.org/10.1016/j.cor.2018.09.006

The 2-Allocation p-Hub Median Problem and a Modified Benders Decomposition Method for Solving Hub Location Problems

Hamid Mokhtar*[1], Mohan Krishnamoorthy†[1,2], and Andreas T. Ernst‡ [3]

[1]School of IT & Electrical Engineering, The University of Queensland QLD 4072, Australia
[2]Department of Mechanical & Aerospace Engineering, Monash University, Clayton VIC 3800, Australia
[3]School of Mathematical Sciences, Monash University, Clayton VIC 3800, Australia

Abstract

We study the *uncapacitated 2-allocation p-hub median problem* (U2ApHMP), which is a special case of the well-studied *hub median problem*. The hub median problem designs a hub network in which the location of p hubs needs to be decided (the hubs are fully interconnected). The other nodes (known as access nodes) in the hub median problem are then allocated to one or many hubs. In the U2ApHMP, each access node is allocated to *exactly* two hubs. We discuss how this problem provides an alternative network design option for well-known p-hub median problems. We show its relevance and usefulness in the context of survivable network design and show that it addresses network survivability, a feature that has often been largely overlooked in hub network design research to date. We show that U2ApHMP is NP-hard even for a fixed/known set of hubs. We propose a mathematical formulation and develop a modified Benders decomposition method for this problem. In this, we convert the corresponding subproblems to minimum cost network flow problems. This allows us to solve large instances efficiently. We believe that, while our resulting method solves the U2ApHMP efficiently, it is also generalisable and can potentially be employed for solving other classes and types of hub location problems too.

keywords: Hub Location, p-Hub Median, Benders Decomposition, Location-Allocation, Survivability

1 Introduction

Hubs are employed in several network design contexts that involve flow interchange between nodes and are often used in the design of, for example, airline networks, parcel delivery networks, and telecommunication networks. Flow between nodes (referred to as *access nodes*) is routed via *hubs*, each of which acts as a consolidator and forwarder. The (volume) flow between the hubs is discounted because of the large volumes that are presumed to accrue from flow consolidation. Given a fixed/known positive integer p, we either get the uncapacitated single allocation p-hub median problem (USApHMP), if each access-node is allocated to exactly one hub, or the uncapacitated multiple allocation p-hub median problem (UMApHMP), if access nodes may be allocated to multiple hubs.

The hub location problem (HLP) has been well-studied in the literature. Following seminal works of O'Kelly [33, 34], a few hub median problems were introduced and formulated by Campbell [6, 7].

*Corresponding author. email: h.mokhtar@uq.edu.au
†email: m.krishnamoorthy@uq.edu.au
‡email: andreas.ernst@monash.edu

The HLP has been studied in several contexts including telecommunications, parcel delivery systems, airline hub design and transportation networks (Çetiner et al. [10], Ernst and Krishnamoorthy [14, 15, 16], Jaillet et al. [20], Klincewicz [24], Powell and Sheffi [37], Sen et al. [39, 40]). There are a few surveys which review early and recent works on HLP classification, modelling, and solution methodologies (Alumur and Kara [3], Campbell et al. [8], Farahani et al. [17]). An alternative network design whose operational costs are between those of USApHMP and UMApHMP is studied by yam [1]. They studied a variation of the p-hub median problem in which any non-hub is allocated to at most r hubs. This problem was called UrApHMP. The problem we study in this paper is a special case of UrApHMP with further considerations. We discuss the differences and other considerations later on in this section.

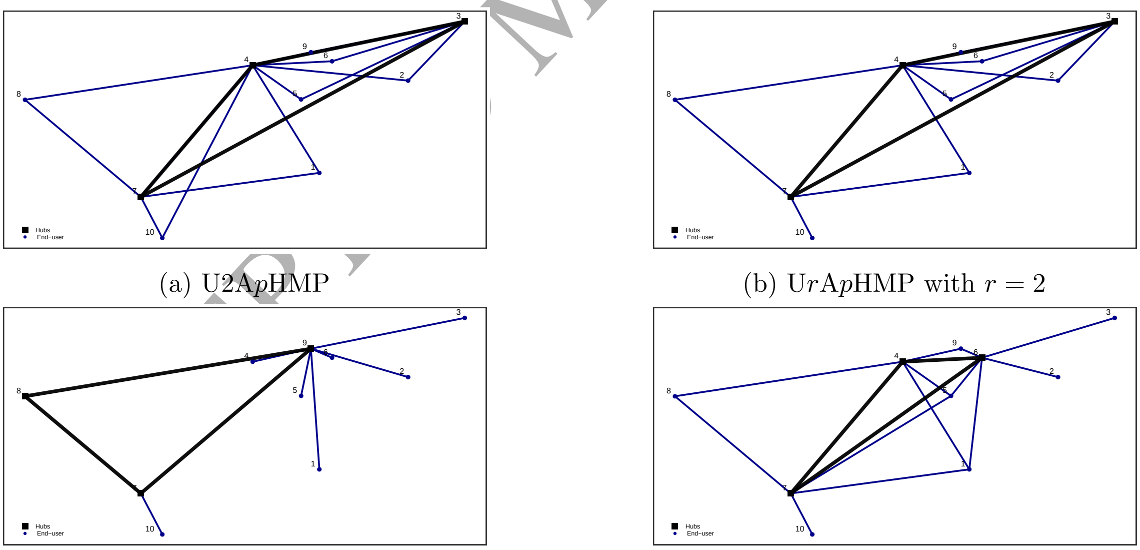
Ernst and Krishnamoorthy [14] present a compact 3-index formulation for USApHMP and UMApHMP and provided exact solution approaches for these problems (Ernst and Krishnamoorthy [14, 15, 16]). Recently, modified Benders decomposition methods have been developed, with remarkable success, for solving some classes of HLPs (de Camargo et al. [12] and Contreras et al. [11]). Through these novel methods it is now possible to solve reasonably large instances of a certain class of HLPs to optimality. Besides exact methods, heuristic approaches have also been used for solving the HLP (see, for example, Çetiner et al. [10], Klincewicz [25], Skorin-Kapov and Skorin-Kapov [41], Smith et al. [42], Yaman [48]).

Despite all the attention in the literature on the study of HLPs, there has not been a significant amount of attention paid to hub network survivability. This requirement is particularly relevant in electrical and telecommunication networks that have a hub topology. While a hub topology achieves decreased costs through flow aggregation, we are not assured that there will always be a path in the network for flow between each origin-destination pair in the network. In particular, in data location problems, content delivery networks employ hub networks to utilise multiple servers for responding to customer demands (Sen and Krishnamoorthy [38], Sen et al. [40]). This context is especially important for Video-on-Demand systems, where content and videos are distributed among multiple servers and end-users are constantly connected to the server network (Sen et al. [39]). In such applications, it is necessary to have survivable networks in the case of component failures.

The design of *survivable hub networks* has, however, started to receive some attention in the literature ever since it was documented that hub networks are vulnerable and suffer from resilience drawbacks (O'Kelly [35]). In fact, there is a large body of research in the network design literature in which k-connectivity of networks for $k \geq 2$ is required to ensure survivability. In the design of telecommunications (hub) networks, the 2-connectivity, also called 'survivability constraint', is widely used to increase fault-tolerance of networks (Cardwell et al. [9], Grötschel et al. [19], Monma and Shallcross [32], Soni et al. [44], Xu [47]). Kim and O'Kelly [22] considered the p-hub median problem in which the expected network throughput is maximised when the probability of disruption for each origin-destination route is taken into the account. Kim [21] and An et al. [4] consider the design of secondary routes for origin-destination pairs through a fixed number of back-up hubs which are different from the primary hubs that are chosen. As we can see, to date, the research thrust has been on the backbone (hub) network, and on the development of back-up hubs and routes through these. This approach, while expensive, may not be an option for many applications because it ignores the vulnerability of end-users that results from disruptions to (and failures of) access links.

In this paper, we examine a variation of a special case of UrApHMP problem (yam [1]). Our problem is, indeed UrApHMP with $r = 2$ in which we require each non-hub node to be allocated to *exactly* 2 hubs and each allocation involves a fixed cost. This modification enables us to generate a hub network that is able to survive *access link* failures. We assume, without loss of generality, that access link failures can only happen one at a time for a relatively short period of time and we further assume that the backbone hub network is robust and is free from failure. Given a graph, $G = (N, A)$, where N is the set of nodes and A is the set of arcs, flow demands between all pairs of nodes, and a fixed number p of hub nodes, our objective is to design a least-cost 2-connected network with minimum transfer and facility establishment costs in which every node is connected to exactly two hubs. The establishment cost is the total fixed cost for hubs and access links. The former is related to providing facilities for hub operations, and the latter is related to constructing or leasing links between end-users and hubs (especially in telecommunication applications). We call this variation of the hub median problems the *uncapacitated 2-allocation p-hub median problem* (U2ApHMP). Any solution of this problem is a 2-connected network.

We could design a hub network with exactly r allocations of non-hubs to hubs, for $r = 2, 3, \ldots, p$. However, we are particularly interested in the 2-allocation problem to ensure 2-connectivity of the network with least establishment costs. Figure 1 illustrates optimal solutions of four variations of the hub median problem. Clearly in this instance, the only 2-connected network is the one for U2ApHMP. In general, U2ApHMP and UrApHMP with $r = 2$ can have different solutions. Figure 2 illustrates an example where the fixed costs of access links are large enough, as compared to routing costs, so that the network designs by U2ApHMP and UrApHMP with $r = 2$ are quite different. However, when access link fixed costs are sufficiently small, a solution by UrApHMP with $r = 2$ can be used to get a solution for U2ApHMP. Defining and addressing the survivability of non-hub nodes is not unique. For instance, the flow for each request could be split into several paths to impose a minimum number of node-disjoint paths carry each flow request. Hence, our approach is a first step in this direction.



(a) U2ApHMP

(b) UrApHMP with $r = 2$

(c) USApHMP

(d) UMApHMP

Figure 1: Solutions of different problems on CAB with $n = 10, p = 3$ and $\alpha = 0.4$

An important motivating application of U2ApHMP is telecommunication networks where constant connectivity is vital. In this problem, we consider a fixed cost for each access link to reflect economic factors for construction or leasing costs. While both access links of each non-hub in optimal solutions of U2ApHMP carry flow in most cases, in some cases, an access link is only reserved for back-up paths without carrying (primary) flow. The choice of this (backup) access link is based on access link fixed

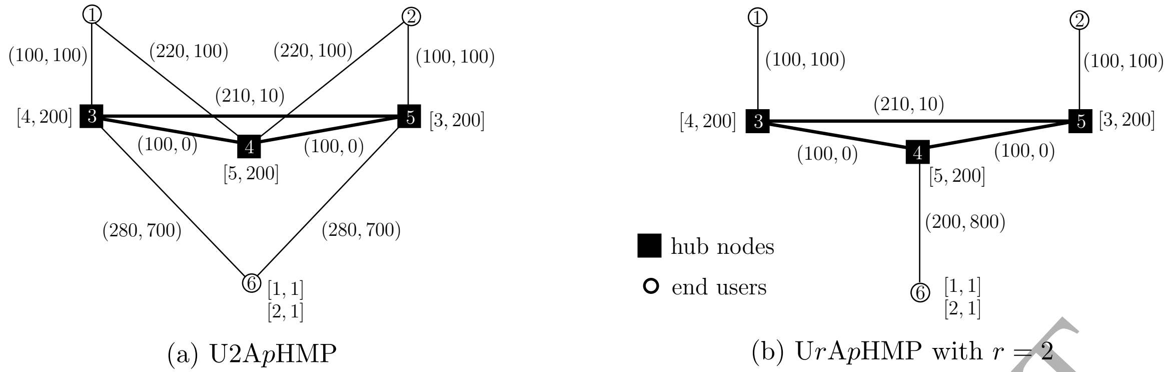


Figure 2: Solutions of U2ApHMP and UrApHMP for an instance. Every none-zero flow demand $[d, f]$ is shown near its origin node, where d and f are respectively the destination node and the amount flow demand. For each link (w, c) shows its weight w and its fixed cost c.

costs. In such cases, since we assume that the duration of link disruptions is relatively small, a back-up path does not need to be optimal in the disrupted network. In fact, the fixed costs for chosen access links during normal situations plays a larger role than the operational cost of back-up routes when an access link is broken. We believe this addresses network survivability for hub median designs in the event of access link failures. It provides a network design for assuring survivability while minimising network operation and establishment costs. We propose a formulation, and show that this problem is NP-hard. We show that the problem remains NP-hard even if the set of hubs is fixed. We develop an acceleration of Benders decomposition for this problem to solve large instances efficiently.

Despite similarities between the 2-allocation problem and the single- or multiple-allocation p-hub median problems, there are key differences between them which make the 2-allocation problem applicable for a survivable and economic network design. There is a trade-off between establishment cost and operating cost of a network (as indicated in yam [1]). Since any 2-allocation optimal set of routings is also feasible for the multiple-allocation problem, and any single-allocation optimal set of routings is feasible for the 2-allocation problem, we should expect the operational cost of U2ApHMP to be somewhere between the single-allocation and the multiple-allocation cases for the same instance. As long as fixed costs are relatively small, the same relative order for optimal solutions of those three problems also holds. In an experiment, the total costs of network designs by U2ApHMP on the CAB data (see Table 7 in Section 4) is on average 4.7% cheaper when compared to USApHMP, and 1% more expensive when compared with UMApHMP. This is even though the number of access links in U2ApHMP is twice that required by USApHMP and, on average, 96% of the number required by UMApHMP. Therefore, the 2-allocation method provides an option for designing a network in which operational costs are between the costs of the two well-known hub models, but with a greater survivability in case of an access link failure. Hence, U2ApHMP can be a cost-effective alternative for USApHMP and UMApHMP with an added survivability feature.

Survivability is a feature of the U2ApHMP design and is an attempt to avoid severe costs of network disruptions (Stoer [45]). Our design provides at least two mutually node-disjoint paths for each flow demand; a need that is overlooked in USApHMP and UMApHMP network designs. In UMApHMP, there is no guarantee that a non-hub node is allocated to more than one hub. Note that UrApHMP (yam [1]) solutions may have access nodes that are allocated to exactly one hub. In UrApHMP, the number of allocations is smaller in hub networks with a smaller transfer factor. Thus, a network design for UrApHMP might be the same network design for USApHMP. In special cases, when the transfer discount factor is very small, the solutions of USApHMP, UrApHMP, and UMApHMP may be the same and each non-hub may be singly-allocated. For cases where the access link fixed costs are relatively small, a solution for U2ApHMP can be obtained from a solution of UrApHMP with $r = 2$, by allocating the lowest cost unused access link as the second allocation for each singly-allocated non-hub.

In general, a 2-allocation hub network design may not necessarily be an optimally robust network design. This is because all flow on the disrupted link has to be (at least momentarily) routed through the second link, and this may not be the best choice for the flow that has been disrupted. However, this does provide an alternative to a much more difficult problem (see, for example, Kim and Ryerson [23], Matisziw et al. [28]) that could be posed which considers the determination of a network design in which all access nodes are connected via the globally best (least cost) second best option in which all flow is routed by the alternate link in the even that the primary link fails. That problem, we believe is a much harder problem than the U2ApHMP that we have posed for designing survivable hub networks. Our approach here is to provide the possibility for restoration of connections when there is a faulty link. The U2ApHMP is an alternative to the UMApHMP and USApHMP network designs with the additional guarantee that the network can survive one access link failure at a time. The design of robust hub networks, as posed above, is a possible extension to the current study.

In this paper, we develop an improved Benders decomposition method for the U2ApHMP. Benders decomposition has been widely implemented on large mixed integer problems. In this method, cuts are added to a relaxation of the problem iteratively. In many cases the choice of these cuts could lead to slow convergence of the method. This issue was noticed and addressed by Magnanti and Wong [27] who introduced the generation of pareto optimal cuts using some 'core point'. This improvement is employed in many implementations of the Benders method, including the one by Contreras et al. [11] for the hub location problem with multiple allocations. In the current paper we take advantage of these pareto optimal cuts. We enhance this approach by choosing better core points and generating stronger cuts. We also come up with a more efficient approach for solving subproblems to generate cuts.

In Section 2, we provide a mathematical formulation of U2ApHMP. We show that U2ApHMP is NP-hard even for a fixed set of hubs. We use a Benders decomposition approach, and then develop a modification of the Benders decomposition that enables us to efficiently solve large problem instances. This is described in Section 3. As we observe, an implementation of original Benders decomposition does not result in an efficient solution algorithm due to high degeneracy of the subproblems which then leads to a slow convergence to the optimal solution. We show that our modified Benders method is more efficient, and by converting the subproblems to minimum cost network flow problems and through the use of more effective Benders cuts, we improve its performance further. Our computational results, presented in Section 4, indicate that our method results in fewer iterations and faster running times. Also it is more efficient than the built-in Benders decomposition of `CPLEX 12.7`, which is the original Benders method implemented in `CPLEX` solver.

2 Problem Statement

We are given a positive integer p, a set of n nodes $N = \{1, 2, \dots, n\}$, and distances between each pair of nodes, where d_{ij} denotes the distance between nodes i and j. A trivial assumption here is that $n \geq p$ and $p \geq 2$ as we require each node to be allocated to exactly 2 hubs. We assume that the triangle inequality for distances between nodes holds. We consider a complete digraph $G = (N, A)$, where A

is the set of arcs (i,j), $i,j \in N$ and $i \neq j$, so that the weight of each link is the distance between its endpoints. We suppose that hubs are connected through a complete graph on the set of hubs, and non-hub nodes are only connected to hubs. For every pair of nodes $(i,j) \in N \times N$, let W_{ij} denote the amount of flow demand from i to j. Assume that $d_{ij} \geq 0$ and $W_{ij} \geq 0$. The establishment of a node as a hub is associated with a fixed cost. Problems with fixed costs are more general since they can always be set to zero.

