Benders Decomposition for Profit Maximizing Hub Location Problems with Capacity Allocation

Gita Taherkhani, Sibel A. Alumur
Department of Management Sciences, University of Waterloo, Waterloo, Ontario, Canada
gita.taherkhani@uwaterloo.ca, sibel.alumur@uwaterloo.ca

Seyed Mojtaba Hosseini
The Paul Merage School of Business, University of California, Irvine, California, USA, smhosse1@uci.edu

This paper models capacity allocation decisions within profit maximizing hub location problems to satisfy demand of commodities from different market segments. A strong deterministic formulation of the problem is presented and two exact algorithms based on a Benders reformulation are described to solve large-size instances of the problem. A new methodology is developed to strengthen the Benders optimality cuts by decomposing the subproblem in a two-phase fashion. The algorithms are enhanced by the integration of improved variable fixing techniques. The deterministic model is further extended by considering uncertainty associated with the demand to develop a two-stage stochastic program. To solve the stochastic version, a Monte-Carlo simulation-based algorithm is developed that integrates a sample average approximation scheme with the proposed Benders decomposition algorithms. Novel acceleration techniques are presented to improve the convergence of the algorithms proposed for the stochastic version. The efficiency and robustness of the algorithms are evaluated through extensive computational experiments. Computational results show that large-scale instances with up to 500 nodes and three demand segments can be solved to optimality, and that the proposed algorithms generate cuts that provide significant speedups compared to using Pareto-optimal cuts. The proposed two-phase methodology for solving the Benders subproblem as well as the variable fixing and acceleration techniques can be used to solve other discrete location and network design problems.

Key words: Hub location, Benders decomposition, variable fixing, acceleration techniques, sample average approximation.

---

1. Introduction

Hubs are special facilities that serve as switching, sorting, connecting, and consolidation points in transportation and telecommunication networks, routing the traffic between origin and destination pairs. Hubs help reduce the cost of establishing a network connecting many origins and destinations, and also consolidate flows to exploit economies of scale. A hub location problem is a network design problem consisting generally of two main decisions: to locate a set of hubs and to allocate the demand nodes to these hubs (Campbell et al. 2002, Alumur and Kara 2008, Contreras 2015).

One of the basic requirements in classical hub location problems is that the demand of each commodity must be fully satisfied. From a revenue management point of view, it may be more advantageous not to serve the demand of some commodities, especially if the cost of serving a commodity is higher than the revenue associated with servicing its demand. In such a setting, the right objective is to maximize profit rather than minimizing cost so that the decision on how much demand of each commodity to satisfy depends on the trade-off between revenue and cost. Profit maximization is more complex than cost minimization as it incurs additional decisions, nevertheless, this is the actual problem setting in many real-life hub networks. Moreover, the demand of commodities usually consists of different classes (e.g., for regular and priority service). The decision maker thus needs to consider how to allocate the available capacity to these different demand segments, while determining the proportion of the demand to serve for each class. Incurring these revenue management decisions surely bring on extra challenges yet this is a much more realistic problem setting.

This study is motivated from airline passenger and freight transportation, and express shipment delivery networks in which the amount of demand of different commodities to serve is very dependent on hub locations. To provide the big picture, in 2017 alone, worldwide airline industry provided services to more than 4.081 billion passengers across the world (IATA 2018); the world's largest package delivery company served 5.1 billion parcels and documents in the global delivery volume (UPS 2018), both by employing hub networks. Even though passengers choose their own routes in airline networks, airlines still have to design their hub networks considering the forecasted demand. Different classes of demand may include the demand for, for example, the first, business, and economy class service. In express shipment delivery networks, on the other hand, there is a demand for services such as priority, express, and standard mail.

In this study, we consider revenue management decisions within hub location problems and determine how to allocate available capacities of hubs to demand of commodities from different market segments. *The profit maximizing capacitated hub location problems* introduced in this study seek to find an optimal hub network structure, maximizing total profit to provide services to a set of commodities while considering the design cost of the network. The decisions to be made are the optimal number and locations of hubs, allocation of demand nodes to these hubs, and the optimal routes of flow of different classes of commodities that are selected to be served. We consider a hub location problem with multiple assignments, and allow a path of an origin-destination pair to pass through at most two hubs. Direct connections between non-hub nodes are not allowed; all commodities must be routed via a set of hubs. Demand is segmented into different classes and revenue is obtained from satisfying demands for the commodities of each class. The model is to decide how much demand to serve from each class considering the available capacity. Each demand class from each commodity can be partially satisfied.

We first present a strong deterministic mixed-integer programming formulation of the profit maximizing capacitated hub location problem. As precise information on demand may not be known in advance, we then incorporate demand uncertainty into the problem, and develop a two-stage stochastic program with recourse considering stochastic demand. The first stage decision is the locations of hubs, while the assignment of demand nodes to hubs, optimal routes of flows, and capacity allocation decisions are made in the second stage.

We present two exact algorithms based on a Benders decomposition of the deterministic formulation. Since the subproblem is inseparable, solving it can be as challenging as solving the original problem. Moreover, due to degeneracy in the subproblem, straightforward implementation of Benders decomposition suffers from low convergence. Alleviating these deficiencies, in this paper, we propose a general methodology for decomposing inseparable subproblems into smaller problems in a two-phase fashion, where optimality and strength of the cuts are guaranteed in phase I and phase II, respectively. More specifically, for the profit maximizing capacitated hub location problem, we prove that the second phase can be solved as a set of LP-relaxations of maximum weighted matching problems, or as a series of LP-relaxations of knapsack problems. We enhance these algorithms by incorporating improved variable fixing techniques.

We integrate the proposed Benders decomposition algorithms with a sample average approximation (SAA) scheme to solve the stochastic problem with a continuous demand distribution and an infinite number of scenarios. Inspired by the repetitive nature of SAA, we additionally propose novel acceleration techniques to improve the convergence of the algorithms employed for the stochastic problem. Moreover, we perform extensive computational experiments to evaluate the efficiency and robustness of the proposed algorithms and solve large-scale instances of the problem.

The major contribution of this paper is the methodology. We develop a new two-phase methodology for solving the subproblem to strengthen the Benders optimality cuts, and propose two new Benders-based algorithms to optimally solve large-scale instances of the profit maximizing capacitated hub location problems. We show that both algorithms result in generating very strong cuts that outperform the best known cuts from the literature (Pareto-optimal cuts; Magnanti and Wong 1981) in terms of both quality and computational efficiency. We also propose several variable fixing techniques improving the convergence. For the stochastic problem, our contribution is to propose a general technique for accelerating Benders decomposition coupled with SAA by using the cuts generated from the previous replications. We show that implementation of this acceleration technique may result in up to five times improvement in CPU times. The decomposition methodology for solving the Benders subproblem as well as the improved variable fixing and the SAA acceleration techniques proposed in this paper can be used to solve several classes of transportation and network optimization problems, including, but not limited to, other facility location and network design problems.

We were able to solve, to optimality, instances of the deterministic model, up to 500 nodes and 750,000 commodities of different demand segments. For the stochastic model, we could solve instances with up to 75 nodes and 16,875 commodities. To the best of our knowledge, these are the largest set of instances that have been solved exactly for any type of capacitated or stochastic hub location problems.

The rest of the paper is organized as follows. In Section 2, we review the related literature. We introduce the notation and formulate the mixed-integer linear programming models in Section 3. Section 4 contains the basic Benders decomposition algorithm and our introduction of several features that improve the convergence and efficiency for the deterministic version of the problem. In Section 5, we present an SAA algorithm, coupled with Benders decomposition, for the problem with stochastic demand. We perform extensive computational experiments in Section 6 to test our mathematical models and evaluate our algorithms. Section 7 provides some concluding remarks along with future research directions. Finally, technical extensions and supplementary numerical results are provided in an Appendix.

2. Literature Review

In the last few decades, many variants of hub location problems have been studied in the literature. The reader may refer to reviews on this area by Campbell et al. (2002), Alumur and Kara (2008), Campbell and O'Kelly (2012), and Contreras (2015). In this section, we focus in particularly on the capacitated hub location studies, hub location problems with profit-oriented objectives, hub location research considering uncertainty, and also on publications employing Benders decomposition for solving hub location problems.

Capacitated versions of hub location problems are first formulated by Campbell (1994) using path-based mixed integer programs that impose capacity constraints on the total incoming flow at hubs (i.e., flow arriving from both hub and non-hub nodes). A variant of this problem arises when capacities are applied only to the traffic arriving directly from non-hub nodes. This variant is motivated from the postal-delivery applications and has been studied by Ebery et al. (2000), Boland et al. (2004), Marín (2005), and Contreras et al. (2012). In some postal-delivery and express shipment networks (e.g. Canada Post, UPS), however, total incoming flow (regardless that is from a hub or a non-hub node) is sorted at every hub-stop. Moreover, the limiting capacity of a hub may not necessarily be on sorting or material handling, but, for example, on the available number of docks or gates. Hence, in this paper, we model the generic case and impose capacity constraints on the total incoming flow at hubs, as introduced in Campbell (1994). As we elaborate in Section 4.3, this generic definition of capacity usage brings on extra challenges for the implementation of Benders decomposition.

There are only a few studies considering a profit-oriented objective in hub location problems. Alibeyg et al. (2016) present a hub network design problem with profits including the decisions on the origin-destination pairs that will be served. They propose different variants of this problem and use the CPLEX solver to perform computational experiments. The companion paper Alibeyg et al. (2018) presents an exact solution algorithm for the profit-oriented models introduced by Alibeyg et al. (2016). They employ Lagrangean relaxation within a branch-and-bound algorithm, and also adopt reduction tests and partial enumeration to reduce the size of the problems as well as the computation times. They are able to solve instances with up to 100 nodes.

Lin and Lee (2018) consider a hub network design problem for time definite LTL freight transportation in which the carrier aims to determine hub locations, under price elasticity of demand, that maximize total profit. They show that profit optimization builds a denser hub network than cost minimization. More recently, Taherkhani and Alumur (2018) provide new formulations for profit maximizing hub location problems. They allow the demand of an origin-destination pair to be shipped through any number of hubs and network connections as necessary. They consider all possible allocation strategies: multiple allocation, single allocation, and r-allocation, and also allow for the possibility of direct connections between non-hub nodes. In the present study, we additionally consider demands of commodities from different market segments and incorporate capacity allocation decisions.

To the best of the authors' knowledge, this is the first paper incorporating demand uncertainty into profit maximizing hub location problems. There are, however, several publications that incorporate uncertainty into other hub location problems. Marianov and Serra (2003) study the problem in the context of airline transportation and derive formulae for the probable numbers of customers in the system, which is later used to formulate a capacity constraint. Yang (2009) proposes a two-stage stochastic programming model to address the issues of air freight hub location and flight-route planning under stochastic demand. He solve the deterministic equivalent of this problem considering three scenarios. In the same year, Sim et al. (2009) present the stochastic p-hub center problem, in which service level considerations are incorporated by using chance constraints when travel times are normally distributed.

Contreras et al. (2011b) study the uncapacitated multiple allocation hub location problem under uncertain demands and transportation costs. When demand is analyzed as the source of uncertainty, they show that the stochastic problem will be equivalent to its associated deterministic expected value problem where uncertain demand is replaced by its expectation. However, this equivalence does not hold when transportation costs are stochastic. The authors develop an SAA method to solve their stochastic problems and apply it to instances involving up to 50 demand nodes. Alumur et al. (2012) consider two sources of uncertainty, set-up costs for the location of hubs and the demands to be transported between the nodes, within both the single and multiple allocation hub location problems. Mathematical models are proposed considering randomness in demand, while the uncertainty in setup costs is handled by a minmax regret formulation.

More recently, Meraklı and Yaman (2016) model the robust uncapacitated multiple allocation p-hub median problem under demand uncertainty. To represent that uncertainty, they use two polyhedral uncertainty sets: hose and hybrid. They adopt a minmax robustness criterion to model this problem. Meraklı and Yaman (2017) extend this study further by incorporating capacity constraints for hubs into the problem. They present a new mathematical formulation and devise two different Benders decomposition algorithms applied to instances with up to 50 nodes.

Profit maximizing hub location problems with capacity allocation are also related to service network design problems, which determine the selection and scheduling of the services to operate as well as the routes of flow to be shipped. An interested reader may refer to reviews on this area by Crainic (2000) and Wieberneit (2008). In this study, unlike the service network design problem, we additionally incorporate decisions on the locations of hubs.

From a methodological point of view, Benders decomposition (BD) has received increased attention, particularly for solving multiple allocation hub location problems. There are several successful BD implementations in the literature. Camargo et al. (2008) is the first work using a Benders reformulation to solve the uncapacitated multiple allocation hub location problem. They present three variants of the algorithm: the classical BD algorithm, where a single cut is generated at each iteration, a multicut version where Benders cuts are generated for each origin-destination pair, and a variant that terminates when an ϵ-optimal solution is obtained. The proposed algorithms were applied to solve instances with up to 200 nodes. Rodriguez-Martin and Salazar-Gonzalez (2008) consider a capacitated multiple allocation hub location problem in which the arcs connecting the hubs are not assumed to create a complete graph. They provide a formulation and design two exact solution algorithms relying on BD. The first one employs the classical BD approach and the second is a branch-and-cut algorithm based on a two-level nested decomposition scheme. The second outperforms the first approach in terms of computational time.

Contreras et al. (2011a) employ a Benders reformulation for the uncapacitated multiple allocation hub location problem which is enhanced through the use of a multicut reformulation, the generation of Pareto-optimal cuts, the integration of reduction tests, and the execution of a heuristic procedure. Contreras et al. (2012) provide an extension of the BD approach proposed by Contreras et al.

(2011a) to solve capacitated multiple allocation hub location problems. They apply Pareto-optimal Benders cuts as well as reduction tests to improve the convergence of the algorithm, and solve instances with up to 300 nodes.

BD is also used to solve other variants of hub location problems, including flow dependent discounts for inter-hub links using a non-linear cost function (Camargo et al. 2009), hub location-routing problems which incorporate routing decisions between non-hub nodes (Camargo et al. 2013), hub location problems with single allocation under congestion (Camargo et al. 2011, Camargo and Miranda 2012), problems with incomplete inter-hub networks (Camargo et al. 2017, de Sá et al. 2018), problems with distinct topologies of inter-hub networks such as tree-star networks (Martins de Sá et al. 2013) and hub-line networks (Martins de Sá et al. 2015), and hub location problems in liner shipping applications (Gelareh and Nickel 2011).

Motivated by successful implementations of BD to solve hub location problems, in this study, we also develop Benders-based algorithms to solve our problems to optimality. To date, the largest capacitated hub location instances that could be solved optimally contain 300 nodes (Contreras et al. 2012). As we will show through our computational experiments, we are able to solve instances of size 500 nodes, while considering an even more difficult problem setting with generic capacity constraints, multiple demand segments, and a profit maximizing objective function. Moreover, we show that the BD algorithms that we introduce in this paper result in generating stronger cuts than those in the previous literature.

