European Journal of Operational Research 302 (2022) 1097–1112

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

Lagrange dual bound computation for stochastic service network design

Xiaoping Jiang[a,b], Ruibin Bai[a,*], Jianfeng Ren[a], Jiawei Li[a], Graham Kendall[c,d]

[a] School of Computer Science, University of Nottingham Ningbo China, Ningbo 315100, China
[b] College of Electronic Engineering, National University of Defense Technology, Hefei 230009, China
[c] School of Computer Science, University of Nottingham, Nottingham NG7 2RD, UK
[d] School of Computer Science, University of Nottingham Malaysia Campus, Semenyih 43500, Malaysia

ARTICLE INFO

Article history:
Received 29 November 2020
Accepted 27 January 2022
Available online 3 February 2022

Keywords:
Transportation
Service network design
Stochastic mixed-integer programming
Lagrangian relaxation
Progressive hedging

ABSTRACT

Information on lower bounds plays an important role in the development of exact and heuristic methods for stochastic service network design (SSND). In this paper, we consider the Lagrange dual problem of SSND for computing lower bounds. The Lagrange dual problem is obtained by introducing scenario bundling into scenario-wise decomposition of the SSND model and dualizing the non-anticipativity constraints in a Lagrangian fashion. Our theoretical analysis establishes the superiority of the resulting optimal Lagrange dual bound over that in the case of scenario-wise decomposition. The Lagrange dual problem is solved from the primal perspective by employing the recently proposed FW-PH algorithm, which combines Progressive Hedging with a Frank-Wolfe-like method. To improve the computing efficiency, scenario-wise decomposition in the FW-PH algorithm is replaced with bundle-wise decomposition, which divides the problem by scenario bundles into multiple-scenario subproblems, rather than by individual scenarios into single-scenario subproblems. Scenario bundles are constructed using Gaussian mixture models. Our convergence analysis shows that this improvement retains the desirable theoretical property of FW-PH about convergence to the optimal Lagrange dual value. Computational experiments on SSND instances demonstrate that the improved FW-PH algorithm is far superior to the basic version, providing either a dramatic saving in the run-time required to achieve convergence or a much tighter lower bound when terminated due to the time limit.

1. Introduction

Over the past few years, less-than-truckload and parcel transportation have experienced tremendous growth on the coattails of soaring electronic and mobile commerce (Bai, Wallace, Li & Chong, 2014). Due to relatively small freight in these two types of transportation (Hewitt, Crainic, Nowak & Rei, 2019), consolidation operations are strongly preferred over customized operations. That is, carriers usually choose to consolidate shipments from different customers with possibly different origins and destinations into common vehicles, rather than tailor shipping services for each customer individually (Crainic, 2000). Given customers' demand, a transportation plan is therefore needed for consolidation carriers to distribute the shipments from many origins to many destinations in an underlying network, such as the national highway system. Since each arc of the underlying network is associated with a fixed cost, which represents the minimum investment required to offer shipping services on an arc (Jiang, Bai, Atkin & Kendall, 2017), the planning entails deciding which arcs to open and how to route each shipment on the chosen arcs, so as to minimize the total cost while satisfying customers' transportation demand. The selected arcs constitute a service network, and the process of building the transportation plan is referred to as service network design (SND) (Lium, Crainic & Wallace, 2009). Because of the discrete nature of design decisions and continuous nature of commodity flow, SND problems are often cast in the form of mixed integer programming models.

1.1. Stochastic service network design

Traditionally, customers' transportation demand is defined in the spatial aspect as a certain volume of goods to be shipped between origin-destination pairs (Crainic, 2000), leaving out its temporal dimension. In the Internet and mobile commerce era, the timing of the delivery of a shipment is, however, often an important determinant of service quality. To ensure timely delivery, the planning of service schedules has become an integral part of SND.

* Corresponding author.
E-mail address: ruibin.bai@nottingham.edu.cn (R. Bai).

https://doi.org/10.1016/j.ejor.2022.01.044
0377-2217/© 2022 Elsevier B.V. All rights reserved.

This is usually addressed by dividing the planning horizon into a certain number of time periods and duplicating each terminal of the underlying network in every time period to produce the so-called space-time network (Lium et al., 2009). An arc therein connecting different terminals in two different periods represents the movement from one terminal at some time period to another terminal at the next period. Compared to static SND without temporal requirements, a time-dependent SND involves a significantly larger network and hence is much harder to solve (Wang, Crainic & Wallace, 2019).

Typically, SND deals with tactical planning of operations for a relatively long period of time (e.g., next season), and future demand generally cannot be accurately predicted. In order to build a robust service network that functions well for various demand realizations, demand uncertainty must be explicitly taken into account (Lium et al., 2009). For SND under demand uncertainty, the set of decisions can be divided into two groups according to the availability of demand information at the decision-making point, giving rise to a two-stage decision problem. In the first stage, decisions about arc selection have to be taken without demand information. After future demand is realized, decisions about the most cost-effective way of flowing commodities are made in the second stage. In most applications, demand uncertainty is described by either continuous probability distributions or discrete distributions with numerous outcomes, making the resulting optimization model intractable (Lium et al., 2009). A common approach to this problem consists in approximating these distributions as a finite number of scenarios, each with a probability of occurrence (Høyland, Kaut & Wallace, 2003). With uncertain demand represented by scenarios, the stochastic SND problem can be formulated as a deterministic model, the objective function of which is to minimize the first-stage cost of network construction, plus the expectation of the second-stage cost with respect to all the scenarios. Compared to deterministic SND, stochastic SND is more difficult to tackle owing to the sheer number of second-stage decisions and the additional complexity of balancing first-stage decisions against various scenarios.

1.2. Lower bound computation

As an NP-hard problem, deterministic SND is generally difficult to solve (Bai, Kendall, Qu & Atkin, 2012). The extension of deterministic static SND to stochastic SND makes the resulting model even more challenging from a computational perspective. Fortunately, the model contains some decomposable structures that can be advantageously exploited to alleviate the computational difficulty. Broadly speaking, the decomposition strategies developed to date can be categorized into (i) stage-wise decomposition where the model is decomposed vertically by time stages, and (ii) scenario-wise decomposition where the model is decomposed horizontally according to scenarios (Bakir, Boland, Dandurand & Erera, 2020; Beier, Venkatachalam, Corolli & Ntaimo, 2015; Watson, Woodruff & Hart, 2012). One advantage of scenario-wise decomposition methods over the stage-wise ones is that they are easily adaptable to parallel implementation, since the scenario subproblems can be solved independently (Guo, Hackebeil, Ryan, Watson & Woodruff, 2015). In this paper, we focus primarily on scenario-wise decomposition methods.

The scenario-wise decomposition strategies can be further classified into exact and heuristic methods (Ryan, Rajan & Ahmed, 2016). The lower bound information plays an important role in both of these methods. Exact methods such as dual decomposition (Carøe & Schultz, 1999) rely on efficient estimation of lower bounds to iteratively prune regions of the search space that will not contain an optimal solution (Barnett, Watson & Woodruff, 2017; Guo et al., 2015). Heuristic methods such as progressive hedging (PH) (Løkketangen & Woodruff, 1996; Watson & Woodruff, 2011) depend on lower bounds to assess the quality of the solutions relative to the global minimum (Gade et al., 2016). One might settle for a suboptimal solution if the corresponding objective function value is close to a known lower bound, since an optimal solution would not be much better (Hooker, 2008). In this paper, we attempt to develop a method of efficiently computing high-quality lower bounds for stochastic SND.

To obtain lower bounds, one of the most widely used methods is Lagrangian relaxation (Fisher, 2004). The main idea is to relax the original problem by removing the hard constraints and incorporating them into the objective function with some weights (called Lagrange multipliers). For any choice of the Lagrange multipliers, the relaxation of the original problem (called *Lagrange dual function*) is usually easier to solve and its optimal value is a lower bound on the optimal value of the original problem. The problem of finding the best lower bound from the Lagrange dual function with respect to the Lagrange multipliers is referred to as the *Lagrange dual problem* (Boyd & Vandenberghe, 2004). In the context of mixed integer programs, as is the case with stochastic SND, it is well known that the optimal value of the Lagrange dual problem is equal to the optimal value of a corresponding *primal characterization*, which is obtained by convexifying the feasible region of the original primal problem (Carøe & Schultz, 1999; Gade et al., 2016; Nemhauser & Wolsey L.A., 1988). The best lower bound from Lagrangian relaxation can thus be computed by solving either the Lagrange dual problem or the primal problem. The solution methods for the former problem include, among others, subgradient methods (Hooker, 2008; Shor, Kiwiel & Ruszczynski, 1985) and proximal bundle methods (Carøe & Schultz, 1999; Kiwiel, 1990). In this paper, we restrict our study to the solution approaches for the latter problem (i.e. the primal problem).

Despite a linear program, the primal characterization turns out to be hard to solve. The main difficulty stems from the challenge of an explicit polyhedral description of the convex hull of the constraints, which is generally not readily available. Gade et al. (2016) applied the progressive hedging (PH) method to the primal problem. It is proved in their work that the sequences of primal solutions and Lagrange multipliers generated by PH at each iteration converge to the optimal solutions for the primal and Lagrange dual problem, respectively. However, due to the absence of an explicit description of the convex hull in practical applications, the authors used the PH to optimize directly over the constraints of the original problem rather than the convex hull, and the PH was meant to provide feasible Lagrange multipliers for the Lagrange dual function to compute lower bounds. Because of the authors' use of nonconvex feasible region, the convergence guarantee no longer holds. To overcome the lack of convergence guarantees for the above use of PH, Boland et al. (2018) proposed a hybrid algorithm, FW-PH, which combines PH with a Frank-Wolfe (FW) method to solve the primal problem. In the FW-PH, the PH is used as a "wrapper", while the FW is used as a subroutine to construct inner approximation of the convex hull over the course of the PH, thereby elegantly circumventing the difficulty in convexification of the constraint set. Since the FW-PH has desirable theoretical properties of convergence to the optimal Lagrangian dual value, we explore the use of FW-PH to derive lower bounds for stochastic dynamic SND in this paper.

1.3. Scenario bundling-based decomposition

While theoretically appealing, the FW-PH exhibits slow convergence speed when applied to the stochastic SND model. To improve on this, we note that scenario bundling has emerged as an effective way to accelerate PH convergence. The key improvement to the PH is the replacement of scenario-wise decomposition with bundle-wise decomposition, where individual scenarios are combined into bundles and the mathematical model is decomposed by the resulting bundles into scenario-bundle subproblems, rather than by individual scenarios into single-scenario subproblems. Compared to scenario-wise decomposition, the marked reduction in the number of subproblems typically leads to improved computing efficiency, provided that the multi-scenario subproblems are still manageable.

As an attractive algorithmic enhancement, scenario bundling has received growing attention in the past few years. Escudero, Garín, Pérez and Unzueta (2013) introduced scenario bundling into the decomposition of the Lagrangian relaxation of the two-stage stochastic mixed 0–1 problem for obtaining strong lower bounds with less computational effort. Gade et al. (2016) applied scenario bundling to the PH-based lower bound computation for stochastic mixed-integer programs, leading to an improvement in the bound quality and a reduction in the number of PH iterations. In the above two pieces of research, scenarios were grouped randomly into bundles of equal size, but the effects of different scenario bundling results were not compared and no guidance was given on which scenario would be grouped together. Ryan et al. (2020) attempted to group scenarios from an optimization perspective and developed a mixed integer formulation of the scenario bundling problem so as to maximize the Lagrangian relaxation bound. Compared with random grouping, this optimization-driven approach was observed to provide stronger bounds, but at the cost of increased computational time. Crainic, Hewitt and Rei (2014) presented a k-means-based scenario bundling method and several variants thereof for the PH-based meta-heuristic in the context of stochastic SND. Empirical evaluations indicated that the proposed methods performed better compared to grouping scenarios randomly or not grouping scenarios. Jiang, Bai, Wallace, Kendall and Landa-Silva (2021) developed a probabilistic treatment of scenario bundling by introducing membership scores to measure the degree to which a scenario belongs to each bundle. The authors applied Gaussian mixture models to calculate membership scores and showed that Gaussian mixture models-based PH achieved nearly equivalent solution quality in a fraction of the run-time of k-means-based PH. In addition to two-stage stochastic programs, scenario bundling has attracted a lot of interest from other research areas, such as multistage stochastic programs (Bakir et al., 2020; Escudero, Garín & Unzueta, 2016) and chance-constrained programs (Ahmed, Luedtke, Song & Xie, 2017; Deng, Jia, Ahmed, Lee & Shen, 2021). Due to its superior performance reported in the literature in improving pH, the method based on Gaussian mixture models is used in this paper to construct scenario bundles, as we shall see later.

