



CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Service Network Design Problem with Quality Targets and Stochastic Travel Times

Giacomo Lanza
Teodor Gabriel Crainic
Walter Rei
Nicoletta Ricciardi

December 2017

CIRRELT-2017-71

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palais-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Service Network Design Problem with Quality Targets and Stochastic Travel Times

Giacomo Lanza¹, Teodor Gabriel Crainic^{2,3,*}, Walter Rei^{2,3}, Nicoletta Ricciardi^{2,4}

¹ Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy

² Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

³ Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

⁴ Department of Statistical Sciences, Sapienza University of Rome, Italy

Abstract. Service network design is one of the fundamental problems that carriers face when planning a consolidation-based service network. Its scope is the selection and scheduling of services and the routing of freight. In this paper, we define a new problem setting in which quality targets are accounted and the uncertainty of travel time is explicitly considered. The goal is to define a cost-efficient transportation plan such that scheduled service arrival times at stops and agreed upon time of freight deliveries are respected as much as possible over time. Particularly challenging in our problem was to define the appropriate definition of decisions and the related information revelation process. To tackle the problem, we propose a two-stage stochastic linear mixed-integer programming formulation where design and routing make up the first stage and the given targets are accounted in the second stage through a set of penalties, once travel time realizations become known. We also propose a meta-heuristic based on the idea of progressive hedging to address larger instances than a well-known commercial solver cannot solve. Numerical results show the attractiveness with respect to given quality targets as well as the effectiveness of the proposed resolution methodology.

Keywords: Service network design, stochastic travel time, quality targets, 2-stage formulation, progressive hedging meta-heuristic.

Acknowledgements. Computations were made on the supercomputer Briarée from University of Montreal, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), the Ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQNT).

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

1 Introduction

Throughout history, transportation has played a vital role in the social, political and economical development of nations by supporting production, trade and consumption activities and ensuring the movement of people and freight timely and efficiently from place to place. The transportation industry operates in a highly competitive and mainly cost-driven environment. Shippers, carriers and logistics service providers continuously seek for ways to minimize the costs of the offered services establishing sets of operating policies to perform the routing of commodity flows and management of available resources (both human and material) in the most efficient and rational way in order to satisfy customers' needs and still making a profit.

Freight consolidation is one of the many ways to lower transportation costs (and consequent service prices), taking advantage of economies of scale. Building a consolidation-based transportation network is normally a rather complex problem which is traditionally faced at a tactical level by carriers and supported by *Service network design* (SND).

The scope of SND is to produce an operations plan achieving the economic and quality targets of a carrier. The plan specifies the services to operate (routes, frequency and schedules) to meet the estimated transportation demand and the routes of freight (services used and terminals passed through). Quality targets normally concerns the *reliability of operations* of the entire service network, meaning the regular on time arrival of services at their stops with respect to a published schedule, and *reliability of deliveries*, meaning the repeatedly on time arrival of freight at destinations with respect to promised due dates. The operations plan is normally made for a certain time interval, called schedule length (e.g., a week) and, then, repeatedly performed for a longer time period (e.g., the next season), called planning horizon [Crainic, 2000, 2003; Crainic and Kim, 2006].

Several efforts have been directed towards the formulation of SND models. All the models aim for economic efficiency to build the service network. Extremely rarely, instead, direct attention is given to the above mentioned quality targets. In addition, most of the proposed formulations assume that all the necessary information to build the service network is available and completely known at the moment of planning. As opposed, planning usually means facing with the challenge of making decisions when only limited information is available. The presence of uncertainty in the network (demand, cost, profit, lead time, reliability of vehicles, customers' locations) is a critical issue to consider explicitly in the decision process to achieve the above mentioned carrier goals, making of SND a very challenging problem to address [Powell and Topaloglu, 2003].

The most studied stochastic phenomenon in the SND literature is customers' demand. One aspect which received little attention is, instead, travel time (time needed to travel between two consecutive stops). The vast majority of proposed model formulations assume, in fact, travel time as a deterministic parameter, commonly built on point forecasts based on historical data. The travel time observed in actual operations may, however, differ from estimations due to a variety of influences such traffic congestion or heavy weather conditions, resulting in potential additional economical costs related to crews and resource utilization and, in addition to them, fines and loss of reputation for not respecting planned arrival times and agreed upon time of deliveries. A deterministic time assumption is, therefore, a strong assumption which not only do not represent a realistic approximation of this phenomenon, but also may lead to poor decisions with respect to economic and quality goals. Despite its importance, only few contributions dealing with design of

transportation services and stochastic time have appeared in the literature till now and none of those focused on the definition of a reliable service network in terms of both schedules and freight deliveries as we do. To the best of our knowledge, this problem has never been considered in the literature before. We define it as the Stochastic Service Network Design problem with Quality Targets.

The first goal of this paper is, therefore, to address the problem of defining a cost-efficient transportation plan such that the selected services and the specified routes of freight respect, respectively, the scheduled arrival times at stops and due dates as much as possible over time. To achieve this purpose, we propose a stochastic SND original methodology, where the uncertainty of travel time is explicitly accounted into the decision process and quality targets are modelled through penalties. Design and routing decisions are made before any travel time realization. The proposed formulation, therefore, takes the form of a two-stage mixed-integer linear stochastic model. In the first stage, design and routing decisions are taken, while in the second stage, when travel time realizations become known, the given targets are accounted and delays for a given travel time realization and a chosen design are properly penalized.