3. Mathematical Formulations

This section first introduces the notation and then presents mathematical formulations for the deterministic and stochastic versions of the problem.

3.1. Notation

Let $G=(N,A)$ be a complete digraph, where N is the set of nodes and A is the set of arcs. Adapting the notation used by Contreras et al. (2012), we define a hub arc as an ordered set $a \in A$ and a loop a as $\{a_1,a_2\}$ if $|a|=2$, and as $\{a_1\}$ if $|a|=1$. Let $H \subseteq N$ be the set of potential hub locations, and K represent the set of commodities whose origin and destination points belong to N. Demand of commodities are segmented into M classes. For each commodity $k \in K$ of class $m \in M$, w_k^m is defined as the amount of commodity k of class m to be routed from the origin $o(k) \in N$ to the destination $d(k) \in N$. Satisfying a unit commodity $k \in K$ of class $m \in M$ produces a per unit revenue of r_k^m. Let f_i and Γ_i denote the installation cost and the available capacity of a hub located at node $i \in H$, respectively. The transportation cost from node $i \in N$ to node $j \in N$ is defined as $c_{ij}=\gamma d_{ij}$, where d_{ij} denotes the distance from node i to node j, and γ is the resource cost per unit distance. Distances are assumed to satisfy the triangle inequality, Each path of an origin-destination pair contains at least one and at most two hubs. Thus, paths are of the form $(o(k), i, j, d(k))$, where $(i, j) \in H \times H$ represents the ordered pair of hubs to which $o(k)$ and $d(k)$ are allocated, respectively. The transportation cost of routing one unit of commodity k along path $(o(k), i, j, d(k))$ can be calculated by $C_{ijk} = \chi c_{o(k)i} + \alpha c_{ij} + \delta c_{jd(k)}$, where χ, α, δ represent the collection, transfer, and distribution costs, respectively. The economies of scale between hubs is reflected by assuming $\alpha < \chi$ and $\alpha < \delta$.

A few characteristics of the multiple allocation hub location problem can be employed in modeling profit maximizing capacitated hub location problems as detailed below:

Property 1 *In any optimal solution, a commodity can be routed via a path containing two distinct hubs only if it is not cheaper to do so using one of the hubs (Boland et al. 2004).*

Property 2 *In any optimal solution, every commodity $k \in K$ uses at most one of the paths $o(k), i, j, d(k)$ and $o(k), j, i, d(k)$; the one with the lower transportation cost (Contreras et al. 2011a).*

Consequently, each commodity defines its own set of potential hub arcs. Henceforth, we replace C_{ijk} with $\hat{C}_{ijk} = \min\{C_{ijk}, C_{jik}\}$, and reduce the set of candidate hub arcs for commodity $k \in K$ to A_k as defined in (1), which can be used to reduce the size of the mathematical formulations.

$$A_k = \{(i,j) \in A : i \leq j, \hat{C}_{ijk} \leq \min\{C_{iik}, C_{jjk}\}\} \tag{1}$$

3.2. Deterministic Model

We first consider the deterministic setting assuming that perfect information on demands is available. Let y_i equal to 1 if a hub is established at node $i \in H$, and 0 otherwise. Moreover, let x^m_{ak} determine the fraction of commodity $k \in K$ of class $m \in M$ that is satisfied through a path with hub arc $a \in A_k$. The *profit maximizing capacitated hub location problem* is then modeled as:

$$\begin{aligned}
\text{Maximize} \quad & \sum_{m \in M} \sum_{k \in K} \sum_{a \in A_k} (r^m_k - \hat{C}_{ak}) w^m_k x^m_{ak} - \sum_{i \in H} f_i y_i && && (2)\\
\text{s.t.} \quad & \sum_{a \in A_k} x^m_{ak} \leq 1 && k \in K, m \in M && (3)\\
& \sum_{a \in A_k : i \in a} x^m_{ak} \leq y_i && i \in H, k \in K, m \in M && (4)\\
& \sum_{m \in M} \sum_{k \in K} \sum_{a \in A_k : i \in a} w^m_k x^m_{ak} \leq \Gamma_i y_i && i \in H && (5)\\
& x^m_{ak} \geq 0 && k \in K, m \in M, a \in A_k && (6)\\
& y_i \in \{0, 1\} && i \in H. && (7)
\end{aligned}$$

The objective function (2) represents net profit, which is calculated by subtracting total cost from total revenue. If demand of a commodity from a given class is to be satisfied, then by constraints (3) flow must be routed via hubs. Each demand class from each commodity can be partially satisfied through different paths. Constraints (4) ensure that demand of commodities can be satisfied only through open hubs. Constraints (5) restrict capacity on the total incoming flow at a hub via both hub and non-hub nodes. Finally, constraints (6) and (7) define the non-negative and binary variables.

Remark 1. For integer values of y, constraints (3) and (5) imply constraints (4). Hence, constraints (4) act as valid inequalities for the mathematical model (2)-(7).

Since our solution method is based on Benders decomposition, and it is known that Benders decomposition performs better with stronger formulations (Magnanti and Wong 1981), we choose to keep these constraints in our mathematical model.

Note that when revenue from satisfying the commodity $k \in K$ of class $m \in M$ is strictly smaller than the unit transportation cost of routing commodity k along a path containing a hub arc $a \in A_k$, no profit can be obtained from satisfying the demand for commodity $k \in K$ of class $m \in M$ through that path. Accordingly, the optimal value for the corresponding variable x_{ak}^m can then be set to zero as noted in Property 3 below:

Property 3 *For every $k \in K$, $m \in M$ and $a \in A_k$, if $r_k^m < \hat{C}_{ak}$, then $x_{ak}^m = 0$ in any optimal solution to* (2)-(7).

3.3. Stochastic Model

We now model the problem with uncertain demand assuming that the uncertainty associated with demands is described by a known probability distribution. As noted before, demand is segmented into different classes and we consider a stochastic demand for each class. To model this problem, let $w_k^m(\xi)$ denote the random variables representing the future demand for commodity $k \in K$ of class $m \in M$ to be shipped from origin $o(k) \in N$ to destination $d(k) \in N$, and assume that $w_k^m(\xi)$ are random variables associated with a known probability distribution. The demands of different commodities are considered as independent random variables, whereas different demand classes of each commodity are assumed to be dependent and thus correlated. If E_ξ denotes the expectation with respect to ξ, and Ξ the support of ξ, then the *profit maximizing capacitated hub location problem with stochastic demand* can be modeled as:

$$\text{Maximize} \quad E_\xi\left[\sum_{m \in M}\sum_{k \in K}\sum_{a \in A_k}(r_k^m - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi)\right] - \sum_{i \in H} f_i y_i \tag{8}$$

$$\text{s.t.} \quad \sum_{a \in A_k} x_{ak}^m(\xi) \le 1 \qquad k \in K, m \in M, \xi \in \Xi \tag{9}$$

$$\sum_{a \in A_k : i \in a} x_{ak}^m(\xi) \le y_i \qquad i \in H, k \in K, m \in M, \xi \in \Xi \tag{10}$$

$$\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k:i\in a} w_k^m(\xi)x_{ak}^m(\xi) \leq \Gamma_i y_i \qquad i\in H, \xi\in\Xi \tag{11}$$

$$x_{ak}^m(\xi) \geq 0 \qquad k\in K, m\in M, a\in A_k, \xi\in\Xi \tag{12}$$

$$y_i \in \{0,1\} \qquad i\in H. \tag{13}$$

The above model forms a two-stage stochastic program. The first stage problem corresponds to strategic hub location decisions. These long-term decisions will not be influenced by demand variations, accordingly, the variables y_i become known in this stage. However, the allocation decisions and the optimal routes of flows through the network, as well as the decision on how much of total capacity should be allocated to demand from different classes, do vary in response to the change of demand and, thus, are influenced by the stochastic demand. These tactical decisions are determined in the second stage depending on the particular realization of the random vector $\xi\in\Xi$. Accordingly, the variables x_{ak}^m become known in the second stage.

The objective function (8) contains a deterministic term which calculates the installation cost of the hubs, and the expectation of the second stage objective which calculates the expected value of revenue and transportation cost.

4. Benders Decomposition

We introduce a Benders decomposition methodology for the solution of our models. Benders decomposition is well suited for hub location problems especially with multiple allocation structure as the problem can be decomposed into linear subproblems by fixing the integer variables for the location of hubs. In this section, we first present the Benders reformulation of the deterministic model and the Benders decomposition algorithm; we then detail our solution strategies for the subproblems.

4.1. Benders Reformulation and Algorithm

Given the mixed integer formulation (2)-(7), in the Benders reformulation of the problem, the hub location decisions are handled in the master problem and the rest is left to the subproblem. By fixing the values of the integer variables $y_i = y_i^e$, we obtain the following linear *primal subproblem* (PS):

$$\text{(PS)} \quad \text{Maximize} \quad \sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k} (r_k^m - \hat{C}_{ak})w_k^m x_{ak}^m \tag{14}$$

$$\text{s.t.} \quad \sum_{a\in A_k} x_{ak}^m \leq 1 \qquad k\in K, m\in M \tag{15}$$

$$\sum_{a\in A_k:i\in a} x_{ak}^m \leq y_i^e \qquad i\in H, k\in K, m\in M \tag{16}$$

$$\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k:i\in a} w_k^m x_{ak}^m \leq \Gamma_i y_i^e \qquad i\in H \tag{17}$$

$$x_{ak}^m \geq 0 \qquad k\in K, m\in M, a\in A_k. \tag{18}$$

We derive the dual of PS by associating the dual variables α_k^m, u_{ik}^m, and b_i to the constraints (15), (16), and (17), respectively. We then have the following *dual subproblem* (DS):

$$\text{(DS)} \quad \text{Minimize} \quad \sum_{k\in K}\sum_{m\in M}\alpha_k^m + \sum_{i\in H} y_i^e(\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m) \tag{19}$$

$$\text{s.t.} \quad \alpha_k^m + u_{ik}^m + u_{jk}^m + w_k^m(b_i+b_j) \geq (r_k^m - \hat{C}_{ijk})w_k^m \quad k\in K, m\in M, (i,j)\in A_k : i\neq j \tag{20}$$

$$\alpha_k^m + u_{ik}^m + w_k^m b_i \geq (r_k^m - \hat{C}_{iik})w_k^m \quad k\in K, m\in M, i\in H \tag{21}$$

$$\alpha_k^m, u_{ik}^m, b_i \geq 0 \quad k\in K, m\in M, i\in H \tag{22}$$

Let P denote the polyhedron defined by (20)-(22), and let $\hat{P}$ be the set of extreme points of P. The primal and dual subproblems are always feasible and bounded, hence an optimal solution of DS is one of the extreme points of P. Note that $\hat{P}$ does not depend on y_i^e; hence, for any arbitrary y, DS can be restated as

$$\min_{(\alpha,u,b)\in\hat{P}} \quad \sum_{k\in K}\sum_{m\in M}\alpha_k^m + \sum_{i\in H} y_i(\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m). \tag{23}$$

Let η denote the overall revenue obtained by satisfying the demand; the Benders *master problem* (MP) can then be formulated as:

$$\text{(MP)} \quad \text{Maximize} \quad \eta - \sum_{i\in H} f_i y_i \tag{24}$$

$$\text{s.t.} \quad \eta \leq \sum_{k\in K}\sum_{m\in M}\alpha_k^m + \sum_{i\in H} y_i(\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m) \quad (\alpha,u,b)\in\hat{P} \tag{25}$$

$$y_i \in \{0,1\} \quad i\in H. \tag{26}$$

Note that MP contains an exponential number of constraints. We can work around this difficulty by employing a cutting-plane method. Starting with an empty set $\hat{P}$, we iteratively solve a relaxed master problem with a small subset of $\hat{P}$ and keep adding new extreme points to $\hat{P}$ by solving dual subproblems, until the optimal solution to MP is found.

An overview of the basic BD algorithm is given in Algorithm 1. In this algorithm, UB, LB, e, z_{MP}^e, and z_{DS}^e stand for the current upper and lower bounds, the iteration counter, and the optimal solutions obtained from the master problem and dual subproblem at iteration e, respectively.

The computational efficiency of the Benders decomposition algorithm generally depends on the number of iterations required to obtain an optimal solution and the computational effort needed to solve MP as well as DS at each iteration. In the following sections, we first describe how the size of the problem can be reduced via variable fixing and then explain how the subproblem can be solved efficiently.

Algorithm 1 Benders decomposition

$UB \leftarrow +\infty, LB \leftarrow -\infty, e \leftarrow 0, \hat{P} \leftarrow \emptyset$
while $(LB < UB)$ **do**
  SOLVE MP and obtain y^e and z^e_{MP}
  $UB \leftarrow z^e_{MP}$
  SOLVE DS and obtain $(\alpha, u, b)^e$ and z^e_{DS}
  $LB \leftarrow \max\{LB, z^e_{DS} - \sum_{i \in H} f_i y^e_i\}$
  ADD $(\alpha, u, b)^e$ to $\hat{P}$
  $e \leftarrow e + 1$
end while

4.2. Variable Fixing

Variable fixing can improve the efficiency of the Benders decomposition algorithm by reducing the computational time of solving the master problems and subproblems due to solution space reduction. The dominance properties presented in Section 3 reduce the size of the model significantly via preprocessing. We can further reduce the size by eliminating hubs that cannot be open in an optimal solution. Contreras et al. (2011a) propose two reduction tests that eliminate such hubs by using the information obtained during the inner iterations of the Benders algorithm. In this paper, we adapt and improve on these tests for eliminating hubs from H and the associated variables from the model.

The first reduction test is based on the primal information obtained by solving the LP-relaxation of MP. Let MP^e_{LP} denote the LP-relaxation of MP at iteration e, z^e_{LP} its optimal value, and rc_i the reduced cost associated with variable y_i for $i \in H$. Let LB be a known lower bound on the optimal value of MP. Since $z^e_{LP} + rc_i$ provides a lower bound on the optimal value of MP, any hub $i \in H$ for which $z^e_{LP} + rc_i < LB$ cannot be open in any optimal solution. Hence, such hubs and their associated variables in the master and subproblem can be discarded in subsequent iterations.

The second reduction test relies on eliminating a set of hubs $Q \subset H$ that are proved to be closed in an optimal solution. Let $\text{MP}^e(Q)$ denote the MP at iteration e with the additional constraint $\sum_{i \in Q} y_i \geq 1$, and $z^e_{MP}(Q)$ its optimal value. Let LB be a lower bound on the optimal value of MP. Note that $\text{MP}^e(Q)$ provides a lower bound on the optimal value of MP^e. Hence, if $z^e_{MP}(Q) < LB$, then none of these hubs can be open in any optimal solution. Therefore, the hubs contained in Q and their associated variables can be discarded in subsequent iterations.

The performance of this reduction test highly depends on the choice of Q. The hubs contained in Q should ideally have the least chance of being open in an optimal solution. Note that it is vital to eliminate potentially non-optimal hubs in the earlier iterations of the Benders algorithm. To increase the chance of a successful test, we make the following two modifications on the settings of the second reduction test of Contreras et al. (2011a).