1.4. Motivations and contributions

The slow convergence of FW-PH we observed for the stochastic SND problem, and the efficacy of scenario bundling in improving PH, motivate us to consider integrating scenario bundling into FW-PH, in the hope of developing a theoretically supported and computationally efficient method to compute lower bounds for stochastic SND. The main contribution of this paper is twofold. First, we present a method that can produce a tighter lower bound for stochastic SND within less run-time. In the proposed method, scenario bundling is introduced into scenario-wise decomposition of the stochastic SND model, and the FW-PH algorithm is improved by replacing scenario-wise decomposition with bundle-wise decomposition. The convergence analysis indicates that this improvement not only retains the desirable property of FW-PH about convergence to the optimal Lagrangian dual value, but also produces tighter lower bounds than the case of scenario-wise decomposition. Second, we perform an experimental assessment of the improved FW-PH in the context of stochastic SND. To construct scenario bundles, we introduce the Gaussian mixture models-based scenario bundling method. The numerical results show that the specific choice of the parameter (i.e., the penalty factor) has a marked effect on the performance of the improved FW-PH. We analyze the behavior of the improved FW-PH under various penalty factors and recommend how to choose appropriate parameter settings. Moreover, the results demonstrate that the improved FW-PH is far superior to the basic version when the penalty factor is not too large, providing either a dramatic saving in the run-time required to achieve convergence or a much improved lower bound when terminated due to the time limit.

The remainder of this paper is organized as follows. In Section 2, we give the mathematical model of SSND. In Section 3, we present the Lagrange dual problem of SSND obtained by bundle-wise decomposition and Lagrangian relaxation. To solve the Lagrange dual problem, we develop the FW-PH method based on bundle-wise decomposition in Section 4. In Section 5, we describe the Gaussian mixture models for scenario bundling. We empirically evaluate the improved FW-PH in Section 6. In Section 7, we conclude with a summary of our results and some directions for further study.

2. Problem formulation

2.1. Notations

Networks

Let $\mathcal{N}$represent the set of physical terminals. The transportation service is scheduled over a time horizon of T periods, denoted by $\mathcal{T} := \{0, 1, \ldots, T-1\}$. We assume that the transport movement between every pair of terminals takes one period and use t^- to represent the departure time of a movement with arrival time t. So $t^- = T-1$ if $t$takes the value 0, otherwise $t^- = t-1$. We can obtain the so-called space-time network by duplicating each terminal in every time period. Every pair of terminals in different time periods is connected by an arc, representing the transport movement from one terminal at a certain time period to another terminal at the next period. We use a triplet (i, j, t) to denote a time-space arc from terminal i at time t^- to terminal j at time t. In particular, an arc connecting the identical terminals in different time periods is referred to as a holding arc, which represents the activities of holding vehicles at a terminal for some time. Each arc (i, j, t) has an associated fixed cost c_{ij} incurred by the transportation or holding service. Except for holding arcs, each arc has a resource capacity denoted by u.

Commodities

Through the underlying space-time network, there are some commodities to be shipped. Let $\mathcal{K}$ be the commodity set. Each commodity $k \in \mathcal{K}$ is characterized by its quantity, which is not known in advance but can be represented by a finite number of scenarios, its origin $o(k)$ and the time $\sigma(k)$ when it is available, as well as its destination $s(k)$ and the delivery deadline $\tau(k)$.

Decisions

The decision-making process of stochastic SND has a two-stage structure. In the first stage, the decisions on the inclusion of an arc in the service network are taken before knowledge of the demand. For an arc (i, j, t), this decision is denoted by a binary variable x^t_{ij}. The vector $\mathbf{x}$ stands for such decisions on all of the arcs. After the first-stage decisions are made, a realization of the uncertain demand is observed. We have a limited number of scenarios s for possible future demand, each with a probability of occurrence p_s. These probabilities are non-negative and sum to 1. The collection of these scenarios is denoted by $\mathcal{S}$. The demand of commodity k

Table 1
List of symbols used in the model for stochastic service network design.

Symbol	Meaning
Network parameters	
$\mathcal{N}$	The set of nodes with elements i or j
(i, j)	The arc from node i to node j
u_{ij}	The capacity of an arc (i, j)
c_{ij}	The fixed cost for providing services on an arc (i, j)
$\mathcal{T}$	The set of time periods with elements $t \in \{0, 1, \cdots, T-1\}$
t^-	The departing time period for a vehicle arriving at time period t
Commodity parameters	
$\mathcal{K}$	The set of commodities with elements k
$\sigma(k)$	The time period when commodity k becomes available
$\tau(k)$	The delivery deadline of commodity k
$\mathcal{S}$	The set of scenarios with elements s
p_s	The probability of scenario s
d_k^s	The demand of commodity k in scenario s
$\mathbf{d}_s$	The vector of d_k^s for all types of commodities
λ	The cost of outsourcing one unit of the commodity
Decision variables	
x_{ij}^t	Binary variables indicating whether an arc (i, j) in period t is chosen
$\mathbf{x}$	The vector of x_{ij}^t
y_{ijk}^{st}	The flow of commodity k on arc (i, j) in period t, scenario s
z_k^s	The amount of outsourcing for commodity k in scenario s

in scenario s is represented by d_k^s and the vector $\mathbf{d}_s$ denotes the demand of all commodities in that scenario. In the second stage, the decisions on the flow of commodity are taken based on the demand realization. We use the decision variable y_{ijk}^{st} to represent the flow of commodity k on arc (i, j, t) in scenarios s. In case the demand exceeds the capacity of the service network, some of the orders can be outsourced to external suppliers. Let the decision variable z_k^s denote the amount of outsourcing for commodity k in scenarios s, and λ stands for the cost of outsourcing one unit of the commodity.

The full list of symbols used in the mathematical formulations of SSND is shown in Table 1.

2.2. Model formulation

With the above notations, stochastic SND can be formulated as the following two-stage mixed integer program (Bai et al., 2014), which we denote by [SSND]. This formulation based on scenario representation of demand uncertainty is also referred to as the extensive form (Crainic et al., 2014) or deterministic equivalent (Boland et al., 2018).

Model [SSND]
Stage 1:

$$\min\left\{\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^t + \lambda\sum_{s\in\mathcal{S}} p_s Q(\mathbf{x}, \mathbf{d}_s)\right\} \tag{1}$$

s.t.

$$\sum_{j\in\mathcal{N}} x_{ji}^{t-} = \sum_{j\in\mathcal{N}} x_{ij}^t \forall i \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2}$$

$$x_{ij}^t \in \{0, 1\} \forall i, j \in \mathcal{N}, \forall t \in \mathcal{T} \tag{3}$$

where
Stage 2:

$$Q(\mathbf{x}, \mathbf{d}_s) = \min \sum_{k\in\mathcal{K}} z_k^s \tag{4}$$

s.t.

$$\sum_{k\in\mathcal{K}} y_{ijk}^{st} \le u x_{ij}^t \forall i, j \in \mathcal{N}, \forall t \in \mathcal{T}, \forall i \ne j \tag{5}$$

$$-\sum_{j\in\mathcal{N}} y_{jik}^{st^-} + \sum_{j\in\mathcal{N}} y_{ijk}^{st} = \begin{cases} d_k^s - z_k^s & \text{if } (i,t) \text{ is supply node for } k \\ -d_k^s + z_k^s & \text{if } (i,t) \text{ is demand node for } k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$
$$\forall i \in \mathcal{N}, \forall t \in \mathcal{T}, \forall k \in \mathcal{K}$$

$$y_{ijk}^{s\tau(k)} = 0 \forall i, j \in \mathcal{N}, \forall k \in \mathcal{K} \tag{7}$$

$$y_{ijk}^{st} \ge 0 \forall i, j \in \mathcal{N}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{8}$$

$$z_k^s \ge 0 \forall k \in \mathcal{K} \tag{9}$$

The objective function (1) minimizes the cost of constructing the service network, plus the expected outsourcing cost across all the scenarios. Constraint (2) is the design balance constraint, which ensures that the number of incoming and outgoing vehicles at each node is balanced. Constraint (3) enforces the binary restrictions on design variables. The objective function (4) minimizes the outsourcing cost for a scenario. Constraint (5) ensures that the total flow along each arc does not exceed its capacity. Constraint (6) makes sure that all the commodities are shipped from their origins to destinations and hence all customer demands are met. Constraint (7) ensures that a commodity must not flow past its delivery deadline. Constraints (8) and (9) are used to guarantee the non-negativity of decision variables.

3. Bundle-wise decomposition and lagrangian dual bound

Suppose that the set of scenarios $\mathcal{S}$ is partitioned into finitely many disjoint scenario bundles, indexed by b. In this paper, the method for bundling scenario is based on Gaussian mixture models, which will be discussed at length in Section 5. Let $\mathcal{B}$represent the collection of scenario bundles and hence $b \in \mathcal{B}$. To perform bundle-wise decomposition, we introduce copies x_{ij}^{tb} of the first-stage decision variable x_{ij}^t for each $b \in \mathcal{B}$and write x_{ij}^t as a sum of the form

$$x_{ij}^t = \sum_{b\in\mathcal{B}} p_b x_{ij}^{tb} \forall i, j \in \mathcal{N}, \forall t \in \mathcal{T} \tag{10}$$

where p_b is the probability of scenario bundle b and satisfies $p_b = \sum_{s\in b} p_s$. The vector of x_{ij}^{tb} on all arcs during all time periods for bundle b is denoted by $\mathbf{x}_b$. Note that these variable copies must satisfy the non-anticipativity constraints (NACs). In the case of scenario-wise decomposition, the NACs require that the scenario copies of the first-stage decisions should be identical, because there is no way to tell which scenario would come true at the decision-making point. For bundle-wise decomposition, the scenarios belonging to the same bundle b share a common first-stage decision variable x_{ij}^{tb}. Therefore, the NACs among the scenarios within the same bundle are implicitly implemented and only the NACs among different bundles need to be explicitly considered. We introduce additional variables x_{ij}^{t} and express the NACs as

$$x_{ij}^{tb} = x_{ij}^{t},\ \forall b \in \mathcal{B} \tag{11}$$

Substituting (10) into (1), we obtain the objective function in the form

$$\min \sum_{b\in\mathcal{B}} p_b \left(\sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{tb} + \lambda \sum_{s\in b} \sum_{k\in\mathcal{K}} \frac{p_s}{p_b} z_k^s \right) \tag{12}$$

We see that the objective function decomposes by scenario bundles. Due to the substitution of x_{ij}^{tb} for x_{ij}^{t}, we need to modify the constraints (2), (3) and (5) involving x_{ij}^{t} into

$$\sum_{j\in\mathcal{N}} x_{ji}^{t-b} = \sum_{j\in\mathcal{N}} x_{ij}^{tb} \forall i \in \mathcal{N}, \forall t \in \mathcal{T}, \forall b \in \mathcal{B} \tag{13}$$

$$x_{ij}^{tb} \in \{0,1\} \forall i, j \in \mathcal{N}, \forall t \in \mathcal{T}, \forall b \in \mathcal{B} \tag{14}$$

$$\sum_{k\in\mathcal{K}} y_{ijk}^{st} \le u x_{ij}^{tb} \forall i, j \in \mathcal{N}, i \ne j, \forall t \in \mathcal{T}, \forall s \in b, \forall b \in \mathcal{B} \tag{15}$$

Let $\mathcal{F}$ be the feasible set defined by constraints (13–15) and (6–9) without taking into account (11) yet. Since these constraints are separable with respect to scenario bundles, we shall denote the subset of $\mathcal{F}$pertaining to a certain bundle b by $\mathcal{F}_b$. Among all of the constraints, the newly added NACs (11) are the only ones that tie together different bundles. By Lagrangian relaxation, we can remove the NACs and bring them into the objective function with associated weights $v_{ij}^{tb} \in \mathbb{R}$ (called Lagrange multipliers), obtaining the Lagrange dual function in the form

$$\phi(\mathbf{v}) := \min \left\{ \begin{array}{c} \sum\limits_{b\in\mathcal{B}} \left[p_b \left(\sum\limits_{i\in\mathcal{N}} \sum\limits_{j\in\mathcal{N}} \sum\limits_{t\in\mathcal{T}} c_{ij} x_{ij}^{tb} + \lambda \sum\limits_{s\in b} \sum\limits_{k\in\mathcal{K}} \frac{p_s}{p_b} z_k^s \right) + \sum\limits_{i\in\mathcal{N}} \sum\limits_{j\in\mathcal{N}} \sum\limits_{t\in\mathcal{T}} v_{ij}^{tb} \left(x_{ij}^{tb} - x_{ij}^{t} \right) \right] : \\ (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \mathcal{F}_b, \forall b \in \mathcal{B}, x_{ij}^{t} \in \mathbb{R} \end{array} \right\} \tag{16}$$

where $\mathbf{v}$ is a vector of length $|\mathcal{N}| * |\mathcal{N}| * |\mathcal{T}| * |\mathcal{B}|$ with elements v_{ij}^{tb}. $\mathbf{y}_b$ and $\mathbf{z}_b$ are the vectors of y_{ijk}^{st} and z_k^s for all the scenarios in bundle b, with length $|\mathcal{N}| * |\mathcal{N}| * |\mathcal{K}| * |\mathcal{T}| * |b|$ and $|\mathcal{K}| * |b|$, respectively. It is often convenient to define $w_{ij}^{tb} := \frac{1}{p_b} v_{ij}^{tb}$ and rewrite (16) as

$$\phi(\mathbf{w}) := \min \left\{ \begin{array}{c} \sum\limits_{b\in\mathcal{B}} p_b \left[\sum\limits_{i\in\mathcal{N}} \sum\limits_{j\in\mathcal{N}} \sum\limits_{t\in\mathcal{T}} c_{ij} x_{ij}^{tb} + \lambda \sum\limits_{s\in b} \sum\limits_{k\in\mathcal{K}} \frac{p_s}{p_b} z_k^s + \sum\limits_{i\in\mathcal{N}} \sum\limits_{j\in\mathcal{N}} \sum\limits_{t\in\mathcal{T}} w_{ij}^{tb} \left(x_{ij}^{tb} - x_{ij}^{t} \right) \right] : \\ (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \mathcal{F}_b, \forall b \in \mathcal{B}, x_{ij}^{t} \in \end{array} \right\} \tag{17}$$

where $\mathbf{w}$ is the vector of w_{ij}^{tb} with the same length as $\mathbf{v}$. Since x_{ij}^{t} is unconstrained, in order for the Lagrange dual function $\phi(\mathbf{w})$ to be bounded from below, the multipliers w_{ij}^{tb} are required to satisfy

$$\sum_{b\in\mathcal{B}} p_b w_{ij}^{tb} = 0 \forall i, j \in \mathcal{N}, \forall t \in \mathcal{T} \tag{18}$$

so that the terms related to x_{ij}^{t} in (17) will vanish. Rearranging (17) then gives:

$$\phi(\mathbf{w}) = \min \left\{ \begin{array}{c} \sum\limits_{b\in\mathcal{B}} p_b \left[\sum\limits_{i\in\mathcal{N}} \sum\limits_{j\in\mathcal{N}} \sum\limits_{t\in\mathcal{T}} \left(c_{ij} + w_{ij}^{tb} \right) x_{ij}^{tb} + \lambda \sum\limits_{s\in b} \sum\limits_{k\in\mathcal{K}} \frac{p_s}{p_b} z_k^s \right] : \\ (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \mathcal{F}_b, \forall b \in \mathcal{B} \end{array} \right\} \tag{19}$$