In practice, the actual cost of flow between different types of nodes is computed with different cost coefficients: the *collection* coefficient corresponds to flow from a non-hub to a hub, the *distribution* coefficient corresponds to flow from a hub to a non-hub, and the *transfer* coefficient corresponds to flow between hubs. These are denoted by χ, δ and α, respectively. Usually $\alpha \leq 1$, $\chi \geq \alpha$ and $\delta \geq \alpha$ in practical applications. The *uncapacitated* 2*-allocation p-hub median problem* (U2ApHMP) is the problem of locating p hubs among n nodes in N, and allocating each non-hub node to exactly 2 hubs with minimum total cost of fulfilling flow demands.

2.1 Mathematical Formulation

Without loss of generality, we assume that all flow must be routed through at most two hubs since using two hubs always reduces the cost when compared to routing flow through three or more hubs because of the triangular inequality assumption. Therefore, any path between i and j must contain three links, (i,k), (k,l), and (l,j), where i and j are connected to hubs k and l respectively. We denote such path by $i-k-l-j$. Then the cost of using the $i-k-l-j$ path, considering the cost coefficients of different link types, is

$$C_{ijkl} = \chi d_{ik} + \alpha d_{kl} + \delta d_{lj}.$$

Note that the costs and flow demands of (i,j) and (j,i) may not be equal, for any $i,j \in N$.

Let F_k be the cost of establishing node k as hub, and G_{ik} be the establishment cost of access link (i,k).

Let binary decision variable $h_k = 1$ if node k is chosen as hub, and $h_k = 0$ otherwise, for all $k \in N$.

Let $z_{ik} = 1$ if node i is connected to hub k, and $z_{ik} = 0$ otherwise, for all $i,k \in N$.

Let x_{ijkl} be the fraction of flow request W_{ij} that is sent on the $i-k-l-j$ path, for all $i,j,k,l \in N$.

We present an integer linear programming formulation of U2ApHMP below:

$$\text{U2A}p\text{HMP:} \quad \min \quad \sum_{k\in N} F_k\, h_k + \sum_{i\in N}\sum_{k\in N} G_{ik}\, z_{ik} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{l\in N} C_{ijkl}\, W_{ij}\, x_{ijkl} \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in N} h_k = p, \tag{2}$$

$$\sum_{k\in N} z_{ik} = 2, \qquad \forall i \in N \tag{3}$$

$$z_{ik} \leq h_k, \qquad \forall i,k \in N \tag{4}$$

$$\sum_{k\in N}\sum_{l\in N} x_{ijkl} = 1, \qquad \forall i,j \in N \tag{5}$$

$$\sum_{l\in N} x_{ijkl} \leq z_{ik}, \qquad \forall i,j,k \in N \tag{6}$$

$$\sum_{k\in N} x_{ijkl} \leq z_{jl}, \qquad \forall i,j,l \in N \quad (7)$$

$$h_k, z_{ik} \in \{0,1\}, x_{ijkl} \geq 0 \qquad \forall i,j,k,l \in N. \quad (8)$$

In the above formulation, equation (2) corresponds to locating p hubs. The set of equations (3) forces the allocation of each node to exactly 2 hubs, while equations (4) ensure that each node is allocated only to hubs. The set of constraints (5)-(7) fulfils commodity flow request from i to j through established links between nodes and hubs. The objective function (1) represents total cost of hub establishments and transfer costs. In this formulation we have $n^4 + n^2 + n$ variables ($n^2 + n$ of these are binary) and $2n^3 + 2n^2 + n + 1$ constrains.

Note that in the above formulation, each hub will have two allocations in the optimal solution, typically to itself and to one other hub. Whether this type of 'backup' allocation makes sense for hubs depends on the details of the application, but makes no difference to the optimal cost as the second allocation will have zero flow. Hence we do not specifically prohibit the number of allocations for hubs.

2.2 Complexity of U2ApHMP

It is known that p-hub median problems are NP-hard in general (Love et al. [26]). While UMApHMP with fixed location of hubs can be solved polynomially (Ernst and Krishnamoorthy [16]), it is known that USApHMP for $p \geq 3$ is NP-hard even for the special case in which the location of hubs are fixed (Love et al. [26], Sohn and Park [43]). In the following we prove U2ApHMP is as hard as USApHMP.

Theorem 2.1. U2ApHMP *is NP-hard, even when the location of hubs are fixed.*

Proof. Suppose we are given an instance of USApHMP in which $\alpha = \delta = \chi = 1$, $d_{ij} = d_{ji}$ for all $i,j \in N$, $G_{ik} = 0$ for all $i,k \in N$, and suppose $H \subset N$ is a set of p fixed hubs for this instance. We show that USApHMP with the set of hubs H is polynomially reducible to an instance of U2ApHMP with a fixed set of hubs. Choose a node v and construct an instance of the uncapacitated 2-allocation problem on $n+1$ distinct nodes $N \cup \{v\}$, such that, for some $\epsilon > 0$,

$$d_{iv} + \epsilon \leq d_{ih} + d_{hv}, \quad \forall h \in H, i \in N, \tag{9}$$

and

$$d_{iv} + d_{jv} > \max_{h,l\in H}\{d_{ih} + d_{hl} + d_{lj}\}, \quad \forall i,j \in N. \tag{10}$$

Such a node v exists, and can be obtained polynomially by choosing a point which does not lie on $(n-p)p$ lines, each of which pass through a hub and a non-hub. Now for $i \in N$, set $W_{vi} = 0$, and set W_{iv} to be large enough such that:

$$\epsilon\, W_{iv} > \bar{d}\, \overline{W}_i,$$

where $\bar{d} = \max_{h,l\in H} d_{hl} + 2\max_{i\in N, h\in H} d_{ih}$, and $\overline{W}_i = \sum_{j=1}^{n} W_{ij}$. Denote this instance of U2ApHMP by P_v (see Figure 3 for a depiction). Furthermore, let $H \cup \{v\}$ be a set of $p+1$ fixed hubs for this instance.

Suppose in an optimal allocation for USApHMP with hubs in H, node i is allocated to $a(i) \in H$. We show that this optimal allocation gives rise to an optimal allocation of i to $\{a(i), v\}$, $1 \leq i \leq n$, for P_v.

Every $i \in N$ must be allocated to v in any optimal allocation of P_v; otherwise, if some $i \in N$ is allocated to h and h', where $h \neq v, h' \neq v$, then, without loss of generality, the total flow cost from i is at least

$$(d_{ih} + d_{hv})W_{iv}.$$

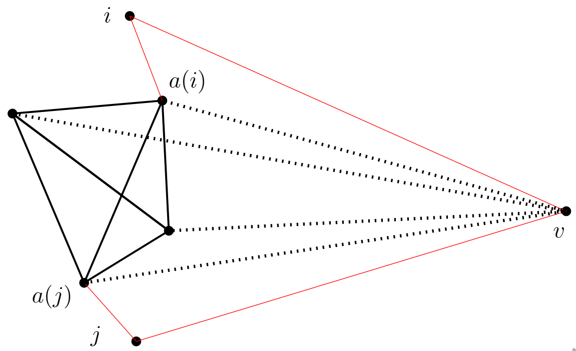


Figure 3: Construction of an instance of U2ApHMP from an instance of USApHMP

However, when node i is allocated to l and v, for some $l \in H$, and j allocated to some $q \in H$, then, by (9) and (10), the total cost of flows from i to all other nodes is

$$\sum_{j=1}^{n}(d_{il} + d_{lq} + d_{qj})W_{ij} + d_{iv}\,W_{iv} \leq \bar{d}\,\overline{W}_i + d_{iv}W_{iv} < \epsilon\,W_{iv} + d_{iv}W_{iv} \leq (d_{ih} + d_{hv})W_{iv}.$$

Hence, i must be allocated to v in any optimal allocation. On the other hand, by (10) and the optimality of allocation of i to $a(i)$ in the single allocation problem, for any $h, l \in H$,

$$\sum_{j\in N}(d_{ia(i)} + d_{a(i)a(j)} + d_{a(j)j})W_{ij} \leq \sum_{j\in N}(d_{ih} + d_{hl} + d_{lj})W_{ij} < \sum_{j\in N}(d_{iv} + d_{vj})W_{ij},$$

which implies that i must be allocated to $a(i)$ in any optimal solution to P_v.

Similarly, any optimal solution of P_v gives rise to an optimal solution for the single allocation problem. Therefore, any USApHMP with a fixed set of hubs can be reduced to a U2A$(p+1)$HMP with some fixed set of hubs. Hence, we have the result. □

By generalising Theorem 2.1, we claim that for any fixed r, $2 \leq r < p$, the uncapacitated r-allocation p-hub median problem (UrApHMP), in which the number of allocations for each node is exactly r, is NP-hard, even for a fixed set of hubs. Assuming U$(r-1)$ApHMP is NP hard, by induction we can construct an instance of UrApHMP from U$(r-1)$ApHMP such that they are polynomially reducible to each other using a proof that is similar to that used in Theorem 2.1.

In our computational experiments, we were not able to solve large instances of U2ApHMP using commercial solvers like `CPLEX`. We struggled to find optimal solutions to U2ApHMP even for instances with only 60 nodes within a reasonable time limit (see Section 4 for more details).

The major difficulty in dealing with this problem is the large number of variables and constraints in its formulation (the flow variables x_{ijkl} and constraints (5)-(7) comprise the majority of the variables and constraints). It is possible to employ more compact formulations, however they are weaker in general [14]. These observations lead us to the idea of solving this problem using Benders decomposition for (1)-(8).

3 Benders Decomposition

Benders decomposition method is a partitioning algorithm applied to mixed integer programming and nonlinear integer programming problems (Benders [5]). This method has been widely used for solving a wide range of difficult problems, including hub median problems. Specifically, Benders decomposition is shown to be effective for solving large instances of hub location problems (Contreras et al. [11], de Camargo et al. [12]).

In this method, the original problem is decomposed into a master problem MP, which may consist of integer variables and corresponding constraints, and a subproblem SP, which consists of the remaining variables and constraints. MP and SP are solved iteratively in a dependant manner. Hence, MP is a relaxation of the original problem, and SP is constructed via a feasible solution of MP at each iteration. If SP is not feasible, then the solution of MP is not feasible for the original problem, and it will be excluded from the MP feasible region by a *feasibility* Benders cut which is generated by the dual of SP. Otherwise, an *optimality* Benders cut will be added to MP to improve the current MP solution, until no further improvement is needed. An advantage of this method is that larger instances of problems can be solved, since MP and SP are often more tractable than the original problem. A solution may be obtained faster even though MP and SP may be solved a number of times. We first describe Benders method for U2ApHMP, and in Sections 3.2 and 3.4, we discuss how we may generate stronger cuts.

In order to apply the Benders decomposition method to U2ApHMP, in each iteration, the location and allocation variables, h_k and z_{ik} respectively, are fixed to some $\hat{h}_k$ and $\hat{z}_{ik}$ respectively, $i, k \in N$. Therefore, we obtain a linear programming subproblem in the iteration corresponding to vectors $\hat{\boldsymbol{h}} = (\hat{h}_k)$ and $\hat{\boldsymbol{z}} = (\hat{z}_{ik})$. A formulation of the corresponding subproblem is given below.

$$\min \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{l\in N} C_{ijkl}\, W_{ij}\, x_{ijkl} \tag{11}$$

$$\text{s.t.} \quad \sum_{k\in N}\sum_{l\in N} x_{ijkl} = 1, \qquad \forall i,j \in N \tag{12}$$

$$\sum_{l\in N} x_{ijkl} \leq \hat{z}_{ik}, \qquad \forall i,j,k \in N \tag{13}$$

$$\sum_{k\in N} x_{ijkl} \leq \hat{z}_{jl}, \qquad \forall i,j,l \in N \tag{14}$$

$$x_{ijkl} \geq 0 \qquad \forall i,j,k,l \in N. \tag{15}$$

The above problem is clearly a routing problem for n^2 pairs of nodes, where the underlying network is defined by $\hat{\boldsymbol{h}}$ and $\hat{\boldsymbol{z}}$. By associating dual variables, f_{ij} to the set of constraints (12), (u_{ijk}) to the set of constraints (13), and (v_{ijl}) to the set of constraints (14), the dual of subproblem (11)-(15) is as follows.

$$\begin{aligned} \max \quad & \sum_{i\in N}\sum_{j\in N} f_{ij} - \sum_{i\in N}\sum_{j\in N}\sum_{k\in N} \hat{z}_{ik} u_{ijk} - \sum_{i\in N}\sum_{j\in N}\sum_{l\in N} \hat{z}_{jl} v_{ijl} \\ \text{s.t.} \quad & f_{ij} - u_{ijk} - v_{ijl} \leq C_{ijkl}\, W_{ij}, && \forall i,j,k,l \in N \\ & u_{ijk}, v_{ijl} \geq 0, f_{ij} \in \mathbb{R} && \forall i,j,k,l \in N. \end{aligned}$$

An optimal solution $(\hat{\boldsymbol{f}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ of the above dual problem gives the following Benders cut:

$$\eta \geq \sum_{i\in N}\sum_{j\in N} \hat{f}_{ij} - \sum_{i\in N}\sum_{j\in N}\sum_{k\in N} z_{ik}\hat{u}_{ijk} - \sum_{i\in N}\sum_{j\in N}\sum_{l\in N} z_{jl}\hat{v}_{ijl}, \tag{16}$$

where η is a real non-negative variable since the right side of (16) is equal to (11) which is always positive. Then this optimality cut is added to the master problem which is formulated below:

$$\begin{aligned} \min \quad & \sum_{k\in N} F_k\, h_k + \sum_{i\in N}\sum_{k\in N} G_{ik}\, z_{ik} + \eta \\ \text{s.t.} \quad & (2)-(4),(8) \\ & \eta + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N} z_{ik}\hat{u}_{ijk} + \sum_{i\in N}\sum_{j\in N}\sum_{l\in N} z_{jl}\hat{v}_{ijl} \geq \sum_{i\in N}\sum_{j\in N} \hat{f}_{ij} \\ & \eta \geq 0 \end{aligned}$$

Clearly, for any valid location of hubs $\hat{\boldsymbol{h}}$ and allocation of non-hubs $\hat{\boldsymbol{z}}$, the subproblem (11)-(15) is always feasible and bounded. In fact, every demand can be served by some path in the network constructed by $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}})$. Thus, we have the following lemma.

Lemma 3.1. *The dual of subproblem* (11)-(15) *is always feasible and bounded for any* $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}})$ *satisfying* (2)-(4) *and* (8)*.*

As a corollary, in the Benders decomposition method for U2ApHMP, only optimality cuts are added to the master problem.

3.1 Generating Multicuts

Note that the subproblem (11)-(15) can be further decomposed into n^2 subproblems. This is because we can find an optimal routing for each pair of nodes separately/independently. In return, the dual of the subproblems can be expressed for any pair of nodes. This decomposition results in n^2 Benders cuts, which may provide tighter cuts for the master problem. Consequently, this could result in faster convergence.

De Camargo et al. [12] reported that the generation of n^2 cuts, and adding them to their MP was not efficient in comparison to having one cut. This was due to a huge increase in the size of the MP. However, another reason for this difficulty could be as a result of using inefficient methods for solving the n^2 subproblems at each iteration. In fact, we observed that generating n^2 cuts is much more effective than using the one cut obtained by (16). The number of iterations is often reduced due to a tightening of the MP formulation in general.

In Section 3.3 we discuss an efficient approach to solve subproblems. The dual problem for any pair of nodes $(i,j) \in N^2$ is as follows:

$$\text{DS}_{ij}: \qquad \max \quad f_{ij} - \sum_{k\in N} \hat{z}_{ik}u_{ijk} - \sum_{l\in N} \hat{z}_{jl}v_{ijl} \tag{17}$$

$$\text{s.t.} \quad f_{ij} - u_{ijk} - v_{ijl} \leq C_{ijkl}\, W_{ij}, \qquad \forall k,l \in N \tag{18}$$

$$u_{ijk}, v_{ijl} \geq 0, f_{ij} \in \mathbb{R} \qquad \forall k,l \in N. \tag{19}$$

By an optimal solution $(\hat{f}_{ij}, \hat{\boldsymbol{u}}_{ij}, \hat{\boldsymbol{v}}_{ij})$ of DS_{ij} for each $(i,j) \in N^2$, a Benders cut is generated. Thus, in each iteration we obtain n^2 Benders cuts:

$$\eta_{ij} \geq \hat{f}_{ij} - \sum_{k \in N} z_{ik} \hat{u}_{ijk} - \sum_{l \in N} z_{jl} \hat{v}_{ijl} \quad \forall i,j \in N, \tag{20}$$

where η_{ij} is a real non-negative variable and $\eta = \sum_{i,j} \eta_{ij}$. By this decomposition, the master problem becomes:

$$\begin{aligned} \text{MP:} \qquad & \min \quad \sum_{k \in N} F_k\, h_k + \sum_{i \in N} \sum_{k \in N} G_{ik}\, z_{ik} + \sum_{i \in N} \sum_{j \in N} \eta_{ij} \\ & \text{s.t.} \quad (2) - (4), (20) \\ & \qquad\;\; h_k, z_{ik}, \eta_{ij} \geq 0 \qquad \forall i,k,j \in N. \end{aligned} \tag{21}$$

Note that the objective in DS_{ij} is to maximise $f_{ij} - \sum_k u_{ijk} - \sum_l v_{ijl}$, where the first sum is taken over k for which $\hat{z}_{ik} = 1$ and the second sum is over l for which $\hat{z}_{jl} = 1$. So, an optimal solution for DS_{ij} is $\hat{u}_{ijk} = 0$ when $\hat{z}_{ik} = 1$, and $\hat{v}_{ijl} = 0$ when $\hat{z}_{jl} = 1$,

$$\hat{f}_{ij} = \hat{\delta}_{ij} := \min_{\hat{z}_{ik} = \hat{z}_{jl} = 1} C_{ijkl}\, W_{ij}, \tag{22}$$

and arbitrary non-negative values for the remaining variables.