The first modification addresses the choice of Q. Rather than setting $Q = H$ at the beginning of the Benders algorithm, we set Q to a certain proportion p_Q (e.g., 75%) of the hubs that are less likely to be open in an optimal solution. A simple greedy way of identifying such hubs is to sort the hubs in a nondecreasing order of their ratio of fixed cost (f) to capacity (Γ). At iteration e, we discard from Q the set of hubs opened in the optimal solution of MP as well as the nodes that satisfy the first reduction test. We then perform the second reduction test and if it fails, we further remove from Q the open hubs in the optimal solution of $\mathrm{MP}^e(Q)$. As the Benders algorithm proceeds, Q may become empty (either because of successful tests or discarding open hubs), and we must thus reset Q. This time, rather than greedily sorting the hubs in nondecreasing order of $f/_\Gamma$, we sort them in nonincreasing order of the number of times the hubs have been opened in previous iterations of the Benders algorithm, and select the first p_Q of the sorted hubs.

The second modification addresses the weakness of the bound obtained by $z^e_{MP}(Q)$. Rather than running a single test at each iteration, we split Q into a certain number n_Q of (not necessarily mutually exclusive) smaller subsets. We then run a reduction test for each subset and remove all the nodes in the subset that results in a successful reduction test. Although this modification requires solving multiple integer programs at each iteration of the Benders algorithm, the time required for solving these programs is negligible and compensated by the improved effectiveness of the resulting tests when proper values of p_Q and n_Q are selected. Our computational experiments show that both of these modifications considerably increase the performance of the second reduction test.

4.3. A Two-Phase Method for Solving the Subproblem

Solving the dual subproblem is the most challenging part of the BD algorithm. Due to different definition of capacity usage of hubs in our problems, unlike Contreras et al. (2012), we can no longer cast the subproblem (19)-(22) as a transportation problem[1]. Moreover, the algorithm may suffer from slow convergence due to weakness of the generated cuts.

Pareto-optimal (PO) cuts (Magnanti and Wong 1981) have been extensively used in the literature for generating strong optimality cuts. This method relies on solving two subproblems. In the first problem, the optimal value of the subproblem is calculated. In the second problem (known as the PO subproblem), using a *core point*, a PO cut is generated. By fixing b at specific values, Contreras et al. (2012) generate good cuts by approximating the PO subproblem (Contreras et al. 2011a). Because of the structural differences of our problem, as we will demonstrate via computational experiments, the method proposed by Contreras et al. (2012) does not generate good enough cuts for our problem, or it comes with a high computational burden. Hence, we seek a method for generating good optimality cuts with less computational effort.

In this paper, we propose a method for solving the subproblem in two sequential phases based on the set of values of the binary variables at which the subproblem is evaluated, and reduce each phase into simpler problems. Our approach for breaking the subproblem into two phases is analogous to the idea of approximating the PO cuts: In Phase I, we obtain the optimal value of the subproblem, whereas in Phase II, we strengthen the cut while preserving the optimality and feasibility of the solution.

At iteration e of the BD algorithm, we obtain an optimal solution y^e from MP. Let $H_1^e = \{i : y_i^e = 1\}$ be the set of open hubs and $H_0^e = \{i : y_i^e = 0\}$ be the set of closed hubs. Moreover, let A_{ke}^1 denote the set of distinct potential open hubs of commodity k at iteration e (i.e., $A_{ke}^1 = \{(i,j) \in A_k \cap H_1^e \times H_1^e : i \neq j\}$.) Note that any feasible value of b_i and u_{ik}^m would be optimal when $i \in H_0^e$. Hence, we can solve the subproblem in two phases. In *Phase I*, we remove the variables b_i and u_{ik}^m and their corresponding constraints from DS associated with $i \in H_0^e$, and compute the values of the remaining variables by solving the following *Phase I* subproblem.

$$
\begin{aligned}
\text{(DS-I)} \quad \text{Minimize} \quad & \sum_{k\in K}\sum_{m\in M} \alpha_k^m + \sum_{i\in H_1^e} (\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m) && (27)\\
\text{s.t.} \quad & \alpha_k^m + u_{ik}^m + u_{jk}^m + w_k^m(b_i + b_j) \geq (r_k^m - \hat{C}_{ijk}) w_k^m \quad && k \in K, m \in M, (i,j) \in A_{ke}^1 \quad (28)\\
& \alpha_k^m + u_{ik}^m + w_k^m b_i \geq (r_k^m - \hat{C}_{iik}) w_k^m && k \in K, m \in M, i \in H_1^e \quad (29)\\
& \alpha_k^m, u_{ik}^m, b_i \geq 0 && k \in K, m \in M, i \in H_1^e \quad (30)
\end{aligned}
$$

Note that α_k^m variables are independent from i, accordingly, solving DS-I results in obtaining the optimal values of all α_k^m variables. Hence, in *Phase II*, we find feasible values of b_i and u_{ik}^m for $i \in H_0^e, k \in K, m \in M$ with respect to constraints (20)-(22), aiming to generate optimality cuts as strong as possible.

Let $A_{ke}^0 = \{(i,j) \in A_k \cap H_0^e \times H_0^e : i \neq j\}$ denote the set of distinct potential closed hub arcs of commodity $k \in K$ at iteration e, and let $H_1^{ei} = \{j \in H_1^e : (i,j) \in A_k \textit{ or } (j,i) \in A_k\}$ be the set of open hubs that together with the closed hub $i \in H_0^e$ form a potential hub arc for commodity $k \in K$. Updating DS by fixing the value of the computed variables and removing the already satisfied constraints, we get the following *Phase II* subproblem:

$$
\begin{aligned}
\text{(DS-II)} \quad \text{Minimize} \quad & \sum_{i\in H_0^e} (\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m) && (31)\\
\text{s.t.} \quad & u_{ik}^m + u_{jk}^m + w_k^m(b_i + b_j) \geq \rho_{ij}^{km} \quad && k \in K, m \in M, (i,j) \in A_{ke}^0 \quad (32)\\
& u_{ik}^m + w_k^m b_i \geq \rho_{ii}^{km} && k \in K, m \in M, i \in H_0^e \quad (33)\\
& u_{ik}^m, b_i \geq 0 && k \in K, m \in M, i \in H_0^e \quad (34)
\end{aligned}
$$

where $\rho_{ij}^{km} = (r_k^m - \hat{C}_{ijk})w_k^m - \alpha_k^m$ and

$$\rho_{ii}^{km} = \max\{\max_{j\in H_1^{ei}}\{(r_k^m - \hat{C}_{ijk})w_k^m - u_{jk}^m - w_k^m b_j\}, (r_k^m - \hat{C}_{iik})w_k^m\} - \alpha_k^m. \tag{35}$$

We propose two algorithms tailored to the special structure of the Phase II subproblem. First, we show that DS-II can be viewed as a set of LP-relaxations of maximum weighted matching problems for a given value of b_i for $i \in H_0^e$. In the second approach, we show that Phase II can be solved as a sequence of LP-relaxations of knapsack problems. In the following sections, we present some theoretical insights and discuss how to solve the two phases efficiently.

4.3.1. Solving the Phase I subproblem. We simplify DS-I (27)-(30) by showing that $u_{ik}^m = 0$, for $k \in K$, $m \in M$ and $i \in H_1^e$, in an optimal solution of DS-I.

PROPOSITION 1. *There exists an optimal solution to DS-I, in which $u_{ik}^m = 0$ for $k \in K$, $m \in M$ and $i \in H_1^e$.*

Proof. For any arbitrary value of b (including the optimal solution) and for each $k \in K$ and $m \in M$, DS-I (27)-(30) can be decomposed into smaller problems of the following form:

$$\text{(DS-I}(km)) \quad \text{Minimize} \quad \alpha_k^m + \sum_{i\in H_1^e} u_{ik}^m \tag{36}$$

$$\text{s.t.} \quad \alpha_k^m + u_{ik}^m + u_{jk}^m \geq (r_k^m - \hat{C}_{ijk} - b_i - b_j)w_k^m \quad (i,j)\in A_{ke}^1 \tag{37}$$

$$\alpha_k^m + u_{ik}^m \geq (r_k^m - \hat{C}_{iik} - b_i)w_k^m \quad i \in H_1^e \tag{38}$$

$$\alpha_k^m, u_{ik}^m \geq 0 \quad i \in H_1^e. \tag{39}$$

Define $\beta_{ij} = (r_k^m - \hat{C}_{ijk} - b_i - b_j)w_k^m$ for $(i,j) \in A_{ke}^1$, and $\beta_{ii} = (r_k^m - \hat{C}_{iik} - b_i)w_k^m$ for $i \in H_1^e$. Let μ_{ij} and μ_{ii} be the dual variables associated with (37) and (38), respectively, for $(i,j) \in A_{ke}^1$ and $i \in H_1^e$. Dual of DS-I(km) can be formulated as:

$$\text{(Dual DS-I}(km)) \quad \text{Maximize} \quad \sum_{a\in \hat{A}_{ke}^1} \mu_a \beta_a \tag{40}$$

$$\text{s.t.} \quad \sum_{a\in \hat{A}_{ke}^1} \mu_a \leq 1 \tag{41}$$

$$\sum_{a\in \hat{A}_{ke}^1 : i\in a} \mu_a \leq 1 \quad i \in H_1^e \tag{42}$$

$$\mu_a \geq 0 \quad a \in \hat{A}_{ke}^1 \tag{43}$$

where $\hat{A}_{ke}^1 = A_{ke}^1 \cup \{(i,i) : i \in H_1^e\}$. Constraints (42) are clearly dominated by (41) for every $i \in H_1^e$, thus can be removed entirely from the dual problem. Accordingly, it is optimal to set the dual variables associated with (42) to zero, i.e. $u_{ik}^m = 0$ for $i \in H_1^e$. $\square$

The outcome of Proposition 1 is that we can compute the optimal solution of DS-I(km) and its dual, by using the corollaries stated below:

COROLLARY 1. *In the optimal solution of DS-I(km),* $\alpha_k^m = \max_{a \in \hat{A}_{ke}^1} \{\beta_a, 0\}$.

COROLLARY 2. *Let* $a^* = \arg\max_{a \in \hat{A}_{ke}^1} \{\beta_a\}$, *where ties are broken arbitrarily. In the optimal solution of Dual DS-I*(km), $\mu_a = 0$ *for* $a \in \hat{A}_{ke}^1 \setminus \{a^*\}$. *If* $\beta_{a^*} > 0$, *then* $\mu_{a^*} = 1$, *otherwise* $\mu_{a^*} = 0$.

Consequently, DS-I can be solved using a nested cutting-plane algorithm, where b is the master problem variable and α is the subproblem variable, and the dual subproblems are solved using Corollary 2.

4.3.2. Solving the Phase II subproblem as maximum weighted matching problems. For a given vector $(b_i)_{i \in H_0^e}$, DS-II can be restated for each $k \in K$ and $m \in M$ as:

$$\begin{aligned}
\text{(DS-II)}(km) \quad & \text{Minimize} \quad \sum_{i \in H_0^e} u_{ik}^m && && (44)\\
& \text{s.t.} \quad u_{ik}^m + u_{jk}^m \geq \rho_{ij}^{km} - w_k^m (b_i + b_j) && (i,j) \in A_{ke}^0 && (45)\\
& \quad u_{ik}^m \geq \rho_{ii}^{km} - w_k^m b_i && i \in H_0^e && (46)\\
& \quad u_{ik}^m \geq 0 && i \in H_0^e. && (47)
\end{aligned}$$

Observe that (46) together with (47) serve as lower bounds on u_{ik}^m. Define $t_{ik}^m = u_{ik}^m - lb_{ik}^m$, where $lb_{ik}^m = \max\{\rho_{ii}^{km} - w_k^m b_i, 0\}$. DS-II($km$) can be restated as:

$$\begin{aligned}
\text{(MWC)}(km) \quad & \text{Minimize} \quad \sum_{i \in H_0^e} t_{ik}^m && && (48)\\
& \text{s.t.} \quad t_{ik}^m + t_{jk}^m \geq \beta_{ij}^{km} && (i,j) \in A_{ke}^0 && (49)\\
& \quad t_{ik}^m \geq 0 && i \in H_0^e. && (50)
\end{aligned}$$

where $\beta_{ij}^{km} = \rho_{ij}^{km} - w_k^m (b_i + b_j) - lb_{ik}^m - lb_{jk}^m$ for $(i,j) \in A_{ke}^0$. Assume without loss of generality that β_{ij}^{km} values are nonnegative, otherwise their corresponding constraints can be dropped. Problem (48)-(50) is commonly known as the *Minimum Weight Cover (MWC)* problem (see e.g. Galil 1986), which is the dual of the LP-relaxation of the *Maximum Weighted Matching (MWM)* problem in the edge-weighted graph $G_{km} = (H_0^e, A_{ke}^0)$, where edges are weighted according to β_{ij}^{km}.

LP-relaxation of MWM can be formulated as

$$\begin{aligned}
\text{(LP-MWM}(km)) \quad & \text{Maximize} \quad \sum_{a \in A_{ke}^0} \mu_a^{km} \beta_a^{km} && && (51)\\
& \text{s.t.} \quad \sum_{a \in A_{ke}^0 : i \in a} \mu_a^{km} \leq 1 && i \in H_0^e && (52)\\
& \quad \mu_a^{km} \geq 0 && a \in A_{ke}^0 && (53)
\end{aligned}$$

where μ_a^{km} is the dual variable associated with (49) and represents the extent to which edge $a \in A_{ke}^0$ is picked in the MWM. Unlike the LP-relaxation of the MWM problem in general graphs, LP-relaxation of the MWM problem in bipartite graphs (MWMB) guarantees integrality of the solutions. Hence, methods proposed for solving the MWMB can also be used for solving its LP-relaxation. In EC.1, we show that the MWM problem (51)-(53) can be transformed into the MWMB problem in an equivalent sparse bipartite graph.

Several algorithms have been proposed for solving the MWMB. (For a review on recent methods, see, e.g., Burkard et al. 2009.) As a byproduct of solving the MWM problem (51)-(53), we eventually need to find the optimal value of (44)-(47). Hence, we adapt a primal-dual algorithm due to Galil (1986), which can be solved in $O(an\log_{\lceil a/n+1\rceil} n)$, where $n=|H_0^e|$ and $a=|A_{km}^0|$ are the number of nodes and edges, respectively. The details of the proposed algorithm are given in EC.1.

The strength of the cut generated using this method depends highly on the value at which vector $(b_i)_{i\in H_0^e}$ is fixed. One could simply set b_i to zero for $i \in H_0^e$, or to the average of b_i in previous iterations of the BD algorithm; however, this will likely result in weak cuts. Moreover, note that larger values of b result in fewer positive β_a^{km} values, thus reducing the number of edges (i.e. $|A_{km}^0|$), and consequently the computational time for solving the MWM problems. In EC.2, we explain how proper values of $(b_i)_{i\in H_0^e}$ can be calculated efficiently using a relaxation of DS-II.