As can be seen from (19), both the objective function and the constraints are now separable with respect to scenario bundles, and hence the Lagrange dual function splits into separate multi-scenario subproblems for each bundle. In fact, scenario-wise decomposition can be viewed as a special case of bundle-wise decomposition where each bundle contains only one scenario, or equivalently, bundle-wise decomposition is a generalization of scenario-wise decomposition. More formally, scenario-wise decomposition corresponds to setting $|\mathcal{B}| = |\mathcal{S}|$, whereas the original problem [SSND] can be obtained by setting $|\mathcal{B}| = 1$.

For any feasible choice of $\mathbf{w}$, the Lagrange dual function $\phi(\mathbf{w})$ in model [SSND-B] gives a lower bound on the original problem [SSND]. We can find the tightest possible lower bound by maximizing (19) over $\mathbf{w}$, leading to our bundle-wise decomposition model:

Model [SSND-B]

$$\varsigma_B^{LD} := \max_{\mathbf{w}} \left\{ \phi(\mathbf{w}) : \sum_{b\in\mathcal{B}} p_b w_{ij}^{tb} = 0 \right\} \tag{20}$$

This is known as the Lagrange dual problem (Boyd & Vandenberghe, 2004) and the vector $\mathbf{w}$ is referred to as dual variables. Moreover, in comparison with scenario-wise decomposition, bundle-wise decomposition enables the Lagrange dual problem to provide an equal or tighter lower bound (Escudero et al., 2013). We summarize these results in Proposition 1.

Proposition 1. $\varsigma_B^{LD}(|\mathcal{B}| = |\mathcal{S}|) \le \varsigma_B^{LD}(1 < |\mathcal{B}| < |\mathcal{S}|) \le \varsigma_B^{LD}(|\mathcal{B}| = 1) = \varsigma^{SSND}$, *where* $\varsigma_B^{LD}(.)$ *is the optimal Lagrange dual bound under a given number of scenario bundles in brackets.*

Proof. The three cases differ from each other only in the number of NACs that are relaxed in $\phi(\mathbf{w})$. When $|\mathcal{B}| = |\mathcal{S}|$, there are$|\mathcal{S}|$NACs relaxed. That number decreases to$|\mathcal{B}|$and 0 for the case of $1 < |\mathcal{B}| < |\mathcal{S}|$ and $|\mathcal{B}| = 1$, respectively. So $\phi(\mathbf{w})$ for the case of $|\mathcal{B}| = |\mathcal{S}|$ is a relaxation of that for the case of $1 < |\mathcal{B}| < |\mathcal{S}|$ or$|\mathcal{B}| = 1$, and $\phi(\mathbf{w})$ for the case of$1 < |\mathcal{B}| < |\mathcal{S}|$ is a relaxation of that for the case of $|\mathcal{B}| = 1$. From this, the result follows.

There exist several methods for directly solving the dual problem (20), such as the subgradient method. The following proposition opens up the possibility of computing the Lagrange dual bound from the primal perspective.

Proposition 2. *The optimal value* ς^{LD} *of the Lagrange dual problem (20) equals the optimal value of a linear program arising from convexification of the original problem [SSND], giving*

$$\varsigma_B^{LD} = \min \left\{ \begin{array}{c} \sum\limits_{b\in\mathcal{B}} p_b \left(\sum\limits_{i\in\mathcal{N}} \sum\limits_{j\in\mathcal{N}} \sum\limits_{t\in\mathcal{T}} c_{ij} x_{ij}^{tb} + \lambda \sum\limits_{s\in b} \sum\limits_{k\in\mathcal{K}} \frac{p_s}{p_b} z_k^s \right) : \\ (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \operatorname{conv}(\mathcal{F}_b), x_{ij}^{tb} = x_{ij}^{t}, \ \forall b \in \mathcal{B} \end{array} \right\} \tag{21}$$

where $\operatorname{conv}(\mathcal{F}_b)$ denotes the convex hull of $\mathcal{F}_b$ for each $b \in \mathcal{B}$.

Proof. When $|\mathcal{B}| = |\mathcal{S}|$, the bundle-wise decomposition reduces to the scenario-wise decomposition. The result then follows from the proof of Proposition 2 in (Carøe & Schultz, 1999).

Compared to the case of $|\mathcal{B}| = |\mathcal{S}|$, the case of $1 < |\mathcal{B}| < |\mathcal{S}|$ differs only in the sizes of some vectors and the values of some coefficients. For example, the size of the vector $\mathbf{z}_b$ for a bundle b changes from $|\mathcal{K}|$ to $|b| * |\mathcal{K}|$. The coefficient for an element z_k^s of the vector $\mathbf{z}_b$ changes from λ to $\lambda * (p_s/p_b)$. By using vector representations, all the formula for the case of $1 < |\mathcal{B}| < |\mathcal{S}|$ can thus be converted to the same form as those for the case of $|\mathcal{B}| = |\mathcal{S}|$. Therefore, the result follows in the case of $1 < |\mathcal{B}| < |\mathcal{S}|$from the proof for the case of $|\mathcal{B}| = |\mathcal{S}|$.

From Proposition 2, we know that the optimal Lagrange dual bound can also be derived by solving a certain primal problem, whose feasible region is the convex hull of the constraints of the original problem. In this work, we explore the Lagrange dual bound computation from the primal perspective instead of the dual angle and focus our efforts on methods for solving the problem (21).

4. FW-PH based on bundle-wise decomposition

Although there exist many efficient methods for linear programs, the primal problem (21) turns out hard to solve. The principal difficulty lies in the fact that an explicit polyhedral description of $\mathrm{conv}(\mathcal{F}_b)$ is generally not available in practice. The recently developed algorithm FW-PH (Boland et al., 2018), which integrates a Frank-Wolfe method with the Progressive Hedging algorithm, is a theoretically desirable method to resolve this difficulty. Essentially, the FW-PH algorithm is a scenario-wise decomposition method, which works by breaking down the potentially difficult problem into manageable single-scenario subproblems. In this section, we improve the FW-PH algorithm by substituting bundle-wise decomposition, which divides the problem according to scenario bundles instead of individual scenarios into multi-scenario subproblems. It is shown in later sections that this improvement significantly improves the computational performance without spoiling the desirable property of optimal convergence.

In the FW-PH algorithm, PH plays the role of a ‘wrapper’, where sequences of subproblem solutions and dual variable values are generated during the iterative process, while FW serves as a subroutine to solve the subproblems in each PH iteration.

The standard PH begins with an augmented Lagrangian representations for problem (21), so that it breaks down into separate subproblems for each scenario. The augmented Lagrangian method is quite similar to the Lagrangian relaxation above, but adds extra squared penalty terms in the objective function. When scenario bundles are considered instead of individual scenarios, applying the augmented Lagrangian method to the NACs in (21) yields

$$\min\left\{\sum_{b\in\mathcal{B}} p_b L_b^{\rho}(\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b,\mathbf{x},\mathbf{w}_b) : (\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b)\in\mathrm{conv}(\mathcal{F}_b),\ \forall b\in\mathcal{B}\right\} \tag{22}$$

where

$$\begin{aligned}L_b^{\rho}(\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b,\mathbf{x},\mathbf{w}_b) = &\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^{tb} + \lambda\sum_{s\in b}\sum_{k\in\mathcal{K}}\frac{p_s}{p_b}z_k^s\\ &+\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} w_{ij}^{tb}\left(x_{ij}^{tb}-x_{ij}^{t}\right)+\frac{\rho}{2}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}}\left(x_{ij}^{tb}-x_{ij}^{t}\right)^2\end{aligned} \tag{23}$$

and $\rho > 0$ is the penalty factor with a fixed value. $\mathbf{x}$ is the vector of x_{ij}^t with length $|\mathcal{N}| * |\mathcal{N}| * |\mathcal{T}|$. $\mathbf{w}_b$ is the vector of w_{ij}^{tb} on all arcs during all the time periods for bundle$b$. Scenario bundles are obtained with a method based on Gaussian mixture models, as we shall see in Section 5.

For a given $\mathbf{x}$, the problem (22) decomposes into separate subproblems for each scenario bundle. Solving these subproblems gives admissible solutions, which do not necessarily satisfy the NACs. Nevertheless, only if these subproblem solutions are also implementable can they be a feasible solution to the problem (21). Implementable solutions are the ones that fulfill the NACs. To addresses this issue, the PH iteratively aggregates all subproblem solutions into an implementable solution and encourages every subproblem solution to move towards this implementable solution until all subproblem solutions agree. Specifically, given values $(\mathbf{x})^{(r-1)}$ and $(\mathbf{w}_b)^{(r)}$ obtained in the $(r-1)$-th iteration, the r-th iteration of the PH involves three key steps stated as follows. Here a variable with a parentheses-enclosed loop index (r) in the superscript represents its value in the r-th iteration.

Step 1 Solve for each scenario bundle b the multi-scenario subproblem of the form

$$\left((\mathbf{x}_b)^{(r)},(\mathbf{y}_b)^{(r)},(\mathbf{z}_b)^{(r)}\right)\in\arg\min\left\{\begin{array}{c}L_b^{\rho}\left(\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b,(\mathbf{x})^{(r-1)},(\mathbf{w}_b)^{(r)}\right):\\ (\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b)\in\mathrm{conv}(\mathcal{F}_b)\end{array}\right\} \tag{24}$$

Step 2 Compute an implementable solution for the first-stage decision variables according to

$$(\mathbf{x})^{(r)}\leftarrow\sum_{b\in\mathcal{B}}p_b(\mathbf{x}_b)^{(r)} \tag{25}$$

Step 3 Update the dual variables for each scenario bundle b according to

$$(\mathbf{w}_b)^{(r+1)}\leftarrow(\mathbf{w}_b)^{(r)}+\rho\left[(\mathbf{x}_b)^{(r)}-(\mathbf{x})^{(r)}\right] \tag{26}$$

The iterative procedure of the PH generates a sequence of implementable solutions and dual variable values that provably converges to the optimal solutions to (21) and (20), respectively (Gade et al., 2016).

The trouble now is that an explicit polyhedral description of $\mathrm{conv}(\mathcal{F}_b)$ is not known, so that the multi-scenario subproblem (24) cannot be tackled directly. Fortunately, the subproblem (24) has two structural characteristics: (1) minimizing a linear function over $\mathrm{conv}(\mathcal{F}_b)$ is much simpler than minimizing $L_b^{\rho}(.)$ over $\mathrm{conv}(\mathcal{F}_b)$ because we can replace $\mathrm{conv}(\mathcal{F}_b)$ with F_b; (2) minimizing $L_b^{\rho}(.)$ over the convex hull of a relatively small number of extreme points of $\mathrm{conv}(\mathcal{F}_b)$is much simpler than minimizing $L_b^{\rho}(.)$ over $\mathrm{conv}(\mathcal{F}_b)$. Under these two conditions, the *simplicial decomposition* method, which is an extension of the classical Frank-Wolfe method, is well suited for solving (24) (Bertsekas, 2015).

The simplicial decomposition method overcomes the lack of an explicit polyhedral description for $\mathrm{conv}(\mathcal{F}_b)$ in (24) by constructing inner approximations of $\mathrm{conv}(\mathcal{F}_b)$. That is, the set $\mathrm{conv}(\mathcal{F}_b)$ is approximated by the convex hull of some extreme points of $\mathrm{conv}(\mathcal{F}_b)$ and a new extreme point of $\mathrm{conv}(\mathcal{F}_b)$is added at each iteration to expand this convex hull. Minimizing $L_b^{\rho}(.)$ over this convex hull is much easier since the decision variables can now be expressed as a convex combination of these extreme points. For the problem (24), given the solutions $((\mathbf{x}_b)^{(r-1)},(\mathbf{y}_b)^{(r-1)},(\mathbf{z}_b)^{(r-1)})$ and the finite set $(\mathcal{V}_b)^{(r-1)}$ of points obtained in the $(r-1)$-th iteration, the r-th iteration of the simplicial decomposition method involves three key steps stated as follows.