SND problems are notoriously NP-Hard. On the one hand small to medium-sized problem instances can be solved optimally, on the other hand heuristic methods are generally needed to find high-quality solution to real-life instances in acceptable time. Our second goal for this paper is to propose an effective progressive hedging-based meta-heuristic (PHM) algorithm to tackle the above mentioned problem. This is the first attempt to apply such a methodology to a SND problem with uncertainty in travel times and, thus, an additional original feature of this contribution. The proposed meta-heuristic modifies the traditional application scheme of the method (applied, for instance in Crainic et al. [2011b]) in order to overcome problems related to a quadratic reformulation and flow-degeneracy which raise, when it is classically applied to our proposed formulation. The numerical results provided in the paper show how the meta-heuristic is able to address larger instances than *CPLEX*.

Therefore, the contributions of this paper are:

- to propose a new SND problem setting where uncertainty in travel times and quality targets are explicitly considered;
- to provide a two-stage stochastic linear mixed-integer programming formulation for the proposed problem;
- to develop an PHM able to effectively find good quality solutions to large-scaled problem instances;
- to show numerical results of the attractiveness of the formulation when the described quality targets are of interest;
- to present numerical results of the performance of the proposed PHM comparing it to an exact resolution method.

The paper is organized as follows. In Section 2, we state the problem and review the associated very little available literature related to SND and stochastic time. In Section 3, we describe the notation adopted in order to introduce our mathematical programming formulation for the proposed problem. In Section 4, we describe the PHM. In section 5, we illustrate the experimental

plan. Two set of experiments are proposed, the first aims at qualifying the model and its benefits, the second the performance of the PHM. Results of both sets are reported in Section 6. Finally conclusions and future research are discussed in the last section (Section 7).

2 Problem Description and Literature Review

We focus on one of the fundamental problems - part of tactical planning - that consolidation-based carriers have to face: set up a cost-efficient and reliable service network which achieve chosen quality targets and satisfy the estimated regular demand [Crainic, 2000, 2003; Crainic and Kim, 2006]. Consolidation-based transportation systems are systems in which freight of several different customers with possibly different origins and destinations is combined and dispatched together into a common vehicle for part of the shipment. It is in contrast with customized service (in which carriers dedicate an entire vehicle to single customers performing consignments tailored exactly to their needs) and usually includes several surrounding activities such as warehousing, sorting, loading into or unloading from vehicles. The structure of consolidation-based transportation systems (typical examples are railways, less-than-truckload motor carriers, container shipping lines) consists of a large and quite complex network of terminals connected by physical (or conceptual) links, also known as a hub-and-spoke network: low-volume demand is first moved to intermediate terminals, or hubs, where is consolidated with loads of other customers and moved together to other hubs. Demand is represented in terms of commodities, that is collection of similar products requiring transportation through the network. Other than by its origin, destination, arrival time at origin (entry date) and required time at destination (due date), each shipment has, normally, several physical characteristics (e.g weight, volume) and may have specific shipment-requirements (e.g. delivery condition, type of vehicle).

Such systems are composed by regular services, which carriers establish selecting them appropriately from a set of *potential* ones that could be activated. Each potential service is defined by its origin and destination terminals, departure time at origin, departure and arrival times at intermediate stops (if any) and arrival time at destination. The selection is performed over a given length of time (the *schedule length*), in which for instance some regularity in demand may be observed (e.g. a week). Selected services are, then, externally proposed by carriers to the customers and operated according to a schedule, repeated periodically for the whole duration of a longer time period, called *planning horizon* (e.g., six months). The schedule specifies for each single offered service: departure time at origin, arrival time at destination and arrival times at/departure times from each intermediate stop (if any).

Internally, instead, carriers define a series of rules that affect the whole system to ensure that the selected services are performed as stated or, at least, as close as possible to the resulting (and published) schedule and that freight routing policy respects demand promised due dates as much as possible over time. For instance, externally a carrier may propose for a given traffic-class deliveries in 24 hours. Considering the uncertainty in duration of the various activities of a carrier (e.g. travel time, operation time for consolidation activities), it is almost impossible to guarantee 100% on-time operations (as externally promised), competitiveness and profitability at the same time. Consequently, internally, a certain level of quality is chosen considering a trade-off between operating costs and service performances (e.g. the selected services activated to carry

freight belonging to that particular traffic-class must ensure on time deliveries at least 90% of the time). Policies related to the penalty when not respecting the promised due date have to also be defined and, normally, customers are significantly involved through their contracts. *Quality targets* define, thus, the minimum degree of conformity to the schedule and demand promised due dates the carrier desires to achieve with the selected service network and freight routing policies. We define them respectively *target of services* and *target of demand*. Targets may easily be expressed through probabilistic conditions. For instance, a service target may be expressed as follows: each service has to respect its planned arrival time at least with probability α , considering all its repetitions during the planning horizon, and delays should be not greater than a pre-specified time amount (e.g., a specific percentile of the travel time probability distribution, the expected value plus the standard deviation), with probability 1.

SND is normally devoted to the construction of such a service network as well as the definition of the routing of freight on the network. Its goal is then the selection of services, their schedule and the routing of the demand in the most cost-efficient way to achieve chosen quality targets and satisfy the estimated regular demand [Crainic, 2000, 2003; Crainic and Kim, 2006].