4.3.3. Solving the Phase II subproblem as knapsack problems. At each iteration of the BD algorithm, we obtain a solution y with a set of open/closed hubs. The open hubs are potentially the ones with desirable properties; e.g., with high capacity, low installation cost, and/or profitable flows through these hubs. Hence, the most frequently opened hubs in the preceding iterations of BD are more likely to be opened again in the subsequent iterations. To strengthen the cut, we must prioritize minimizing the coefficients of these hubs over the hubs that are less likely to be open. In light of this observation, solving DS-II at iteration e, we sequentially minimize the coefficient of y_i for one closed hub $i \in H_0^e$ at a time, in a particular order of H_0^e, rather than minimizing the summation of coefficients of all closed hubs simultaneously.

Let $O_i^e = \sum\limits_{h:h\le e} y_i^h$ denote the number of times that hub i has been opened prior to or at iteration e of BD. A higher value of O_i^e implies a higher chance for hub i to be open at iteration $e+1$. We first solve DS-II (31)-(34) as if $i = \arg\max\limits_{j\in H_0^e}\{O_j^e\}$ is the only hub in H_0^e:

$$
\begin{aligned}
\text{(DS-II}(i)) \quad \text{Minimize} \quad & \Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m && && (54)\\
\text{s.t.} \quad & u_{ik}^m + w_k^m b_i \ge \rho_{ii}^{km} && k\in K, m\in M && (55)\\
& u_{ik}^m, b_i \ge 0 && k\in K, m\in M. && (56)
\end{aligned}
$$

Note that since i is assumed to be the only hub in H_0^e, constraints (32) do not appear in this model, but will be satisfied by the next closed hubs in the sequence by means of updating the respective ρ_{ii}^{km} values using (35).

Upon solving DS-II(i), we obtain the values of all dual variables associated with hub i (i.e., u_{ik}^m and b_i). Consequently, we fix these values, add i to H_1^e and remove i from H_0^e. We continue this procedure with the next closed hub $i = \arg\max_{j \in H_0^e} \{O_j^e\}$ with ρ_{ii}^{km} values updated according to the new sets H_1^e and H_1^{ei}. We replicate this procedure until values of all dual variables are computed; i.e., until H_0^e becomes empty. An overview of this procedure is presented in Algorithm 2.

Algorithm 2 Solving DS-II via sequential knapsack problems

while ($H_0^e \neq \emptyset$) **do**
  $i \leftarrow \arg\max_{j \in H_0^e} \{O_j^e\}$
  Compute ρ_{ii}^{km} values using (35) for the selected i and each $k \in K$ and $m \in M$ with respect to the updated H_1^e and H_1^{ei}.
  SOLVE DS-II(i) and obtain b_i and u_{ik}^m for $k \in K$ and $m \in M$.
  $H_0^e \leftarrow H_0^e \setminus \{i\}$
  $H_1^e \leftarrow H_1^e \cup \{i\}$
end while

Observe that DS-II(i) is the dual of the LP-relaxation of a knapsack problem (KP) with knapsack capacity Γ_i, and items $(k,m) \in K \times M$ with weight w_k^m and profit ρ_{ii}^{km}, as formulated in (57)-(59):

$$\text{(LP-KP}(i)) \quad \text{Maximize} \quad \sum_{(k,m) \in K \times M} \rho_{ii}^{km} \mu_k^m \tag{57}$$

$$\text{s.t.} \quad \sum_{(k,m) \in K \times M} w_k^m \mu_k^m \leq \Gamma_i \tag{58}$$

$$0 \leq \mu_k^m \leq 1 \qquad (k,m) \in K \times M \tag{59}$$

where μ_k^m is the dual variable associated with (55), and represents the extent to which item $(k,m) \in K \times M$ is picked. Note that the items with a non-positive profit (i.e., $\rho_{ii}^{km} \leq 0$) can be discarded from the problem. Dantzig (1957) showed that the optimal solution to this problem can be found by filling the knapsack in a non-increasing order of profit-to-weight ratio ρ_{ii}^{km}/w_k^m of the items, until we reach the first item that does not fit into the knapsack. This item is called the break item $(\bar{k}, \bar{m})$, which is partially picked according to the residual capacity. Using the complementary-slackness conditions, it can be verified that the optimal value of b_i is the profit-to-weight ratio of the break item; i.e.,

$$b_i = \rho_{ii}^{\bar{k}\bar{m}} / w_{\bar{k}}^{\bar{m}}. \tag{60}$$

Moreover, using (55) and (56), the optimal value of u_{ik}^m can be calculated by setting $u_{ik}^m = \max\{\rho_{ii}^{km} - w_k^m b_i, 0\}$. Balas and Zemel (1980) showed that the LP-relaxation of knapsack problem can be solved in $O(n)$, where $n = |K||M|$, by finding the break item as a weighted median, rather than by explicitly sorting the items. This implies that the optimal values of b and u can be found in the same time bound.

5. Solution Scheme for the Stochastic Model

In this section, we present an algorithm to solve the profit maximizing capacitated hub location problem with stochastic demand. The methodology integrates a sampling technique, named as *sample average approximation* (SAA) algorithm (the reader may refer to Shapiro and Homem-de Mello 1998, Mak et al. 1999, Kleywegt et al. 2002) with the BD algorithm detailed in the following sections.

5.1. Sample Average Approximation

SAA is a Monte Carlo simulation based approach to stochastic discrete optimization problems. The main idea of this method is to reduce the size of the problem by generating a random sample and approximating the expected value of the corresponding sample average function. The sample average optimization problem is then solved (using the BD algorithm in our case), and the procedure is repeated. The SAA scheme has previously been applied to stochastic supply chain design as well as hub location problems with a large number of scenarios (see, e.g., Santoso et al. 2005, Schütz et al. 2009, and Contreras et al. 2011b).

The main challenge in solving the stochastic problem (8)-(13) is the evaluation of the expected value of the objective function (Kleywegt et al. 2002). To deal with this problem, we use SAA scheme in which a random sample of realizations of the random vector $n \in \mathcal{N}$ is generated, and the second-stage expectation

$$E_\xi \Big[\sum_{m\in M} \sum_{k\in K} \sum_{a\in A_k} (r_k^m - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) \Big]$$

is approximated by the sample average function

$$\tfrac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \sum_{m\in M} \sum_{k\in K} \sum_{a\in A_k} (r_k^m - \hat{C}_{ak}) w_k^{mn} x_{ak}^{mn}$$

where w_k^{mn} and x_{ak}^{mn} denote the amount of commodity $k \in K$ of class $m \in M$ to be shipped from origin $o(k) \in N$ to destination $d(k) \in N$ under sample $n \in \mathcal{N}$, and the fraction of commodity $k \in K$ of class $m \in M$ that is satisfied through a hub link $a \in A_k$ under scenario $n \in \mathcal{N}$, respectively. Accordingly, the approximated form of the stochastic problem by the SAA algorithm is modeled as:

$$\text{Maximize} \quad \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \sum_{m\in M} \sum_{k\in K} \sum_{a\in A_k} (r_k^m - \hat{C}_{ak}) w_k^{mn} x_{ak}^{mn} - \sum_{i\in H} f_i y_i \tag{61}$$

$$\text{s.t.} \quad \sum_{a \in A_k} x_{ak}^{mn} \leq 1 \qquad k \in K, m \in M, n \in \mathcal{N} \tag{62}$$

$$\sum_{a \in A_k : i \in a} x_{ak}^{mn} \leq y_i \qquad i \in H, k \in K, m \in M, n \in \mathcal{N} \tag{63}$$

$$\sum_{m \in M} \sum_{k \in K} \sum_{a \in A_k : i \in a} w_k^m x_{ak}^{mn} \leq \Gamma_i y_i \qquad i \in H, n \in \mathcal{N} \tag{64}$$

$$x_{ak}^{mn} \geq 0 \qquad k \in K, m \in M, a \in A_k, n \in \mathcal{N} \tag{65}$$

$$y_i \in \{0, 1\} \qquad i \in H. \tag{66}$$

Hereafter, we use the above approximated model (61)-(66) as the mathematical model for the profit maximizing capacitated hub location problem with stochastic demand.

The optimal solution and the optimal value of the SAA problem (61)-(66) converge with probability one to their true counterpart (8)-(13) as the sample size increases (Kleywegt et al. 2002). Assuming that the SAA problem is solved within an optimality gap of $\delta > 0$, and by letting $\varepsilon > \delta$ and $\alpha \in (0, 1)$, then a sample size of

$$|\mathcal{N}| \geq \frac{3\sigma_{max}^2}{(\varepsilon - \delta)^2} \log(\frac{|Y|}{\alpha}) \tag{67}$$

guarantees that the SAA solution is an ε-optimal solution to the true problem with a probability of at least $1 - \alpha$, where σ_{max}^2 is the maximal variance of certain function differences (the readers may refer to Kleywegt et al. 2002 for details).

To choose $\mathcal{N}$ in practice, one should take into account the trade-off between the quality of the solution obtained from the SAA problem and the computational time required to solve it. Hence, it can be more efficient to solve the SAA problem (61)-(66) with independent samples repeatedly rather than increasing the sample size $\mathcal{N}$. We now describe our procedure:

1. Generate $\mathcal{M}$ independent samples each of size $\mathcal{N}$; i.e., $\xi_j^1, \ldots, \xi_j^{|\mathcal{N}|}$, for $j \in \mathcal{M}$ and solve the corresponding SAA problem (61)-(66) for each sample $\mathcal{N}_j$ to optimality employing the BD algorithm detailed in Section 4. Let $\mathcal{V}^{\mathcal{N}_j}$ and $\hat{y}^{\mathcal{N}_j}$, $j \in \mathcal{M}$, be the corresponding optimal objective value and an optimal solution, respectively.

2. Calculate the average of all optimal solution values from the SAA problems and their variance:

$$\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}} = \tfrac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \mathcal{V}^{\mathcal{N}_j}$$

$$\sigma_{\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}}^2 = \tfrac{1}{(|\mathcal{M}|-1)|\mathcal{M}|} \sum_{j \in \mathcal{M}} (\mathcal{V}^{\mathcal{N}_j} - \bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}})^2$$

The expected value of $\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}$ provides an upper statistical bound for the optimal value of the original problem, and $\sigma_{\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}}^2$ is an estimate of the variance of this estimator.

3. Pick a feasible solution $\hat{y} \in Y$ for problem (8)-(13), for example, use one of the previously computed solutions $\hat{y}^{\mathcal{N}_j}$. Estimate the objective function value of the original problem by using this solution as follows:

$$\mathcal{V}_{\mathcal{N}'}(\hat{y}) = \frac{1}{|\mathcal{N}'|}\left[\sum_{k\in K}\sum_{a\in A_k}\sum_{m\in M}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn}\right] - \sum_{i\in H} f_i\hat{y}_i$$

where $\xi_j^1,\ldots,\xi^{|\mathcal{N}'|}$ is a sample of size $\mathcal{N}'$ generated independently of the samples used in the SAA problems. Note that since the first-stage variables are fixed, one can take much larger number of scenarios for $|\mathcal{N}'|$ than the sample size $|\mathcal{N}|$ used to solve the SAA problems. The estimator $\mathcal{V}_{\mathcal{N}'}(\hat{y})$ serves as a lower bound on the optimal objective function value. We can estimate the variance of $\mathcal{V}_{\mathcal{N}'}(\hat{y})$ as follows:

$$\sigma^2_{\mathcal{N}'}(\hat{y}) = \frac{1}{(|\mathcal{N}'|-1)|\mathcal{N}'|}\sum_{n\in\mathcal{N}'}\left([\sum_{k\in K}\sum_{a\in A_k}\sum_{m\in M}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn}] - \sum_{i\in H} f_i\hat{y}_i - \mathcal{V}_{\mathcal{N}'}(\hat{y})\right)^2$$

4. Calculate the estimators for the optimality gap and its variance. Employing the estimators computed in steps 2 and 3, we get:

$$gap_{\mathcal{N},\mathcal{M},\mathcal{N}'}(\hat{y}) = \bar{\mathcal{V}}^{\mathcal{N}}_{\mathcal{M}} - \mathcal{V}_{\mathcal{N}'}(\hat{y})$$

$$\sigma^2_{gap} = \sigma^2_{\bar{\mathcal{V}}^{\mathcal{N}}_{\mathcal{M}}} + \sigma^2_{\mathcal{N}'}(\hat{y})$$

We can then use these estimators to construct a confidence interval for the optimality gap.

5.2. Benders Decomposition for the SAA Problem

In this section, we present a BD algorithm coupled with SAA to solve the profit maximizing capacitated hub location problem with stochastic demand. By approximating the second-stage expectation via the sample average function as described in the previous section, the stochastic problem can be formulated as (61)-(66).

This problem can be solved using an L-shaped method (Van Slyke and Wets 1969). We apply the same procedure described in Section 4.1 and assume that the hub location decisions are handled in the master problem, while the rest is left to the subproblem. For a given sample $\mathcal{N}$ and fixed value of the integer variables $y_i = y_i^e$, the *primal subproblem* (PS($\mathcal{N}$)) reads as

$$(\text{PS}(\mathcal{N}))\quad \text{Maximize}\quad \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn} \tag{68}$$

$$\text{s.t.}\quad \sum_{a\in A_k} x_{ak}^{mn} \le 1 \qquad k\in K, m\in M, n\in\mathcal{N} \tag{69}$$

$$\sum_{a\in A_k: i\in a} x_{ak}^{mn} \le y_i^e \qquad i\in H, k\in K, m\in M, n\in\mathcal{N} \tag{70}$$

$$\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k: i\in a} w_k^{mn}x_{ak}^{mn} \le \Gamma_i y_i^e \qquad i\in H, n\in\mathcal{N} \tag{71}$$

$$x_{ak}^{mn} \ge 0 \qquad k\in K, m\in M, a\in A_k, n\in\mathcal{N}. \tag{72}$$

Observe that PS($\mathcal{N}$) can be decomposed into $|\mathcal{N}|$ independent subproblems of the form PS (14)-(18) for each $n\in\mathcal{N}$. Consequently, the dual subproblem associated with each n can be formulated as DS (19)-(22) and solved using the techniques proposed in Section 4.3. In the following, we denote the DS under scenario $n \in \mathcal{N}$ by DS($\mathcal{N}, n$), in which w_k^m is replaced with w_k^{mn}, for each $k \in K$ and $m \in M$.

Let $\hat{P}_{\mathcal{N}}^n$ for $n \in \mathcal{N}$ denote the set of extreme points of the polyhedron defined by feasible region of DS($\mathcal{N}, n$). The master problem can then be stated as

$$(\text{MP}(\mathcal{N})) \quad \text{Maximize} \quad \frac{1}{|\mathcal{N}|}\sum_{n \in \mathcal{N}} \eta^n - \sum_{i \in H} f_i y_i \tag{73}$$

$$\text{s.t.} \quad \eta^n \leq \sum_{k \in K}\sum_{m \in M} \alpha_k^{mn} + \sum_{i \in H} y_i(\Gamma_i b_i^n + \sum_{k \in K}\sum_{m \in M} u_{ik}^{mn}) \qquad n \in \mathcal{N}, (\alpha^n, u^n, b^n) \in \hat{P}_{\mathcal{N}}^n \tag{74}$$

$$y_i \in \{0,1\} \qquad i \in H \tag{75}$$

As described in Section 4.1, this problem also contains an exponential number of constraints, which can be tackled by employing a cutting plane method, where a sequence of relaxed master problems and dual subproblems are solved, until the optimal solution is found.