Step 1 Generate $(\hat{\mathbf{x}}_b,\hat{\mathbf{y}}_b,\hat{\mathbf{z}}_b)$ as an extreme point of $\mathrm{conv}(\mathcal{F}_b)$ by solving a modification of the problem (24), in which the quadratic objective $L_b^{\rho}(.)$ is linearized using a first-order approximation. The mathematical description of the modified problem can be found in the lines 14 and 15 of Algorithm 1.

Step 2 Expand the set of extreme points according to $(\mathcal{V}_b)^{(r)}\leftarrow(\mathcal{V}_b)^{(r-1)}\cup\{(\hat{\mathbf{x}}_b,\hat{\mathbf{y}}_b,\hat{\mathbf{z}}_b)\}$.

Step 3 Generate solutions $((\mathbf{x}_b)^{(r)},(\mathbf{y}_b)^{(r)},(\mathbf{z}_b)^{(r)})$ by solving a modification of the problem (24), in which the feasible set $\mathrm{conv}(\mathcal{F}_b)$ is replaced by $\mathrm{conv}((\mathcal{V}_b)^{(r)})$. The mathematical description of the modified problem can be found in the line 17 of Algorithm 1.

The solutions generated in Step 3 of the simplicial decomposition method provably converge to the optimal solutions to the convex problem (24) in a finite number of iterations (Bertsekas, 2015).

The FW-PH algorithm integrates the simplicial decomposition method into PH by substituting Step 1 of each PH iteration with one iteration of the simplicial decomposition method. A complete description of the FW-PH algorithm based on bundle-wise decomposition is given in Algorithm 1. Here, $\mathcal{V}_b$ is a finite set of points $(\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b)$. Initialization of the set $\mathcal{V}_b$ in the lines 4–7 is crucially important for convergence of the FW-PH algorithm. The goal is to

Algorithm 1
FW-PH based on bundle-wise decomposition for the problem (21).

```
// Initialization in the lines 1–11.
1: $r \leftarrow 0$, $(w_{ij}^{tb})^{(0)} \leftarrow 0$ for all $i, j \in \mathcal{N}, t \in \mathcal{T}, b \in \mathcal{B}$
2: for $b \in \mathcal{B}$ do
    3: $((\mathbf{x}_b)^{(0)}, (\mathbf{y}_b)^{(0)}, (\mathbf{z}_b)^{(0)}) \in argmin\{\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^{tb} + \lambda\sum_{s\in b}\sum_{k\in\mathcal{K}}\frac{p_s}{p_b}z_k^s : (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \mathcal{F}_b\}$
    4: $(\mathcal{V}_b)^{(0)} \leftarrow \{((\mathbf{x}_b)^{(0)}, (\mathbf{y}_b)^{(0)}, (\mathbf{z}_b)^{(0)})\}$
    5: if $b$ is not a certain pre-specified $\tilde{b} \in \mathcal{B}$ then
        6: $((\bar{\mathbf{y}}_b)^{(0)}, (\bar{\mathbf{z}}_b)^{(0)}) \in argmin\{\lambda\sum_{s\in b}\sum_{k\in\mathcal{K}}\frac{p_s}{p_b}z_k^s : ((\mathbf{x}_{\tilde{b}})^{(0)}, \mathbf{y}_b, \mathbf{z}_b) \in \mathcal{F}_b\}$
        7: $(\mathcal{V}_b)^{(0)} \leftarrow (\mathcal{V}_b)^{(0)} \cup \{((\mathbf{x}_{\tilde{b}})^{(0)}, (\bar{\mathbf{y}}_b)^{(0)}, (\bar{\mathbf{z}}_b)^{(0)})\}$
    8: end if
9: end for
10: $(\mathbf{x})^{(0)} \leftarrow \sum_{b\in\mathcal{B}} p_b(\mathbf{x}_b)^{(0)}$
11: $(\mathbf{w}_b)^{(1)} \leftarrow (\mathbf{w}_b)^{(0)} + \rho[(\mathbf{x}_b)^{(0)} - (\mathbf{x})^{(0)}]$
// Iteration updates in the line 12.
12: $r \leftarrow r + 1$
13: for $b \in \mathcal{B}$ do
    // Steps 1, 2 and 3 of simplicial decomposition in the lines 14–17.
    14: $\hat{w}_{ij}^{tb} \leftarrow (w_{ij}^{tb})^{(r)} + \rho((x_{ij}^{tb})^{(r-1)} - (\mathbf{x}_{ij}^{t})^{(r-1)})$ for all $i, j \in \mathcal{N}, t \in \mathcal{T}, b \in \mathcal{B}$
    15: $(\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b) \in argmin\{\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}}(c_{ij} + \hat{w}_{ij}^{tb})x_{ij}^{tb} + \lambda\sum_{s\in b}\sum_{k\in\mathcal{K}}\frac{p_s}{p_b}z_k^s : (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \mathcal{F}_b\}$
    16: $(\mathcal{V}_b)^{(r)} \leftarrow (\mathcal{V}_b)^{(r-1)} \cup \{(\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b)\}$
    17: $((\mathbf{x}_b)^{(r)}, (\mathbf{y}_b)^{(r)}, (\mathbf{z}_b)^{(r)}) \in argmin\{L_b^{\rho}(\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b, (\mathbf{x})^{(r-1)}, (\mathbf{w}_b)^{(r)}) : (\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b) \in \mathrm{conv}((\mathcal{V}_b)^{(r)})\}$
    // Lower bound computation in the line 18.
    18: $(\phi_b)^{(r)} \leftarrow \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}}(c_{ij} + \hat{w}_{ij}^{tb})\hat{x}_{ij}^{tb} + \lambda\sum_{s\in b}\sum_{k\in\mathcal{K}}\frac{p_s}{p_b}\hat{z}_k^s$
19: end for
// Steps 2 and 3 of the pH in the lines 20–21.
20: $(\mathbf{x})^{(r)} \leftarrow \sum_{b\in\mathcal{B}} p_b(\mathbf{x}_b)^{(r)}$
21: $(\mathbf{w}_b)^{(r+1)} \leftarrow (\mathbf{w}_b)^{(r)} + \rho[(\mathbf{x}_b)^{(r)} - (\mathbf{x})^{(r)}]$
// Lower bound computation in the line 22.
22: $(\phi)^{(r)} \leftarrow \sum_{b\in\mathcal{B}} p_b(\phi_b)^{(r)}$
23: If the termination criteria are met, then stop. Otherwise, go to line 12.
```

ensure that the initial set $(\mathcal{V}_b)^{(0)}$ for each bundle has a common $\mathbf{x}_b$. This is accomplished by arbitrarily choosing one bundle $\tilde{b} \in \mathcal{B}$ and adding an extra point to the initial set for every bundle except $\tilde{b}$. In all the extra points, the element $\mathbf{x}_b$ is set equal to the initial solution $(\mathbf{x}_{\tilde{b}})^{(0)}$ for bundle $\tilde{b}$, and the elements $\mathbf{y}_b$ and $\mathbf{z}_b$ are obtained by solving the problem in line 6 of Algorithm 1, with the first-stage decision variables $\mathbf{x}_b$ fixed at the initial solutions $(\mathbf{x}_{\tilde{b}})^{(0)}$ for bundle $\tilde{b}$. The termination criteria are based mainly on convergence of the $\mathbf{x}_b^{(r)}$ to a common $\mathbf{x}^{(r)}$, for which we require the Euclidean distance $\sqrt{\sum_{b\in\mathcal{B}} p_b\|\mathbf{x}_b^{(r)} - \mathbf{x}^{(r)}\|^2}$ to be less than some prespecified tolerance. If needed, Algorithm 1 may also terminate when a predetermined limit on the run-time or the number of iterations is exceeded.

We now extend the convergence analysis of the FW-PH algorithm performed for the case of scenario-wise decompositions in (Boland et al., 2018) to the more general case of bundle-wise decomposition. The following proposition shows that the value $(\phi)^{(r)}$ calculated at each iteration of the bundle-wise decomposition-based FW-PH provides a lower bound on the optimal objective value ς^{LD} of the problem (21). Moreover, the sequence of lower bound $(\phi)^{(r)}$ converges to ς^{LD} in the limit.

Proposition 3. *Suppose Algorithm 1 is applied to the problem (21). For the value $(\phi)^{(r)}$ calculated at line 22 at iteration r of Algorithm 1, the following hold:*

(a) $(\phi)^{(r)} \leq \varsigma_B^{LD}$ for any $r \geq 1$.
(b) $\lim_{r\to\infty}(\phi)^{(r)} = \varsigma_B^{LD}$.

Proof. When $|\mathcal{B}| = |\mathcal{S}|$, the bundle-wise decomposition reduces to the scenario-wise decomposition. By Proposition 3 and 4 in (Boland et al., 2018), the results (a) and (b) hold in this case. Note that the assumption of Proposition 3 in (Boland et al., 2018) that $\sum_{b\in\mathcal{B}} p_b(w_{ij}^{tb})^{(0)} = 0$ is guaranteed by the initialization $(w_{ij}^{tb})^{(0)} \leftarrow 0$ in the line 1 of Algorithm 1. Note also that the precondition of Proposition 4 in (Boland et al., 2018) that the initial set $(\mathcal{V}_b)^{(0)}$ for each bundle has a common $\mathbf{x}_b$ is satisfied by the initialization of the set $\mathcal{V}_b$ in the lines 4–7 of Algorithm 1.

When $1 < |\mathcal{B}| < |\mathcal{S}|$, as we have explained in the proof of Proposition 2, all the formulas for the case of $1 < |\mathcal{B}| < |\mathcal{S}|$ can be converted to the same form as those for the case of $|\mathcal{B}| = |\mathcal{S}|$ by vector representations. Therefore, the results (a) and (b) follow in the case of $1 < |\mathcal{B}| < |\mathcal{S}|$ from the proof for the case of $|\mathcal{B}| = |\mathcal{S}|$.

5. Scenario bundling based on Gaussian mixture models

In this section, we consider methods for grouping individual scenarios into bundles. As reported in the literature, Gaussian mixture models (GMMs) outperform other prevalent methods for scenario bundling, such as randomly grouping scenarios and *k*-means, in improving the computing efficiency of the PH algorithm. In view of the close connection between the PH algorithm and the FW-PH algorithm, we employ GMMs to construct scenario bundles for the FW-PH algorithm in this paper. Here we give only a brief description of GMMs in the context of scenario bundling. For more information on GMMs, see (Bishop, 2006) and (Jiang et al., 2021).

Before scenario bundling, we need to specify the scenario features based on which scenario similarity is measured, as well as the number of desired bundles. Because of its superior performance reported in (Crainic et al., 2014), we choose as the feature of a scenario s the $|\mathcal{K}|$-dimensional vector $\mathbf{d}_s$, which represents the demand volumes of all the commodities in scenario s. As in (Jiang et al., 2021), the number of bundles is determined by rounding the square root $\sqrt{|\mathcal{S}|}$ to the nearest integer, so as to achieve a proper balance between the number and the size of the associated subproblems.

Suppose we wish to partition the data set comprising $|\mathcal{S}|$ data points $\mathbf{d}_s$ into $|\mathcal{B}|$ disjoint bundles b. In order to use GMMs for data grouping, we first need to fit a GMM to the given data set. A GMM is a linear combination of $|\mathcal{B}|$ multivariate Gaussian densities, given by

$$\sum_{b\in\mathcal{B}} \varphi_b G(\mathbf{d}_s|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) \tag{27}$$

The parameters φ_b are called mixing coefficients, which satisfy $0 \leq \varphi_b \leq 1$ and $\sum_{b\in\mathcal{B}} \varphi_b = 1$. Each Gaussian density $G(\mathbf{d}_s|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$ is called a component of the mixture and has its own $|\mathcal{K}|$-dimensional mean $\boldsymbol{\mu}_b$ and $|\mathcal{K}|$-by-$|\mathcal{K}|$ covariance matrix $\boldsymbol{\Sigma}_b$. For a $|\mathcal{K}|$-dimensional data point $\mathbf{d}_s$, the multivariate Gaussian density takes the form

$$G(\mathbf{d}_s|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) = \frac{1}{(2\pi)^{|\mathcal{K}|/2}}\frac{1}{(\det(\boldsymbol{\Sigma}_b))^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{d}_s - \boldsymbol{\mu}_b)^{\top}\boldsymbol{\Sigma}_b^{-1}(\mathbf{d}_s - \boldsymbol{\mu}_b)\right\} \tag{28}$$

where $\det(\boldsymbol{\Sigma}_b)$ denotes the determinant of $\boldsymbol{\Sigma}_b$.