There are two computational issues with the above decomposition. First, in many implementations of this method for U2ApHMP, its convergence is very slow and a large number of iterations is required to reach optimality. Second, the computational effort to solve n^2 subproblems, each of which with n^2 variables and n^2 constraints in each iteration of Benders method is very expensive. To tackle these issues, we first develop a modification of Benders decomposition. For the second issue, we model subproblems as minimum cost network flow problems to solve them more efficiently.

3.2 Accelerating the Benders Decomposition Approach for Solving HLPs

The optimal solution of DS_{ij} is not unique since the subproblem is degenerate. As a result, Benders cuts exist for the MP, with different strengths. The strength of Benders cuts (20) is dependent on the choice of optimal solutions of DS_{ij}. Magnanti and Wong [27] proposed an acceleration of the Benders method, in which a second LP is constructed from the dual of the subproblem to maximise a weighted summation of the dual variables among optimal solutions. For this purpose, they used a point in the relative interior of the convex hull of the master problem, called a *core point*, to define a weight for the dual variables. They showed that the optimal solution of that LP results in a 'pareto optimal cut' [27, Theorem 1]. A cut is pareto optimal if it is not dominated by any other cut and a cut obtained by $(\hat{f}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ is dominated by a cut obtained by $(\tilde{f}, \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ if for all feasible $\boldsymbol{z}$ we have

$$\tilde{f} - \sum_{k \in N} z_{ik} \tilde{u}_k - \sum_{l \in N} z_{jl} \tilde{v}_l \geq \hat{f} - \sum_{k \in N} z_{ik} \hat{u}_k - \sum_{l \in N} z_{jl} \hat{v}_l,$$

and at least for one feasible $\boldsymbol{z}$ the above inequality holds strictly. Fischetti et al. [18] also judiciously chose the dual objective function of the SP and converted the SP to a 'minimal infeasible subsystem' which resulted in a more effective choice of Benders cuts.

Let m_{ik}, m_{jl} for $k, l \in N$ be non-negative real parameters and m_0 be a real parameter. For $(m_0, \boldsymbol{m_{ij}}) = (m_0, m_{i1}, \ldots, m_{in}, m_{j1}, \ldots, m_{jn})$, we consider the following problem to generate cuts:

$$\max \quad m_0 f_{ij} - \sum_{k \in N} m_{ik} u_{ijk} - \sum_{l \in N} m_{jl} v_{ijl} \tag{23}$$

$$\text{s.t.} \quad f_{ij} - \sum_{k \in N} \hat{z}_{ik} u_{ijk} - \sum_{l \in N} \hat{z}_{jl} v_{ijl} = \hat{\delta}_{ij} \tag{24}$$

$$(18) - (19)$$

Any feasible point in the above problem is an optimal solution for (17)-(19). Furthermore, when $m_0 = 1$ and $(\boldsymbol{h}', \boldsymbol{m})$ are such that (2)-(4) are satisfied, $0 < h_k < 1$ for all $k \in N$, and $0 < m_{ik} < 1$ for every pair $(i, k) \in N^2$, then $(\boldsymbol{h}', \boldsymbol{m})$ is a point in the relative interior of the convex hull of our master problem. For such a choice of $(m_0, \boldsymbol{m_{ij}})$, the Benders cuts generated by an optimal solution of the above problem is a pareto optimal cut (Magnanti and Wong [27]).

Note that for a fixed allocation by $\hat{\boldsymbol{z}}$, the shortest path from i to j is among the 4 paths through allocated hubs of i and j, namely $i - k_1 - l_1 - j$, $i - k_1 - l_2 - j$, $i - k_2 - l_1 - j$ and $i - k_2 - l_2 - j$, where $\hat{z}_{ik_1} = \hat{z}_{ik_2} = \hat{z}_{jl_1} = \hat{z}_{jl_2} = 1$. Let $\hat{k}$ and $\hat{l}$ denote the allocated hubs of i and j respectively, such that the shortest path among these 4 paths for (i, j) is through those hubs. Let $K^1_{ij} = \{k \in N : \hat{z}_{ik} = 1, k \neq \hat{k}\}$, $K^0_{ij} = \{k \in N : \hat{z}_{ik} = 0\}$, $L^1_{ij} = \{l \in N : \hat{z}_{jl} = 1, l \neq \hat{l}\}$, and $L^0_{ij} = \{l \in N : \hat{z}_{jl} = 0\}$. So (24) is equivalent to

$$f_{ij} - \sum_{k \in K^1_{ij} \cup \{\hat{k}\}} u_{ijk} - \sum_{l \in L^1_{ij} \cup \{\hat{l}\}} v_{ijl} = \hat{\delta}_{ij}.$$

Note that by the above equation, we have

$$u_{ijk} = 0, \qquad \forall k \in K^1_{ij} \tag{25}$$

$$v_{ijl} = 0, \qquad \forall l \in L^1_{ij}, \tag{26}$$

since $u_{ijk}, v_{ijl} \geq 0$ and $f_{ij} - u_{ij\hat{k}} - v_{ij\hat{l}} \leq C_{ij\hat{k}\hat{l}}\, W_{ij} = \hat{\delta}_{ij}$ by (18). Hence, (24) becomes

$$f_{ij} - u_{ij\hat{k}} - v_{ij\hat{l}} = C_{ij\hat{k}\hat{l}}\, W_{ij}. \tag{27}$$

Thus, by (25) and (27), the problem (23)-(24), (18)-(19) can be rewritten as follows.

$$\begin{array}{llll}
\text{BDS}_{ij}: & \max & \multicolumn{2}{l}{m_0 f_{ij} - m_{i\hat{k}} u_{ij\hat{k}} - m_{j\hat{l}} v_{ij\hat{l}} - \sum_{k \in K^0_{ij}} m_{ik}\, u_{ijk} - \sum_{l \in L^0_{ij}} m_{jl}\, v_{ijl}} \\
\text{s.t.} & f_{ij} - u_{ij\hat{k}} - v_{ij\hat{l}} & = C_{ij\hat{k}\hat{l}}\, W_{ij} & \\
& f_{ij} - u_{ijk} - v_{ijl} & \leq C_{ijkl}\, W_{ij}, & k \in K^0_{ij} \cup \{\hat{k}\}, l \in L^0_{ij} \cup \{\hat{l}\}, (k, l) \neq (\hat{k}, \hat{l}) \\
& f_{ij} - u_{ijk} & \leq C_{ijkl}\, W_{ij}, & k \in K^0_{ij} \cup \{\hat{k}\}, l \in L^1_{ij} \\
& f_{ij} \qquad - v_{ijl} & \leq C_{ijkl}\, W_{ij}, & k \in K^1_{ij}, l \in L^0_{ij} \cup \{\hat{l}\}, \\
& f_{ij} & \leq C_{ijkl}\, W_{ij}, & k \in K^1_{ij}, l \in L^1_{ij}, \\
& u_{ijk}, v_{ijl} \geq 0, & & f_{ij} \in \mathbb{R} k, l \in N.
\end{array}$$

In Section 4 we show that for an appropriate choice of $(m_0, \boldsymbol{m})$, the generated cuts using optimal solutions of BDS_{ij} are stronger and result in fewer required iterations in the modified Benders decomposition method.

3.3 Solving Subproblems BDS$_{ij}$ Efficiently

The second issue we address in this section is the computationally expensive issues of the generation of Benders cuts. Using the simplex method to solve n^2 linear programs in each Benders iteration can be very time consuming. As shown in Figure 6-(a), more than 90% of the average computational time (in a few implementations) is consumed for the generation of cuts by solving BDS$_{ij}$ using the simplex method. In this part, we develop an algorithm to obtain Benders cuts more efficiently.

For a fixed pair (i, j), let r_{kl}^{ij} be the dual variable of constraint with indices (k, l) in BDS$_{ij}$. For simplicity we use r_{kl} instead of r_{kl}^{ij} when the context is clear. Then the dual of BDS$_{ij}$ is:

$$\begin{aligned}
\min \quad & \sum_{k\in N}\sum_{l\in N} C_{ijkl}\, W_{ij}\, r_{kl} \\
\text{s.t.} \quad & \sum_{k\in N}\sum_{l\in N} r_{kl} = m_0, \\
& \sum_{l\in N} r_{kl} \le m_{ik}, && k \in K_{ij}^0 \cup \{\hat{k}\} \\
& \sum_{k\in N} r_{kl} \le m_{jl}, && l \in L_{ij}^0 \cup \{\hat{l}\} \\
& r_{\hat{k}\hat{l}} \in \mathbb{R}, r_{kl} \ge 0 && (k,l) \in N^2 \setminus \{(\hat{k},\hat{l})\}.
\end{aligned}$$

Note that, by the choice of $K_{ij}^1, L_{ij}^1, \hat{k}$ and $\hat{l}$, in any optimal solutions of the above problem we have $r_{kl} = 0$ for $k \in K_{ij}^1, l \in L_{ij}^1 \cup \{\hat{l}\}$. Thus, by constraints of the problem,

$$\sum_{k\in K_{ij}^0}\sum_{l\in N} r_{kl} \le \sum_{k\in K_{ij}^0} m_{ik} \text{ and } \sum_{k\in K_{ij}^1\cup\{\hat{k}\}}\sum_{l\in L_{ij}^0} r_{kl} \le \sum_{l\in L_{ij}^0} m_{jl}.$$

Suppose $\sum_{k\in K_{ij}^0} m_{ik} = \Gamma_1$ and $\sum_{l\in L_{ij}^0} m_{jl} = \Gamma_2$, for some positive Γ_1 and Γ_2. Then,

$$\sum_{k\in N}\sum_{l\in N} r_{kl} \le \sum_{k\in N} m_{ik} + \sum_{l\in N} m_{jl} + r_{\hat{k}\hat{l}} \le \Gamma_1 + \Gamma_2 + \min\{m_{i\hat{k}}, m_{j\hat{l}}\}.$$

To eliminate the unsigned variable $r_{\hat{k}\hat{l}}$, we rewrite $r_{\hat{k}\hat{l}} = r_{\hat{k}\hat{l}}^+ - r_{\hat{k}\hat{l}}^-$. Therefore,

$$r_{\hat{k}\hat{l}}^- = \sum_{k\in N}\sum_{l\in N,(k,l)\neq(\hat{k},\hat{l})} r_{kl} + r_{\hat{k}\hat{l}}^+ - m_0 \le S,$$

where $S = \Gamma_1 + \Gamma_2 - m_0$. Obviously, $S \ge 0$ must hold. Now let $g = S - r_{\hat{k}\hat{l}}^-$. So $g \ge 0$ and $r_{\hat{k}\hat{l}} = r_{\hat{k}\hat{l}}^+ + g - S$. In the following problem, $r_{\hat{k}\hat{l}}$ denotes $r_{\hat{k}\hat{l}}^+ + g$ to make the presentation of the problem easier. This means that $r_{\hat{k}\hat{l}}$ has different meanings in the above problem and the (equivalent) following problem:

$$\min \quad -S\, C_{ij\hat{k}\hat{l}}\, W_{ij} + \sum_{k\in N}\sum_{l\in N} C_{ijkl}\, W_{ij}\, r_{kl} \tag{28}$$

$$\text{s.t.} \quad \sum_{k\in N}\sum_{l\in N} r_{kl} = \Gamma_1 + \Gamma_2, \tag{29}$$

$$\sum_{l\in N} r_{kl} \leq\ m_{ik}, \qquad k \in K_{ij}^0 \tag{30}$$

$$\sum_{l\in N} r_{\hat{k}l} \leq\ m_{i\hat{k}} + S, \tag{31}$$

$$\sum_{k\in N} r_{kl} \leq\ m_{jl}, \qquad l \in L_{ij}^0 \tag{32}$$

$$\sum_{k\in N} r_{k\hat{l}} \leq\ m_{j\hat{l}} + S, \tag{33}$$

$$r_{kl} \geq 0 \qquad k, l \in N. \tag{34}$$

The above problem is a minimum cost network flow problem (MCNF) in an auxiliary network with $2n+2$ nodes and n^2+2n arcs. Let s_i and t_j denote the corresponding source and sink nodes respectively, (which represent i and j respectively), and $1, 2, \ldots, n, n+1, \ldots, 2n$ denote two copies of all nodes of the original network. In the auxiliary network, links are (s_i, k), (k, l), and (l, t_j), for $k = 1, 2, \ldots, n$, and $l = n+1, n+2, \ldots, 2n$. The supply and demand of all nodes are zero except for s_i and t_j with supplies $\Gamma_1+\Gamma_2$ and $-\Gamma_1-\Gamma_2$ respectively. The capacity of arcs is given by constraints (30)-(34) (see Figure 4). Obviously, the total flow must be at most the total capacity of links with an endpoint i, or the total capacity of those with an endpoint j; that is, $\Gamma_1+\Gamma_2 \leq \min\{\sum_{k\in K_{ij}^0} m_{ik}+S+z_{i\hat{k}}, \sum_{l\in L_{ij}^0} m_{jl}+S+z_{j\hat{l}}\}$.

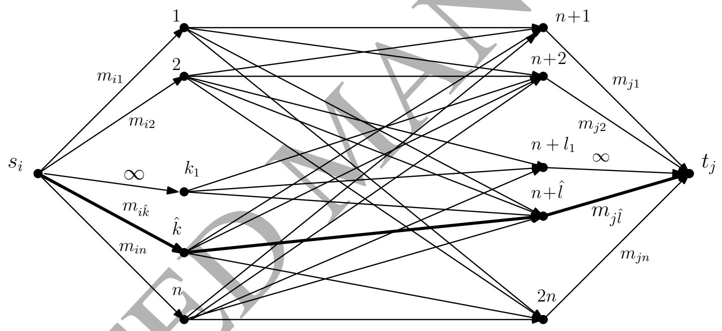


Figure 4: An equivalent minimum cost network flow problem to the dual of BDS_{ij}

Intuitively, the residual network of the subproblem network (the dual of BDS_{ij}) in which $r_{\hat{k}\hat{l}}$ is equal to its lower bound (that is, $-S$) is the above auxiliary network. A model of the minimum cost network flow is given below, where r_{s_ik} is the flow variable from s_i to k and r_{lt_j} is the flow variable from l to t_j.

$$\text{MCNF:}\ \min\ -S\,C_{ij\hat{k}\hat{l}}\,W_{ij} + \sum_{k=1}^{n} \chi d_{ik}\,W_{ij}\,r_{s_ik} + \sum_{k=1}^{n}\sum_{l=n+1}^{2n} \alpha d_{kl}\,W_{ij}\,r_{kl} + \sum_{l=n+1}^{2n} \delta d_{(l-n)j}\,W_{ij}\,r_{lt_j} \tag{35}$$

$$\text{s.t.}\quad \sum_{k\in N} r_{s_ik} = \Gamma_1 + \Gamma_2 \tag{36}$$

$$\sum_{l=n+1}^{2n} r_{lk} - r_{s_ik} = 0, \qquad k = 1, \ldots, n \tag{37}$$

$$r_{lt_j} - \sum_{k\in N} r_{kl} = 0, \qquad l = n+1, \ldots, 2n \tag{38}$$

$$-\sum_{l=n+1}^{2n} r_{lt_j} = -\Gamma_1 - \Gamma_2, \quad (39)$$

$$r_{s_ik} \leq m_{s_ik} \quad k \in K_{ij}^0 \quad (40)$$

$$r_{s_i\hat{k}} \leq m_{s_i\hat{k}} + S \quad (41)$$

$$r_{(l+n)t_j} \leq m_{t_jl} \quad l \in L_{ij}^0 \quad (42)$$

$$r_{(\hat{l}+n)t_j} \leq m_{t_j\hat{l}} + S \quad (43)$$

$$r_{s_ik}, r_{lt_j}, r_{kl} \geq 0 \quad (44)$$

We used the Floyd-Warshall algorithm to obtain $\hat{k}$ and $\hat{l}$ for each pair (i, j), and the successive shortest path algorithm to solve minimum cost network flow problems (Ahuja et al. [2]). Thus, the complexity of obtaining a solution for (28)-(34) is $O(n^3 + n^2 \log n)$ for each pair (i, j). Using the successive shortest path algorithm, the potentials of each node are also computed for any optimal solution. Thus, the optimal values of the dual variables can be obtained efficiently (Ahuja et al. [2]).

Let π_k, $k \in \{s_i, 1, 2, \ldots, 2n, t_j\}$, α_{ijk} for $k \in K_{ij}^0 \cup \{\hat{k}\}$, and β_{ijl} for $l \in L_{ij}^0 \cup \{\hat{l}\}$ be the dual variables of the set of constraints (36)-(39), (40)-(41), and (42)-(43) respectively. Then the dual of the above problem is

$$\max \quad -S\, C_{ij\hat{k}\hat{l}} W_{ij} + (\Gamma_1 + \Gamma_2)\pi_{s_i} - (\Gamma_1 + \Gamma_2)\pi_{t_j} - (m_{i\hat{k}} + S)\alpha_{ij\hat{k}} - (m_{j\hat{l}} + S)\beta_{ij\hat{l}} - \sum_{k \in K_{ij}^0} m_{ik}\, \alpha_{ijk} - \sum_{l \in L_{ij}^0} m_{jl}\, \beta_{ijl} \quad (45)$$

$$\text{s.t.} \quad \pi_{s_i} - \pi_k - \alpha_{ijk} \leq \chi d_{ik}\, W_{ij}, \quad k = 1, \ldots, n \quad (46)$$

$$\pi_k - \pi_l \leq \alpha d_{k(l-n)}\, W_{ij}, \quad k = 1, \ldots, n, l = n+1, \ldots, 2n \quad (47)$$

$$\pi_l - \pi_{t_j} - \beta_{ijl} \leq \delta d_{(l-n)j}\, W_{ij}, \quad l = n+1, \ldots, 2n \quad (48)$$

$$\alpha_{ijk}, \beta_{ijl} \geq 0, \pi_k, \pi_l \in \mathbb{R} \quad k = 1, \ldots, n, l = n+1, \ldots, 2n. \quad (49)$$

Noting that π_k is the potential of node k in the corresponding network for the MCNF problem, we can immediately obtain α_{ijk} and β_{ijk}:

$$\alpha_{ijk} = \max\{0, \pi_{s_i} - \pi_k - \chi d_{ik}\, W_{ij}\}, \quad k = 1, \ldots, n,$$

$$\beta_{ijl} = \max\{0, \pi_l - \pi_{t_j} - \delta d_{(l-n)j}\, W_{ij}\}, \quad l = n+1, \ldots, 2n.$$

Note that α_{ijk} and β_{ijl} will be equal to zero if flows in their associated links in the auxiliary network are strictly less than their corresponding upper bounds. Therefore, for $k \in K_{ij}^1$ and $l \in L_{ij}^1$, we have $\alpha_{ijk} = 0$ and $\beta_{ij(l+n)} = 0$ since $i - \hat{k} - \hat{l} - j$ is shorter than $i - k - l - j$.