5.3. Acceleration Techniques for the SAA Problem

The variable fixing techniques presented in Section 4.2 can be applied to each sample of the stochastic model. We further enhance the convergence of our SAA algorithm as detailed below.

Because of the repetitive structure of the SAA algorithm, we must solve $|\mathcal{M}|$ replications of problem (61)-(66). Consequently, upon solving MP($\hat{\mathcal{N}}$) for a specific sample $\hat{\mathcal{N}}$, we obtain dual solutions $(\hat{\alpha}^{\hat{n}}, \hat{u}^{\hat{n}}, \hat{b}^{\hat{n}}) \in \hat{P}_{\hat{\mathcal{N}}}^{\hat{n}}$ for each $\hat{n} \in \hat{\mathcal{N}}$. Now, assume that we want to solve MP($\mathcal{N}$) for a different sample $\mathcal{N}$. Solving MP($\mathcal{N}$) with initially empty $\hat{P}_{\mathcal{N}}^n$ sets would disregard the fact that the optimal solution of MP($\hat{\mathcal{N}}$) is potentially a near-optimal solution to MP($\mathcal{N}$). We can exploit this property and retrieve feasible solutions $(\alpha^n, u^n, b^n) \in \hat{P}_{\mathcal{N}}^n$ for scenario n of sample $\mathcal{N}$ from the solutions contained in $\hat{P}_{\hat{\mathcal{N}}}^{\hat{n}}$ for $\hat{n} \in \hat{\mathcal{N}}$.

Given a feasible solution for DS($\hat{\mathcal{N}}, \hat{n}$), the following proposition provides a feasible solution for DS($\mathcal{N}, n$).

Proposition 2. *Let $(\hat{\alpha}^{\hat{n}}, \hat{u}^{\hat{n}}, \hat{b}^{\hat{n}}) \in \hat{P}_{\hat{\mathcal{N}}}^{\hat{n}}$ be a feasible solution for DS($\hat{\mathcal{N}}, \hat{n}$), and $\hat{w}_k^{m\hat{n}}$ be the demand for commodity $k \in K$ of class $m \in M$ under scenario $\hat{n} \in \hat{\mathcal{N}}$. (α^n, u^n, b^n) defined by (76)-(78) is feasible for DS($\mathcal{N}, n$):*

$$b_i^n = \hat{b}_i^{\hat{n}} \qquad i \in H \tag{76}$$

$$\alpha_k^{mn} = \frac{w_k^{mn}}{\hat{w}_k^{m\hat{n}}}\hat{\alpha}_k^{m\hat{n}} \qquad k \in K, m \in M \tag{77}$$

$$u_{ik}^{mn} = \frac{w_k^{mn}}{\hat{w}_k^{m\hat{n}}}\hat{u}_{ik}^{m\hat{n}} \qquad k \in K, m \in M, i \in H \tag{78}$$

Proof. From (77) and (78), we obtain $\hat{\alpha}_k^{m\hat{n}} = \frac{\hat{w}_k^{m\hat{n}}}{w_k^{mn}}\alpha_k^{mn}$ and $\hat{u}_{ik}^{m\hat{n}} = \frac{\hat{w}_k^{m\hat{n}}}{w_k^{mn}}u_{ik}^{mn}$, respectively. Feasibility of (α^n, u^n, b^n) for DS($\mathcal{N}, n$) can easily be verified by replacing $\hat{b}_i^{\hat{n}}$, $\hat{\alpha}_k^{m\hat{n}}$, and $\hat{u}_{ik}^{m\hat{n}}$ respectively with b_i^n, $\frac{\hat{w}_k^{m\hat{n}}}{w_k^{mn}}\alpha_k^{mn}$, and $\frac{\hat{w}_k^{m\hat{n}}}{w_k^{mn}}u_{ik}^{mn}$, in constraints (20)-(22). $\square$

Corollary 3. *The solution obtained by* (76)-(78) *provides a valid cut for MP*($\mathcal{N}$).

Note that for a given scenario $n \in \mathcal{N}$, each scenario $\hat{n} \in \hat{\mathcal{N}}$ can provide a valid cut for MP($\mathcal{N}$). To avoid adding too many cuts, we select and add the best potential cut, which belongs to scenario $\hat{n}^* \in \hat{\mathcal{N}}$ with the least demand deviation from the demand under scenario $n \in \mathcal{N}$, i.e.

$$\hat{n}^* = \arg\min_{\hat{n} \in \hat{\mathcal{N}}} \left\{ \sum_{(k,m) \in K \times M} |\hat{w}_k^{m\hat{n}} - w_k^{mn}| \right\}. \tag{79}$$

It should, however, be noted that when generating valid cuts from sample $\hat{\mathcal{N}}$ for sample $\mathcal{N}$, both instances should have the same set of hubs. In other words, if in the process of solving MP($\hat{\mathcal{N}}$), we eliminate a number of hubs via variable fixing, we will not calculate the dual variables associated with the eliminated hubs, hence the incomplete solution obtained by (76)-(78) may not provide a valid cut for MP($\mathcal{N}$). To tackle this problem, we sacrifice the first sample of the SAA algorithm without using any variable fixing to ensure that the resulting cuts can be used for the subsequent samples of SAA. Once these solutions are obtained, we add the respective cuts to the master problem of the subsequent samples and continue with performing variable fixing as introduced in Section 4.2.

6. Computational Experiments

We use the well-known Australia Post (AP) dataset to test our models and algorithms. AP dataset contains postal service data of 200 cities in Australia and it is the most commonly used dataset in hub location literature (Ernst and Krishnamoorthy 1996). The distances (d_{ij}) and the postal flow between pairs of cities (w_k) are provided in OR Library, and a computer code is presented to generate smaller subsets of the data by grouping cities (Beasley 1990). We assume that the demand of commodities are segmented into 3 classes; i.e., $|M| = 3$, where $w_k^1 = 0.2w_k$, $w_k^2 = 0.3w_k$, and $w_k^3 = 0.5w_k$. Motivated from the postal delivery applications, where the price of sending a parcel depends on its size and the distance between the origin-destination, and also considering revenue elasticity of demand, the revenue per unit demand is taken to be dependent on the distance, class, and the amount of commodity to be shipped. Hence, for the revenue from commodity $k \in K$ of class $m \in M$, we generate random values as $r_k^m = \gamma^m \frac{c_k}{w_k^m}$, where $\gamma^1 \sim U[50, 60]$, $\gamma^2 \sim U[40, 50]$, and $\gamma^3 \sim U[30, 40]$. Collection, transfer, and distribution costs per unit are taken as $\chi = 2$, $\alpha = 0.75$, and $\delta = 3$ as defined in the AP dataset (Beasley 1990). We test instances with $|H| \in \{10, 20, 25, 40, 50, 75, 100, 200\}$.

There are two different sets for installation costs and capacities of hubs available on the AP dataset referred to as loose and tight. The opening cost of hubs in the instances with tight (T) installation costs is larger than those with loose (L) installation costs. In contrast, instances with tight (T) capacities have smaller available capacities compared to the instances with loose (L) capacities. Hence, there are four instances for a given node size corresponding to different combinations of installation costs and capacities. We denote each instance as $nf\Gamma$ where n is the instance size, f is the installation costs, and Γ is the capacity.

Computational experiments were carried out on a workstation that contains: Intel Core i7-3930K 2.61GHz CPU, and 39 GB of RAM. The algorithms were coded in *C#* and the time limit was set to 15 hours. The master problems of all versions of the Benders decomposition algorithms as well as the master problems of the inner cutting-plane algorithm of Phase I subproblems were solved using the callable library of CPLEX 12.7. We used $p_Q = 75\%$ and $n_Q = 6$ in all our tests, which are the best values for the corresponding parameters based on the results provided in Online Appendix EC.3.

The organization of this section is as follows: In the first part of the computational experiments, we compare our results obtained from different versions of the Benders algorithm for the deterministic problem to evaluate the effectiveness of the proposed algorithmic features. The second part of the experiments is devoted to the algorithms and the solution methodology proposed for the stochastic model.

6.1. Analysis of Algorithmic Performance

In this section, we first evaluate the performance and effectiveness of the algorithms proposed in Section 4 for the deterministic model. We implemented two different versions of Algorithm 1 referred to as BD1 and BD2, corresponding to different solution strategies of Phase II. In BD1, Phase II is solved as LP-relaxations of maximum weighted matching problem, whereas in BD2, Phase II is solved as LP-relaxations of knapsack problem. For comparison, we also implemented Pareto-optimal cuts (PO), the best known cuts from the literature, to solve our subproblems. We use the two reduction tests as described in Section 4.2 to eliminate candidate hubs within all of the three algorithms: PO, BD1, and BD2.

The detailed results of the comparison between these Benders algorithms using the AP instances are provided in Table 1. The first column represents the name and size of the instance. The next four columns labeled "Total time (sec)" present the computational time of instances (in seconds) obtained from solving the problems to optimality by using CPLEX, PO, BD1, and BD2, respectively. The next three columns labeled "Iterations" provide the required number of iterations for the convergence of the algorithms PO, BD1, and BD2, respectively. The columns labeled "% hubs elim." present the percentage of the total candidate hubs eliminated by algorithms PO, BD1, and BD2, respectively. The last two columns labeled "Optimal solution" indicate the maximum net profit and the locations of hub nodes, respectively, for the optimal solution found for each of the considered instances. Whenever an algorithm is not able to solve an instance within the time

Table 1 Comparison of Benders reformulations and CPLEX with the AP dataset for the deterministic model.

	Total time (sec)				Iterations			% hubs elim.			Optimal solution	
$\|H\|$	CPLEX	PO	BD1	BD2	PO	BD1	BD2	PO	BD1	BD2	Profit	Open hubs
10LL	1.02	1.37	0.13	0.01	7	4	4	50.0	60.0	60.0	20,417	5,6,9,10
10LT	0.89	0.94	0.03	0.01	5	4	4	60.0	80.0	80.0	3,336	5
10TL	1.00	0.81	0.03	0.07	9	6	6	50.0	60.0	60.0	13,488	5,9
10TT	1.16	0.26	0.03	0.03	4	3	3	80.0	90.0	90.0	2,682	5
20LL	9.64	35.64	1.70	2.11	51	11	11	70.0	65.0	80.0	100,443	7,9,10,19
20LT	11.38	45.26	0.97	1.34	60	7	7	65.0	65.0	65.0	57,139	5,10,12,14,19
20TL	10.34	21.52	0.84	0.99	37	9	9	75.0	80.0	85.0	49,559	5,7,10
20TT	6.17	4.51	0.22	0.29	11	3	3	70.0	95.0	95.0	10,135	10
25LL	15.42	68.03	7.91	6.31	48	18	18	76.0	72.0	72.0	125,390	7,14,17,23
25LT	12.82	157.51	3.27	2.35	93	7	7	68.0	72.0	72.0	88,022	6,9,10,12,14,25
25TL	14.21	53.80	2.12	2.09	40	9	9	76.0	72.0	72.0	76,933	6,9,14,23
25TT	12.30	100.76	2.02	1.92	71	10	10	80.0	76.0	76.0	35,121	6,10,14,25
40LL	79.23	261.31	52.57	42.47	48	64	58	70.0	82.5	85.0	76,995	12,22,26,29
40LT	83.18	623.02	39.58	28.93	66	42	38	82.5	82.5	82.5	66,860	12,14,26,29,30,38
40TL	41.25	71.96	9.53	5.26	13	14	11	85.0	82.5	82.5	62,960	14,19,29
40TT	39.21	209.68	6.80	5.49	24	11	11	87.5	82.5	90.0	49,938	14,19,25,38
50LL	142.18	466.78	64.34	39.92	49	42	38	76.0	88.0	86.0	73,054	15,28,33,35
50LT	138.98	666.23	33.69	26.67	24	19	17	76.0	86.0	82.0	68,904	6,26,32,46
50TL	115.73	188.11	11.01	8.06	14	11	13	88.0	88.0	88.0	54,009	3,26,45
50TT	105.01	303.43	7.42	5.23	20	7	7	88.0	88.0	92.0	45,506	17,26,48
75LL	Mem	8,979.00	525.81	387.66	111	105	88	85.3	89.3	86.7	142,551	14,23,35,37,56
75LT	Mem	8,327.27	137.00	87.58	73	23	15	90.7	92.0	90.7	110,194	14,25,32,35,38,59
75TL	Mem	3,918.76	69.85	71.12	26	13	13	92.0	92.0	96.0	89,562	14,35,37
75TT	Mem	4,986.99	42.92	33.84	36	12	11	93.3	90.7	90.7	81,697	25,32,38,59
100LL	Mem	32,872.67	904.00	665.94	58	53	54	94.0	94.0	94.0	1,777,224	29,55,64,73
100LT	Mem	Time	1,078.75	684.20	-	63	56	-	92.0	92.0	1,775,001	29,44,54,68,96
100TL	Mem	4,236.89	87.64	76.50	13	12	13	94.0	93.0	95.0	1,724,826	5,52,95
100TT	Mem	6,743.21	156.57	159.85	13	13	13	95.0	93.0	93.0	1,712,163	5,34,44,52,95
200LL	Mem	Time	11,435.00	6,430.31	-	40	39	-	96.0	96.0	1,102,073	43,95,159
200LT	Mem	Time	12,404.83	7,120.03	-	53	46	-	95.0	95.5	1,087,014	41,96,168,171
200TL	Mem	Time	953.94	879.52	-	12	12	-	97.5	97.5	1,068,431	54,95,186
200TT	Mem	Time	783.13	673.33	-	6	6	-	97.5	97.5	1,055,760	52,54,115,168,186

limit (15 hours of CPU time) to optimality, we write "Time" in the corresponding entry of the table. If an algorithm runs out of memory, we write "Mem".

Our goal of presenting the results of PO in Table 1 is to compare the strength of the cuts generated by BD1 and BD2 with this method. We use the decomposition scheme proposed by Contreras et al. (2012) for solving the Pareto-optimal subproblem. In this method, at iteration e of Algorithm 1, by fixing the value of $(b_i)_{i\in H_1^e}$ at the optimal values obtained from the original subproblem, and by setting $(b_i)_{i\in H_0^e}$ to zero, they decompose the PO subproblem into $|K|$ (here $|K||M|$) independent problems. These problems can be solved using an LP solver, however, as Contreras et al. (2011a) argue, for computational tractability, they sacrifice the strength of the cuts by solving the resulting problems via an approximation technique. Since our goal of implementing PO method is to compare the strength of the resulting cuts with our proposed methods, after decomposing the PO subproblem into $|K||M|$ independent problems, rather than employing the approximation technique, we solve the dual of the resulting problems using the CPLEX LP solver. Note that the number of iterations cannot be reduced by the approximation algorithm, hence the number of iterations of PO as presented in Table 1 provides a lower bound on the number of iterations that we would obtain using the approximation technique proposed by Contreras et al. (2011a).