The vast majority of proposed SND formulations assume travel time as deterministic parameter, built on point forecasts based on available travel time duration historical data (e.g., a statistical estimation or a function of the distance). It is, however, not guaranteed that the travel time observed in actual operations always respects that forecast. In many real-life applications, in fact, a considerable degree of variability in travel time can be observed and a deterministic time assumption (in this case, the perfect knowledge of future time realizations) does not represent an accurate and realistic approximation of actual travel times. Unexpected time fluctuations eventually cause delays, which, for carriers, result in potential additional economical costs related to crews and resource utilization and, in addition to them, fines and loss of reputation and reliability for not respecting promised freight arrival times at customer locations.

Despite its importance, only few contributions dealing with design of transportation services and stochastic time have appeared in the literature. Klibi et al. [2010] and Klibi and Martel [2012] characterize decision-making situations based on the quality of the available information under which decisions are made. Following their classification, our uncertainty-profile belongs to *Randomness*. Randomness describes events which probability of occurrence can be estimated exploiting historical and accessible data through classical forecasting and statistical analysis methods. This information, then, can be used to estimate the probability distribution of the random events disrupting business-as-usual operations, which in our case are travel times. Typical examples of randomness in our context are fluctuations caused by traffic congestion (in particular for road or rail transportation) or heavy weather conditions (in particular for ship and aircraft transportation).

To the best of authors knowledge, when considering SND problems and stochastic travel times, only contributions related to this type of uncertainty are available in the scientific literature, highlighting a lack of research on the topic. Demir et al. [2016] consider a multi-modal consolidation-based transportation system, where road transportation services have to be planned in order to catch available (rail and maritime) transportation services operating according to fixed schedules to perform part of the whole shipment. The goal of the problem is to select truck services and route freight minimizing costs and exploiting as much as possible the available services in order to reduce gas emission related to truck transportations. They, thus, propose a multi-objective SND

formulation and consider a chance constrain approach to ensure with a given probability safe connections between the selected truck services and the available “green” services. A similar problem is addressed by Van Hui et al. [2014]. Here when connections are missed additional costs have to be paid in order to still fulfill demand by alternative services. A “common industrial practice” in dealing with consolidation is applied, upon observing a delay to an upcoming shipment: breaking the consolidation that involves the tardy shipment, release the on-time shipments following the start-up plan and ship the tardy shipment through the faster and available route. The start-up plan is, then, built by minimizing the expected costs of such adjustments considering a two-stage approach where the start-up plan is addressed in the first stage (planning) and re-shipments are addressed in the second stage (recourse). In Wang and Meng [2012a] and Wang and Meng [2012b], the optimization of sailing speed of a fleet of container shipping lines is taken into account by also analysing the characteristics of bunker consumption in order to achieve target arrival times at a sequence of ports. There is a fixed schedule to respect, but late departure times are possible caused by longer than as planned port operations and thereby affecting the available sailing time for the subsequent voyage leg. As opposed as in our case, here the focus is on operation time uncertainty. The model they proposed is a mixed-integer non-linear convex stochastic minimization model. A similar problem is also considered in Song et al. [2015], where a stochastic multi-objective optimization problem has been formulated.

The scarce literature availability on such type of SND highlights the novelty of the problem and a substantial need of further research on the field. The problem we propose in this paper, differs from all the available contributions for considering jointly service and demand targets. Beside differences in modeling and source of time uncertainty, in the latter works, in fact, only service targets are accounted through the respect of a given schedule imposed either by the need to catch other external and given services operating according to already fixed own schedules or by the need to respect given and fixed time slots to perform terminal operations at intermediate stops. In their setting, also, a sort of dependency by external factors (such those earlier described) is imposed on the networks that have to be built, which is not present in our problem.

Some assumptions are here made:

- the travel time variations we consider are those that may be observed in “normal” and “smooth” conditions (Randomness as defined by [Klibi et al., 2010; Klibi and Martel, 2012]);
- service time at terminals (for loading/unloading sorting and consolidation operations) is assumed deterministic and constant;
- travel time random variables are assumed to be independent with known probability distributions;
- all services are of the same type (move at the same speed, have the same capacity);
- early arrivals of services at terminals are allowed and do not imply extra costs;
- although a service arrives at a stop earlier than planned, terminal operations cannot start earlier than scheduled;
- if a service arrives later than planned, terminal operations begin as soon as the service arrives and connections are not missed.

3 Mathematical Formulation

The physical network on which the carrier operates is represented by a graph $G_{phys} = (N_{phys}, A_{phys})$, which nodes in set N_{phys} represent the physical terminals composing the physical network and arcs in A_{phys} represent the physical connections between terminals on which the services move. To each arc is associated a point forecast of the usual travel time and a travel time random variable.

As for many scheduled SND problems, our problem is addressed through a *time-space network*, $G = (N, A)$. We assumed that the schedule length is divided in $T + 1$ time instants. The set of nodes of G_{phys} is, thus, replicated $T + 1$ times. The resulting set of replicated nodes is the set N . Each node of N represents one of the physical freight terminals that composes the physical transportation network at different time instants $(0, \dots, T)$.