Given $|\mathcal{S}|$ data points, the values of the parameters $\boldsymbol{\varphi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, which denote the set of φ_b, $\boldsymbol{\mu}_b$ and $\boldsymbol{\Sigma}_b$ respectively, are determined using the maximum likelihood method. If the data points are assumed to be drawn independently from this mixture of Gaussian densities, the likelihood function of the $|\mathcal{S}|$ data points is given by

Algorithm 2
Scenario bundling based on Gaussian mixture models.

```
// Initialization in the lines 1–2.
1: $r \leftarrow 0$
2: Initialize $\boldsymbol{\varphi}^{(0)}$, $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\Sigma}^{(0)}$.
// Iteration updates in the line 3.
3: $r \leftarrow r+1$
// E-step of the EM algorithm in the lines 4–8.
4: for $s \in \mathcal{S}$ do
5:   for $b \in \mathcal{B}$ do
6:     $(\delta_{sb})^{(r)} \leftarrow \frac{(\varphi_b)^{(r-1)} G(\mathbf{d}_s | (\boldsymbol{\mu}_b)^{(r-1)}, (\boldsymbol{\Sigma}_b)^{(r-1)})}{\sum_{b \in \mathcal{B}} (\varphi_b)^{(r-1)} G(\mathbf{d}_s | (\boldsymbol{\mu}_b)^{(r-1)}, (\boldsymbol{\Sigma}_b)^{(r-1)})}$
7:   end for
8: end for
// M-step of the EM algorithm in the lines 9–13.
9: for $b \in \mathcal{B}$ do
10:   $(\varphi_b)^{(r)} \leftarrow \frac{\sum_{s \in \mathcal{S}} (\delta_{sb})^{(r)}}{|\mathcal{S}|}$
11:   $(\boldsymbol{\mu}_b)^{(r)} \leftarrow \frac{\sum_{s \in \mathcal{S}} (\delta_{sb})^{(r)} \mathbf{d}_s}{\sum_{s \in \mathcal{S}} (\delta_{sb})^{(r)}}$
12:   $(\boldsymbol{\Sigma}_b)^{(r)} \leftarrow \frac{\sum_{s \in \mathcal{S}} (\delta_{sb})^{(r)} (\mathbf{d}_s - (\boldsymbol{\mu}_b)^{(r)})(\mathbf{d}_s - (\boldsymbol{\mu}_b)^{(r)})^{\top}}{\sum_{s \in \mathcal{S}} (\delta_{sb})^{(r)}}$
13: end for
14: if $\ell(\boldsymbol{\varphi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ does not converge, then
      Go to line 3.
15: end if
// Scenario bundling in the lines 15–17.
16: for $s \in \mathcal{S}$ do
17:   Assign $s$ to the bundle that satisfies $\arg\max_b (\delta_{sb})^{(r)}$.
18: end for
```

$$\prod_{s \in \mathcal{S}} \left(\sum_{b \in \mathcal{B}} \varphi_b G(\mathbf{d}_s | \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) \right) \tag{29}$$

The maximum likelihood method estimates the parameter values by maximizing the log of the likelihood function (29) in the form

$$\ell(\boldsymbol{\varphi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{s \in \mathcal{S}} \log \left\{ \sum_{b \in \mathcal{B}} \varphi_b G(\mathbf{d}_s | \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) \right\} \tag{30}$$

so that the $|\mathcal{S}|$ data points are most probable. An elegant method for maximizing (30) is the iterative Expectation-Maximization (EM) algorithm (Bishop, 2006). Given values $\boldsymbol{\varphi}^{(r-1)}$, $\boldsymbol{\mu}^{(r-1)}$ and $\boldsymbol{\Sigma}^{(r-1)}$ obtained in the $(r-1)$-th iteration, the r-th iteration of the EM algorithm involves two key steps stated as follows.

E Step For every possible combination of datapoints and components, compute the posterior probability $(\delta_{sb})^{(r)}$ that the datapoint $\mathbf{d}_s$ is from the component $G(\mathbf{d}_s | \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$. The formula of the posterior probability can be found in the line 6 of Algorithm 2.

M Step Update the component means $\boldsymbol{\varphi}^{(r)}$, covariance matrices $\boldsymbol{\mu}^{(r)}$ and mixing coefficients $\boldsymbol{\Sigma}^{(r)}$using the posterior probabilities $(\delta_{sb})^{(r)}$. The update rules can be found in the lines 10–12 of Algorithm 2.

The EM algorithm iterates over these steps until convergence of the log likelihood $\ell(\boldsymbol{\varphi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. A fitted GMM can then be obtained by substituting the parameter values at convergence into (27).

With a fitted GMM, grouping data points is fairly straightforward because the posterior probabilities for a data point indicate the probability of it belonging to each component. Since we consider disjoint bundles in this paper, we assign each data point to the Gaussian component yielding the highest posterior probability. A complete description of scenario bundling based on Gaussian mixture models is given in Algorithm 2. Regarding the initialization in Algorithm 2, every initial mixing coefficient is set to be the uniform probability $1/|\mathcal{B}|$. We select $|\mathcal{B}|$ data points $\mathbf{d}_s$ as the initial means using the k-means++ algorithm (Arthur &, Vassilvitskii, 2007). To allow for different covariance structures among scenario bundles, we chose independent full covariance matrices for every Gaussian component. The initial covariance matrices for all Gaussian components are diagonal, where an element on the diagonal is the variance of an $|\mathcal{S}|$-dimensional vector formed by d_s^k across different scenarios.

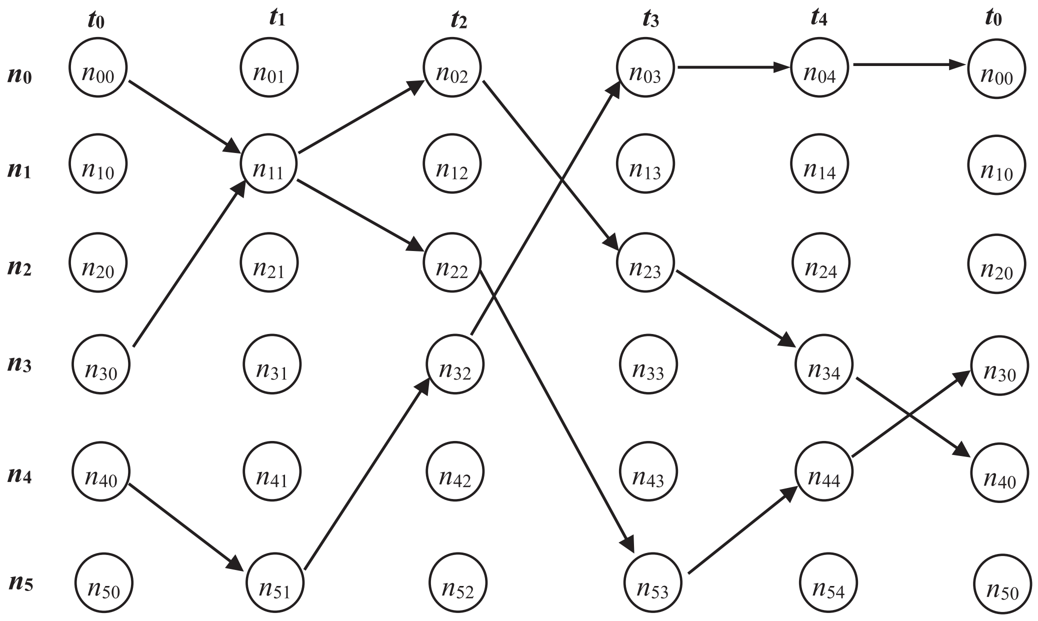
6. Computational experiments

6.1. Test instances and experimental settings

In this section, we examine the impact of scenario bundling on the convergence speed of FW-pH for stochastic service network design problems. Two sets of test instances are included in this study. The first set is taken directly from Jiang et al. (2021), and an adaption thereof forms the second set. For most instances in the first set, neither FW-pH nor our proposed bundle-based method, GMM-FW-pH, can converge within the three-hour time limit, whereas the final convergence is observed for many instances in the second set. With these two sets, we are therefore able to see if the effects of scenario bundling generalize in different conditions.

Network configuration
Both sets of test instances use a fully connected network consisting of 6 terminals, but they have different types of cost matrices. To establish the cost matrices, we labeled these terminals consecutively from 1 to 6 and partitioned them into three groups, namely {{1,2}, {3,4}, {5,6}}. We attached a relatively higher fixed cost to the arcs connecting two between-group terminals than the arcs that link two within-group terminals. While the between-group costs are the same in the first set, we specified different values for the between-group costs in the second set. For consistency with the names in (Jiang et al., 2021), we refer to these matrices, which are shown in Table 2, as Type 2 and Type 3 cost matrix respectively. In our experiments, all the services are scheduled over a planning horizon of 5 time periods. Since each terminal is repeated in every time period, the resulting space-time network comprises 30 ($= 6 \times 5$) nodes and 180 ($= 6 \times 6$*5) arcs.

Scenario representation of demand uncertainty
In each test instance, there are 12 commodities to be conveyed, but the volumes of these commodities are uncertain. Similar to the procedure in Lium et al. (2009) and Bai et al. (2014), demand uncertainty in our experiment is described by marginal distributions of each commodity, along with correlations between all pairs of commodities. We specified the same probability distribution for all the commodities, which is given by the symmetric triangular distribution Tri(4, 12, 8) where 3 parameters are min, max, and mode, respectively. We considered three different correlation settings: (1) every pair of commodities are positively correlated, (2) a mix of positively and negatively correlated commodities, (3) all the commodities are uncorrelated. For later use in the instance identifiers, the three correlation settings are denoted in turn by uppercase letters C, M and U. For each correlation setting, we employed the open-source scenario-generating tool from Høyland et al. (2003) to construct two scenario trees, one containing 20 scenarios and the other including 40 scenarios. The scenario-generating tool is based on the moment-matching algorithm, where scenarios are generated to match the first four marginal moments and the correlation matrix of the given uncertain demand (Høyland & Wallace, 2001; Høyland et al., 2003; Kaut & Wallace, 2007). For the two scenario trees, in-sample stability (see (Kaut & Wallace, 2007; Lium et al., 2009) for details) was verified, with the difference between the highest and lowest objective function values across scenario trees less than 1.5%. Other parameters can be found in Table 2.



t_0
t_1
t_2
t_3
t_4
t_0
n_0
n_1
n_2
n_3
n_4
n_5
n_{00} n_{01} n_{02} n_{03} n_{04} n_{00}
n_{10} n_{11} n_{12} n_{13} n_{14} n_{10}
n_{20} n_{21} n_{22} n_{23} n_{24} n_{20}
n_{30} n_{31} n_{32} n_{33} n_{34} n_{30}
n_{40} n_{41} n_{42} n_{43} n_{44} n_{40}
n_{50} n_{51} n_{52} n_{53} n_{54} n_{50}

Fig. 1. The network configuration of the optimal solution for U4035 in Table 3.

Table 2
Problem specifications for the test instances. In each cost matrix, the main diagonal element in the i-th row represents the fixed cost of holding service at terminal i, whereas the off-diagonal element in the i-th row and j-th column represents the fixed cost of transportation service from terminal i to j.

Parameters	Values	Type 2 cost matrix						Type 3 cost matrix					
$\lvert\mathcal{N}\rvert$	6	50	100	250	250	250	250	50	100	250	250	550	550
T	5	100	50	250	250	250	250	100	50	250	250	550	550
$\lvert\mathcal{K}\rvert$	12	250	250	50	100	250	250	250	250	50	100	250	250
u	20	250	250	100	50	250	250	250	250	100	50	250	250
λ	100(250)*	250	250	250	250	50	100	550	550	250	250	50	100
#scenarios	20(40)**	250	250	250	250	100	50	550	550	250	250	100	50

* : For the instances with type 2 and 3 cost matrices, the outsourcing costs are 100 and 250, respectively.
** : Each scenario has an equal probability of occurrence, i.e. 1/20 and 1/40 for the cases of 20 and 40 scenarios, respectively.

Table 3
Performance results of FW-PH and GMM-FW-PH under various penalty values on the test instance U4035, for which both algorithms can converge within 3 h. The letter “T” in the column “Run-time” indicates that the algorithm hit the time limit before convergence.

CPLEX Direct Solving			Expected Cost 2657.89		Run-time(s) 729	
Theoretical bound Via Decomposition					2657.89	
Penalty factor	Optimality Gap (%)		# Iterations		Run-time (s)	
	FW-PH	GMM-FW-PH	FW-PH	GMM-FW-PH	FW-PH	GMM-FW-PH
0.2	1.17	0.15	165	67	T	T
0.5	0.82	0.00	146	53	T	9224
1	0.51	0.00	130	30	T	5487
2	0.27	0.00	117	17	T	3083
5	0.06	0.01	102	9	T	1619
10	0.00	0.00	71	10	7703	1783
25	0.00	0.00	48	15	5168	2783
50	0.00	0.00	37	14	3651	2367
100	0.00	0.00	37	18	3805	3106
200	0.00	0.00	48	22	5220	3361
400	0.00	0.00	50	31	5283	4786
800	0.00	0.00	62	41	7190	7616

Combining the three correlation settings and two different numbers of scenarios, together with the two types of cost matrices, we obtained a total of 12 ($= 3 \times 2^*2$) test instances. For convenience, we supplied a unique identifier for each instance. For example, the test instance characterized by positive correlation, 40 scenarios, Type 3 cost matrix and commodity set 5 is designated ‘C4035’. For an instance with 20 (or 40) scenarios, its deterministic equivalent contains 180 integer decision variables, 43,440 (or 86,880) continuous decision variables and 19,350 (or 38,670) constraints. While the problem size is far from large by the standards of real-world applications, more than half of the test instances are large enough to create serious difficulties for FW-PH to converge within three hours. Since our primary objective is to see if scenario bundling helps FW-PH achieve faster convergence, or produce a better lower bound at termination, these instances serve adequately as the testbed for our purposes.