Theorem 3.2. *Given an optimal solution $(\boldsymbol{\pi}, \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})$ of problem (45)-(49), $(f_{ij}, \boldsymbol{u_{ij}}, \boldsymbol{v_{ij}})$ is a feasible and optimal solution of BDS$_{ij}$, where $f_{ij} = \pi_{s_i} - \pi_{t_j}$, $u_{ijk} = \alpha_{ijk}$, and $v_{ijl} = \beta_{ij(l+n)}$ for $k, l \in N$.*

Proof. First note that by feasibility of $(\boldsymbol{\pi}, \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})$, we have

$$\pi_{s_i} - \pi_{t_j} - \alpha_{ijk} - \beta_{ij(l+n)} \leq (\chi d_{ik} + \alpha d_{kl} + \delta d_{lj})W_{ij} = C_{ijkl}W_{ij}, \qquad \forall k, l \in N. \quad (50)$$

By optimality of $(\boldsymbol{\pi}, \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})$, we have $\pi_{s_i} - \pi_{\hat{k}} - \alpha_{ij\hat{k}} = \chi d_{i\hat{k}} W_{ij}$, $\pi_{\hat{k}} - \pi_{\hat{l}} = \alpha d_{\hat{k}(\hat{l}-n)}\, W_{ij}$, and $\pi_{\hat{l}} - \pi_{t_j} - \beta_{ij\hat{l}} = \delta d_{(\hat{l}-n)j} W_{ij}$. For otherwise, if $\pi_{s_i} - \pi_{\hat{k}} - \alpha_{ij\hat{k}} < \chi d_{i\hat{k}} W_{ij}$, then $(\boldsymbol{\pi}, \boldsymbol{\alpha}'_{ij}, \boldsymbol{\beta}_{ij})$ with $\alpha'_{ij\hat{k}} = \alpha_{ij\hat{k}} - \epsilon$

for some $\epsilon > 0$, and $\alpha'_{ijk} = \alpha_{ijk}$ for $k \neq \hat{k}$, is feasible and has strictly larger objective value, which is a contradiction with the optimality of $(\boldsymbol{\pi}, \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})$. Similarly if $\pi_{\hat{l}} - \pi_{t_j} - \beta_{ij\hat{l}} < \delta d_{(\hat{l}-n)j} W_{ij}$ then we get the same contradiction. If $\pi_{\hat{k}} - \pi_{\hat{l}} < \alpha d_{\hat{k}(\hat{l}-n)} W_{ij}$, then $(\boldsymbol{\pi'}, \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}'_{ij})$ with $\pi'_{\hat{l}} = \pi_{\hat{l}} - \epsilon$ and $\beta'_{ij\hat{l}} = \beta_{ij\hat{l}} - \epsilon$ for some $\epsilon > 0$, and $\pi'_l = \pi_l$ and $\beta'_{ijl} = \beta_{ijl}$ for $l \neq \hat{l}$ is feasible while it has a strictly larger objective value, which is the same contradiction. Therefore, we have

$$\pi_{s_i} - \pi_{t_j} - \alpha_{ij\hat{k}} - \beta_{ij(\hat{l}+n)} = C_{ij\hat{k}\hat{l}}\, W_{ij}. \tag{51}$$

Therefore, for $f_{ij} = \pi_{s_i} - \pi_{t_j}$, $u_{ijk} = \alpha_{ijk}$ for $k \in N$, and $v_{ijl} = \beta_{ij(l+n)}$ for $l \in N$, $(f_{ij}, \boldsymbol{u_{ij}}, \boldsymbol{v_{ij}})$ is feasible for BDS_{ij} by (50)-(51), and the fact that $\alpha_{ijk} = 0$ for $k \in K^1_{ij}$, and $\beta_{ij(l+n)} = 0$ for $l \in L^1_{ij}$.

On the other hand, by (51) and since $S = \Gamma_1 + \Gamma_2 - m_0$, the objective value of $(\boldsymbol{\pi}, \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})$ for problem (45)-(49) is

$$m_0(\pi_{s_i} - \pi_{t_j}) - m_{i\hat{k}}\alpha_{ij\hat{k}} - m_{j\hat{l}}\beta_{ij\hat{l}} - \sum_{k \in K^0_{ij}} m_{ik}\, \alpha_{ijk} - \sum_{l \in L^0_{ij}} m_{jl}\, \beta_{ijl},$$

which is exactly the objective value of BDS_{ij} for $(f_{ij}, \boldsymbol{u_{ij}}, \boldsymbol{v_{ij}})$. Since these two problems have the same optimal values by invoking the strong duality theorem, $(f_{ij}, \boldsymbol{u_{ij}}, \boldsymbol{v_{ij}})$ is an optimal solution of BDS_{ij}. □

Note that by using MCNF for generating cuts, we (i) avoid numerical instability arising from constraint (24), and (ii) solve the subproblems much more efficiently, so that generating n^2 cuts is not a hindrance to obtaining tight cuts for the MP formulation – an observation that was made previously for the HLP problem in de Camargo et al. [12]. Algorithm 1 summarises the steps we use to solve U2ApHMP with our accelerated Benders decomposition method.

Algorithm 1: Modified Benders decomposition algorithm for U2ApHMP

1 Set $UB = \infty$;
2 Solve Master Problem MP for optimal solution $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}})$ and optimal value $\hat{Z}_{MP}$;
3 **if** *MP is infeasible* **then** Stop. U2ApHMP is infeasible;
4 Choose Γ_1, Γ_2 and m_0 so that $S \geq 0$;
5 Using successive shortest path algorithm and $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}})$, solve subproblem MCNF for optimal solution $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})$ and optimal value $\hat{Z}_{SP}$;
6 Let $UB := \min\{\sum_{k \in N} F_k \hat{h}_k + \sum_{i,k \in N} G_{ik}\, z_{ik} + \hat{Z}_{SP}, UB\}$;
7 **if** $UB - \hat{Z}_{MP} < \epsilon$ **then** Stop. Optimal solution for U2ApHMP is $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{x}})$;
8 Using Theorem 3.2, generate cuts (20) and add them to MP;
9 Go to Line 2.

3.4 Choice of Core Points

The strength of the Benders cuts in similar acceleration techniques (as discussed in Section 3.2) is directly related to the slope of the dual objective function of SP. In the literature, this slope is defined by some core point. A better choice of the points results in stronger cuts by defining more effective facets for cuts, which in turn improves the convergence rate of branch and bound for MP. We now discuss two methods for choosing this slope.

Magnanti and Wong [27] showed that any core point can be used to generate pareto optimal cuts. For instance, Contreras et al. [11] set and fixed integer variables of MP to 0.1 in each iteration in their HLP implementation. We equivalently set and fix $m_0 = 1$, and

$$m_{ik} = 2/n \qquad \forall i, k \in N,$$

in each iteration as our core point. Generated cuts by $\boldsymbol{m}$ are pareto optimal, however, they might not be the strongest cuts. We observed in most cases (across a few test implementations we ran) that a modification of BDS_{ij} – in which the objective function is minimised – results in stronger pareto optimal cuts, fewer branch and bound iterations, and faster convergence than those obtained by BDS_{ij} (see Figure 5). In fact, there might not be any method to realise a core point for the strongest cut as observed by Mercier et al. [29]. So pareto optimality is not sufficient to measure the strength of Benders cuts.

Our observation gave a motivation to choose more effective core points. Recently, Papadakos [36] showed that even if we drop (24) from the second LP, the generated cut is still pareto optimal. They defined a 'Magnanti-Wong point' to be any point for which the second LP gives a pareto optimal cut, and further showed that it is enough to use any convex combination of a Magnanti-Wong point and a feasible point of MP to generate a pareto optimal cut (Papadakos [36, Theorem 8]). In our approach, there is no need to drop (24) since it is included in MCNF without causing any instability in the solution. In MCNF we set $\Gamma_1 = 0.7, \Gamma_2 = 2.7, m_0 = 0, S = \Gamma_1 + \Gamma_2 - m_0$, and

$$m_{ik} = \Gamma_1/n \quad \forall k \in K^0_{ij}, \qquad m_{jl} = \Gamma_2/n \quad \forall l \in L^0_{ij},$$

$m_{i\hat{k}} = m_{j\hat{l}} = 1/n$, and $m_{ik} = \Gamma_2$ for $k \in K^1_{ij}$, and $m_{jl} = \Gamma_1$ for $l \in L^1_{ij}$. This choice of parameters is an empirical conclusion. By this combination, we increase the coefficient of z_{ik} and z_{jl} in Benders cuts when $\hat{z}_{ik} = 0$ and $\hat{z}_{jl} = 0$, respectively, (see the objective function of BDS_{ij}) in order to generate cuts which remove more space which is not recognised as being close to optimal points by the optimal solution $(\hat{\boldsymbol{h}}, \hat{\boldsymbol{z}})$ of the MP. We show in Section 4 that, on average, this choice of core point substantially reduces the number of iterations and the computational time.

Since our approach is developed for a new hub location problem, we are not able to compare our results with previous Benders decomposition approaches for solving hub location problems. The closest related research to U2ApHMP in the HLP literature is that of Contreras et al. [11] who report state-of-the-art results on a hub location problem with multiple allocations. Their results have been a marked improvement to various other results in the literature. They were able to solve large instances exactly using their method. There are key differences between our research assumptions: (i) for a fixed set of hubs, the hub location problem with multiple allocation is polynomially solvable, whereas U2ApHMP is NP-hard even for a fixed set of hubs, and (ii) they fixed $\chi = \delta = 1$ in the calculation of costs, whereas we did not impose this restriction. Note that when $\chi = \delta$, the number of subproblems is halved since the optimal route for any pair (i, j) is the same as that of the pair (j, i) with opposite orientation. These differences makes our problem quite challenging. On the other hand, they set the coordinates of the core point to 0.1 in each iteration, and because they found it very expensive to optimally solve the second LP for generating Benders cuts, Contreras et al. [11] were satisfied with approximations to the optimal solutions of the subproblems. This might be troublesome since a suboptimal solution of the subproblem is prone to numerically unbounded Benders cuts as discussed in Papadakos [36, Example 5]. In contrast, we proposed a versatile choice of core points, and used MCNF to solve the dual of the subproblems for cut generations.

We believe that our method is more efficient than existing similar methods in the literature. This claim is supported by our tests on the uncapacitated hub location problem with single or multiple allocation.

This is achieved through a more careful and judicious choice of the core point. We presented the results of our approach on USApHMP and UMApHMP in [31]. Furthermore, we believe that our results can be extended to UrApHMP for any $3 \leq r \leq p-1$. However, the investigation of this claim is not in the scope of this paper. So we leave it as a subsequent exercise for other researchers. Nevertheless, in order to show the efficiency of our approach for other problems, we also briefly present the results of our examination of this method and a second method for choosing core points on USApHMP.

By substituting (3) with the following, we obtain a formulation for USApHMP (Campbell [7]):

$$\sum_{k \in N} z_{ik} = 1, \qquad \forall i \in N$$

The MP in the modified Benders decomposition is different with the one in U2ApHMP, but their subproblems are similar. By this change, the corresponding K^1_{ij} and L^1_{ij} in Section 3.2 are empty sets. But, this has no effect on the SP, and the Benders cuts for USApHMP can be obtained by solving (28)-(34). Hence, all the previous discussions and methods for subproblems are valid for USApHMP. Recall that for a fixed set of hubs USApHMP is NP-hard. The choices of parameters and m_{ik} for this problem are the same as those for U2ApHMP.

4 Computational Results

In this section we present the computational results of our experiments on U2ApHMP using a branch and bound method, and discuss Benders decomposition methods that we developed in this paper. We first provide some notations that we will use in the succeeding discussions in Table 1.

notation	**description**
n	the number of nodes
p	the number of hubs
Bitr	the number of Benders iterations
Bcuts	the number generated Benders cuts
B&B	branch and bound using CPLEX 12.7
BnsAuto	built-in Benders decomposition method in CPLEX 12.7
Bns-SPX	modified Benders decomposition, where n^2 BDS$_{ij}$ are solved using the simplex method at each iteration
Bns-MCNF	modified Benders decomposition where MCNF is solved using the successive shortest path algorithm
$\boldsymbol{m}$-Bns-MCNF	Bns-MCNF with the second method of defining the dual objective function as in Section 3.4

Table 1: Notations used in computational results

We observe that the modified Benders decomposition method is very efficient for solving U2ApHMP, and that our choice of core points significantly improves the convergence rate. Additionally, it also reduces the number of iterations required. The effectiveness of our modified Benders decomposition method for general HLPs is also demonstrated for solving the USApHMP, UMApHMP, UMAHLP and USAHLP [30, 31].

In order to test the efficiency of our method for solving U2ApHMP, our computational experiments were carried out on three well-known datasets in the HLP literature. We examined different discount factors, namely $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$ and experimented with the number of hubs $p = 3, 4, 5, 6$ and 7. We used a factor (10^{-3}) of access link lengths to set fixed costs for corresponding experiments of Tables 3-6. We used 80 instances of the Civil Aeronautics Board (CAB) dataset (presented first in O'Kelly [34]) with 10, 15, 20 and 25 nodes. For all instances of CAB, we set $\chi = \delta = 1$. There is no fixed cost data for CAB dataset. We used the method of Ebery et al. [13] to generate fixed costs datasets for CAB.

The second dataset we used is the Australia Post dataset (AP) which contains a maximum of 200 nodes. This dataset was introduced by Ernst and Krishnamoorthy [14], who also made an application available online in order to generate smaller-sized datasets. This application consolidates subsets of ordered nodes to fewer nodes. We implemented algorithms on AP instances with 30, $40, 50, 60, 70, 80$, 90, 100, $125, 150, 175$ and 200 nodes. The fixed cost datasets were available for instances of size $10, 20, 25, 40, 50$, 100 and 200. There are tight and loose variants of fixed cost data, where the tight one is known to result in harder problems in general. We used the tight fixed cost datasets, and generalised their method of generating smaller size instances to produce fixed cost datasets for instances of size $30, 60, 70, 80, 90, 125, 150$ and 175 too. For all AP instances, we set $\chi = 3$ and $\delta = 2$.

To test the efficacy of our methods even further, we also tested our methods using the Turkish Cargo Delivery dataset that was provided by Tan and Kara [46]. This dataset contains 81 nodes. Using the same method that we used for the AP dataset, we produced smaller instances of the Turkish data with $30, 40, 50, 60, 70$ nodes. For all these instances, we set $\chi = \delta = 1$.

Thus, in total, we tested our methods on 340 instances. The results are presented in Tables 2-8. All computations were performed on a computer with Xeon(R) 2.30GHz processors and 64 GiB memory, with 64-bit `Linux Red Hat 4.4` operating system. All methods were coded in `C++` using the `Concert Technology CPLEX 12.7`. The time limit for all computations was fixed to 3 hours (10800 seconds), and the number of computation threads for all the Benders decomposition methods was fixed to one. If within the specified time limit, a method is able to find the optimal solution, the corresponding CPU time is presented in seconds (sec). If the method only finds a feasible non-optimal solution, the gap between the best solution is presented. If, however, a method does not find any feasible solution, then 'm' or 't' is displayed to (respectively) indicate that the method stopped because of insufficient memory or because it reached the time limit.

The computational results of U2ApHMP tests on the CAB dataset are presented in Table 2. All procedures were able to optimally solve all instances in a relatively short period of time. The results indicate that $\boldsymbol{m}$-`Bns-MCNF` is the fastest method with fewest `Bitr` and `Bcuts` on average. The two slowest methods are `BnsAuto` and `Bns-SPX`. This issue is because of their inefficient solutions of the SPs for a factor of `Bitr` or `Bcuts` many times. On average, `Bns-SPX` and `Bns-MCNF` produced a similar number of `Bitr`. This demonstrates that computational times for this problem are significantly impacted by efficiency of solution methods for SPs in Benders approaches. For all methods, the complexity and running time grow with the number of nodes n.