Let K be the set of commodities that have to be transported. Each commodity $k \in K$, requires the transport of a certain volume $w(k)$ from an origin $o(k)$ to a destination $d(k)$ in the time-space network in according to its origin, destination, entry and due dates in the physical system. Let R be the set of potential services that the carrier may use to answer demand, which capacity is denoted by u_r . Each service $r \in R$ has a route in the physical network. Following [Crainic et al., 2011a], we define each route with the ordered set of visited terminals $\sigma(r) = \{z_n \in N_{phys}, n = 1, \dots, |\sigma(r)|\}$, where $n < m$ implies r visits terminal n before terminal m . On the same physical route may move different services, having the same set of stops but different leaving times at origins (and, consequently, different arrival time at destination) or services having the same origin and destination terminals but not the same set of intermediary stops. The path segment between two consecutive stops z_l and z_{l+1} of service r is called *service leg* and is denoted by $l(r)$. Each service leg is composed by one or more arcs of A_{phys} and its usual travel time and associated travel time random variable is respectively the sum of the usual travel times and the convolution of travel time random variables (note the independence assumption) of the arcs in A_{phys} making up that service leg.

The set of potential services R is represented in the time-space network through a set of arcs. The set A of G is, in fact, composed by two sets: A^H and A^M . An arc $(i, j) \in A^H$ links two nodes representing the same physical terminal in two consecutive time instants and is used to model *idle time* at terminal for freight or *operation time* at terminal for services. These arcs are also often referred to as *holding arcs*. An arc $(i, j) \in A^M$ links nodes representing different physical terminals in different time instants and is used to model the *movement of a service* between two different terminals at certain point in time. Each movement arc between two nodes $i \in N$ and $j \in N$ represents a specific service leg of a potential service in time. We sometimes refer to such arcs as $i(r) \in L(r)$, instead of $(i, j) \in A^M$, where $L(r)$ is the set composed by the service legs of service $r \in R$ in the time-space network. The travel time point forecast, denoted by $\hat{\tau}_{i(r)}$, and the travel time random variable, denoted by $\tau_{i(r)}$, are associated to each service leg $i(r)$, according to the leg of the physical network it represents in the time-space network. Finally, we denoted by t the deterministic time amount required for terminal operations.

Three types of costs are taken into account. The first is the fixed cost f_r which captures all the expenses of including service $r \in R$ in the final plan. The second, is a cost that varies proportionally with the volume of commodity $k \in K$ moved in the network on arc $(i, j) \in \mathcal{A}$. It is denoted by c_{ij}^k and specifically represent:

- the cost associated with the transport of commodity k , if $(i, j) \in A^M$;

- the cost associated with the handling of commodity k at terminals, if $(i, j) \in \mathcal{A}^H$.

The third is the cost in which the carrier incurs if a delay in operations or consignment is observed. It will be described in detail later on.

The *information revelation process* defines how and when the values of stochastic parameters, in our case travel times, are observed. Decisions made without any information about future travel time realizations are design and routing decisions. Once such decisions are made and travel time realizations are available, an evaluation of the reliability (in terms of both service and demand targets) of the set up network may be performed. The evaluation is defined in terms of delays, that is comparing observed and scheduled service arrival times at terminals and freight arrival times at destination, resulting by operating a given service network. To enforce reliability such delays are penalized. Note that, once the service network is established, it cannot be modified, regardless the values of observed travel times (only penalties are paid). The problem may then be represented as a two-stage stochastic optimization model with simple recourse.

In the first stage, planning decisions are made considering their future effects: the selection of the services and the routing of freight are determined with the objective of minimizing the fixed service-selection and variable demand-routing costs, plus the expected costs following the application of the chosen plan to the observed realizations of travel times. Two sets of first stage decision variables are defined, which model selection of services and routing of demand:

- binary variables $y_r \in \{0, 1\}$, $\forall r \in R$ represent whether a service r is selected ($y_r = 1$) in the final plan or not ($y_r = 0$);
- non-negative and continuous variables x_{ij}^k , $\forall k \in \mathcal{K}$, $\forall (i, j) \in \mathcal{A}$ represent the flowing of commodity k in the network. In particular, the amount of commodity k transported on arc $(i, j) \in \mathcal{A}^M$ or waiting at a terminal, if $(i, j) \in \mathcal{A}^H$.

The second stage addresses how to deal with delays for a given travel time realization and a chosen design. Let Ω define the set of possible outcomes of the random variable travel time and let ω be a random element in that set. We use here a leg-notation. A travel time realization of service leg $i(r) \in L(r)$ is denoted $\tau_{i(r)}(\omega)$, $\forall i(r) \in L(r)$, $\forall r \in R$. Three sets of second stage variables are defined, which model for a travel time realization ω leaving times and arrival times of each service from/to each terminal of its route and arrival time of each commodity at destination:

- non-negative and continuous variables $\delta_{i(r)}(\omega)$, $\forall i(r) \in L(r)$, $\forall r \in R$ represent the time instant in which service $r \in R$ begins its movement on service leg $i(r) \in L(r)$;
- non-negative and continuous (dummy) variables $\eta_{i(r)}(\omega)$, $\forall i(r) \in L(r)$, $\forall r \in R$ represent the time instant in which service $r \in R$ ends its movement on service leg $i(r) \in L(r)$;
- non-negative and continuous variables $\varepsilon_k(\omega)$, $\forall k \in K$ represent the time instant in which commodity $k \in K$ arrives at its destination.