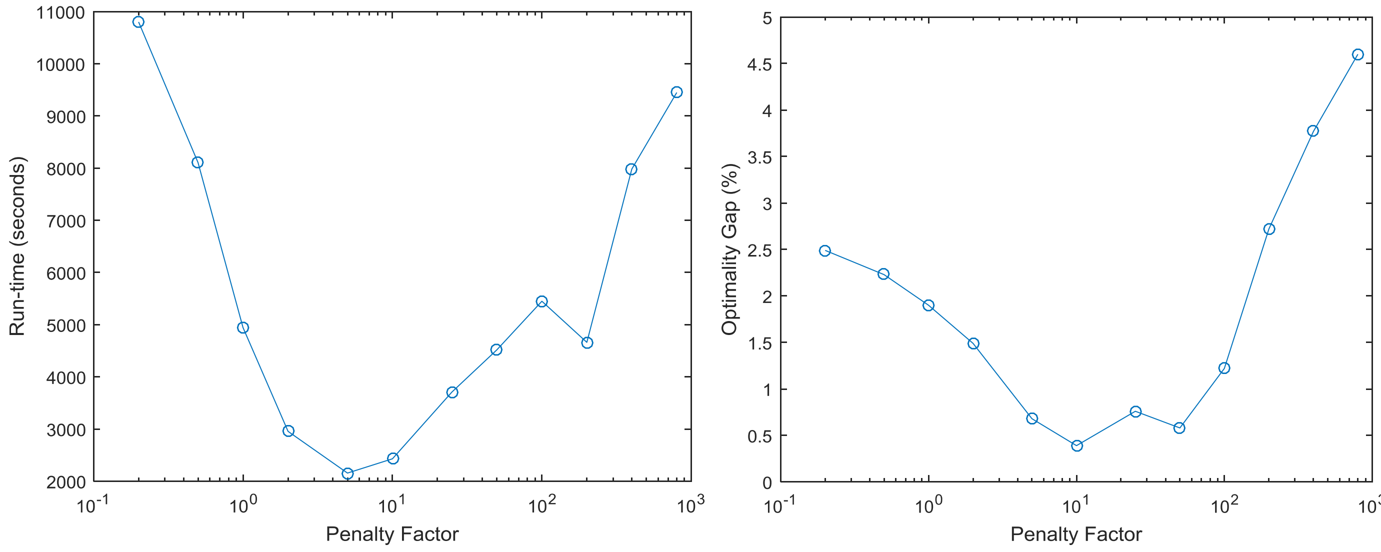


Fig. 2. The impacts of the penalty factor on the performance of GMM-FW-PH. The left plot shows the run-time versus the penalty factor on the instance M4035, whereas the right-hand plot shows the optimality gap versus the penalty factor on the instance C4023. Both plots use a base 10 logarithmic scale for the penalty factor.

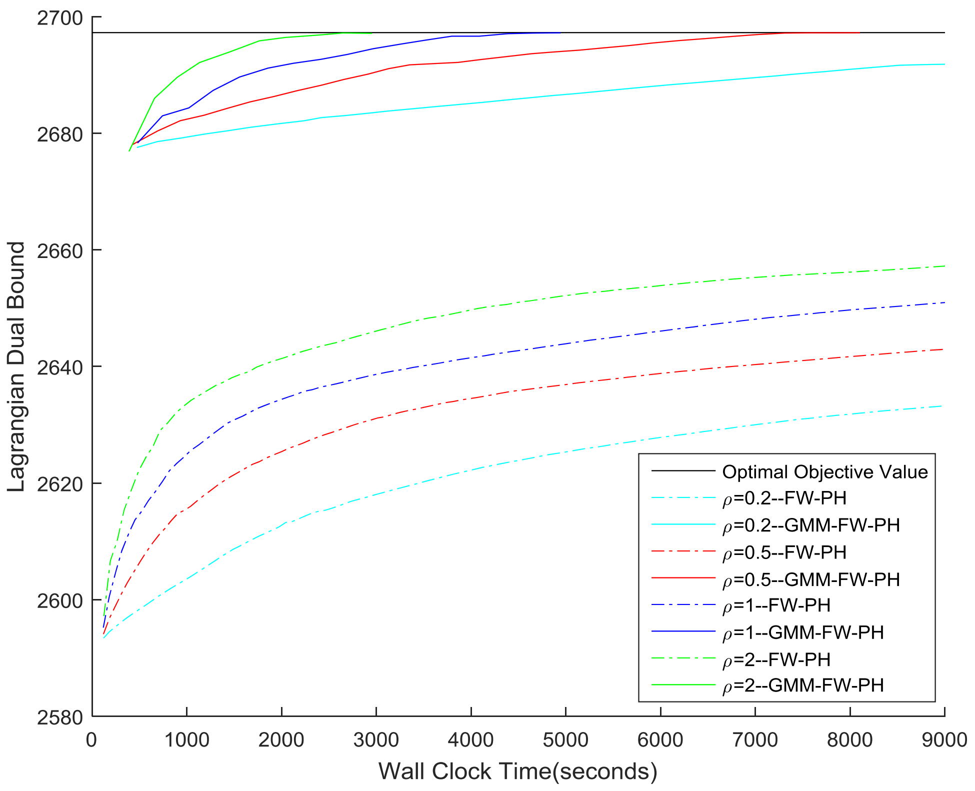


Fig. 3. Evolution of the lower bounds with time for M4035, obtained by FW-PH and GMM-FW-PH under four different penalty values {0.2, 0.5, 1, 2}.

We implemented Algorithm 1 in Microsoft Visual Studio 2010 using C++, with the optimization problems at line 3, 6,15 and 17 modelled by ILOG Concert Technology and solved by the CPLEX MIP optimizer in version 12.6.2. For all the experiments performed, we initialized the dual variables ω_{ij}^{tb} to zeros. Apart from a relative MIP gap tolerance of 0.5%, all the CPLEX parameters were left at their default settings. Regarding the termination criteria, we set the convergence tolerance at 0.001. Algorithm 1 also terminates after 3 h of wall-clock time or 190 iterations, whichever condition is met first. As the preprocessing step, Algorithm 2 was implemented in MATLAB R2015b, with the membership score calculation at the E- and M-step performed by the built-in function for Gaussian Mixture Models. Unless otherwise indicated, all the experiments were performed on a PC with eight 3.60 GHz Intel Core i7 CPUs and 16GB of RAM, under a 64-bit Windows 7 operating system. All of the algorithms in the experiments were executed sequentially and we did not consider parallel implementation of these algorithms here.

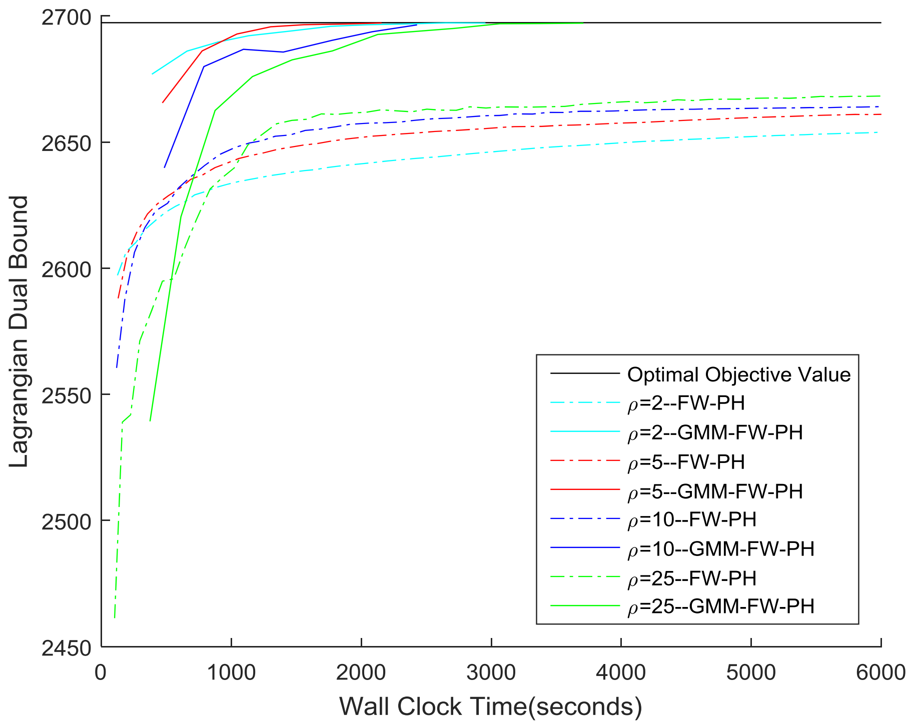


Fig. 4. Evolution of the lower bounds with time for M4035, obtained by FW-PH and GMM-FW-PH under four different penalty values {2, 5, 10, 25}.

Table 4
Performance results of FW-PH and GMM-FW-PH under various penalty values on the test instance M4035, for which only one algorithm can converge within 3 h. The letter "T" in the column "Run-time" indicates that the algorithm hit the time limit before convergence.

CPLEX Direct Solving			Expected Cost 2697.25		Run-time(s) 717	
Theoretical bound Via Decomposition					2697.25	
Penalty factor	Optimality Gap (%)		# Iterations		Run-time (s)	
	FW-PH	GMM-FW-PH	FW-PH	GMM-FW-PH	FW-PH	GMM-FW-PH
0.2	2.30	0.16	143	42	T	T
0.5	1.94	0.00	130	31	T	8104
1	1.64	0.00	115	17	T	4945
2	1.42	0.01	102	10	T	2955
5	1.25	0.00	89	7	T	2156
10	1.15	0.03	90	7	T	2429
25	0.85	0.01	92	12	T	3709
50	0.63	0.04	89	16	T	4517
100	0.27	0.00	85	20	T	5448
200	0.06	0.00	87	17	T	4661
400	0.00	0.00	94	25	T	7981
800	0.00	0.00	109	31	T	9448

6.2. Experimental results and analysis for small-size instances

Tables 3-5 provide the performance results of FW-PH and GMM-FW-PH on the test instances for stochastic service network design. According to the convergence status at termination, we organized the 12 test instances into three groups: (i) both algorithms converged, (ii) only one algorithm converged, and (iii) neither algorithm converged. Although each group was found to contain several instances, only the results of one instance were displayed in each table due to space limitations. For comparison purposes, we included the optimal objective values obtained by directly solving the instances by CPLEX, along with the computational time. The theoretical bounds via decomposition were also reported. As shown in each table, we chose up to 12 different values of the penalty factor for each test instance, ranging between 0.2 and 800, so as to study its influence on the convergence of FW-pH and GMM-FW-PH. In the column "Optimality Gap (%)", we reported the quality of the incumbent lower bound ϕ produced by FW-PH and GMM-FW-PH at termination, which was measured against a known optimal objective value ζ^* for the test instance and was calculated as $(\zeta^* - \phi)/\zeta^* * 100\%$. The total number of iterations and the elapsed computing time (rounded to the nearest integer) were recorded in the columns "# Iterations" and "Run-time", respectively. The network configuration of the optimal solution for U4035 in Table 3 is given in Fig. 1. It can be seen that, to effectively handle the demand uncertainty, each node is visited at least twice over the planning duration of 5 periods. Additionally, the consolidation at node 1, period 1 is also a common practice to enhance robustness of the network under uncertainties.

Analyzing the experimental results presented in Tables 3-5, we first observe that the specific choice of the penalty factor has a marked effect on the performance of both FW-PH and GMM-FW-

Table 5
Performance results of FW-PH and GMM-FW-PH under various penalty values on the test instance C4023, for which neither algorithm can converge within 3 h. The letter "T" in the column "Run-time" indicates that the algorithm hit the time limit before convergence.