For a fixed n, the computational effort increases with α on average. A reason is that with lower transfer costs, non-hubs tend to be allocated to the closest hub, which makes the competition for allocations easier. For a given n, the computational effort by `B&B` and `BnsAuto` decreases when p increases. Intuitively, the increased number of hubs results in less required cuts to resolve the competition of choosing hubs. However, this trend is not realised for `Bns-SPX`, `Bns-MCNF`, and $\boldsymbol{m}$-`Bns-MCNF`. For example, the instance with $n = 25, p = 4, \alpha = 0.4$ (for brevity 25.4.(0.4)) is one of the most time consuming problems for `Bns-SPX` and $\boldsymbol{m}$-`Bns-MCNF`. A reason for this difference might be that the modified Benders methods are not restricted to the chosen hubs and allocations specified by MP for

		B&B	BnsAuto	Bns-SPX			Bns-MCNF			$\boldsymbol{m}$-Bns-MCNF				B&B	BnsAuto	Bns-SPX			Bns-MCNF			$\boldsymbol{m}$-Bns-MCNF		
n p	α	time	time	time	Bcuts	Bitr	time	Bcuts	Bitr	time	Bcuts	Bitr	α	time	time	time	Bcuts	Bitr	time	Bcuts	Bitr	time	Bcuts	Bitr
10 3	0.2	0.22	2.73	0.63	455	7	0.16	446	9	0.11	396	10	0.6	0.20	2.74	0.69	483	9	0.12	474	9	0.09	428	9
4		0.22	2.34	0.69	371	9	0.14	478	11	0.14	425	13		0.22	2.41	0.90	565	14	0.15	534	14	0.09	326	8
5		0.20	2.36	0.58	403	8	0.15	563	16	0.07	352	9		0.20	2.26	0.72	497	10	0.14	534	13	0.10	233	5
6		0.09	2.30	0.70	455	10	0.11	428	10	0.04	198	5		0.19	2.22	1.00	662	18	0.13	476	14	0.08	240	8
7		0.18	2.24	0.64	477	9	0.12	431	16	0.04	224	6		0.18	2.21	0.86	569	14	0.17	463	18	0.06	212	7
15 3		1.14	12.71	3.01	1168	9	0.54	1033	12	0.35	861	8		0.96	11.11	2.61	1108	9	0.50	1013	12	0.47	982	11
4		1.00	11.79	2.40	832	7	0.48	881	10	0.40	553	5		0.98	10.76	3.75	1238	10	0.51	975	12	0.52	963	11
5		0.99	11.47	3.23	1354	10	0.45	1054	12	0.18	573	7		1.06	11.54	3.95	1657	14	0.66	1072	12	0.56	1063	15
6		1.05	10.22	2.95	1160	10	0.36	921	9	0.21	687	8		1.04	11.32	3.31	1497	12	0.55	1008	9	0.41	875	13
7		0.83	11.30	2.79	1160	11	0.43	1141	14	0.22	667	8		1.83	28.11	3.58	1602	13	0.70	1207	15	0.33	878	12
20 3		5.50	32.74	9.90	1540	7	0.97	1490	7	0.80	1464	6		4.53	32.72	11.31	2141	9	0.97	1402	8	1.22	1183	8
4		4.65	27.70	13.74	2415	11	1.01	1642	14	0.80	1137	6		9.63	60.03	24.95	4376	21	2.07	1778	13	1.50	1914	18
5		3.39	30.25	11.21	1878	8	0.96	1675	11	0.67	1106	8		4.04	26.62	18.79	2826	16	1.54	1718	11	0.93	1382	11
6		2.78	30.90	19.63	3016	18	1.03	1962	12	0.98	1331	9		5.69	52.93	22.14	3730	19	1.92	2544	21	0.94	1455	11
7		2.65	28.85	14.21	2743	13	1.09	1815	11	0.67	1378	9		4.32	29.74	19.52	3791	19	1.40	2012	15	0.88	1411	10
25 3		17.69	76.71	47.17	3019	7	2.59	3726	11	2.10	3280	8		12.48	75.99	75.19	4533	12	3.14	3287	15	1.90	3007	10
4		14.40	76.09	59.52	2989	9	2.60	3273	11	2.20	3287	9		19.47	115.12	71.42	5418	16	4.91	3959	18	3.73	3495	18
5		13.96	123.62	60.86	4421	12	2.21	2849	8	1.47	2032	9		13.37	119.19	99.32	6184	19	4.67	3380	16	2.71	2716	13
6		9.45	170.56	71.01	5360	17	3.84	4661	25	1.44	2140	8		11.96	164.85	85.86	6753	22	5.44	4244	23	2.38	3256	15
7		9.81	76.02	75.79	6219	18	2.67	3712	20	1.31	2004	9		10.35	78.32	93.64	6547	24	3.28	3825	16	2.54	2622	15
10 3	0.4	0.23	2.71	0.69	599	10	0.15	426	9	0.09	363	8	0.8	0.34	2.82	0.65	483	9	0.17	438	14	0.14	353	11
4		0.22	2.37	0.90	526	13	0.16	541	13	0.09	347	8		0.57	2.57	0.94	658	13	0.21	550	14	0.11	319	9
5		0.20	2.31	0.62	373	8	0.14	587	16	0.10	294	8		0.21	2.27	0.88	565	13	0.20	507	17	0.09	315	8
6		0.19	2.26	0.70	514	10	0.14	522	20	0.07	300	9		0.38	12.68	1.08	814	19	0.18	457	17	0.06	309	10
7		0.19	2.19	0.68	432	10	0.10	433	15	0.05	237	9		0.43	2.47	1.23	717	22	0.12	427	15	0.08	230	7
15 3		0.61	10.78	3.04	1166	9	0.59	1050	12	0.29	783	8		1.67	22.00	3.92	1466	12	0.88	1162	15	0.53	930	11
4		0.93	11.54	2.94	1037	9	0.45	953	11	0.27	774	10		2.31	24.58	3.39	1730	12	0.94	1190	17	0.66	1035	14
5		1.02	10.98	3.06	1191	10	0.51	1175	12	0.23	696	9		4.48	27.61	5.36	2323	20	0.98	1166	15	0.93	1237	22
6		0.96	9.79	4.01	1492	14	0.46	950	11	0.31	720	10		2.52	31.23	6.84	2350	27	1.24	1674	30	0.58	946	12
7		0.96	11.48	2.98	1119	9	0.67	1434	20	0.36	804	11		2.25	24.93	5.26	2248	20	1.02	1287	22	0.60	1192	19
20 3		5.52	32.89	10.13	1537	7	1.00	1425	7	0.87	1056	5		3.65	28.36	11.19	2090	10	1.16	1371	10	0.97	1163	6
4		4.11	26.54	13.32	2437	11	1.27	1755	14	1.21	1291	11		9.90	61.72	25.41	3816	19	2.55	2389	21	2.14	2495	22
5		3.70	27.04	15.13	2402	12	1.01	1753	10	0.71	1375	8		5.83	52.76	20.05	3877	17	2.67	1972	14	1.52	1550	10
6		2.64	27.50	18.81	3411	18	1.28	1802	11	0.75	1367	9		8.10	68.10	31.61	4926	25	3.12	2259	18	1.63	1845	18
7		2.76	29.40	18.83	3233	17	1.54	2171	19	0.77	1619	11		6.62	55.38	35.55	6620	33	2.21	2552	23	1.25	1694	14
25 3		19.07	75.05	40.69	3017	8	3.10	3475	13	1.67	2391	8		12.34	74.37	54.34	4745	13	3.12	3475	24	2.22	3083	12
4		13.06	76.17	127.85	5820	17	2.83	3013	10	2.23	2552	8		46.15	133.34	93.25	8123	19	7.89	4706	23	10.30	4856	26
5		11.73	122.62	78.43	5833	17	3.10	3059	13	1.99	2124	7		51.85	187.90	228.84	13177	40	12.26	5271	22	8.22	4554	21
6		9.80	171.84	111.54	6265	19	2.88	3167	12	1.41	2099	7		28.85	227.93	161.68	12398	35	8.74	4642	21	6.12	3845	19
7		9.47	75.96	80.81	4794	15	2.37	3474	13	1.60	1960	10		28.67	107.41	127.56	8822	25	9.15	3860	19	4.10	2800	16

Table 2: U2ApHMP on CAB dataset

generating cuts and take into account any shorter paths.

Table 3 presents the computational results of U2ApHMP tests on the problem instances in the AP dataset with n between 30 and 60.

The B&B, Bns-MCNF, and $\boldsymbol{m}$-Bns-MCNF methods were able to optimality solve all the instances within the time limit (except one instance by B&B). However, BnsAuto was not able to load instances with $n = 60$, and Bns-SPX was not able to optimally solve instances with $n = 60$, for which the gap of the best solutions is presented in the 'time' column. $\boldsymbol{m}$-Bns-MCNF outperforms all other methods in terms of computational time, and also it requires fewer Bitr and Bcuts for almost all instances. In general, the computational times of the modified Benders approaches are dependant on Bitr and Bcuts, without being impacted by the value of p. The computational times of B&B, BnsAuto and Bns-SPX grow exponentially with n, due to the growth of the size of instances and/or the increased number of required Bitr or Bcuts. B&B outperforms BnsAuto for all instances. In contrast, the increase of computational times of Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF with growth of n are not as fast, which is because of efficient solutions of SPs and fewer Bitr. $\boldsymbol{m}$-Bns-MCNF outperforms Bns-MCNF by more than 55% in average computational times, and it outperforms other methods by more than almost 90%.

Tables 4-5 present the computational results of U2ApHMP tests on the instances in the AP dataset, where n is between 70 and 200. Since B&B, BnsAuto, and Bns-SPX were unable to load instances due to shortage of memory, or unable to obtain optimal (or even feasible) solutions for these instances in our test conditions, we do not include details of computational tests for these methods in Table 4-5. When a method finds a feasible solution which is not optimal, the gap of its best solution and the best

		B&B	BnsAuto	Bns-SPX			Bns-MCNF			$\boldsymbol{m}$-Bns-MCNF				B&B	BnsAuto	Bns-SPX			Bns-MCNF			$\boldsymbol{m}$-Bns-MCNF		
n p	α	time	time	time	Bcuts	Bitr	time	Bcuts	Bitr	time	Bcuts	Bitr	α	time	time	time	Bcuts	Bitr	time	Bcuts	Bitr	time	Bcuts	Bitr
30 3	0.2	20.4	142.8	235.9	7014	13	3.7	4408	11	2.2	3503	9	0.6	21.1	172.0	259.5	6692	11	3.8	4446	14	2.0	3170	10
4		19.4	181.4	300.0	7105	14	3.1	4531	10	2.1	3597	9		18.9	145.8	246.6	6839	13	3.6	4562	13	3.0	3531	10
5		15.4	178.3	317.4	5562	16	4.0	4954	13	2.4	3126	11		16.7	173.8	252.4	6140	14	3.8	5496	16	2.3	3214	10
6		13.1	186.3	321.1	7993	19	3.4	3743	11	2.4	2829	10		11.5	182.8	354.9	7904	23	4.6	4217	14	1.5	3042	10
7		12.4	183.2	259.1	6067	15	3.5	4884	13	2.5	3717	14		14.0	142.1	357.7	7888	19	5.6	5327	18	1.5	3086	12
40 3		396.6	572.3	1730.7	16357	16	12.4	10006	13	8.3	6945	9		252.3	518.9	3097.1	21740	24	10.1	8419	13	9.6	7416	12
4		266.4	2339.5	1596.5	11751	12	16.2	9661	13	9.1	7026	9		225.9	2468.6	1271.9	9538	10	17.6	10138	16	7.2	6204	8
5		196.5	1972.8	1999.9	14663	19	27.7	11804	23	11.6	7036	8		201.1	2383.6	2866.7	18621	25	15.4	9878	22	15.2	8984	20
6		118.1	2332.2	1790.3	14743	17	20.6	10752	17	9.6	8358	11		167.2	2391.8	4401.5	23910	38	19.8	10669	25	12.2	8713	17
7		205.1	2409.4	3240.0	21200	28	36.6	12226	26	12.9	9633	19		191.2	1880.1	3043.2	21011	28	33.4	11054	23	14.4	10451	19
50 3		1590.9	7392.8	8208.2	29075	18	39.9	17903	15	26.1	14829	10		1667.5	7240.0	5300.8	20734	12	36.9	18154	14	24.8	13615	12
4		1578.3	6824.0	8664.7	26591	22	85.1	22793	20	27.6	14112	11		1485.3	5693.9	8757.7	38760	26	62.1	23334	28	30.3	14075	16
5		618.0	5828.9	0.0%	31998	27	47.2	18967	18	22.3	13193	10		1055.9	6186.9	7354.4	27959	17	62.2	20743	36	29.2	10821	12
6		567.2	5248.2	6572.0	26031	21	37.3	15124	16	17.5	10318	9		1002.4	4811.5	0.0%	38496	32	68.0	17639	27	28.1	14465	16
7		369.9	5774.8	7981.3	38091	27	57.3	15244	12	23.9	12065	15		965.9	4896.2	0.0%	43308	31	93.5	20031	31	37.2	15488	16
60 3		6762.5	m	23.0%	17968	8	143.3	31872	19	95.8	23691	13		t	m	25.0%	20829	9	211.6	33296	41	189.3	29152	27
4		4453.1	m	16.0%	20881	8	154.2	30675	25	114.6	20226	10		9577.3	m	46.0%	24967	8	191.9	32814	20	201.8	28747	36
5		2575.8	m	3.0%	27228	11	107.8	24302	18	68.0	18938	15		5650.3	m	15.0%	28996	10	231.2	29457	27	72.2	18160	16
6		1667.9	m	4.0%	26885	9	206.9	31506	27	127.8	23367	22		3416.5	m	14.0%	25809	10	201.3	30810	35	191.8	27646	36
7		1267.6	m	6.0%	26534	10	233.2	30939	20	138.1	29016	28		4133.1	m	16.0%	21334	11	279.9	26353	22	173.3	26281	36
30 3	0.4	24.8	187.3	423.0	8185	17	4.8	5395	16	3.9	4153	12	0.8	21.2	168.7	352.8	8680	17	3.9	5519	15	3.0	3121	10
4		20.7	186.9	405.6	7185	16	4.7	5938	23	2.6	3239	11		20.9	164.7	223.3	7816	13	4.0	4538	12	2.9	2969	9
5		15.5	183.8	319.3	7136	17	4.4	4628	11	2.3	3851	12		13.2	139.0	248.4	6611	14	5.4	4811	15	2.3	3068	10
6		14.3	179.8	353.2	7509	16	3.7	4518	15	2.2	2647	8		14.4	175.5	318.0	7886	18	4.0	4061	17	2.6	3771	12
7		10.5	180.3	282.2	6614	17	4.2	4308	15	2.3	3420	11		12.1	183.2	329.4	6439	15	5.8	4298	15	2.1	3352	10
40 3		323.1	513.5	3758.2	26084	30	10.5	8694	12	6.5	6027	10		283.4	563.6	2510.5	19102	21	12.2	8535	13	7.5	5844	8
4		234.0	2455.3	2812.5	24154	26	17.0	9608	16	12.2	7597	16		218.8	2533.3	1591.6	14234	15	13.6	8714	12	9.5	6162	8
5		211.1	2533.6	2146.9	13292	17	16.2	8292	13	12.1	9490	18		149.3	2386.5	1784.5	16669	18	18.1	9942	14	14.1	8142	19
6		165.9	2052.4	4508.9	26791	45	22.9	9241	17	10.7	7414	15		163.9	2515.3	2780.3	20658	24	19.1	10122	16	12.3	9878	17
7		172.4	2469.4	3581.1	21982	32	36.2	10661	22	14.1	10937	27		143.3	2622.2	4477.4	24829	39	25.4	9500	17	11.9	8728	12
50 3		1722.7	5742.3	8259.6	24067	18	35.9	17054	19	25.1	11825	9		1771.3	5319.8	0.0%	36659	26	103.0	28347	43	33.4	15861	14
4		1206.7	5137.0	6102.7	26666	16	49.3	19793	22	36.6	15471	16		1302.7	5323.6	10929.8	37886	26	70.0	19535	21	51.9	18025	21
5		903.5	5212.4	8227.5	27342	19	57.0	20747	26	23.5	11394	14		1143.4	4669.7	0.0%	32438	32	53.1	16139	19	31.9	14217	21
6		414.9	5875.4	0.0%	28681	28	47.0	15283	21	17.5	10849	11		1508.1	5785.9	0.0%	40824	34	53.7	14875	16	36.8	14185	24
7		564.2	5770.0	0.0%	36814	31	60.9	20836	25	23.0	11685	14		1783.3	t	0.0%	39467	35	91.3	17444	19	46.2	15611	15
60 3		7960.4	m	9.0%	21381	8	132.9	26928	22	86.9	25456	11		9353.4	m	6.0%	27017	11	117.5	23432	19	131.5	28591	22
4		6319.9	m	t			203.8	35150	34	99.7	22377	18		9784.0	m	14.0%	24042	9	229.9	32188	41	304.6	29526	33
5		2465.4	m	12.0%	20464	11	154.7	24908	21	76.8	16574	21		8667.8	m	12.0%	20057	9	259.0	24595	27	128.1	19200	25
6		2381.4	m	19.0%	24003	10	172.4	26240	23	77.7	21034	22		5460.5	m	7.0%	24216	10	284.3	31914	41	174.6	29146	30
7		3727.5	m	20.0%	26041	10	221.8	30316	32	102.1	21082	22		7305.9	m	15.0%	27262	11	335.3	37828	54	283.8	30653	31

Table 3: U2ApHMP on Australia Post dataset (part 1)

lower bound is presented in the 'time' column. Within the time limit, Bns-MCNF was able to solve 60% of these problems to optimality, and could not obtain any feasible solution for 3 instances (indicated by 't' in the 'time' column). $\boldsymbol{m}$-Bns-MCNF performed better and was able to optimally solve 65% of instances. In general, the computational efforts of Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF decrease as α grows for a fixed n. $\boldsymbol{m}$-Bns-MCNF was able to solve instances with $70 \leq n \leq 100$ faster than Bns-MCNF by 47%, with 20% fewer Benders iterations on average. This is rooted in the stress of our cut generation method with $\boldsymbol{m}$-slope of DS_{ij} on non-hubs and non-allocated links. The gap of the best solutions obtained by $\boldsymbol{m}$-Bns-MCNF is 2.5% lower than those by Bns-MCNF for instances with $125 \leq n \leq 200$ on average.