The quality targets may be easily expressed through probabilities. Regarding the target related to arrival time, for a direct service $r \in R$ that should achieve at least an $\alpha \cdot 100\%$ on-time arrivals at destination, the target can be expressed as the probability of arriving at destination before the usual arrival time instant, defined $e_{i(r)}$, that is, $P(\eta_{i(r)}(\omega) \leq e_{i(r)}) \geq \alpha$. Similar expressions may also be used to represent the targets of services with intermediary stops (the expression has to hold

independently for each intermediary stop) and for on-time delivery of demand. In our formulation, we do not consider probabilistic constraints to control the satisfaction of targets, rather we model the same underlying significance of them by penalizing appropriately observed lateness. Lateness of a service is considered as soon as the observed arrival time at a stop exceeds the usual arrival time at that stop. A penalty proportional to the difference between $\eta_{i(r)}(\omega)$ and $e_{i(r)}$ is then applied. The penalty to pay if service r is late on service leg $i(r)$ is denoted by $\lambda_{i(r)}^r$. The same idea is followed to model target of services with intermediary stops and the target of demand. The latter is modeled by penalizing the excess of time between the actual arrival time of commodity k at destination, $\varepsilon_k(\omega)$, and its due date $b(k)$. The penalty cost is denoted by λ^k .

Regarding the target related to the maximum delay, the target of a direct service can be expressed through probabilities as well: $P(\eta_{i(r)} \leq B) = 1$, where B represents the maximum acceptable (or long) delay. These targets are modeled through penalties as well in the model. A very high penalty proportional to the difference between $\eta_{i(r)}(\omega)$ and B is applied. The penalty cost is denoted $\Lambda_{i(r)}^r$. The same idea is followed to model the target of maximum delay of demand: a penalty proportional to the difference between the actual arrival time of commodity k , $\varepsilon_k(\omega)$, and its maximum allowed delay B^k is applied. This fixed penalty cost is denoted by Λ^k .

Different penalties may be considered if the delay regards a service or the transported demand. Note that, the lateness of a service at a particular stop does not always imply that the transported demand is also late (in fact, demand could be shipped in advance with respect to its due date to its final destination), but implies, as said, a loss of reputation and reliability for the carrier, as well as potentially, additional costs for crews and resource utilization. At the same time, it could happen that some demand is late at its particular destination also when the service on which it is transported is not.

In order to introduce the model, we define for each node $i \in N$ its set of successor nodes $\mathcal{N}^+(i) = \{j \in \mathcal{N} : (i, j) \in \mathcal{A}\}$ and predecessor nodes, $\mathcal{N}^-(i) = \{j \in \mathcal{N} : (j, i) \in \mathcal{A}\}$. The goal of the SSND-SDT model is to select the services and route the freight in order to satisfy the demand and the quality targets in the most efficient way, that is, minimizing fixed service selecting costs, variable moving costs and the expected extra costs if delays are observed when applying the chosen plan. The two-stage formulation may be written as follows.

$$\min \sum_{r \in R} f_r y_r + \sum_{(i,j) \in \mathcal{A}} \sum_{k \in K} c_{ij}^k x_{ij}^k + \mathbb{E}_{\tau_{i(r)}, i(r) \in L(r), r \in R} [Q(y, x; \tau_{i(r)}(\omega))] \quad (1)$$

$$\sum_{j \in \mathcal{N}^+(i)} x_{ij}^k - \sum_{j \in \mathcal{N}^-(i)} x_{ji}^k = \begin{cases} w(k) & \text{if } i = o(k) \\ 0 & \text{if } i \neq o(k), i \neq d(k) \\ -w(k) & \text{if } i = d(k) \end{cases} \quad \forall i \in N, \forall k \in K \quad (2)$$

$$\sum_{k \in K} x_{i(r)}^k \leq u_r y_r \quad \forall i(r) \in L(r), \forall r \in R \quad (3)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, \forall (i, j) \in \mathcal{A} \quad (4)$$

$$y_r \in \{0, 1\} \quad \forall r \in R \quad (5)$$

where

$$\begin{aligned}
 Q(y, x; \tau_{i(r)}(\omega)) = & \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)}(\omega) - e_{i(r)})_+ + \\
 & \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)}(\omega) - B)_+ + \\
 & \sum_{k \in K} \lambda^k (\varepsilon_k(\omega) - b(k))_+ + \sum_{k \in K} \Lambda^k (\varepsilon_k(\omega) - B^k)_+
 \end{aligned} \tag{6}$$

The objective (1) is to minimize the total cost of the system, which consists of three elements: the fixed cost of operating services, the transportation costs for routing commodities and the expected cost of recourse for applying the chosen plan. The function $Q(y, x; \tau_{i,j}(\omega))$ is dependent on both design decisions and routing decisions and, in addition, on the realizations of the random variable $\tau_{i(r)}$. Equations (2) are the commodity flow conservation constraints. Equations (3) are linking-capacity constraints, which state that a commodity flow may be positive on movement arc $i(r) \in L(r)$ but not exceed the capacity of the service r travelling on it, only if r is selected, that is $y_r = 1$, and have to be 0 otherwise. We assume not capacity restriction at terminals. Relations (4) and (5) are non negativity and binary constraints which define the domains of the decision variables.

The second stage, composed by (6), computes the total penalty costs of service and demand late arrivals, where the first two terms relate to targets of services, the last two to targets of demand and where the operator $(x - y)_+$ returns the difference between x and y if positive and 0 otherwise.