Table 1 shows that both algorithms BD1 and BD2 outperform CPLEX in terms of computational time and the number of instances solved to optimality. Additionally, the results of Table 1 clearly indicate that our algorithms (BD1 and BD2) outperform PO, with the only exception of instance 40LL.

Each of the Benders algorithms proposed in this paper is able to solve all considered instances to optimality within an hour of CPU time, with the exception of instances 200LL and 200LT, which take approximately 3.3 hours for BD1 and 1.9 hours for BD2, respectively. The columns % hubs elim. show that a large percentage of candidate hubs can be eliminated by variable fixing. The columns Total time (sec) and Iterations indicate that the convergence of the Benders algorithm is improved by solving Phase II as LP-relaxations of knapsack problem, especially for the larger size instances. This implies that BD2 outperforms BD1; hence, we use BD2 for the rest of the computational experiments.

Table 1 also shows the locations of installed hubs in optimal solutions, where the optimal number of hubs to locate varies between one and six. It seems that, in these particular instances, the number of installed hubs does not depend on the size of the instance; it is rather more dependent on hub installation costs and capacities. For example, in the instances with tight installation costs and loose capacities, the problem tends to result in locating fewer hubs.

Table 2 Percentage of total demand satisfied for each demand class.

Instance type	Demand class 1	Demand class 2	Demand class 3	Average
LL	98.32%	94.58%	71.74%	88.21%
LT	98.32%	87.95%	55.83%	80.70%
TL	98.32%	93.45%	59.99%	83.92%
TT	93.00%	75.45%	39.64%	69.37%

In Table 2, we observe the percentages of satisfied demand from different market segments. The averages for each demand class are calculated over instances from Table 1 with the same type of installation costs and capacities. The last column provides the average percentages of total satisfied demand. Among the three demand classes, the first class is the one with the highest percentages of satisfied demand, as serving this class of demand yields the highest revenue. On the other hand, for the instances with the same configuration of hub installation costs and capacities, the third demand class, having the least revenues, has the least percentages of satisfied demand as expected.

Table 3 **Computational results for the deterministic model using BD2 with Sets I, II, and III instances.**

Instance	Total time (sec)	Iterations	% hubs elim.	Profit	#open hubs
Set I					
50L	22.30	31	88.0	21,765	6
50T	15.49	19	90.0	16,943	4
100L	45.43	36	94.0	30,273	4
100T	50.18	23	95.0	30,215	4
150L	648.31	65	93.3	77,667	7
150T	523.05	25	94.7	74,952	6
200L	1,734.22	30	96.0	155,097	8
200T	3,160.84	33	95.5	144,874	7
250L	19,441.16	69	96.0	325,657	9
250T	4,861.76	37	97.2	181,027	5
300L	20,132.06	75	96.0	388,322	11
300T	14,885.61	52	97.7	188,944	6
Set II					
50L	64.29	35	88.0	85,327	6
50T	50.78	27	86.0	60,174	5
100L	224.41	48	92.0	201,451	7
100T	448.40	58	90.0	199,819	8
150L	3,785.39	121	91.3	458,915	12
150T	2,904.36	38	92.7	438,771	9
200L	1,442.94	25	95.0	236,247	8
200T	3,556.37	21	94.5	216,241	9
250L	10,414.33	62	95.2	437,819	11
250T	5,589.51	47	96.0	354,688	10
300L	17,367.18	68	96.0	1,532,224	10
300T	16,399.61	49	97.0	986,373	7
Set III					
50L	32.50	37	88.0	19,037	5
50T	19.93	32	90.0	16,353	5
100L	736.18	21	88.0	511,879	9
100T	530.51	16	91.0	39,672	7
150L	5,945.02	28	91.3	129,155	12
150T	5,327.03	27	92.7	964,137	9
200L	7,920.45	49	90.0	144,503	15
200T	6,738.52	45	93.5	103,987	11
250L	12,166.41	41	92.8	992,585	16
250T	11,057.31	32	95.2	710,562	11
300L	22,565.61	39	93.3	1,004,591	17
300T	18,349.47	34	95.0	903,773	14

Moreover, instances with loose capacities (LL and TL) result in higher percentages on average compared to the instances with tight capacities (LT and TT).

To better understand the performance of the proposed algorithms from a computational point of view, we also present computational results with larger-size instances introduced by Contreras et al. (2011a), and later extended by Contreras et al. (2012). There are three different sets of instances, referred as Set I, Set II, and Set III, which are constructed by considering three different levels of magnitude for the amount of flow originating at a given node: low-level (LL) nodes, medium-level (ML) nodes, and high-level (HL) nodes. The total outgoing flow of LL, ML and HL nodes are obtained from the interval [1,10], [10,100], and [100,1000], respectively. Capacities of hubs are generated by using the formula provided in Ebery et al. (2000) in which parameter ρ is taken to be 0.5 and 1.5 for the loose (L) and tight (T) types of capacities, respectively. The other sets of parameters are as described in the beginning of Section 6. For Sets I, II, and III, we test instances with $|H| \in \{50, 100, 150, 200, 250, 300\}$. The detailed results are provided in Table 3 where the column titles have the same meanings as in the previous table.

All of the instances presented in Table 3 from Sets I, II, and III are solved to optimality. The most time-consuming instance in Sets I, II, and III took around 6, 5, and 6 hours, respectively, to solve to optimality. The averages of the computational times reported in Table 3 for the Sets I, II, and III are 1.5, 1.4, and 2.1 hours, respectively.

Lastly, we present additional runs from Sets I and II with $|H| \in \{350, 400, 500\}$ to analyze the limit of our algorithm. We have extended the CPU time limit to 24 hours for these instances. The results are presented in Table 4. Our algorithm is able to solve all of the instances to optimality within the time limit, except for the instance 500L in Set II. That particular instance resulted in an optimality gap of 2.52%. These results further confirm the efficiency and robustness of the BD2 algorithm when considering more challenging and larger-size instances.

Table 4 Computational results of Sets I and II instances with $|H| = \in \{350, 400, 500\}$.

Instance	Total time (sec)	Iterations	% hubs elim.	Profit	#open hubs
Set I					
350L	28,118.49	19	96.9	188,759	11
350T	21,881.74	15	97.7	147,501	7
400L	43,942.63	16	97.0	248,458	12
400T	35,763.21	14	97.5	186,239	9
500L	83,149.54	15	95.4	1,137,769	23
500T	54,731.43	11	97.0	821,439	14
Set II					
350L	31,025.56	31	95.7	682,514	15
350T	23,149.81	26	96.6	530,697	11
400L	40,209.23	29	95.3	852,627	19
400T	33,195.23	22	95.3	671,734	13
500L	Time	17	71.8	965,113	25
500T	82,296.42	14	96.6	583,129	17

6.2. Computational Results for the Stochastic Model

In this section, we first focus on the practical convergence of the SAA scheme using the stochastic model, we then test the performance of our methods on the instances involving up to 75 nodes. All computational experiments with the stochastic model are performed using BD2 coupled with SAA.

6.2.1. Sample generation. We generate independent samples for demands of commodities using a normal distribution parameterized as follows: Let $\bar{w}_k^m$ be the demand of commodity k of class m in the deterministic case. Moreover, let $\bar{w}_k = \sum_{m \in M} \bar{w}_k^m$ be the total demand of commodity k, and $\rho_m^k = \frac{\bar{w}_k^m}{\bar{w}_k}$ be the proportion of demand of segment m of commodity k. We assume that the total demand of k (i.e. w_k) is drawn from a normal distribution in which the mean demand is set to $\bar{w}_k$ and the standard deviation is equal to $\sigma_k = \nu \bar{w}_k$, where ν is the coefficient of variation. Consequently, once the total demand of commodity k is realized, the correlated demand of class m is computed as $w_k^m = \rho_m^k w_k$.

6.2.2. Practical convergence of the SAA algorithm. The aim of this section is to analyze the practical convergence of the SAA scheme in order to choose a sample size $|\mathcal{N}|$ and the number of replications $|\mathcal{M}|$ that provide the best trade-off between solution quality and computational time. To this end, we perform computational tests with sample sizes $|\mathcal{N}| \in \{50, 100, 500, 1000\}$ and a number of replications $|\mathcal{M}| \in \{10, 20, 40, 60, 80\}$. We select two 10-node instances of the AP dataset and generate independent samples as explained above, with ν set to 0.5. The sample size of $|\mathcal{N}'| = 10,000$ is used to evaluate the SAA gap.

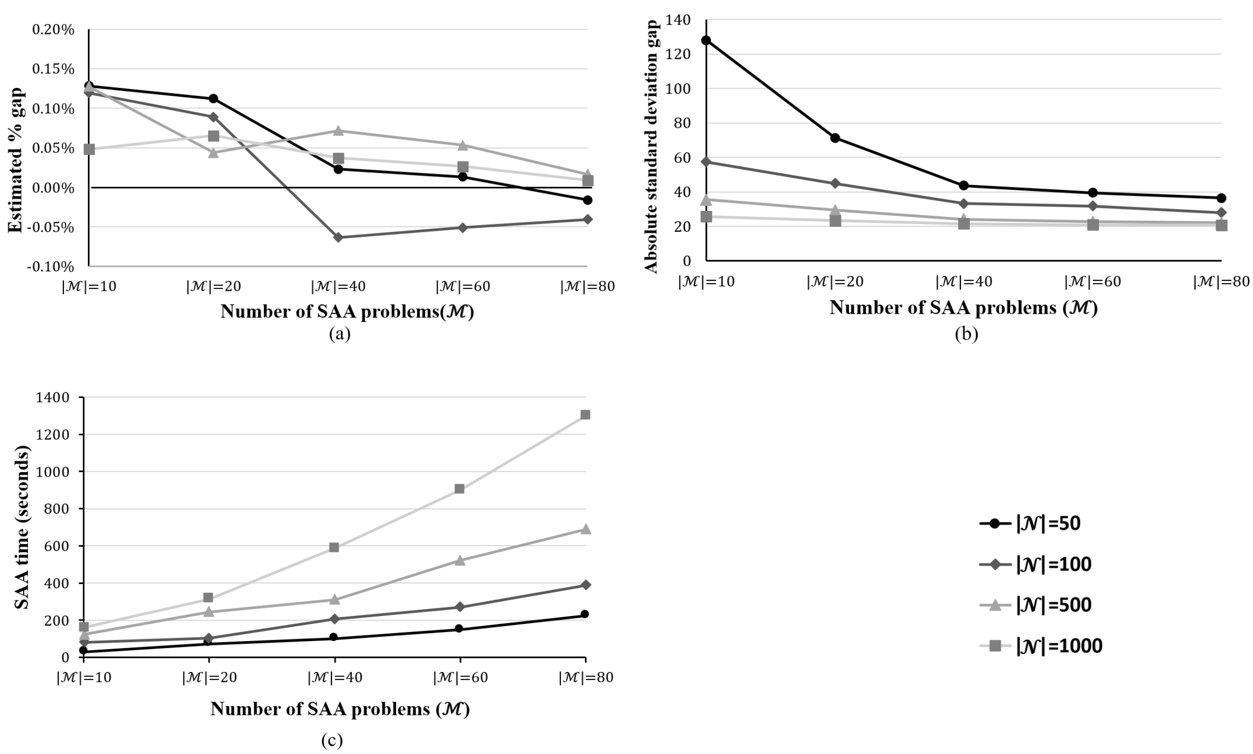

Figure 1 **Optimality gap, standard deviation for the optimality gap, and the total CPU time required for the SAA algorithm for different sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$ with AP10LL.**

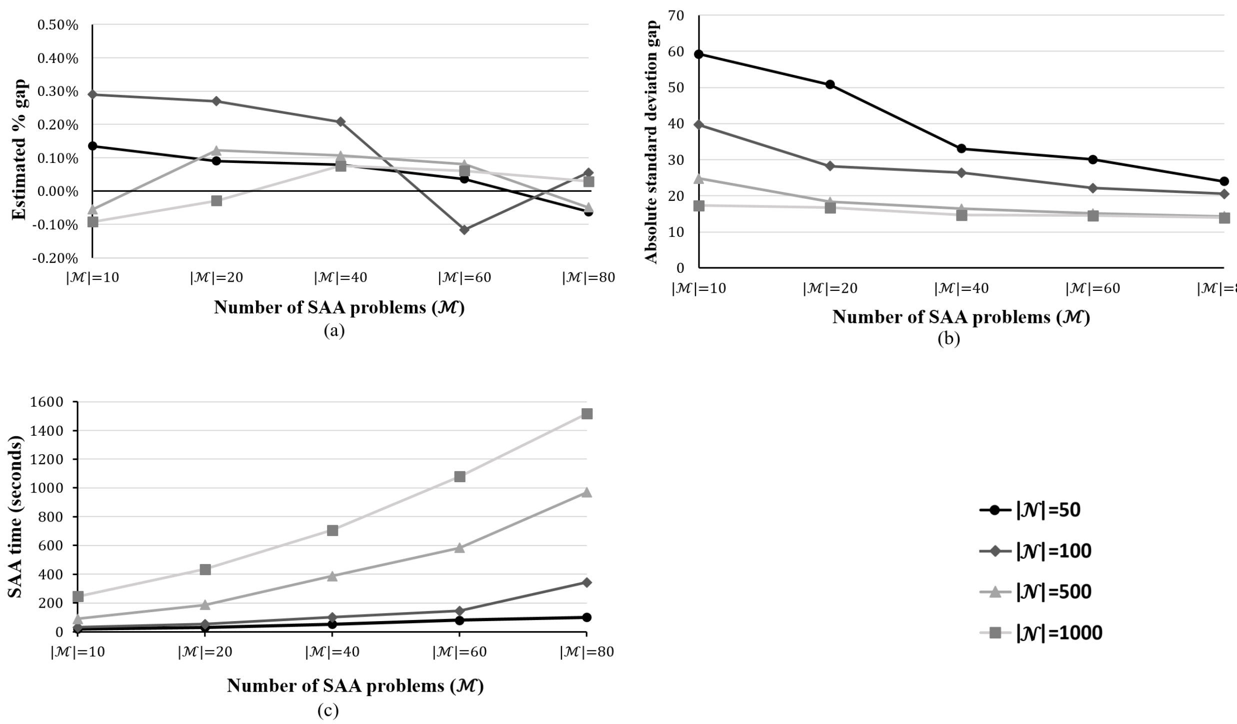

Figure 2 **Optimality gap, standard deviation for the optimality gap, and the total CPU time required for the SAA algorithm for different sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$ with AP10TL.**

Figures 1 and 2 plot the optimality gap, standard deviation for the optimality gap, and the computational time required for the SAA algorithm for different sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$, with AP10LL and AP10TL, respectively. Figures 1(a) and 2(a) clearly indicate that larger sample size result in smaller optimality gap on average. It is also observed that as the sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$ increase, the corresponding standard deviation for the optimality gap decreases (Figures 1b and 2b), whereas the corresponding computation time increases significantly (Figures 1c and 2c), for both AP10LL and AP10TL instances. In general, the largest sample size $|\mathcal{N}| = 1000$ provides the best average SAA gap with the least variation, and the sample size $|\mathcal{N}| = 50$ is the best in terms of the trade-off between solution quality and computational time. For this reason, we use sample sizes $|\mathcal{N}| = 50$ and $|\mathcal{M}| = 60$ during the rest of our computational experiments.