Table 6 presents the computational results of all U2ApHMP tests on the Turkish dataset. B&B was able to obtain optimal solutions for all instances with up to 40 nodes, and most of instances with $n = 50$. This method was only able to optimally solve a fifth of the instances with $n \geq 60$, among which none of them has 81 nodes within the time limit. BnsAuto was able to give optimal solutions for instances only up to 40 nodes, and unable to solve most instances with $n = 50$ especially for larger discount factors α. It encountered memory shortage for $n \geq 60$ in our test conditions. $\boldsymbol{m}$-Bns-MCNF was able to solve all problems to optimality, and outperform B&B, BnsAuto, and Bns-SPX by more than 85%, and Bns-MCNF by 56% in the average computational times. Table 6 shows that its computational times are reduced by the efficient solution of SPs and through generating stronger cuts, which is evident from fewer Bcuts/Bitr. Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF were able to solve (almost) all instances within the time limit in this experiment. However, $\boldsymbol{m}$-Bns-MCNF required around 25% fewer Benders iterations and Benders cuts than Bns-MCNF. The computational time of Bns-MCNF is at least twice that of $\boldsymbol{m}$-Bns-MCNF on average. As in the other tested datasets, the average of computational efforts grows with the number

			Bns-MCNF			*m*-Bns-MCNF				Bns-MCNF			*m*-Bns-MCNF		
n	p	α	time	Bcuts	Bitr	time	Bcuts	Bitr	α	time	Bcuts	Bitr	time	Bcuts	Bitr
70	3	0.2	272.44	43540	24	224.96	36862	15	0.4	420.47	42911	30	266.88	40004	21
	4		588.61	46419	24	296.04	43321	26		467.52	35537	12	282.75	37305	27
	5		366.36	47950	25	269.42	39129	19		530.69	48751	29	310.91	32703	16
	6		332.41	37667	24	205.20	34456	22		226.07	29611	14	222.54	32955	24
	7		616.97	52998	46	130.91	25526	12		592.34	45574	33	198.43	30154	22
80	3		567.50	55814	21	509.55	45554	13		661.14	62778	34	332.41	40559	16
	4		1009.00	80186	35	321.34	46780	14		742.58	61094	30	470.70	49310	25
	5		492.50	48367	19	233.99	30802	10		661.38	48721	29	417.13	44670	24
	6		533.30	49511	25	686.70	55929	38		664.84	53302	24	302.65	35103	16
	7		868.92	50789	24	581.99	48244	24		818.01	46291	24	735.97	53632	43
90	3		1254.12	76260	20	659.62	54696	17		748.67	56556	13	1239.47	86582	33
	4		984.92	74202	30	734.54	65373	21		1044.55	83412	29	751.81	62709	24
	5		1526.97	70471	28	957.49	63884	23		2278.31	63814	29	791.65	54623	27
	6		1619.50	70766	27	574.77	43324	19		2655.76	73494	31	697.32	49481	22
	7		2262.46	73325	28	1399.61	58979	29		4385.24	91700	47	898.63	53064	24
100	3		1323.48	90505	21	1226.75	60782	19		1200.94	78013	19	1567.24	80956	24
	4		2059.02	106704	27	1742.64	106089	25		1958.42	78000	19	1997.37	80251	28
	5		1648.27	80657	25	637.05	46626	13		2141.89	100388	35	927.52	49398	23
	6		2692.40	86209	35	1236.50	71316	23		2696.63	104529	28	1311.31	67185	28
	7		5622.00	115162	37	1429.63	69383	21		4953.79	119414	34	2911.98	91949	39
125	3		3487.45	150235	22	2038.97	92238	11		3696.56	86909	15	2494.87	68878	14
	4		3962.02	107383	20	2215.89	85929	15		3415.98	121096	17	3548.82	111807	24
	5		6337.98	135589	33	6392.57	124667	42		6302.00	117829	27	3752.19	99035	26
	6		10517.41	155356	36	3325.59	105500	26		6428.66	130810	26	7199.55	132662	55
	7		2%	155404	27	9633.33	176204	36		1%	183790	42	0%	172282	43
150	3		8%	190506	15	4775.98	127094	9		9189.72	153038	20	0%	152417	29
	4		8641.77	177568	17	t				2%	166665	30	9246.01	173529	27
	5		5%	157402	32	10352.82	171124	28		13%	171708	22	9048.46	145009	24
	6		9%	147501	15	0%	159738	32		2%	193872	28	12%	137835	12
	7		2%	175550	20	1%	139186	17		19%	158520	27	1%	190000	24
175	3		1%	181219	12	2%	150104	14		5%	158169	15	2%	135545	10
	4		12%	128926	10	3%	162706	8		1%	144543	14	12%	119333	11
	5		26%	148861	12	9%	129987	12		29%	129811	15	17%	120679	11
	6		17%	178584	13	2%	129470	12		16%	192971	13	2%	163189	13
	7		19%	193690	15	12%	145753	14		16%	198397	12	8%	130961	15
200	3		26%	150025	7	23%	158030	5		t			32%	155890	7
	4		9%	206931	12	14%	155976	6		4%	164926	10	12%	171367	9
	5		26%	203334	10	51%	119542	4		48%	146521	5	13%	213862	7
	6		t			7%	118380	5		15%	119744	4	13%	147743	9
	7		17%	184131	10	t				16%	157563	5	23%	119483	5

Table 4: U2A*p*HMP on Australia Post dataset (part 2)

of nodes or the transfer discount. In general for a given n, when p increases or when α decreases, the computational time of B&B decreases, and those of Bns-SPX, Bns-MCNF, and *m*-Bns-MCNF increase.

Tables 2-6 indicate that B&B and BnsAuto as general purpose tools are not as efficient as the modified Benders methods for our hub location problem to solve large instances. B&B is not very efficient since it is very dependant on the capability of computational environment. Its computational effort grows exponentially, so that it is unable to solve every instance with $n \geq 60$ in our experiments. In general, its performance is highly influenced by the size of the problem. According to our computational results, BnsAuto is not an efficient method in general since only one cut is added to the MP in each iteration. In addition, it requires large RAM memory for relatively medium instances. Note that the modified Benders methods are able to find optimal solutions (or feasible solutions in some cases), whereas BnsAuto cannot load the problem due to the shortage of memory. However, the computational times of Bns-SPX are larger than those of BnsAuto for most cases (see Figure 7).

Due to the degeneracy of the SPs, the generated cuts in each iteration are not unique. The convergence rate of the Benders approach is highly dependant on the choice of cuts. As shown in Figure 5 and

			Bns-MCNF			$\boldsymbol{m}$-Bns-MCNF				Bns-MCNF			$\boldsymbol{m}$-Bns-MCNF		
n	p	α	time	Bcuts	Bitr	time	Bcuts	Bitr	α	time	Bcuts	Bitr	time	Bcuts	Bitr
70	3	0.6	252.31	35183	24	263.14	24573	18	0.8	337.60	38940	35	472.43	43573	38
	4		441.62	39697	32	486.48	47158	35		461.18	47404	49	243.46	24881	24
	5		757.08	44179	62	189.72	27664	14		612.01	50489	52	306.15	31548	20
	6		281.95	34518	23	157.70	30936	14		482.43	37257	37	366.54	36032	42
	7		846.52	48646	48	301.99	31383	26		1116.60	56216	38	445.36	36048	23
80	3		740.49	58430	37	458.44	50135	28		447.21	43061	26	430.03	40310	26
	4		691.71	55457	30	423.81	48714	25		948.97	54971	34	367.83	41875	18
	5		1347.31	63052	39	786.51	42747	31		871.12	61774	40	561.00	35163	19
	6		893.77	55924	27	658.50	48990	37		1316.41	57982	46	718.46	44986	30
	7		1211.16	53548	39	977.72	49484	47		1728.25	48971	21	1627.16	57376	35
90	3		686.46	56012	19	1285.79	70023	36		944.95	62426	26	1295.25	75091	29
	4		583.84	57040	24	1003.31	49561	28		1100.42	63724	33	711.18	42654	21
	5		1928.53	76749	35	1168.41	66848	33		1374.35	68986	32	1023.69	49214	31
	6		3292.04	84316	66	707.66	53730	25		2589.41	70574	51	1223.22	50476	33
	7		2120.44	73876	46	1697.27	76755	47		5226.40	89118	43	2147.01	78243	54
100	3		1281.84	84046	18	869.15	69089	14		1391.49	76712	24	1581.76	91822	25
	4		1910.48	80352	21	1667.03	57158	23		2596.91	88826	41	1838.74	64422	21
	5		2775.97	98360	39	1077.40	53798	16		3006.49	97259	47	1520.71	56047	19
	6		2867.54	91289	33	925.23	53999	20		4555.27	113668	43	2066.96	68737	33
	7		6070.26	107040	49	3054.90	87928	38		t			3984.57	91899	35
125	3		2476.75	94968	16	2975.88	64602	20		4451.72	135687	34	6046.55	134748	41
	4		5078.91	138503	46	3044.06	85393	22		7034.84	149321	47	3768.46	115333	26
	5		8537.01	135263	26	3110.27	91207	19		9566.48	144519	38	5187.52	127238	28
	6		0%	161253	39	3555.69	102055	23		0%	172197	68	7322.35	121563	35
	7		1%	147254	33	1%	139797	41		1%	150121	28	1%	126275	36
150	3		6856.67	143136	15	9025.34	155853	25		0%	162363	31	0%	111149	36
	4		7%	162270	25	1%	165872	25		1%	161600	26	10597.32	163260	35
	5		1%	164689	27	2%	150927	20		4%	151280	25	2%	167066	25
	6		3%	162743	22	0%	135489	22		5%	158253	23	12%	142334	30
	7		17%	157042	21	10%	141563	25		17%	176927	27	2%	156856	19
175	3		6%	112188	12	0%	173152	13		2%	166523	11	2%	134015	10
	4		8%	146711	10	10%	128322	12		9%	123723	17	11%	104565	14
	5		17%	117601	13	4%	155376	13		12%	158014	12	15%	113645	17
	6		15%	159425	10	8%	117879	15		14%	133947	22	14%	100007	14
	7		14%	151862	9	13%	107537	16		21%	122024	6	3%	140908	13
200	3		30%	165247	8	5%	154627	6		27%	158489	5	t		
	4		8%	157871	6	29%	156756	5		5%	208366	8	10%	182023	7
	5		14%	203388	9	22%	117791	4		18%	160111	6	8%	148075	5
	6		12%	156055	5	10%	157336	8		21%	119998	4	t		
	7		12%	152070	6	15%	118683	4		16%	178488	6	22%	118929	4

Table 5: U2A*p*HMP on Australia Post dataset (part 3)

indicated earlier in Section 3.4, we observed in most cases (across a few test implementations we ran) that a modification of BDS_{ij} – in which the objective function is minimised – results in stronger pareto optimal cuts, fewer Bitr, and faster convergence than those obtained by BDS_{ij}. As an instance, the computational time for 40.5(0.4) in the Turkish dataset in this experiment is decreased by 60%. However, in general, as we observed in Tables 3-6 for different procedures which use core points for acceleration of Benders decomposition (that is Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF), we are not able to realise the best core point for the strongest cut for all cases. This is also observed by Mercier et al. [29].

It is clear from the tables that a judicious choice of parameters for generating Benders cuts improves the performance of the Benders method. While the modified Benders approaches strengthen the Benders cuts through choosing a different slope of objective functions for SPs, Bns-SPX and Bns-MCNF do not take into the account the information of shortest paths for each pair of nodes. In contrast, $\boldsymbol{m}$-Bns-MCNF used this information to determine the slope of the Benders cuts. This resulted in an almost monotonic and effective convergence, with less than half the Bcuts of that of the other methods on average.

Figure 6 shows that on average, 94% of total computational times in Bns-SPX is dedicated to solve subproblems using the simplex method. The large computational time for solving n^2 subproblems

			B&B	BnsAuto	Bns-SPX			Bns-MCNF			m-Bns-MCNF				B&B	BnsAuto	Bns-SPX			Bns-MCNF			m-Bns-MCNF		
n	p	α	time	time	time	Bcuts	Bitr	time	Bcuts	Bitr	time	Bcuts	Bitr	α	time	time	time	Bcuts	Bitr	time	Bcuts	Bitr	time	Bcuts	Bitr
30	3	0.2	35	191	260	4854	10	5.5	4540	12	4.0	4517	12	0.6	26	192	251	6725	13	3.6	4481	10	2.9	3860	10
	4		31	195	284	6816	15	5.5	4980	14	4.3	3344	9		28	180	259	7732	16	6.8	5398	21	4.1	3749	14
	5		25	180	300	5398	12	4.5	4348	11	2.7	2886	7		29	176	354	8078	20	5.2	4023	12	2.2	3053	12
	6		15	183	302	8029	16	5.5	5201	12	2.8	3961	10		33	283	557	13517	30	5.4	4832	13	2.6	4131	10
	7		15	185	281	7348	17	5.9	5353	14	2.3	3499	9		38	239	511	9799	26	9.8	6096	19	3.8	3864	14
40	3		280	596	1798	12875	13	21.8	12009	15	15.6	8937	11		215	582	1526	9962	12	21.3	9295	16	11.2	6516	9
	4		267	2497	1003	10231	9	24.2	13189	20	16.4	6895	12		242	2628	2294	18144	21	24.2	8306	13	18.3	6179	16
	5		94	2429	1273	10154	12	12.7	8902	13	7.5	5983	10		211	2551	3611	24855	28	35.7	10437	33	12.0	6484	11
	6		105	2387	2897	19108	25	16.3	9015	15	7.0	5972	8		382	2554	2761	23450	23	61.0	13646	41	16.2	6610	12
	7		91	2489	2113	14169	18	19.4	8854	14	8.5	6925	10		442	2053	4879	31484	38	117.8	12370	23	23.1	8971	14
50	3		3715	t	2%	38613	26	95.2	20364	16	87.2	20461	19		2766	5136	9840	43384	27	66.8	16068	22	64.0	13898	20
	4		1005	5905	6787	19426	14	81.2	15549	18	28.0	8668	7		1433	5326	0%	37442	26	94.0	17170	20	37.8	10738	14
	5		488	5582	8786	25323	20	59.5	13938	11	21.8	8602	9		1490	6147	0%	39089	27	146.5	18474	31	24.4	10913	11
	6		348	4662	3%	31103	32	69.1	17420	16	16.4	9223	9		2014	6385	2%	37382	27	452.8	21536	58	57.6	16660	20
	7		596	4936	6988	24952	18	89.3	17434	16	23.9	10448	11		t	t	1%	44514	29	764.4	24663	50	118.7	14678	18
60	3		t	m	10%	18404	8	235.7	32530	22	328.3	31189	25		t	m	13%	20090	9	275.5	30264	28	275.8	27931	20
	4		6585	m	12%	22200	10	203.3	33215	23	220.5	27768	21		8201	m	7%	23730	10	338.6	28291	23	130.9	18651	22
	5		3626	m	16%	19365	10	184.2	27351	21	86.3	20651	15		t	m	9%	27264	12	218.2	20326	15	215.3	21336	24
	6		5170	m	15%	20952	10	288.9	32878	25	111.6	24529	20		t	m	14%	23001	10	1213.0	31113	37	269.5	25416	21
	7		4533	m	18%	24812	12	346.4	30459	27	112.7	21852	21		t	m	7%	28037	10	1536.7	38365	54	396.5	28612	33
70	3		t	m	46%	14556	4	438.4	47010	25	496.5	48739	27		t	m	24%	14559	4	354.5	36092	20	129.2	20474	6
	4		t	m	50%	9730	3	569.7	48657	32	234.9	29150	17		t	m	30%	14550	4	539.8	36096	27	506.5	34217	28
	5		t	m	18%	14554	4	780.3	48805	24	533.2	42653	43		t	m	38%	14506	4	454.8	36342	27	419.9	34567	27
	6		9374	m	40%	19379	5	373.1	32930	24	143.0	23152	17		t	m	36%	14550	4	1065.6	47543	54	282.8	25688	16
	7		7564	m	29%	14559	4	390.3	43129	34	144.9	24878	18		10173	m	19%	19339	5	1246.1	40676	35	198.3	24517	15
81	3		t	m	t			1459.4	78251	22	1530.4	69446	33		t	m	t			636.7	50251	21	456.1	36159	25
	4		t	m	t			3733.3	83526	45	1293.8	76287	31		t	m	t			3361.6	73917	53	1604.2	58071	33
	5		t	m	t			2262.0	83229	57	824.2	54765	31		t	m	t			4969.8	70966	49	2216.5	54330	43
	6		t	m	t			1275.1	56042	26	334.3	32113	12		t	m	t			3054.1	67502	48	1515.2	40575	30
	7		t	m	t			1192.1	43843	19	632.0	49965	27		t	m	t			3328.3	50929	31	1073.7	37241	26
30	3	0.4	30	188	176	4301	8	5.0	4680	9	3.8	2895	6	0.8	44	244	408	9828	22	4.4	4434	13	4.4	3585	12
	4		32	196	319	6642	15	4.9	4896	15	3.4	4020	14		72	255	428	11125	21	8.6	4541	16	6.5	4770	19
	5		26	169	262	6634	15	4.3	4686	11	3.1	3880	13		77	258	492	13025	27	10.2	5347	16	7.3	3826	14
	6		14	186	315	8647	21	4.8	4756	16	2.9	4172	10		89	458	800	22703	42	20.5	6491	24	9.1	5319	24
	7		14	181	495	10530	29	6.0	5370	15	2.4	3326	10		89	364	902	19675	39	30.9	5804	22	8.2	4830	20
40	3		221	584	778	7420	7	18.3	9923	13	15.7	8570	13		438	581	2080	18070	21	17.4	8434	14	12.9	7101	12
	4		231	2185	2811	19562	21	23.5	11016	14	15.5	8338	13		562	2200	4013	34375	33	26.5	11109	19	32.3	10350	27
	5		116	2427	2088	14161	18	12.3	7653	13	9.0	5981	10		905	2408	5535	41233	48	61.6	11967	26	42.2	10879	29
	6		159	2419	2664	17771	21	35.3	11484	20	6.7	5688	11		3135	3253	7028	48169	49	195.8	13628	32	130.6	13311	35
	7		113	2345	2333	19164	22	42.2	10240	18	7.3	6453	11		6073	3536	t			436.5	15898	45	150.3	10825	29
50	3		2017	6743	6802	25413	15	67.0	17710	18	51.0	15964	16		2372	t	5031	24256	12	71.4	14796	19	56.7	14284	18
	4		1011	6113	5802	22597	14	44.4	14264	12	20.3	9125	9		5073	t	1%	44622	28	201.0	21384	45	66.0	14502	19
	5		865	5794	8945	25398	20	59.6	16077	22	29.3	10584	10		4611	t	0%	47874	28	237.5	21740	44	85.3	14498	16
	6		1012	t	1%	36498	27	102.9	19086	37	23.7	11399	10		t	t	1%	49267	27	672.3	23615	48	289.0	17460	35
	7		2208	t	2%	32888	29	207.6	19921	23	76.2	16897	28		t	t	4%	47299	27	1687.5	28458	56	694.7	23077	44
60	3		t	m	11%	21776	10	358.2	35240	41	319.1	26270	24		t	m	7%	20674	9	337.3	28132	31	231.6	23379	23
	4		5794	m	4%	24256	10	133.1	23975	22	121.3	17959	25		t	m	6%	27067	11	412.6	31317	48	417.9	21544	20
	5		8227	m	15%	23493	11	353.2	31202	39	333.4	30308	50		t	m	11%	23085	11	1158.9	32831	44	400.4	23907	41
	6		t	m	19%	24044	11	571.2	26110	26	184.6	22776	18		t	m	17%	30969	13	2312.4	32295	45	1006.6	28402	35
	7		t	m	13%	20688	9	1003.3	34367	45	211.3	22768	21		t	m	11%	27598	11	6092.7	52504	68	3398.6	41304	54
70	3		t	m	37%	14555	4	316.1	36139	16	257.8	32338	23		t	m	18%	14559	4	347.8	29975	30	174.6	24980	18
	4		t	m	42%	14560	4	465.1	37460	30	402.9	40028	31		t	m	11%	18873	5	834.7	43393	48	631.7	34915	29
	5		t	m	42%	14557	4	857.3	45362	24	409.0	31232	36		t	m	36%	14296	4	3726.4	56078	63	744.0	36576	46
	6		t	m	29%	14556	4	747.4	46046	38	275.3	30796	21		t	m	36%	14485	4	1956.9	46242	35	877.9	31133	39
	7		8835	m	53%	14553	4	478.7	38739	31	256.4	28786	31		t	m	37%	14553	4	—	—		1091.0	34274	34
81	3		t	m	t			974.8	61919	25	1156.0	60422	31		t	m	t			656.0	47632	32	815.3	43890	27
	4		t	m	t			2661.6	79313	46	1363.2	52104	34		t	m	t			6152.6	87645	66	3562.8	65119	55
	5		t	m	t			2060.6	67522	39	1240.2	54757	33		t	m	t			6422.7	71088	29	2341.7	51958	27
	6		t	m	t			2421.3	49753	41	621.2	33804	20		t	m	t			8393.9	61897	70	2273.5	47354	34
	7		t	m	t			2644.0	60331	27	682.3	42406	29		t	m	t			0.9%	63633	42	3308.7	49113	40