The following equation (7) describes how to compute $\eta_{i(r)}(\omega)$ for each service where we defined $o(r)$ the origin terminal of service r and $\zeta_{o(r)}$ the scheduled departure time from it:

$$\eta_{i(r)}(\omega) = \delta_{i(r)}(\omega) + \max(\hat{\tau}_{i(r)}, \tau_{i(r)}(\omega)) \quad \forall i(r) \in L(r) \tag{7}$$

and

$$\delta_{i(r)}(\omega) = \begin{cases} \zeta_{i(r)} & \text{if } i = o(r) \\ \eta_{i^-(r)}(\omega) + t & \text{if } i \neq o(r) \end{cases} \quad \forall i(r) \in L(r), \forall r \in R \tag{8}$$

If the observed travel duration $\tau_{i(r)}(\omega)$ is lower than the “usual” one, the service arrives early but have to wait to begin terminal operations, the actual travel time is then considered as that one of the point forecast ($\hat{\tau}_{i(r)}$). If travel duration is higher than the “usual” one, terminal operations begin as soon as the service r finishes its movement on that leg ($\delta_{i(r)}(\omega) + \tau_{i(r)}(\omega)$). This time instant is directly related to the moment in which the movement on that leg may start. Equations (8) define those instants. It is easy to compute for direct services or, at least, for each initial leg, since it is equal to the planned leaving time from service origin (first part of (8)). If a service has an intermediate stop, instead, its leaving time from it is dependent on what happened on the previous service leg, denoted $i^-(r)$, and it is computed as the summation of the arrival time at that stop (that is, the ending time instant of the previous leg $i^-(r)$) plus the deterministic service time t .

It is worth to notice that the model may be easily modified if the interest is only focused on one of the considered targets, by considering only the penalties related to that target. That is, if

the focus is only the target of demand, then the penalties related to the target of services have to be fixed to 0 still maintaining the penalties for the target of demand.

As often done in stochastic programming, the random probability distribution of the stochastic phenomenon is approximated by a set of scenarios, a set of possible realizations, in our case, of travel times, that reasonably are representative of the future. By modeling uncertainty through scenarios the stochastic problem becomes a deterministic mixed integer linear program, which may be solved exploiting technique used in deterministic optimization, even though it may become generally of very large dimensions.

Let S represent the set of scenarios and let s be an element of S . Each scenario s has dimension $|A_{phys}|$. A probability p_s is assigned to each scenario, such that $p_s \leq 1, \forall s \in S$ and $\sum_{s \in S} p_s = 1$. The above mentioned expected costs of applying a chosen plan in the objective function is, then, computed considering the latter probabilities and the delays calculated in the second stage, considering the time realizations of set S , denoted $\tau_{i(r)}(s)$. The sets of second stage variables are, thus, defined, as follows:

- non-negative and continuous variables $\delta_{i(r)s}, \forall i(r) \in L(r), \forall r \in R$ represent the time instant in which service $r \in R$ begins its movement on service leg $i(r) \in L(r)$ in scenario $s \in S$;
- non-negative and continuous (dummy) variables $\eta_{i(r)s}, \forall i(r) \in L(r), \forall r \in R$ represent the time instant in which service $r \in R$ ends its movement on service leg $i(r) \in L(r)$ in scenario $s \in S$;
- non-negative and continuous variables $\varepsilon_{ks}, \forall k \in K, \forall s \in S$ represent the time instant in which commodity $k \in K$ arrives at its destination in scenario $s \in S$.

The two-stage formulation may be written as follows.

$$\min \sum_{r \in R} f_r y_r + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{s \in S} p_s [Q(y, x; \tau_{i(r)}(s))] \quad (9)$$

$$\sum_{j \in N^+(i)} x_{ij}^k - \sum_{j \in N^-(i)} x_{ji}^k = \begin{cases} w(k) & \text{if } i = o(k) \\ 0 & \text{if } i \neq o(k), i \neq d(k) \\ -w(k) & \text{if } i = d(k) \end{cases} \quad \forall i \in N, \forall k \in K \quad (10)$$

$$\sum_{k \in K} x_{i(r)}^k \leq u_r y_r \quad \forall i(r) \in L(r), \forall r \in R \quad (11)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, \forall (i, j) \in A \quad (12)$$

$$y_r \in \{0, 1\} \quad \forall r \in R \quad (13)$$

where

$$\begin{aligned} Q(y, x; \tau_{i(r)}(s)) = & \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)s} - e_{i(r)})_+ + \\ & \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)s} - B)_+ + \\ & \sum_{k \in K} \lambda^k (\varepsilon_{ks} - b(k))_+ + \sum_{k \in K} \Lambda^k (\varepsilon_{ks} - B^k)_+ \end{aligned} \quad (14)$$

4 A Progressive Hedging-Based Meta-heuristic

The meta-heuristic we propose is based on the scenario-decomposition idea along the *principle of progressive hedging* as determined by Rockafellar and Wets [Rockafellar and Wets, 1991]. The idea is to decompose the original multi-scenario stochastic problem into sub-problems according to its set of scenarios and exploit this reduction to come up with a “well hedged” solution for the original problem.

While the convergence of such an approach to a global optimum has been proved for continuous stochastic programs [Rockafellar, 1982], it may not converge in the integer case: every attempt to apply this approach to integer programming formulations, thus, results as a heuristic. The method, however, has been proven to be computationally efficient in various type of integer and mixed-integer problems as operation planning [Gonçalves et al., 2012], lot-sizing [Haugen et al., 2001], portfolio management [Mulvey and Vladimirov, 1991], unit commitment and server location [Guo et al., 2015; Gade et al., 2016], scheduling [Carpentier et al., 2013], resource allocation [Watson and Woodruff, 2011] and network design problems [Mulvey and Vladimirov, 1991; Hvattum and Løkketangen, 2009; Fan and Liu, 2010].