6.2.3. Solving larger-size instances. We first evaluate the performance of the acceleration techniques proposed for SAA. The results are provided in Table EC.2 in Online Appendix EC.3. In our computational experiments, the algorithm runs up to five times faster, and more than two times faster on the average, with the implementation of the acceleration techniques. Hence, all computational experiments with the stochastic model are carried out using the acceleration techniques.

We now analyze and evaluate the performance of the SAA algorithm on larger-size instances with up to 75 nodes from the AP dataset. For each instance, we consider two values 0.5 and 1 as

Table 5 Computational results for the stochastic model with 48 instances of the AP dataset.

Instance			Optimal solution		% Gap	CI for SAA % gap at	
$\lvert H\rvert$	ν	Profit	Open hubs	Time (sec)	SAA	95%	99%
10LL	0.5	21,380	5,6,9,10	50.82	0.08	(-6.16, 11.73)	(-8.74, 14.31)
	1	22,529	5,6,9,10	41.76	-0.04	(-4.91, 13.47)	(-7.47, 16.03)
10LT	0.5	3,344	5	15.39	0.06	(-9.96, 13.67)	(-13.45, 17.16)
	1	3,358	5,6	18.33	0.18	(-7.24, 15.50)	(-9.29, 17.55)
10TL	0.5	13,890	4,5,9	33.89	0.14	(-14.02, 7.87)	(-18.39, 12.24)
	1	14,586	4,5,9	53.48	-0.04	(-11.05, 8.37)	(-12.19, 9.51)
10TT	0.5	2,690	5	14.97	0.03	(-9.96, 13.67)	(-13.56, 17.27)
	1	2,683	5	13.11	0.14	(-6.42, 16.57)	(-10.60, 20.75)
20LL	0.5	105,282	7,9,10,19	1,553.42	0.04	(-15.15, 14.31)	(-16.73, 15.89)
	1	113,599	7,9,10,14,19	2,713.17	0.03	(-9.92, 15.72)	(-14.67, 20.47)
20LT	0.5	59,727	5,9,10,12,14,19	1,442.36	0.05	(-4.45, 8.72)	(-9.21, 13.48)
	1	63,731	5,9,10,12,14,19	1,907.56	0.04	(-3.72, 10.74)	(-7.05, 14.07)
20TL	0.5	53,035	5,7,10	1,089.42	0.08	(-11.67, 10.97)	(-16.07, 15.37)
	1	57,511	5,7,10	1,173.79	0.06	(-8.60, 15.09)	(-11.86, 18.35)
20TT	0.5	13,672	10	101.11	0.02	(-10.62, 10.07)	(-11.90, 11.35)
	1	13,984	10	76.78	-0.03	(-8.08, 12.76)	(-10.94, 15.62)
25LL	0.5	133,240	7,14,17,23	1,418.24	-0.03	(-10.32, 13.46)	(-13.03, 16.17)
	1	140,164	7,14,17,23	2,014.32	-0.02	(-6.47, 14.83)	(-11.33, 19.69)
25LT	0.5	92,042	6,10,12,14,25	3,218.51	-0.01	(-6.23, 10.47)	(-11.13, 15.37)
	1	98,570	9,10,12,14,19,25	3,832.22	0.01	(-9.87, 11.51)	(-14.58, 16.22)
25TL	0.5	81,755	6,9,14,23	1,127.83	-0.02	(-8.48, 10.26)	(-9.95, 11.73)
	1	87,625	6,9,14,23	1,613.32	-0.02	(-5.98, 12.38)	(-10.64, 17.04)
25TT	0.5	36,956	6,10,14,25	1,420.91	0.00	(-9.16, 8.70)	(-11.60, 11.14)
	1	39,992	6,9,10,14,25	1,711.33	-0.01	(-7.18, 10.91)	(-8.92, 12.65)
40LL	0.5	80,696	12,22,26,29	15,219.35	-0.03	(-18.92, 17.71)	(-22.80, 21.59)
	1	86,456	9,22,26,29,38	17,612.49	0.02	(-15.39, 19.48)	(-16.88, 20.97)
40LT	0.5	71,192	12,14,26,29,30,38	20,369.86	-0.04	(-19.32, 22.57)	(-21.37, 24.62)
	1	76,989	5,14,19,26,29,30,38	22,928.64	0.04	(-16.86, 23.36)	(-18.55, 25.05)
40TL	0.5	65,621	14,19,29	7,549.28	-0.03	(-14.74, 19.37)	(-18.99, 23.62)
	1	71,406	14,19,29	8,084.24	0.05	(-12.37, 21.22)	(-14.30, 23.15)
40TT	0.5	52,843	14,19,25,38	9,672.36	-0.04	(-17.53, 19.75)	(-19.06, 21.28)
	1	57,349	5,19,25,30	10,836.71	0.05	(-15.95, 23.42)	(-20.59, 28.06)
50LL	0.5	77,216	15,28,33,35	13,565.38	0.03	(-19.46, 25.37)	(-27.00, 32.91)
	1	83,180	5,15,28,33,35	15,738.46	-0.03	(-17.11, 26.23)	(-25.07, 34.19)
50LT	0.5	73,467	6,26,32,46	11,874.35	0.05	(-15.83, 21.64)	(-22.38, 28.19)
	1	79,923	6,19,26,30,46	14,969.23	-0.06	(-13.26, 23.83)	(-16.90, 27.47)
50TL	0.5	58,384	3,26,45	5,595.46	0.05	(-16.30, 16.99)	(-20.26, 20.95)
	1	60,756	3,14,29,45	7,304.05	-0.09	(-12.61, 18.28)	(-17.92, 23.59)
50TT	0.5	48,376	6,26,48	6,393.29	0.12	(-14.67, 16.36)	(-20.39, 22.08)
	1	53,261	6,26,48	7,141.92	-0.07	(-12.71, 18.49)	(-17.16, 22.94)
75LL	0.5	145,792	14,23,35,37,56	43,955.24	0.10	(-17.54, 12.63)	(-23.37, 18.46)
	1	198,962	5,14,19,26,29,30,38	46,085.51	0.04	(-12.46, 18.49)	(-16.55, 22.58)
75LT	0.5	113,510	14,25,32,35,38,59	36,666.45	-0.08	(-14.29, 18.81)	(-18.50, 23.02)
	1	122,007	14,26,32,35,46,59	39,157.63	0.04	(-11.39, 19.41)	(-16.60, 24.62)
75TL	0.5	92,609	14,35,37	24,531.79	0.16	(-21.74, 19.83)	(-29.04, 27.13)
	1	96,829	14,35,37	25,742.90	0.07	(-18.25, 22.49)	(-24.14, 28.38)
75TT	0.5	81,697	25,32,38,59	32,559.79	0.14	(-23.67, 16.74)	(-31.43, 24.50)
	1	84,157	25,26,32,38,59	34,637.15	-0.10	(-17.34, 18.51)	(-22.33, 23.50)

the coefficient of variation to represent the amount of uncertainty in the stochastic demand. The computational results are summarized in Table 5. The first two columns provide the number of nodes and the coefficient of variation. The next three columns labeled “Optimal solution” present the net profit ($\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}$), the best hub locations, and the run time of instances (in seconds) obtained from solving the SAA algorithm, respectively. The next column labeled "% Gap" provides the percent optimality gap relative to the best solution obtained by the SAA algorithm. The last two columns labeled "CI for SAA % gap at" give the 95% and 99% confidence interval for the optimality gap of the best solution obtained by the SAA algorithm, respectively.

The results provided in Table 5 indicate that the estimated optimality gaps obtained by the SAA algorithm are always below 0.2%, and that the corresponding confidence intervals for the optimality gaps are quite narrow for both 95% and 99%. These confirm the efficiency of the SAA algorithm proposed for the problem with stochastic demand, and also imply that the solutions produced by our algorithm are good enough to be used in practical applications.

We next observe the effects of variability in uncertain demands on the solutions reported in Table 5. When the ν value increases, that is, when the variability in the uncertain demand increases, the net profit values and the computation time required for the SAA algorithm also increase. Note that the best found hub locations do not change significantly under these variations. We can identify a few instances in which hub locations change by demand variation, and in most of the instances, the locations of the hubs obtained with the deterministic and stochastic models are identical (Tables 1 and 5). It seems that, in these particular instances, the long-term location decisions are dependent more on the configuration of hub installation costs and capacities than the demand.

7. Conclusion

In this paper, we defined the profit maximizing hub location problem with capacity allocation by incorporating revenue management decisions, and embedding more realistic and challenging capacity constraints for hubs. We presented a strong path-based deterministic mixed integer programming formulation of the problem. We further addressed demand uncertainty and developed a two-stage stochastic program. We described two Benders-based algorithms to solve large-scale instances of the problem by developing a new decomposition methodology for solving the Benders subproblems. We proved that the subproblems can be viewed as a set of LP-relaxations of maximum weighted matching problems (BD1), or as a series of LP-relaxations of knapsack problems (BD2). We further enhanced the algorithms by incorporating improved variable fixing techniques.

For the stochastic problem, we presented a solution method that integrates the proposed Benders decomposition algorithms with the SAA scheme to obtain solutions to problems with a large number of scenarios. We also developed novel acceleration techniques to improve the convergence of the algorithms employed for the stochastic problem, which resulted in up to five times improvement in CPU times.

We performed extensive computational experiments on the well-known AP dataset, and also on larger-size instances from the literature, to analyze the performance of the proposed algorithms. In view of our computational results, both algorithms outperform the best known cuts (Pareto-optimal cuts), in terms of computational efficiency and also quality.

The results further show that BD2 outperforms BD1 particularly on larger instances. BD2 succeeded to optimally solve instances with up to 500 nodes and 750,000 commodities of different demand classes. Integrating BD2 with the SAA scheme, instances involving up to 75 nodes and 16,875 commodities were solved within an estimated optimality gap of lower than 0.2%. These results clearly confirm the efficiency and robustness of our algorithms.

The results obtained from the stochastic model provided several observations, which can be used as a guideline in the design of optimal hub networks to maximize profit. For example, the amount of net profit is very sensitive to the variance of the stochastic demand. However, it was observed that the long-term location decisions do not change significantly under these variations.

We leave several extensions of our problem open for future research. An interesting avenue of research is addressing uncertainty in costs and revenues. Another future research direction would be to consider network congestion, for example, with capacitated arcs.

Endnotes

1. There exists a Lagrangian relaxation of the subproblem which converts it into a transportation problem. However, this transportation problem would be different than that of Contreras et al. (2012). Moreover, due to difficulties with convergence, solving this problem using a sub-gradient method would computationally be challenging.

Acknowledgments

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) with grant RGPIN-2015-05548. This support is gratefully acknowledged.

References

Alibeyg A, Contreras I, Fernández E (2016) Hub network design problems with profits. *Transportation Research Part E: Logistics and Transportation Review* 96:40–59.

Alibeyg A, Contreras I, Fernández E (2018) Exact solution of hub network design problems with profits. *European Journal of Operational Research* 266(1):57–71.

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

Alumur SA, Nickel S, Saldanha-da Gama F (2012) Hub location under uncertainty. *Transportation Research Part B: Methodological* 46(4):529–543.

Balas E, Zemel E (1980) An algorithm for large zero-one knapsack problems. *Operations Research* 28(5):1130–1154.

Beasley JE (1990) OR Library: Hub location. http://people.brunel.ac.uk/mastjjb/jeb/ orlib/phubinfo.html.

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

Burkard RE, Dell'Amico M, Martello S (2009) *Assignment problems*, volume 125 (SIAM).

Camargo RS, de Miranda G, Løkketangen A (2013) A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling* 37(12-13):7465–7480.

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

Camargo RS, de Miranda Jr G, Luna HPL (2009) Benders decomposition for hub location problems with economies of scale. *Transportation Science* 43(1):86–97.

Camargo RS, de Miranda Jr G, OKelly ME, Campbell JF (2017) Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints. *Applied Mathematical Modelling* 51:274–301.

Camargo RS, Miranda G (2012) Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications* 39(3):3385–3391.

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

Campbell JF (1994) Integer programming formulations of discrete hub location problems. *European Journal of Operational Research* 72(2):387–405.

Campbell JF, Ernst AT, Krishnamoorthy M (2002) Hub location problems. Drezner Z, Hamacher HW, eds., *Facility Location: Applications and Theory*, 373–407 (Springer).

Campbell JF, O'Kelly ME (2012) Twenty-five years of hub location research. *Transportation Science* 46(2):153–169.

Contreras I (2015) Hub location problems. Laporte G, Nickel S, Saldanha-da-Gama F, eds., *Location Science*, 311–344 (Springer).

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

Contreras I, Cordeau JF, Laporte G (2011b) Stochastic uncapacitated hub location. *European Journal of Operational Research* 212(3):518–528.

Contreras I, Cordeau JF, Laporte G (2012) Exact solution of large-scale hub location problems with multiple capacity levels. *Transportation Science* 46(4):439–459.

Crainic TG (2000) Service network design in freight transportation. *European Journal of Operational Research* 122(2):272–288.

Dantzig GB (1957) Discrete-variable extremum problems. *Operations Research* 5(2):266–288.

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

Ebery J, Krishnamoorthy M, Ernst A, Boland N (2000) The capacitated multiple allocation hub location problem: Formulations and algorithms. *European Journal of Operational Research* 120(3):614–631.

Ernst AT, Krishnamoorthy M (1996) Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science* 4(3):139–154.

Galil Z (1986) Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)* 18(1):23–38.

Gelareh S, Nickel S (2011) Hub location problems in transportation networks. *Transportation Research Part E: Logistics and Transportation Review* 47(6):1092–1111.

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

Lin CC, Lee SC (2018) Hub network design problem with profit optimization for time-definite LTL freight transportation. *Transportation Research Part E: Logistics and Transportation Review* 114:104–120.

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* 29(3):464–484.

Mak WK, Morton DP, Wood RK (1999) Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* 24(1-2):47–56.

Marianov V, Serra D (2003) Location models for airline hubs behaving as M/D/c queues. *Computers & Operations Research* 30(7):983–1003.

Marín A (2005) Formulating and solving splittable capacitated multiple allocation hub location problems. *Computers & Operations Research* 32(12):3093–3109.

Martins de Sá E, Camargo RS, Miranda G (2013) An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research* 226(2):185–202.

Martins de Sá E, Contreras I, Cordeau JF, Saraiva de Camargo R, de Miranda G (2015) The hub line location problem. *Transportation Science* 49(3):500–518.