Table 6: U2ApHMP on Turkish Mail dataset

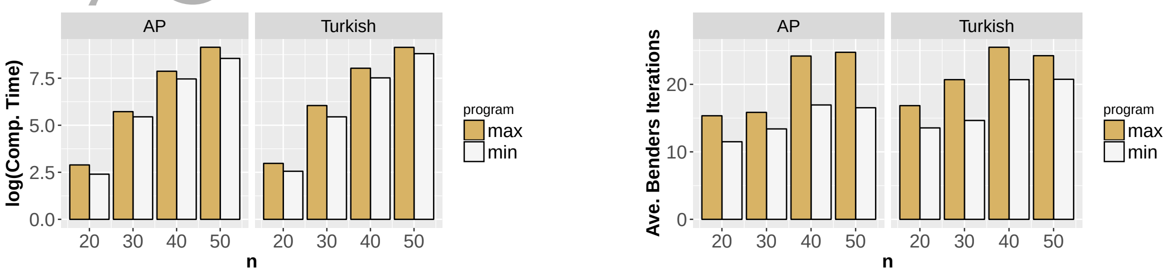
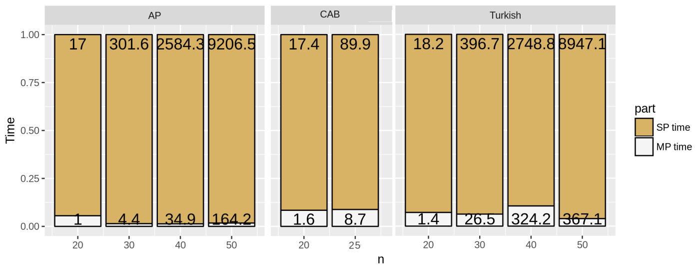
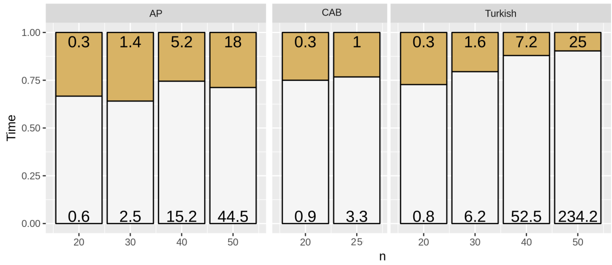


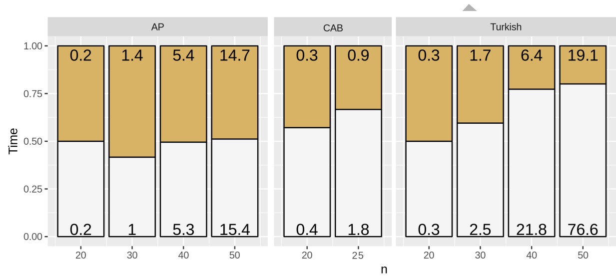
Figure 5: Computational efforts of `Bns-SPX` for $20 \leq n \leq 50$ when the objective function of BDS_{ij} is minimised/maximised.



(a) Bns-SPX



(b) Bns-MCNF



(c) $\boldsymbol{m}$-Bns-MCNF

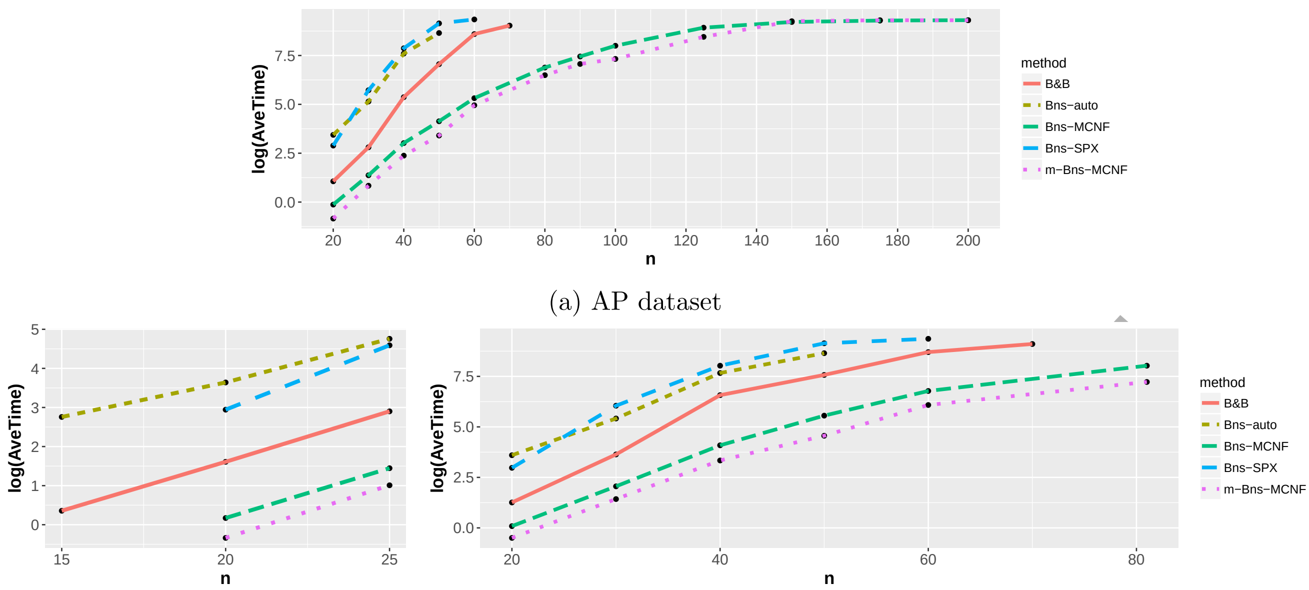
Figure 6: The portion of computational efforts for solving SPs and MP for $20 \leq n \leq 50$ for Bns-SPX, Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF. Average CPU time for each portion is shown in the related bar.

BDS_{ij} is not surprising since the simplex method is not the most efficient way to solve them. As is clear from computational results for Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF, using the successive shortest path algorithm, and Dijkstra algorithm (i) significantly reduces the computational times, and (ii) requires much less memory. The implementation of Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF on all instances used at most 8 Gb memory. Thus, the modified Benders decomposition together with efficient method of solving subproblems significantly improves our ability to tackle large instances of U2ApHMP.

On the other hand, by comparison of computational results of Bns-MCNF and $\boldsymbol{m}$-Bns-MCNF in Tables 2-6, our choice of the objective function slope of DS_{ij} has a significant impact on the number of iterations and computational times. $\boldsymbol{m}$-Bns-MCNF outperforms Bns-MCNF in computational times by 40%, and in Bitr by 22% on average. In general, $\boldsymbol{m}$-Bns-MCNF generates stronger Benders cuts which results in much shorter MP solution times as shown in Figure 6. Note that $\boldsymbol{m}$-Bns-MCNF outperforms Bns-MCNF in average computational effort or in the best solution gap for large instances.

Note that the hub location problems are mostly used for high-level strategic planning. Also real-world problems are large instances generally. Meta-heuristic methods may not guarantee the optimality of solutions which could result in a huge cost. Therefore, $\boldsymbol{m}$-Bns-MCNF is a much more efficient and reliable method than the other methods to tackle real-world problems.

Tables 7-8 present results for network designs of U2ApHMP and UrApHMP for $1 \leq r \leq p$. We analyse implications of different values of r and α on the operational costs and the number of allocations. In these tables, '#A' denotes the number of allocations, 'obj' denotes the optimal cost of U2ApHMP, and 'obj/U2A' denotes the proportion of the optimal value of UrApHMP to the optimal value of U2ApHMP. Note that UrApHMP with $r = 1$ and UrApHMP with $r = p$ are respectively equivalent to USApHMP and UMApHMP. In this experiment we used small fixed costs for access links. Hence, in a few cases in Table 8, the total costs of network designs of USApHMP are lower than those of U2ApHMP because of twice allocations. For UrApHMP, the number of allocations grows with α when



(a) AP dataset

(b) CAB dataset

(c) Turkish dataset

Figure 7: Comparison of average solution times on U2ApHMP with respect to the number of nodes

n and p are fixed. Also, for smaller transfer cost α, the number of allocations for more non-hubs is one since the lowest cost solution is to allocate each non-hub to its nearest hub. Note that the operational costs and the number of allocations by U2ApHMP are upper bounds for those values by UrApHMP with $r = 2$. However, for smaller values of α, U2ApHMP and UrApHMP with $r = 2$ have larger gaps in those values. The gap between the optimal values for the same instance is only for the access link fixed costs. In many cases, the number of allocations by UrApHMP with $r > 2$ is smaller than that of U2ApHMP, while the operational costs are smaller. This indicates that some non-hubs in network designs by UrApHMP are singly allocated. Also, for larger values of r and especially for larger values of α, the number of allocations by UrApHMP increases in favour of reducing the operational costs as compared to U2ApHMP.

For instance 20.5.(0.8) in the CAB dataset, as an example, U2ApHMP provides 0.1% higher optimal value than UrApHMP with $r = 2$ for 2 additional access links which keeps the network survivable. In this example, UrApHMP with $r = 3$ and $r = 4$, respectively, have 10 and 14 more access links (without guaranteeing the network is 2-connected) than U2ApHMP, which result in 2.1% and 2.8% lower costs, respectively. However, UrApHMP network designs do not provide ultimate solutions when survivability is essential. In general, a solution by U2ApHMP is similar to that of UrApHMP with $r = 2$ with slightly larger total costs due to the fixed costs associated with, probably, a few more access links. In this experiment, since the fixed costs of access links are sufficiently small, a solution of U2ApHMP can be obtained from a solution of UrApHMP with $r = 2$ by allocating singly allocated non-hubs to their nearest unused hubs.

5 Conclusions

In this paper, we have introduced the U2ApHMP. This is a variant of hub location problems that captures survivability of networks. Although the manner in which we model survivability is a proxy or

			U2ApHMP		UrApHMP													
					$r=1$		$r=2$		$r=3$		$r=4$		$r=5$		$r=6$		$r=7$	
n	p	α	#A	obj	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A
20	3	0.2	34	4837871645	17	101.102%	21	99.680%	21	99.680%								
		0.4	34	5365681823	17	104.373%	25	99.784%	25	99.784%								
		0.6	34	5835733939	17	108.118%	28	99.850%	29	99.799%								
		0.8	34	6236303446	17	112.544%	32	99.924%	36	99.571%								
	4	0.2	32	4275868363	16	101.237%	22	99.696%	22	99.696%	22	99.696%						
		0.4	32	5002008793	16	103.777%	24	99.785%	27	99.225%	27	99.225%						
		0.6	32	5665425831	16	106.395%	27	99.843%	31	98.027%	31	98.027%						
		0.8	32	6184103897	16	110.690%	29	99.904%	38	97.522%	38	97.522%						
	5	0.2	30	3886883844	15	100.995%	21	99.713%	22	99.657%	22	99.657%	22	99.657%				
		0.4	30	4707486453	15	103.329%	23	99.788%	25	99.244%	26	98.986%	26	98.986%				
		0.6	30	5475744684	15	105.909%	25	99.845%	29	98.492%	31	98.066%	31	98.066%				
		0.8	30	6174332880	15	109.072%	28	99.901%	40	97.873%	44	97.207%	44	97.207%				
	6	0.2	28	3857574090	14	100.541%	19	99.764%	20	99.692%	21	99.653%	21	99.653%	21	99.653%		
		0.4	28	4723681189	14	101.874%	21	99.829%	23	99.322%	24	99.219%	24	99.219%	24	99.219%		
		0.6	28	5532563209	14	103.782%	25	99.887%	29	98.709%	32	98.496%	32	98.496%	32	98.496%		
		0.8	28	6266827565	14	106.456%	27	99.926%	36	98.284%	42	97.884%	42	97.884%	42	97.884%		
	7	0.2	26	3890292159	13	100.255%	17	99.771%	18	99.662%	19	99.615%	19	99.615%	19	99.615%	19	99.615%
		0.4	26	4780444700	13	101.491%	21	99.861%	23	99.399%	24	99.296%	24	99.296%	24	99.296%	24	99.296%
		0.6	26	5599400794	13	103.387%	24	99.900%	29	99.065%	32	98.854%	32	98.854%	32	98.854%	32	98.854%
		0.8	26	6374043309	13	105.520%	25	99.926%	33	98.730%	38	98.363%	39	98.317%	39	98.317%	39	98.317%
25	3	0.2	44	7326753535	22	101.301%	27	99.720%	27	99.720%								
		0.4	44	8234566445	22	104.176%	31	99.798%	31	99.798%								
		0.6	44	9006625381	22	107.791%	36	99.872%	37	99.817%								
		0.8	44	9648569755	22	111.793%	41	99.932%	46	99.606%								
	4	0.2	42	6465260232	21	101.123%	28	99.723%	28	99.723%	28	99.723%						
		0.4	42	7658613368	21	103.426%	30	99.783%	33	99.283%	33	99.283%						
		0.6	42	8699118340	21	106.189%	34	99.855%	40	98.911%	39	98.900%						
		0.8	42	9545191372	21	110.156%	38	99.922%	53	98.215%	53	98.215%						
	5	0.2	40	6025840995	20	100.814%	26	99.728%	27	99.692%	27	99.692%	27	99.692%				
		0.4	40	7319356917	20	102.754%	29	99.800%	31	99.382%	32	99.194%	32	99.194%				
		0.6	40	8490224981	20	105.572%	33	99.871%	37	98.831%	39	98.526%	39	98.526%				
		0.8	40	9518074631	20	109.246%	37	99.950%	53	98.126%	59	97.608%	59	97.608%				
	6	0.2	38	5826458485	19	100.862%	25	99.770%	26	99.732%	26	99.732%	26	99.732%	26	99.732%		
		0.4	38	7164231902	19	102.799%	28	99.836%	30	99.410%	32	99.224%	32	99.224%	32	99.224%		
		0.6	38	8410765446	19	105.216%	32	99.894%	36	98.844%	39	98.496%	39	98.496%	39	98.496%		
		0.8	38	9493453537	19	108.802%	36	99.941%	52	98.092%	61	97.473%	61	97.473%	61	97.473%		
	7	0.2	36	5784401827	18	100.548%	24	99.780%	25	99.742%	25	99.742%	25	99.742%	25	99.742%	25	99.742%
		0.4	36	7109892375	18	102.667%	28	99.875%	30	99.504%	32	99.409%	32	99.409%	32	99.409%	32	99.409%
		0.6	36	8392731734	18	104.601%	32	99.921%	36	98.917%	41	98.631%	41	98.631%	41	98.631%	41	98.631%
		0.8	36	9527483370	18	107.669%	35	99.957%	49	98.359%	58	97.780%	58	97.780%	58	97.780%	58	97.780%

Table 7: A comparison of network designs by U2ApHMP and UrApHMP with various values of r on the CAB dataset

a surrogate to true/robust survivable hub model development, we believe that our effort represents a first step towards more general hub network models that consider survivability. The new problem that we present here (that is, U2ApHMP) is shown to be NP-hard even when the hub locations are known *a priori*. Thus, our conjecture is that any new model that is developed for complete, exhaustive and robust survivable hub network designs is going to be significantly harder than the model that we have developed in this paper. Thus, in some sense, we believe that this contribution may open up a new thrust of research in hub location models and develop more insightful and deep contributions towards survivable hub network design.