In [Crainic et al., 2011b], a multicommodity SND problem with stochastic demand has been presented alongside with a two-stage stochastic programming formulation where design decisions make up the first stage and routing of commodities make up the second stage, according to observed demand. As a resolution method, they proposed a two phases algorithm, the first phase being directly inspired by Rockafellar and Wets [Rockafellar and Wets, 1991].

First a scenario-based decomposition is applied to the original multi-scenario stochastic problem. Scenario decomposition, as traditionally done, is performed by relaxing first stage variables’ non-anticipativity constraints through the augmented Lagrangean method (we refer to [Bertsekas, 2014; Rockafellar, 1982] for more details on it). This yields to a set of deterministic sub-problems, one for each scenario. At each iteration of the first phase, scenario sub-problems are solved and their solutions (most likely defining different designs) are exploited to have an estimation of the solution of the original problem, through an aggregation operator. This estimation, called overall solution and defined as the expectation over the current local solutions, is only used as an indicator of existing trends among scenarios. Overall and sub-problem (or local) solutions are, then, compared. Non-anticipativity is gradually enforced by the appropriate modification of the augmented Lagrangean multipliers and fixed costs based on the deviations of local solutions from the overall solution. The modification rewards the proximity to or penalizes the distance from it, trying to consolidate local solutions into a unique one, iteration after iteration, for the original problem. The goal of this first phase is to recognize trends among scenario solutions identifying a subset of arcs for which “consensus” appears possible. Cost adjustment, therefore, is only used to guide local scenario designs toward consensus, without forcing it. The second phase of the meta-heuristic is performed after a certain amount of iterations of the first phase, if consensus has only been reached for a subset of first stage variables. It solves a the multi-scenario formulation reduced on the design arcs for which consensus has not been found through the iterative search phase, fixing those design arcs for which instead consensus has been reached.

When considering a progressive hedging approach, thus, three aspects have to be carefully defined:

- the *decomposition strategy* (which may involve building sub-problems for each scenario or for

a cluster of scenarios);

- the *aggregation process* to synthesize local solutions in the estimated solution (traditionally, the average over the current local solutions);
- the *search of consensus* strategy, which taking advantage of local information yielded by sub-problem solutions, drives the search mechanism toward a unique solution for the original problem (it normally involves adjusting multipliers or costs appropriately).

In our case, the formulation of the problem involves both design and routing variables as first stage decisions (note that, routing in the previous work was not considered as a first stage variable). The before described standard application of the methodology, in our case, leads to an non-easy reformulation to deal with. Therefore, the introduction of some modifications has been required, firstly, to allow to still work with a linear reformulation, as later described in detail, instead of a quadratic reformulation and, second, to avoid degeneration problems during the search of consensus strategy. A new methodological framework is, thus, here proposed. In the following, we first describe the standard development of the methodology applied to our problem and, then our proposal.

In order to have scenario separability, a reformulation of the original model is needed. Firstly, a copy of each first stage decision variable has to be defined for each scenario. In our case, it involves creating copies of design and routing variables: $y_{rs}, \forall r \in R, \forall s \in S$ and $x_{ijs}^k, \forall k \in K, \forall (i, j) \in A, \forall s \in S$. This yields to the following reformulation of the problem:

$$\begin{aligned} \min \sum_{s \in S} p_s \left(\sum_{r \in R} f_r y_{rs} + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ijs}^k + \right. \\ \left. \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)s} - e_{i(r)})_+ + \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)s} - B)_+ + \right. \\ \left. \sum_{k \in K} \lambda^k (\varepsilon_{ks} - b(k))_+ + \sum_{k \in K} \Lambda^k (\varepsilon_{ks} - B^k)_+ \right) \end{aligned} \quad (15)$$

$$\sum_{j \in N^+(i)} x_{ijs}^k - \sum_{j \in N^-(i)} x_{jis}^k = \begin{cases} w(k) & \text{if } i = o(k) \\ 0 & \text{if } i \neq o(k), i \neq d(k) \\ -w(k) & \text{if } i = d(k) \end{cases} \quad \forall i \in N, \forall k \in K, \forall s \in S \quad (16)$$

$$\sum_{k \in K} x_{i(r)s}^k \leq u_r y_{rs} \quad \forall i(r) \in L(r), \forall r \in R, \forall s \in S \quad (17)$$

$$x_{ijs}^k = x_{ijs}^{k'} \quad \forall k \in K, \forall (i, j) \in A, \forall s, s' \in S, s \neq s' \quad (18)$$

$$y_{rs} = y_{rs'} \quad \forall r \in R, \forall s, s' \in S, s \neq s' \quad (19)$$

$$x_{ijs}^k \geq 0 \quad \forall k \in K, \forall (i, j) \in A, \forall s \in S \quad (20)$$

$$y_{rs} \in \{0, 1\} \quad \forall r \in R, \forall s \in S \quad (21)$$

Constraints (18) and (19) are the *non-anticipativity* (or implementability) constraints. They make sure that design and routing variables are not tailored for each single scenario, rather define a unique and implementable solution. Whereas such condition is implicit in the formulation (9) - (13), in the reformulation (15) - (21) is explicitly stated as constraints through (18) and (19). That is, the two formulations are equivalent. Scenario separability is, then, achieved through the relaxation of these constraints.