Meraklı M, Yaman H (2016) Robust intermodal hub location under polyhedral demand uncertainty. *Transportation Research Part B: Methodological* 86:66–85.

Meraklı M, Yaman H (2017) A capacitated hub location problem under hose demand uncertainty. *Computers & Operations Research* 88:58–70.

Rodriguez-Martin I, Salazar-Gonzalez JJ (2008) Solving a capacitated hub location problem. *European Journal of Operational Research* 184(2):468–479.

Santoso T, Ahmed S, Goetschalckx M, Shapiro A (2005) A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research* 167(1):96–115.

Schütz P, Tomasgard A, Ahmed S (2009) Supply chain design under uncertainty using sample average approximation and dual decomposition. *European Journal of Operational Research* 199(2):409–419.

Shapiro A, Homem-de Mello T (1998) A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming* 81(3):301–325.

Sim T, Lowe TJ, Thomas BW (2009) The stochastic p-hub center problem with service-level constraints. *Computers & Operations Research* 36(12):3166–3177.

Taherkhani G, Alumur S (2018) Profit maximizing hub location problems. *OMEGA* https://doi.org/10.1016/j.omega.2018.05.016.

Van Slyke RM, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* 17(4):638–663.

Wieberneit N (2008) Service network design for freight transportation: A review. *OR Spectrum* 30(1):77–112.

Yang TH (2009) Stochastic air freight hub location and flight routes planning. *Applied Mathematical Modelling* 33(12):4424–4430.

Technical Extensions and Numerical Study Supplements

EC.1. Bipartite Transformation of the MWM Problem

Here, we show how LP-MWM(km) (51)-(53) can be transformed into an equivalent MWM problem in a bipartite graph. Define bipartite graph $G^B_{km} = (V_1, V_2, B)$, where $V_1 = \{v^1_i\}_{i\in H^e_0}$, $V_2 = \{v^2_i\}_{i\in H^e_0}$, and $B = \{(v^1_i, v^2_j) \in V_1 \times V_2 : (i,j) \in A^0_{ke}\}$. Let μ^B be a maximum weighted matching in G^B_{km}, and let

$$\mu^{km}_{ij} = \frac{1}{2}(\mu^B_{(v^1_i,v^2_j)} + \mu^B_{(v^1_j,v^2_i)}) \quad \forall (i,j) \in A^0_{ke}. \tag{EC.1}$$

Proposition EC.1. *$\mu^{km} = (\mu^{km}_{ij})_{(i,j)\in A^0_{ke}}$ defined in* (EC.1) *is optimal to LP-MWM(km)* (51)-(53).

Proof. μ^B being a matching in G^B_{km} requires

$$\sum_{j\in H^e_0:(i,j)\in A^0_{ke}} \mu^B_{(v^1_i,v^2_j)} \le 1 \quad \forall v^1_i \in V_1 \text{ (or equivalently } i \in H^e_0) \tag{EC.2}$$

$$\sum_{j\in H^e_0:(i,j)\in A^0_{ke}} \mu^B_{(v^1_j,v^2_i)} \le 1 \quad \forall v^2_i \in V_2 \text{ (or equivalently } i \in H^e_0). \tag{EC.3}$$

Summing (EC.2) and (EC.3) and dividing the resulting inequality by 2 yields

$$\sum_{j\in H^e_0:(i,j)\in A^0_{ke}} \frac{1}{2}(\mu^B_{(v^1_i,v^2_j)} + \mu^B_{(v^1_j,v^2_i)}) = \sum_{j\in H^e_0:(i,j)\in A^0_{ke}} \mu^{km}_{ij} \le 1 \quad \forall i \in H^e_0, \tag{EC.4}$$

which implies (52), hence μ^{km} defined in (EC.1) is feasible for LP-MWM(km). Now we show that μ^{km} is also optimal for LP-MWM(km). By contradiction assume that μ^{km} is not optimal for LP-MWM(km), hence there exists a solution $\hat{\mu}^{km}$ with a strictly higher total weight. We show that this contradicts with optimality of μ^B. Define $\hat{\mu}^B$ as

$$\hat{\mu}^B_{(v^1_i,v^2_j)} = \hat{\mu}^B_{(v^1_j,v^2_i)} = \hat{\mu}^{km}_{ij} \quad \forall (i,j) \in A^0_{ke}. \tag{EC.5}$$

$\hat{\mu}^B$ clearly satisfies (EC.2) and (EC.3), hence is a feasible matching for the bipartite graph G^B_{km}. Moreover, note that

$$\sum_{(i,j)\in A^0_{ke}} (\hat{\mu}^B_{(v^1_i,v^2_j)} + \hat{\mu}^B_{(v^1_j,v^2_i)}) = 2\sum_{(i,j)\in A^0_{ke}} \hat{\mu}^{km}_{ij} > 2\sum_{(i,j)\in A^0_{ke}} \mu^{km}_{ij} = \sum_{(i,j)\in A^0_{ke}} (\mu^B_{(v^1_i,v^2_j)} + \mu^B_{(v^1_j,v^2_i)}), \tag{EC.6}$$

which implies that $\hat{\mu}^B$ is a better matching than μ^B, contradicting the optimality of μ^B. $\square$

The optimal solution to MWC(km) (48)-(50) can be obtained in a similar manner. Let t^B be a minimum weight cover in G^B_{km}, and let

$$t^m_{ik} = \frac{1}{2}(t^B_{v^1_i} + t^B_{v^2_i}) \quad \forall i \in H^e_0. \tag{EC.7}$$

Proposition EC.2. *$t_k^m = (t_{ik}^m)_{i \in H_0^e}$ defined in* (EC.7) *is optimal to MWC*(km) (48)-(50).

Proof. t^B being a cover in G_{km}^B implies that

$$t_{v_i^1}^B + t_{v_j^2}^B \geq \beta_{ij}^{km} \quad \forall (i,j) \in A_{ke}^0 \tag{EC.8}$$

$$t_{v_j^1}^B + t_{v_i^2}^B \geq \beta_{ji}^{km} \quad \forall (j,i) \in A_{ke}^0. \tag{EC.9}$$

Since β^{km} is symmetric, summing (EC.8) and (EC.9) and dividing the resulting inequality by 2 implies (49); hence, t_k^m defined in (EC.7) is feasible for MWC(km). Now, by contradiction assume that t_k^m is not optimal, hence there exists a solution $\hat{t}_k^m$ with a strictly lower cover weight. Define $\hat{t}^B$ as

$$\hat{t}_{v_i^1}^B = \hat{t}_{v_i^2}^B = \hat{t}_{ik}^m \quad \forall i \in H_0^e. \tag{EC.10}$$

$\hat{t}^B$ clearly satisfies (EC.8) and (EC.9), and has a lower cover weight than t^B, contradicting the optimality of t^B. □

Propositions EC.1 and EC.2 indicate that it is enough to find a MWM in bipartite graph G_{km}^B to solve MWC(km) and LP-MWM(km).

EC.2. Hybrid MWM-Knapsack Method

We show how proper values of $(b_i)_{i \in H_0^e}$ can be found via a relaxation of DS-II (31)-(34). Observe that by relaxing constraints (32), relaxed DS-II can be decomposed into $|H_0^e|$ independent smaller subproblems for each $i \in H_0^e$:

$$\text{(Relaxed DS-II}(i)) \quad \text{Minimize} \quad \Gamma_i b_i + \sum_{k \in K} \sum_{m \in M} u_{ik}^m \tag{EC.11}$$

$$u_{ik}^m + w_k^m b_i \geq \rho_{ii}^{km} \qquad (k,m) \in K \times M \tag{EC.12}$$

$$u_{ik}^m, b_i \geq 0 \qquad (k,m) \in K \times M \tag{EC.13}$$

Similar to problem (54)-(56), this problem is the dual of the LP-relaxation of a knapsack problem with knapsack capacity Γ_i, and items $(k,m) \in K \times M$ with weight w_k^m and profit ρ_{ii}^{km}, and can be solved in the same fashion. As explained in Section 4.3.3, b_i is the profit-to-weight ratio of the break item $(\bar{k}, \bar{m})$

$$b_i = \rho_{ii}^{\bar{k}\bar{m}} / w_{\bar{k}}^{\bar{m}}. \tag{EC.14}$$

Note that even though problem (54)-(56) has a formulation similar to problem (EC.11)-(EC.13), they are inherently different: problem (54)-(56) is a restriction of DS-II, whereas problem (EC.11)-(EC.13) is a relaxation of DS-II, hence the solution obtained by (EC.11)-(EC.13) may not be feasible to DS-II. Therefore, once the value of $(b_i)_{i\in H_0^e}$ is calculated, to obtain a complete feasible solution we calculate the value of u using the MWM algorithm.

It is worth mentioning that the major computational effort required for solving this problem is due to calculating ρ_{ii}^{km} values, while the time required for solving the knapsack problem itself is almost negligible even for the largest instances. On the other hand, note that we need to calculate ρ_{ii}^{km} as part of preparing DS-II (31)-(34), no matter what the value of b is. This means using this hybrid MWM-Knapsack method, we can find promising values of b for the MWM problems (hence stronger cuts) without incurring additional computational cost.

EC.3. Supplementary Numerical Results

To determine the best values to use for the parameters p_Q (proportion of hubs that are less likely to be open in the optimal solution) and n_Q (number of subsets into which Q is partitioned) in our variable fixing tests, we selected medium- and large-size instances with 40-100 nodes from the AP dataset and took runs under different p_Q and n_Q values. For each instance, we performed computational tests with $p_Q \in \{50\%, 75\%, 100\%\}$ and $n_Q \in \{1, 2, \ldots, 10\}$, respectively. (The $p_Q = 100\%$ and $n_Q = 1$ combination corresponds to the tests introduced in Contreras et al. 2011a.) The computational results are summarized in Table EC.1.

The first two columns of Table EC.1 report the instance size and the number of hub elimination subsets (n_Q) used, respectively. The columns labeled "Average time" and "Average % hubs elim." report the average computational times in seconds and the average percentage of the total candidate hubs that were eliminated under the three different p_Q values. The averages are calculated over four same-size instances with different installation costs and capacities.

When the values presented in Table EC.1 are averaged, the combination $p_Q = 75\%$ and $n_Q = 6$ provides the best results in terms of average computational time and average percentage of hubs eliminated in the optimal solution. Hence, we used the aforementioned values for the corresponding parameters in our computational experiments.

To evaluate the performance of the acceleration techniques proposed for SAA, we took runs with and without the implementation of the acceleration techniques on the 10–25 node instances from the AP dataset. We compare the computational times in Table EC.2. The first two columns report the instance size and the coefficient of variation. The third and fourth columns labeled "Time (sec)" report the computation times in seconds without and with the implementation of the acceleration techniques, respectively. The last two rows report the averages.

The results provided in Table EC.2 indicate that the algorithm performs more than two times faster on average with the implementation of the proposed acceleration techniques. In particular, for the larger-size instances, the improvement in CPU times goes up to five times.

Table EC.1 **Computational results for different p_Q and n_Q values with 16 instances of the AP dataset.**

Instance		$p_Q = 100\%$		$p_Q = 75\%$		$p_Q = 50\%$	
$\|H\|$	n_Q	Average time	Average % hubs elim.	Average time	Average % hubs elim.	Average time	Average % hubs elim.
40	1	19.60	70.63	19.66	79.38	19.45	81.88
	2	19.77	82.50	19.47	84.38	19.45	83.13
	3	20.05	81.88	19.90	84.38	19.53	83.13
	4	20.19	85.00	19.99	85.00	19.93	80.63
	5	20.35	85.63	20.48	83.75	20.20	80.00
	6	21.05	85.63	20.55	83.13	20.65	76.25
	7	20.83	85.63	21.22	79.38	20.36	74.38
	8	21.35	85.00	21.38	79.38	20.72	68.13
	9	21.41	81.25	21.22	76.25	20.91	66.25
	10	21.58	80.63	21.46	74.38	20.74	63.75
50	1	21.39	83.50	20.36	85.50	20.98	86.50
	2	20.77	86.00	20.02	85.00	20.15	86.50
	3	20.26	89.50	19.83	88.50	20.58	86.50
	4	20.28	90.00	20.03	88.00	20.25	83.50
	5	19.78	90.00	20.03	87.00	20.08	83.00
	6	19.86	90.00	20.28	86.00	21.04	83.50
	7	19.88	89.00	20.04	85.00	20.58	78.00
	8	20.46	89.00	20.11	84.50	20.83	77.00
	9	20.86	88.00	20.56	82.00	21.04	75.00
	10	20.25	84.00	20.97	78.00	20.54	70.50
75	1	148.00	86.00	152.76	89.33	148.11	89.00
	2	149.38	89.33	148.21	92.00	150.00	90.00
	3	146.08	92.33	146.32	91.33	144.23	90.33
	4	144.27	92.00	145.04	91.33	144.91	87.00
	5	149.96	92.00	145.80	91.00	142.15	88.67
	6	151.31	92.67	144.93	90.67	146.04	88.00
	7	159.46	92.00	146.52	88.67	149.52	85.00
	8	156.55	90.67	146.69	89.67	150.75	84.67
	9	149.52	91.00	146.40	88.00	145.76	83.33
	10	157.78	90.67	145.13	86.00	146.11	81.67
100	1	489.43	85.50	483.90	88.50	481.12	89.75
	2	441.71	92.25	430.57	92.00	441.58	92.25
	3	422.44	93.75	427.37	93.75	424.23	92.25
	4	421.21	94.25	407.14	93.25	412.89	91.75
	5	405.94	94.25	407.84	93.50	407.36	91.00
	6	419.81	94.50	394.60	93.50	413.73	89.75
	7	414.71	94.00	397.70	91.00	424.58	88.50
	8	415.16	93.00	401.41	90.75	424.00	86.00
	9	411.12	92.50	397.78	90.75	417.26	85.25
	10	407.30	91.25	405.67	89.50	420.62	84.25

Table EC.2 Computational times for the stochastic model with and without the implementation of the acceleration techniques for SAA.

$\|H\|$	ν	Time (sec) Without acceleration	Time (sec) With acceleration
10LL	0.5	55.29	50.82
	1	47.38	41.76
10LT	0.5	18.42	15.39
	1	21.80	18.33
10TL	0.5	38.63	33.89
	1	53.46	53.48
10TT	0.5	17.34	14.97
	1	15.10	13.11
20LL	0.5	8,723.45	1,553.42
	1	6,601.72	2,713.17
20LT	0.5	2,892.03	1,442.36
	1	3,303.61	1,907.56
20TL	0.5	3,528.43	1,089.42
	1	4,202.39	1,173.79
20TT	0.5	249.41	101.11
	1	197.85	76.78
25LL	0.5	5,494.94	1,418.24
	1	6,653.34	2,014.32
25LT	0.5	15,617.96	3,218.51
	1	8,327.14	3,832.22
25TL	0.5	3,577.96	1,127.83
	1	4,382.44	1,613.32
25TT	0.5	5,053.09	1,420.91
	1	3,152.68	1,711.33
Average	0.5	3,772.25	957.24
	1	3,079.91	1,264.10