CPLEX Direct Solving			Expected Cost		Run-time(s)	
			2071.32		1261	
Theoretical bound Via Decomposition				2071.32		
Penalty factor	Optimality Gap (%)		# Iterations		Run-time (s)	
	FW-PH	GMM-FW-PH	FW-PH	GMM-FW-PH	FW-PH	GMM-FW-PH
0.2	4.76	2.49	48	14	T	T
0.5	4.21	2.23	42	13	T	T
1	3.6	1.9	37	13	T	T
2	2.91	1.49	33	12	T	T
5	2.22	0.68	28	11	T	T
10	1.81	0.39	26	11	T	T
25	1.55	0.76	26	10	T	T
50	1.15	0.58	30	11	T	T
100	1.27	1.22	31	11	T	T
200	2.57	2.72	36	13	T	T
400	3.12	3.77	50	15	T	T
800	6.56	4.6	59	19	T	T

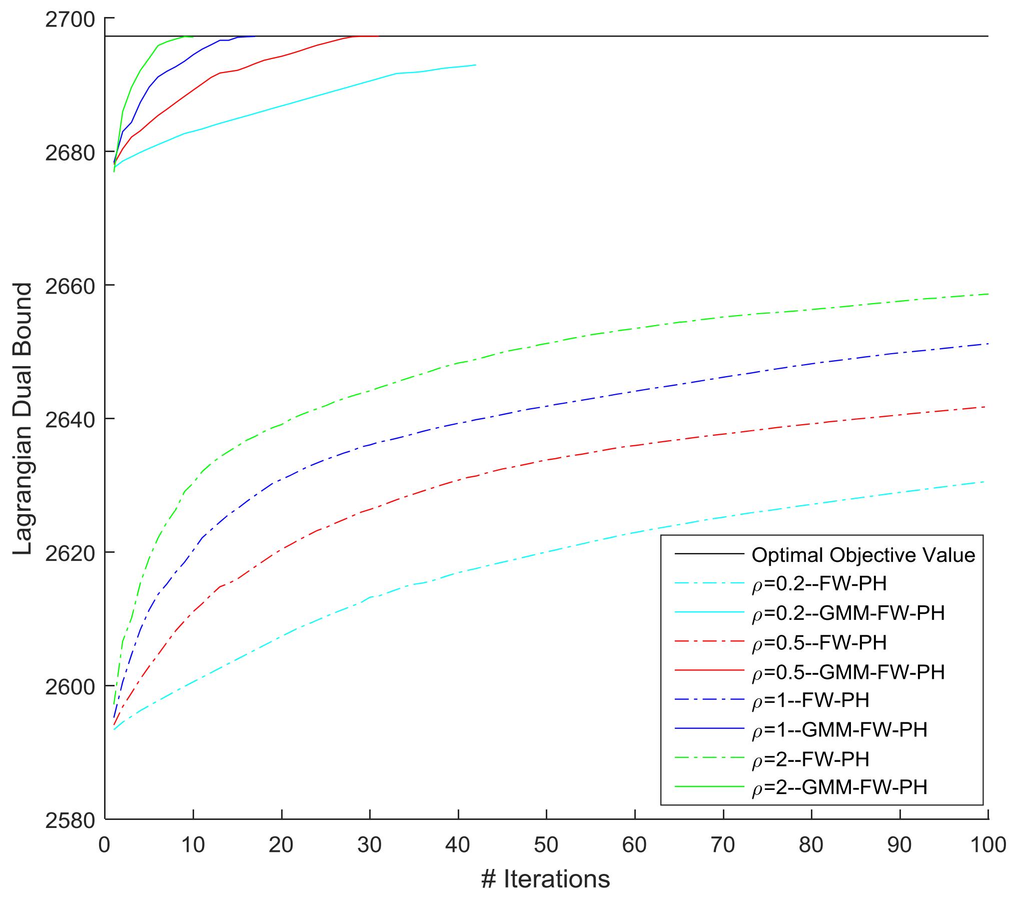


Fig. 5. Evolution of the lower bounds with iterations for M4035, obtained by FW-PH and GMM-FW-PH under four different penalty values {0.2, 0.5, 1, 2}.

PH. When we play around with different values of the penalty factor, it can be seen from Tables 3 and 4 that the time required by GMM-FW-PH to achieve convergence varies considerably from half an hour or so to more than 3 h. The optimality gaps in Table 5 show that the quality of the lower bound that GMM-FW-PH produces after hitting the time limit also varies significantly according to the penalty factor, with the difference between the maximum and minimum greater than 4%.

The impacts of the penalty factor can be seen more clearly by plotting the run-times (Tables 3 and 4) or optimality gaps (Table 5) against the penalty factor, as shown in Fig. 2. We see that small values of the penalty factor lead to relatively poor performance of the GMM-FW-PH, giving comparatively long run-times or large optimality gaps. As the penalty factor increases, the run-times and optimality gaps reduce dramatically, thereby producing substantial improvements in GMM-FW-PH's performance. However, further increase in the penalty factor would cause a sharp drop in the performance of the GMM-FW-PH. On one hand, as we shall see shortly, smaller values of the penalty factor are more reliable since larger values tend to render the lower bounds very poor in early

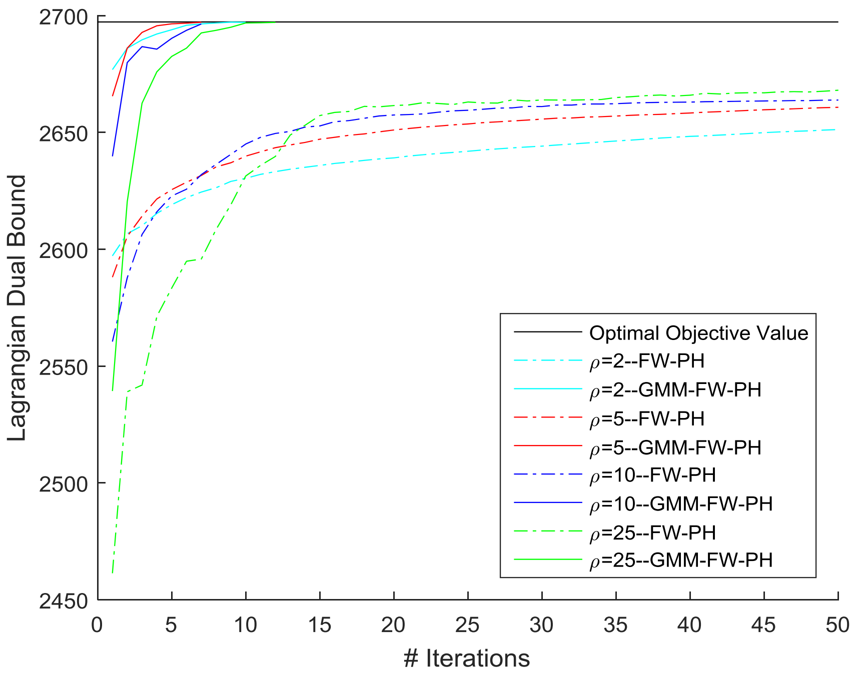


Fig. 6. Evolution of the lower bounds with iterations for M4035, obtained by FW-PH and GMM-FW-PH under four different penalty values {2, 5, 10, 25}.

iterations and widely oscillating throughout. On the other hand, if we take too small a value (e.g., several orders of magnitude smaller than the fixed cost) for the penalty factor, the dual variables will be improved only very marginally per iteration, which in turn will significantly delay the convergence.

We next study the impact of scenario bundling by comparing the performance results of FW-PH and GMM-FW-PH in Tables 3-5. With very few exceptions, GMM-FW-PH gives better performance than FW-PH for any choice of the penalty factor in any of the test instances. In particular, when the penalty factor is not too large, GMM-FW-PH offers vastly superior performance, providing either a dramatic saving in run-time or a much improved lower bound at termination.

Note that the performance results in Tables 3-5 are obtained under certain termination conditions (e.g., a time limit of 3 h). As a matter of fact, these results may vary with the parameter setting of the termination criteria. In order to validate the above research findings at various termination settings, it is necessary to keep track of how the lower bound evolves during the iterative process. Take the test instance M4035, for example. The evolution of its lower bound with time under 12 different penalty factors is illustrated in Figs. 3-4. Due to space limitations, plots showing the evolution with iterations are provided in Appendix A. For the sake of clarity, no more than 4 penalty factors are considered in each figure but the trends behaves similarly across all penalty ranges. It can be seen that when the penalty factor is set at appropriate levels, the lower bound from FW-PH or GMM-FW-PH rises steeply in the early period and then levels out, evolving very slowly in the remaining and majority of run-time towards convergence. From Figs. 2, we see that there is not much difference in the initial lower bounds generated by FW-PH or GMM-FW-PH between small values of the penalty factor. However, larger values of the penalty factor significantly improve the convergence speed of both FW-PH and GMM-FW-PH.

Unlike the cases of very small penalty factors in Figs. 2, it can be seen from Fig. 4 that the quality of the initial lower bounds generated by FW-pH or GMM-FW-pH declines dramatically as the penalty factor increases. Furthermore, among relatively small ones of these penalty factors, a larger value is still much better than a smaller one, since the disadvantages of poorer initial lower bounds are more than offset by the benefits of increased convergence speed.

Overall, the best value of the penalty factor will be given by a trade-off between high-quality initial lower bounds and fast convergence speed. Too strong a focus on either of the two aspects will yield poor results. Basically, these observations support our previous findings from Tables 3-5 that the best performance is obtained for some intermediate value of the penalty factor.

Finally, we consider the comparison between the lower bound curves of FW-PH and GMM-FW-PH when the penalty factor is not too large. Given the same value of the penalty factor, there are no significant differences in the convergence speed between FW-PH and GMM-FW-PH, as evidenced by the absolute change in the lower bound during the early 10 iterations or so. However, the initial lower bound generated by GMM-FW-PH is much higher than that generated by FW-PH under the same penalty value, with the absolute difference being greater than 75. Consequently, for any of the penalty values considered, the lower bound curve of GMM-FW-PH lies completely above that of FW-PH, indicating that the lower bounds obtained by GMM-FW-PH are consistently of better quality than those obtained by FW-PH throughout the iterative process. This can be attributed to the fact that GMM-FW-PH takes more time per iteration than FW-PH as a result of scenario bundling and hence yields the first lower bound at a later time. Aside from the initial short period, GMM-FW-PH consistently gives better lower bounds than FW-PH for the entire run-time. Fundamentally, these observations further reinforce our previous findings from Tables 3-5: GMM-FW-PH is far superior to FW-PH when the penalty factor is not too large, providing either a dramatic saving in run-time or a much improved lower bound at termination.

6.3. Results for large-size instances

In this section, we run the experiments for 3 larger instances with 100 scenarios (C10022, M10022 and U10022). For the comparison purposes, the three instances were also solved to opti-

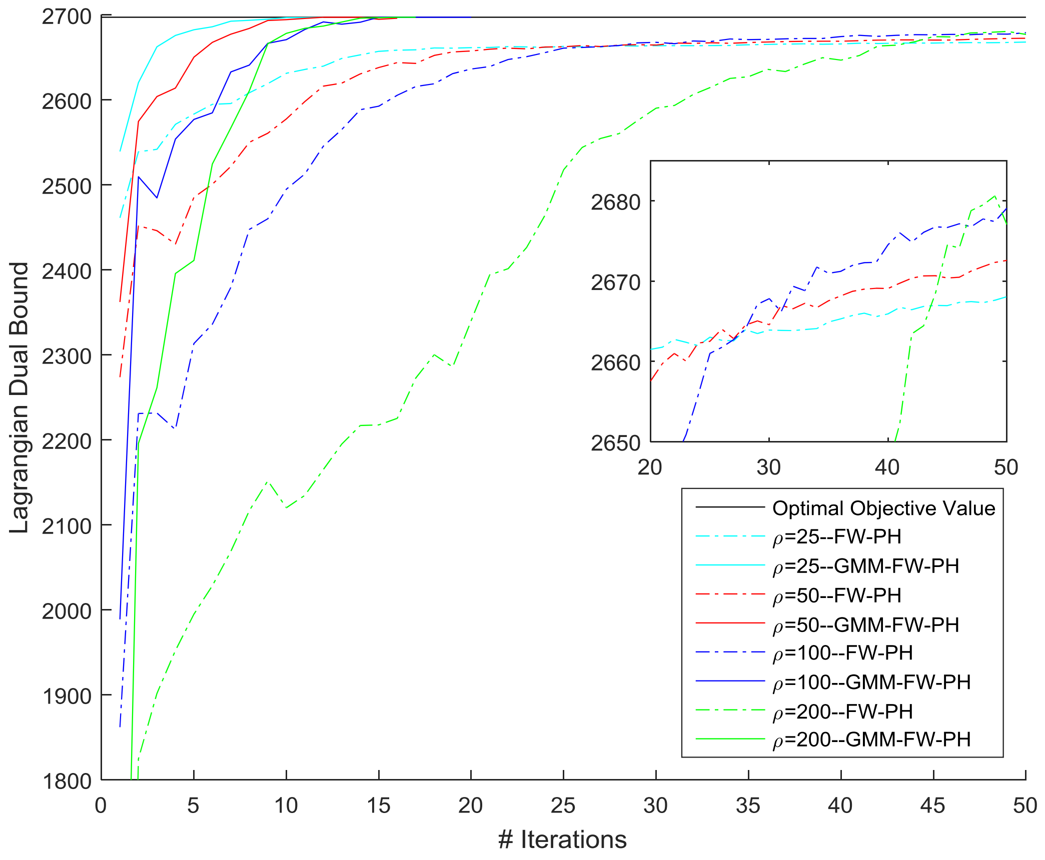


Fig. 7. Evolution of the lower bounds with iterations for M4035, obtained by FW-PH and GMM-FW-PH under four different penalty values {25, 50, 100, 200}.

Table 6
The computational results of GMM-FW-PH for large problem instances with 100 scenarios.

Instances	CPLEX direct solving	GMM-FW-PH		
		cost	gap%	Run-time(s)
M10022	2320.61	2259.55	2.6%	T
C10022	2349.15	2258.55	3.9%	T
U10022	2253.81	2253.81	0.0%	8890

mality by calling CPLEX MIP solver directly. The results and run-times are summarized in Table 6. Although the algorithm fails to converge within the given time limit for two instances (M10022 and C10022), it still generates reasonably tight bounds that can be very useful to algorithmic development. For U10032, the proposed method converges to the optimal solution within 8890 s.

In this study, we chose to use GMM as the method to cluster the scenarios into bundles. In our previous study (Jiang et al., 2021), various bundling methods are investigated, including random bundling and machine learning based bundling (K-means, C-Means). It was found that, based on the empirical results, a small level of overlaps in membership can speed up the convergence considerably without worsening the solution quality.

7. Conclusions and future directions

We have introduced scenario bundling into the decomposition of the SSND problem, so as to obtain tighter lower bounds within less run-time. Instead of scenario-wise decomposition, we consider bundle-wise decomposition to divide the problem by scenario bundles into multi-scenario subproblems. Dualizing the NACs arising from bundle-wise decomposition produces the Lagrange dual problem, whose objective function value provides a lower bound on the SSND problem. We have proven that the optimal Lagrange dual bound is tighter than or equal to that in the case of scenario-wise decomposition.