It has been only recently shown that large instances of hub location problems can be solved using exact methods. With our (i) new approach for efficiently solving subproblems, and (ii) a more judicious and effective choice of core-points in the acceleration of Benders method, the boundary for solving large hub location instances using exact methods is likely to be pushed even further. Since the nature of other hub location problems (and the subproblems in their Benders decompositions) are similar to that of the U2ApHMP that we have considered in this paper, we expect our approach to be efficient for

			U2ApHMP		UrApHMP													
					$r=1$		$r=2$		$r=3$		$r=4$		$r=5$		$r=6$		$r=7$	
n	p	α	#A	obj	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A	#A	obj/U2A
AP																		
20	4	0.2	32	112059700	16	100.170%	19	99.739%	19	99.739%	19	99.739%						
		0.4	32	119881868	16	100.919%	21	99.787%	22	99.776%	22	99.776%						
		0.6	32	127509656	16	101.667%	21	99.799%	24	99.589%	24	99.589%						
		0.8	32	134832172	16	102.560%	24	99.846%	27	99.375%	27	99.375%						
	5	0.2	30	97668214	15	100.042%	18	99.760%	18	99.760%	18	99.760%	18	99.7601%				
		0.4	30	106158519	15	101.011%	20	99.813%	21	99.802%	21	99.802%	21	99.8018%				
		0.6	30	114381245	15	101.956%	21	99.833%	24	99.580%	25	99.580%	25	99.5796%				
		0.8	30	122308258	15	102.974%	23	99.870%	26	99.346%	27	99.306%	27	99.3061%				
	6	0.2	28	83759366	14	100.055%	17	99.725%	17	99.725%	17	99.725%	17	99.7254%	17	99.725%		
		0.4	28	92930541	14	101.166%	18	99.764%	19	99.751%	19	99.751%	19	99.7509%	19	99.751%		
		0.6	28	101935601	14	102.134%	19	99.792%	21	99.606%	21	99.606%	21	99.6059%	21	99.606%		
		0.8	28	110293458	14	102.854%	22	99.845%	25	99.446%	26	99.423%	26	99.4230%	26	99.423%		
	7	0.2	26	73584155	13	99.915%	17	99.749%	17	99.749%	17	99.749%	17	99.7490%	17	99.749%	17	99.749%
		0.4	26	82812079	13	100.887%	18	99.779%	18	99.779%	18	99.779%	18	99.7788%	18	99.779%	18	99.779%
		0.6	26	91923506	13	101.792%	19	99.809%	21	99.773%	21	99.773%	21	99.7731%	21	99.773%	21	99.773%
		0.8	26	100894683	13	102.677%	21	99.853%	24	99.626%	25	99.619%	25	99.6189%	25	99.619%	25	99.619%
40	4	0.2	72	126008496	36	99.588%	38	99.514%	38	99.514%	38	99.514%						
		0.4	72	132957814	36	100.302%	44	99.634%	45	99.518%	46	99.506%						
		0.6	72	138684280	36	100.968%	48	99.674%	49	99.515%	50	99.485%						
		0.8	72	144156800	36	101.761%	49	99.695%	52	99.444%	53	99.400%						
	5	0.2	70	113576643	35	99.621%	37	99.561%	37	99.561%	37	99.561%	37	99.5614%				
		0.4	70	120994765	35	100.258%	45	99.662%	46	99.577%	46	99.577%	46	99.5771%				
		0.6	70	127703891	35	101.381%	47	99.704%	50	99.531%	50	99.531%	50	99.5311%				
		0.8	70	134053795	35	102.415%	50	99.744%	55	99.442%	56	99.431%	56	99.4315%				
	6	0.2	68	103331165	34	99.610%	37	99.534%	37	99.534%	37	99.534%	37	99.5344%	37	99.534%		
		0.4	68	111690022	34	100.120%	43	99.621%	45	99.495%	46	99.480%	46	99.4802%	46	99.480%		
		0.6	68	119043737	34	101.081%	46	99.678%	48	99.489%	49	99.455%	49	99.4547%	49	99.455%		
		0.8	68	126120380	34	101.915%	48	99.707%	52	99.402%	53	99.352%	53	99.3516%	53	99.352%		
	7	0.2	66	95744927	33	99.564%	36	99.518%	35	99.513%	35	99.513%	35	99.5127%	35	99.513%	35	99.513%
		0.4	66	104476812	33	100.233%	43	99.641%	44	99.560%	44	99.560%	44	99.5597%	44	99.560%	44	99.560%
		0.6	66	112421729	33	101.140%	45	99.687%	46	99.560%	46	99.560%	46	99.5601%	46	99.560%	46	99.560%
		0.8	66	119942768	33	101.968%	49	99.752%	54	99.529%	55	99.527%	55	99.5267%	55	99.527%	55	99.527%
Turkish																		
20	4	0.2	32	1137279747	16	99.802%	19	99.285%	19	99.285%	19	99.285%						
		0.4	32	1412588566	16	101.900%	23	99.569%	23	99.569%	23	99.569%						
		0.6	32	1662799404	16	104.209%	26	99.719%	28	99.510%	29	99.450%						
		0.8	32	1884805959	16	105.568%	28	99.832%	31	99.157%	33	99.002%						
	5	0.2	30	893437782	15	100.266%	19	99.201%	19	99.201%	19	99.201%	19	99.201%				
		0.4	30	1214099947	15	101.647%	20	99.448%	21	99.390%	21	99.390%	21	99.390%				
		0.6	30	1524249957	15	103.157%	24	99.669%	28	99.156%	29	99.105%	29	99.105%				
		0.8	30	1797294845	15	105.719%	27	99.808%	32	98.569%	34	98.419%	34	98.419%				
	6	0.2	28	812761970	14	99.295%	16	99.160%	16	99.160%	16	99.160%	16	99.160%	16	99.160%		
		0.4	28	1148088899	14	100.670%	18	99.466%	20	99.415%	21	99.411%	21	99.411%	21	99.411%		
		0.6	28	1468379336	14	102.317%	23	99.704%	26	99.299%	29	99.242%	30	99.184%	30	99.184%		
		0.8	28	1753769720	14	104.693%	25	99.832%	30	98.872%	32	98.735%	36	98.645%	36	98.645%		
	7	0.2	26	745189079	13	99.397%	15	99.251%	15	99.251%	15	99.251%	15	99.251%	15	99.251%	15	99.251%
		0.4	26	1087515315	13	100.903%	17	99.546%	18	99.523%	19	99.516%	19	99.516%	19	99.516%	19	99.516%
		0.6	26	1416761759	13	102.267%	21	99.752%	24	99.282%	27	99.095%	28	99.035%	28	99.035%	28	99.035%
		0.8	26	1714429049	13	104.363%	23	99.841%	30	99.077%	32	98.936%	33	98.846%	33	98.846%	33	98.846%
40		0.4	72	6899266260	36	102.156%	52	99.811%	54	99.512%	54	99.512%						
		0.6	72	7608891542	36	104.732%	55	99.855%	62	98.992%	64	98.974%						
		0.8	72	8226329370	36	107.012%	61	99.900%	70	98.375%	75	98.211%						
	5	0.2	70	5175192373	35	100.677%	44	99.726%	46	99.578%	46	99.578%	46	99.578%				
		0.4	70	6229793093	35	102.643%	53	99.836%	61	99.174%	61	99.174%	61	99.174%				
		0.6	70	7150519975	35	105.251%	59	99.899%	69	98.552%	71	98.533%	71	98.533%				
		0.8	70	7934761604	35	107.330%	65	99.953%	82	97.453%	90	97.204%	90	97.204%				
	6	0.2	68	4681428594	34	100.503%	42	99.698%	44	99.545%	44	99.545%	44	99.545%	44	99.545%		
		0.4	68	5830108974	34	102.439%	51	99.825%	59	99.155%	59	99.155%	59	99.155%	59	99.155%		
		0.6	68	6868417931	34	104.699%	57	99.896%	69	98.372%	70	98.363%	70	98.363%	70	98.363%		
		0.8	69	7751927668	34	106.874%	64	99.951%	83	97.441%	93	97.129%	94	97.129%	94	97.129%		
	7	0.2	66	4303099420	33	100.460%	47	99.760%	48	99.729%	48	99.729%	48	99.729%	48	99.729%	48	99.729%
		0.4	66	5505170515	33	102.583%	52	99.844%	60	99.373%	60	99.373%	60	99.373%	60	99.373%	60	99.373%
		0.6	66	6613191655	33	104.185%	57	99.899%	70	98.372%	72	98.354%	72	98.354%	72	98.354%	72	98.354%
		0.8	66	7589067230	33	106.218%	62	99.947%	87	97.452%	95	97.092%	101	96.888%	102	96.887%	102	96.887%

Table 8: A comparison of network designs by U2ApHMP and UrApHMP with various values of r on the AP and Turkish datasets

implementing on other hub location problems, including the multiple allocation p-hub median problems [14, 16], and also the multiple allocation hub location problems [6].

The computational results on other variations of HLPs [30, 31] also confirms that our approach can be implemented on other hub location problems with a view to (i) solving larger instances, and (ii) improving the state-of-the-art in terms of computational efficiency. Of course, further research is needed to document how efficient is our new Benders decomposition based approach for solving various hub location problems that have already been looked at in the literature. In addition to the above contribution, in this paper, we also introduced U2ApHMP, an important new problem to the literature.

Acknowledgement The authors are grateful for the many constructive comments from the anonymous referees and the guest editors of the special issue. Their comments improved the content and presentation of this paper significantly.

References

[1] Allocation strategies in hub networks. *Eur J Oper Res*, 211(3):442–451, 2011.
[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network flows: theory, algorithms, and applications. 1993.
[3] S. Alumur and B.Y. Kara. Network hub location problems: The state of the art. *Eur J Oper Res*, 190(1):1–21, 2008. doi: 10.1016/j.ejor.2007.06.008.
[4] Y. An, Y. Zhang, and B. Zeng. The reliable hub-and-spoke design problem: Models and algorithms. *Transport Res B-Meth*, 77:103–122, 2015.
[5] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4(1): 238–252, 1962.
[6] J.F. Campbell. Integer programming formulations of discrete hub location problems. *Eur J Oper Res*, 72(2):387–405, 1994. doi: 10.1016/0377-2217(94)90318-2.
[7] J.F. Campbell. Hub location and the p-hub median problem. *Oper. Res.*, 44(6):923–935, 1996.
[8] J.F. Campbell, A.T. Ernst, and M. Krishnamoorthy. Hub location problems. *Facility location: applications and theory*, 1:373–407, 2002.
[9] R.H. Cardwell, C.L. Monma, and T-H Wu. Computer-aided design procedures for survivable fiber optic networks. *IEEE Journal on Selected Areas in Communications*, 7(8):1188–1197, 1989.
[10] S. Çetiner, C. Sepil, and H. Süral. Hubbing and routing in postal delivery systems. *Ann. Oper. Res.*, 181(1):109–124, 2010.
[11] I. Contreras, J.-F. Cordeau, and G. Laporte. Benders decomposition for large-scale uncapacitated hub location. *Oper. Res.*, 59(6):1477–1490, 2011.
[12] R.S. de Camargo, G.D. Miranda, and H. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Comput. Oper. Res.*, 35(4):1047–1064, 2008.
[13] J. Ebery, M. Krishnamoorthy, A.T. Ernst, and N. Boland. The capacitated multiple allocation hub location problem: Formulations and algorithms. *Eur J Oper Res*, 120(3):614–631, 2000.
[14] A.T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, 4(3):139–154, 1996.
[15] A.T. Ernst and M. Krishnamoorthy. An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS J. Comput.*, 10(2):149–162, 1998.
[16] A.T. Ernst and M. Krishnamoorthy. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *Eur J Oper Res*, 104(1):100–112, 1998.
[17] R.Z. Farahani, M. Hekmatfar, A.B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Comput. Ind. Eng.*, 64(4):1096–1109, 2013. doi: 10.1016/j.cie.2013.01.012.
[18] M. Fischetti, D. Salvagnin, and A. Zanette. A note on the selection of benders cuts. *Math Prog*, 124(1):175–182, 2010.
[19] M. Grötschel, C.L. Monma, and M. Stoer. Design of survivable networks. In M.O. Ball, editor, *Handbooks in operations research and management science*, volume 7, pages 617–672. Elsevier, 1995.
[20] P. Jaillet, G. Song, and G. Yu. Airline network design and hub location problems. *Location science*, 4(3):195–212, 1996.
[21] H. Kim. p-hub protection models for survivable hub network design. *J Geogr Syst*, 14(4):437–461, 2012.
[22] H. Kim and M.E. O'Kelly. Reliable p-hub location problems in telecommunication networks. *Geogr Anal*, 41(3): 283–306, 2009.

[23] H. Kim and M.S. Ryerson. The q-ad hoc hub location problem for multi-modal networks. *Networks and Spatial Economics*, 17(3):1015–1041, 2017. doi: 10.1007/s11067-017-9357-y.
[24] J.G. Klincewicz. Hub location in backbone/tributary network design: a review. *Location Science*, 6(1):307–335, 1998.
[25] J.G. Klincewicz. Enumeration and search procedures for a hub location problem with economies of scale. *Ann. Oper. Res.*, 110(1):107–122, 2002.
[26] R.F. Love, J.G. Morris, and G.O. Wesolowsky. *Facilities location.* North-Holland, Amsterdam, 1988.
[27] T.L. Magnanti and R.T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.*, 29(3):464–484, 1981.
[28] T.C. Matisziw, A.T. Murray, and T.H." Grubesic. Strategic network restoration. *Networks and Spatial Economics*, 10(3):345–361, 2010. doi: 10.1007/s11067-009-9123-x.
[29] A. Mercier, J.-F. Cordeau, and F. Soumis. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Comput. Oper. Res.*, 32(6):1451–1476, 2005.
[30] H. Mokhtar, M. Krishnamoorthy, and A.T. Ernst. A new benders decomposition acceleration procedure for large scale multiple allocation hub location problems. In *22nd International Congress on Modelling and Simulation*, pages 340–346. The Modelling & Simulation Society of Australia and New Zealand (MSSANZ), 2017.
[31] H. Mokhtar, M. Krishnamoorthy, and A.T. Ernst. A modified Benders method for the single and multiple allocation p-hub median problem. In *Operations Research Proceedings 2017*. Springer, 2018.
[32] C.L. Monma and D.F. Shallcross. Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research*, 37(4):531–541, 1989.
[33] M.E. O'Kelly. The location of interacting hub facilities. *Transport Sci*, 20(2):92–106, 1986.
[34] M.E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *Eur J Oper Res*, 32(3): 393–404, 1987. doi: 10.1016/S0377-2217(87)80007-3.
[35] M.E. O'Kelly. Network hub structure and resilience. *Networks and Spatial Economics*, 15(2):235–251, 2015.
[36] N. Papadakos. Practical enhancements to the Magnanti–Wong method. *Oper. Res. Lett.*, 36(4):444–449, 2008.
[37] W.B. Powell and Y. Sheffi. The load planning problem of motor carriers: Problem description and a proposed solution approach. *Transportation Research Part A: General*, 17(6):471–480, 1983.
[38] G. Sen and M. Krishnamoorthy. Discrete particle swarm optimization algorithms for two variants of the static data segment location problem. *Appl Intell (to appear)*, pages 1–20, 2017. doi: 10.1007/s10489-017-0995-z.
[39] G. Sen, M. Krishnamoorthy, N. Rangaraj, and V. Narayanan. Exact approaches for static data segment allocation problem in an information network. *Comput. Oper. Res.*, 62:282–295, 2015.
[40] G. Sen, M. Krishnamoorthy, N. Rangaraj, and V. Narayanan. Mathematical models and empirical analysis of a simulated annealing approach for two variants of the static data segment allocation problem. *Networks*, 68(1):4–22, 2016. doi: 10.1002/net.21675.
[41] D. Skorin-Kapov and J. Skorin-Kapov. On tabu search for the location of interacting hub facilities. *Eur J Oper Res*, 73(3):502–509, 1994.
[42] K. Smith, M. Krishnamoorthy, and M. Palaniswami. Neural versus traditional approaches to the location of interacting hub facilities. *Location Science*, 4(3):155–171, 1996.
[43] J. Sohn and S. Park. The single allocation problem in the interacting three-hub network. *Networks*, 35(1):17–25, 2000.
[44] S. Soni, R. Gupta, and H. Pirkul. Survivable network design: The state of the art. *Information Systems Frontiers*, 1(3):303–315, 1999.
[45] M. Stoer. Design of survivable networks. 2006.
[46] P.Z. Tan and B.Y. Kara. A hub covering model for cargo delivery systems. *Networks*, 49(1):28–39, 2007.
[47] J. Xu. *Topological structure and analysis of interconnection networks*, volume 7. Springer Science & Business Media, 2013.
[48] H. Yaman. Star p-hub median problem with modular arc capacities. *Comput. Oper. Res.*, 35(9):3009–3019, 2008.