In order to solve the Lagrange dual problem, we have improved the FW-PH algorithm by replacing scenario-wise decomposition with bundle-wise decomposition. The disjoint scenario bundles are constructed using Gaussian mixture models. As with the basic FW-PH algorithm, the improved version is guaranteed to converge to the optimal Lagrange dual bound in theory. However, the two versions differ significantly in the computational performance. Extensive computational experiments demonstrate that the improved FW-PH algorithm is far superior to the basic version when the penalty factor is not too large, providing either a dramatic saving in the run-time required to achieve convergence or a much tighter lower bound when terminated due to the time limit. Moreover, the computational evidence suggests that the specific choice of the penalty factor has a marked effect on the performance of both the basic and the improved FW-PH algorithm. In particular, the computational evidence does not support the use of very large penalty factors, because in that case it would take prohibitively long before the quality of the lower bound reaches an acceptable level, owing to the very poor initial lower bound and the continual oscillation. Note that the method proposed in this paper is served to compute a high quality lower bound of the original SSND problem. The results can be useful to develop more efficient bundling based SSND algorithms that were compared extensively in (Xiang et al. 2021).

One limitation of the proposed algorithm is that its computational performance gets poor when addressing very large-scale problems (involving, say, over 100 nodes, 1000 arcs, 24 time periods, 800 commodities and 1000 scenarios). This is not surprising if we consider the fact that deterministic network design problems

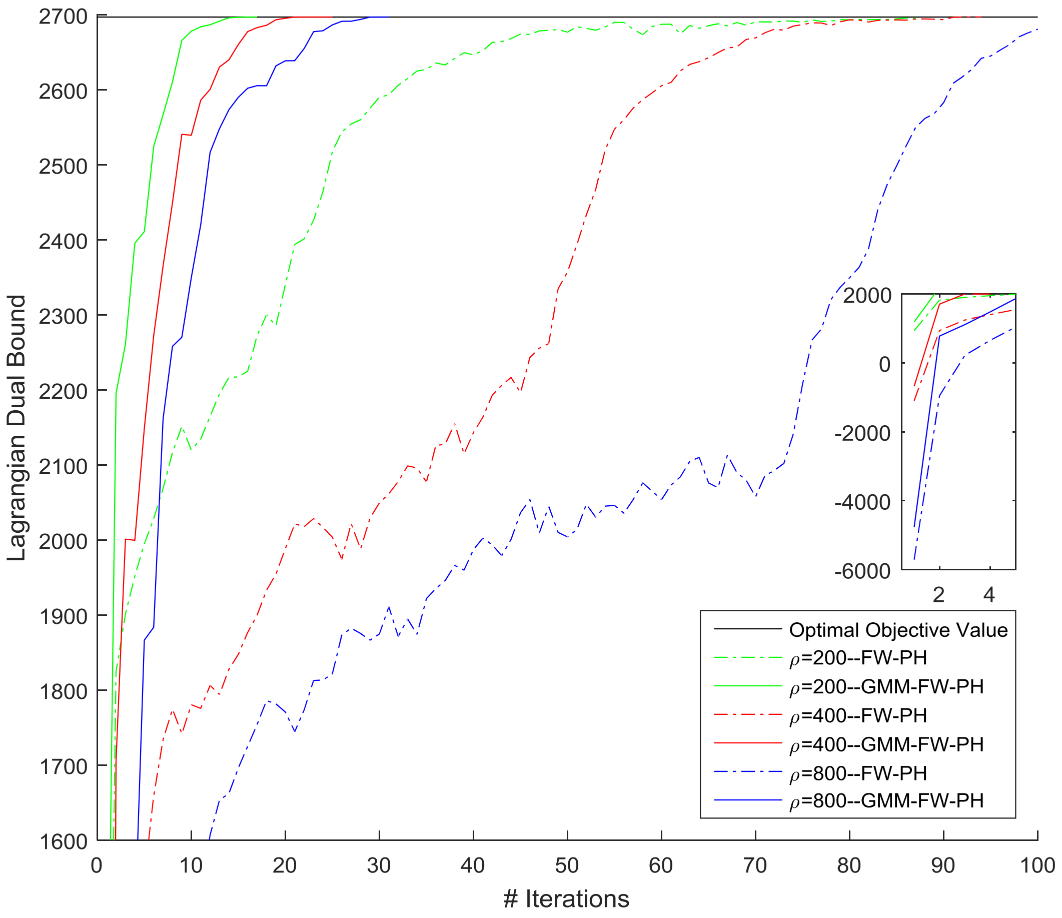


Fig. 8. Evolution of the lower bounds with iterations for M4035, obtained by FW-PH and GMM-FW-PH under four different penalty values {200, 400, 800}.

are NP-hard and modeling demand uncertainty with scenarios further increases the size of the network design problem. Given the current large size of LTL transportation networks, and the trends for them to become even larger in the era of ecommerce, further research into solution approaches (e.g., parallel computation) is needed. As for the model discussed, the primary limitation concerns model assumptions. In this study, it is assumed that customers' demands are uncertain only in commodity volumes. In real-world applications, however, not only the commodity volumes, but the origins, destinations, ready times and delivery deadlines of the shipments are uncertain at the decision-making point. These other uncertainties are left out or inadequately treated in the models.

For future research, we envision three possible directions. First, the improved FW-PH algorithm offers great opportunity for parallel computation, since the multi-scenario subproblems can be solved independently. Because parallelization has the potential to dramatically reduce the run-time, a parallel implementation of the extended FW-PH warrants careful study, in a similar effort by Boland, Christiansen, Dandurand, Eberhard and Oliveira (2019). Second, the improved FW-PH algorithm can be naturally used as a bounding procedure for obtaining optimal solutions to SSND. New exact methods can thus be developed by, for example, integrating the improved FW-PH algorithm into a branch-and-bound framework. The rapid improvement in the Lagrange dual bound during early iterations of the improved FW-PH algorithm may be beneficial for accelerating the exact methods. Also, the improved FW-PH algorithm may be used in conjunction with heuristic techniques for SSND to enable an assessment of the solution quality. Lastly, besides commodity volumes, customers' demands are also uncertain in terms of commodity origins and destinations, as well as the starting time and deadline for delivery. These uncertainties have become even more pronounced with the rising popularity of electronic and mobile commerce. In addition, travelling time is often an important source of uncertainties in practice. Therefore, it is meaningful to investigate the impact of these uncertainties on service network design in the future and develop some machine learning-based methods which have shown promising results in recent studies (Bai, Chen & Chen, 2022; Zhang, Bai, Qu, Tu & Jin, 2022)".

Declaration of Competing Interest

None.

Acknowledgements

This work is supported by the National Natural Science Foundation of China [grant number NSFC 72071116, 72101266] and the Ningbo Science and Technology Bureau [grant number 2019B10026]. We are also grateful to the financial support from the International Doctoral Innovation centre.

Appendix A. Evolution of the lower bounds with iterations

Appendix para

References

Ahmed, S., Luedtke, J., Song, Y., & Xie, W. (2017). Nonanticipative duality, relaxations, and formulations for chance-constrained stochastic programs. *Mathematical Programming, 162*(1), 51–81.

Arthur, D., & Vassilvitskii, S. (2007). K-Means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), January 07-09* (pp. 1027–1035).
Bai, R., Kendall, G., Qu, R., & Atkin, J. (2012). Tabu assisted guided local search approaches for freight service network design. *Information Sciences, 189*, 266–281.
Bai, R., Wallace, S. W., Li, J., & Chong, A. Y.-L. (2014). Stochastic service network design with rerouting. *Transportation Research Part B, 60*, 50–65.
Bai, R., Chen, X., & Chen, Z. L.others. (2022). Analytics and Machine Learning in Vehicle Routing Research. *International Journal of Production Research. In press.*. https://doi.org/10.1080/00207543.2021.2013566.
Bakir, I., Boland, N., Dandurand, B., & Erera, A. (2020). Sampling Scenario Set Partition Dual Bounds for Multistage Stochastic Programs. *INFORMS Journal on Computing, 32*(1), 145–163.
Barnett, J., Watson, J.-. P., & Woodruff, D. L. (2017). BBPH: Using progressive hedging within branch and bound to solve multi-stage stochastic mixed integer programs. *Operations Research Letters, 45*(1), 34–39.
Beier, E., Venkatachalam, S., Corolli, L., & Ntaimo, L. (2015). Stage- and scenario-wise Fenchel decomposition for stochastic mixed 0-1 programs with special structure. *Computers & Operations Research, 59*, 94–103.
Bertsekas, D. P. (2015). *Convex optimization algorithms*. Belmont, Massachusetts: Athena Scientific.
Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer-Verlag.
Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., et al. (2018). Combining progressive hedging with a frank-wolfe method to compute Lagrangian dual bounds in stochastic mixed-integer programming. *SIAM Journal on Optimization, 28*(2), 1312–1336.
Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., & Oliveira, F. (2019). A parallelizable augmented Lagrangian method applied to large-scale non-convex-constrained optimization problems. *Mathematical Programming Series A, 175*(1–2), 503–536.
Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. New York: Cambridge University Press.
Carøe, C. C., & Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters, 24*(1–2), 37–45.
Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research, 122*(2), 272–288.
Crainic, T. G., Hewitt, M., & Rei, W. (2014). Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research, 43*, 90–99.
Deng, Y., Jia, H., Ahmed, S., Lee, J., & Shen, S. (2021). Scenario grouping and decomposition algorithms for chance-constrained programs. *INFORMS Journal on Computing, 33*(2), 757–773.
Escudero, L. F., Garín, M. A., Pérez, G., & Unzueta, A. (2013). Scenario cluster decomposition of the lagrangian dual in two-stage stochastic mixed 0-1 optimization. *Computers & Operations Research, 40*(1), 362–377.
Escudero, L. F., Garín, M. A., & Unzueta, A. (2016). Cluster Lagrangean decomposition in multistage stochastic optimization. *Computers & Operations Research, 67*, 48–62.
Fisher, M. L. (2004). The lagrangian relaxation method for solving integer programming problems. *Management Science, 50*(12), 1861–1871.
Gade, D., Hackebeil, G., Ryan, S. M., Watson, J.-. P., Wets, R. J.-B. &, & Woodruff, D. L. (2016). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming, 157*(1), 47–67.
Guo, G., Hackebeil, G., Ryan, S. M., Watson, J.-. P., & Woodruff, D. L. (2015). Integration of progressive hedging and dual decomposition in stochastic integer programs. *Operations Research Letters, 43*(3), 311–316.
Hewitt, M., Crainic, T. G., Nowak, M., & Rei, W. (2019). Scheduled service network design with resource acquisition and management under uncertainty. *Transportation Research Part B: Methodological, 128*, 324–343.
Hooker, J. N. (2008). Integer Programming: Lagrangian Relaxation. In C. Floudas, & P. Pardalos (Eds.), *Encyclopedia of optimization*. Boston: Springer.
Høyland, K., Kaut, M., & Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications, 24*(2), 169–185.
Høyland, K., & Wallace, S. W. (2001). Generating scenario trees for multistage decision problems. *Management Science, 47*(2), 295–307.
Jiang, X., Bai, R., Atkin, J., & Kendall, G. (2017). A scheme for determining vehicle routes based on arc-based service network design. *Information Systems and Operational Research, 55*(1), 16–37.
Jiang, X., Bai, R., Wallace, S. W., Kendall, G., & Landa-Silva, D. (2021). Soft clustering-based scenario bundling for a progressive hedging heuristic in stochastic service network design. *Computers & Operations Research, 128*, Article 105182.
Kaut, M., & Wallace, S. W. (2007). Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization, 3*(2), 257–271.
Kiwiel, K. C. (1990). Proximity control in bundle methods for convex nondifferentiable optimization. *Mathematical Programming, 46*(1–3), 105–122.
Lium, A.-. G., Crainic, T. G., & Wallace, S. W. (2009). A study of demand stochasticity in service network design. *Transportation Science, 43*(2), 144–157.
Løkketangen, A., & Woodruff, D. L. (1996). Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. *Journal of Heuristics, 2*(2), 111–128.
Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization*. New York: Wiley-Interscience.
Ryan, K., Ahmed, S., Dey, S. S., Rajan, D., Musselman, A., & Watson, J. P. (2020). Optimization-driven scenario grouping. *INFORMS Journal on Computing, 32*(3), 805–821.
Ryan, K., Rajan, D., & Ahmed, S. (2016, May). Scenario decomposition for 0-1 stochastic programs: Improvements and asynchronous implementation. In *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International* (pp. 722–729).
Shor, N. Z., Kiwiel, K. C., & Ruszczynski, A. (1985). *Minimization methods for non-differentiable functions*. Berlin: Springer.
Wang, X., Crainic, T. G., & Wallace, S. W. (2019). Stochastic network design for planning scheduled transportation services: The value of deterministic solutions. *INFORMS Journal on Computing, 31*(1), 153–170.
Watson, J.-. P., & Woodruff, D. L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science, 8*(4), 355–370.
Watson, J.-. P., Woodruff, D. L., & Hart, W. E. (2012). PySP: Modeling and solving stochastic programs in Python. *Mathematical Programming Computation, 4*(2), 109–149.
Zhang, Y., Bai, R., Qu, R., Tu, C., & Jin, J. (2022). A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *European Journal of Operational Research. In press.*. https://doi.org/10.1016/j.ejor.2021.10.032.