The number of non-anticipativity constraints, however, may become quite large given the size of the set S (as observed in [Crainic et al., 2011b]). Therefore, another way to express the non-anticipativity constraints can be considered. Let \bar{y}_r and \bar{x}_{ij}^k define respectively a feasible design and routing solutions. An equivalent way to impose that all designs and routings must be equal to each other ((18) and (19)) is to require that each scenario design and routing must be equal to the latter mentioned feasible and fixed design and routing solutions. Therefore, (18) and (19) may be replaced by

$$y_{rs} = \bar{y}_r \quad \forall r \in R, \forall s \in S \quad (22)$$

$$\bar{y}_r \in \{0, 1\} \quad \forall r \in R \quad (23)$$

$$x_{ijs}^k = \bar{x}_{ij}^k \quad \forall k \in K, \forall (i, j) \in A, \forall s \in S \quad (24)$$

$$\bar{x}_{ij}^k \geq 0 \quad \forall k \in K, \forall (i, j) \in A \quad (25)$$

Constraints (22) and (24) require each scenario design and each scenario routing to be equal, respectively, to a fixed design and to a fixed routing solution, which as imposed by constraints (23) and (25), satisfy the traditional binary and non-negativity conditions. We define in the following the \bar{y}_r and \bar{x}_{ij}^k respectively as overall design and overall routing solutions.

By considering the latter constraints for the non-anticipative requirements and performing relaxation of (22) and (24) through the augmented Lagrangean method, the following objective function is, then, obtained:

$$\begin{aligned} \min \sum_{s \in S} p_s & \left(\sum_{r \in R} f_r y_{rs} + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ijs}^k + \right. \\ & \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)s} - e_{i(r)})_+ + \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)s} - B)_+ + \\ & \sum_{k \in K} \lambda^k (\varepsilon_{ks} - b(k))_+ + \sum_{k \in K} \Lambda^k (\varepsilon_{ks} - B^k)_+ + \\ & \sum_{r \in R} \phi_{rs} (y_{rs} - \bar{y}_r) + \frac{1}{2} \sum_{r \in R} \rho (y_{rs} - \bar{y}_r)^2 + \\ & \left. \sum_{(i,j) \in A} \phi_{ijs}^k (x_{ijs}^k - \bar{x}_{ij}^k) + \frac{1}{2} \sum_{(i,j) \in A} \psi (x_{ijs}^k - \bar{x}_{ij}^k)^2 \right) \end{aligned} \quad (26)$$

where ϕ_{rs} and ϕ_{ijs}^k are respectively the Lagrangean multipliers used to relax constraints (22) and (24), and ρ and ψ are penalty ratios. Note that the differences between each scenario solution and overall solutions can be penalized individually. The latter objective function may be reduced by taking advantage of the binary requirements of the service design variables, as follows:

$$\begin{aligned}
 & \min \sum_{s \in S} p_s \left(\sum_{r \in R} (f_r + \phi_{rs} + \frac{1}{2}\rho - \rho\bar{y}_r)y_{rs} + \right. \\
 & \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k + \phi_{ijs}^k + \frac{1}{2}\psi x_{ijs}^k - \psi\bar{x}_{ij}^k)x_{ijs}^k + \\
 & \quad \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)s} - e_{i(r)})_+ + \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)s} - B)_+ + \\
 & \quad \left. \sum_{k \in K} \lambda^k (\varepsilon_{ks} - b(k))_+ + \sum_{k \in K} \Lambda^k (\varepsilon_{ks} - B^k)_+ \right)
 \end{aligned} \tag{27}$$

Formulation (27), (16), (17), (20) and (21) is scenario separable, once an overall design \bar{y}_r and an overall routing \bar{x}_{ij}^k are fixed to given values. The model, then, decomposes according to the scenarios of set S .

The traditional application of the progressive hedging approach to our problem highlights two disadvantages. First, even though the difficulty related to the size of the original problem is split and distributed among sub-problems, each sub-problem assumes now a non-linear reformulation. In fact, if on one side the binary requirements of the service design variables allow a reduction of part of the objective function (26), the continuous nature of routing variables does not allow it and the reformulation ends up having quadratic routing variables, needing appropriate resolution methods.

In addition, such an approach involves the search of consensus of both service and routing decisions. In general, SND problems are degenerative in the sense that for a given network design several equivalent - in terms of costs - but different - in terms of paths - routings may be defined. Similarly, the same degeneracy may be observed during the first phase of the algorithm: if consensus is reached for design variables, several equivalent flow path solutions could be found for it, lengthening the convergence of phase one. The search of consensus seems, thus, to be inconvenient for such variables. In order to have a linear reformulation and overcome degeneracy problems, we do not consider all the first stage variables for consensus. Rather, we introduce a hierarchy of ‘‘importance’’ on first stage variables, looking for consensus only on a subset of them. In particular, consensus is still sought for design variables. Non-anticipativity design constraints (19) are relaxed at the beginning through the augmented Lagrangean method and enforced during the execution of the algorithm. As opposed, to the non-anticipativity routing constraints (18) a linear relaxation is applied. This leads to a linear reformulation of the problem, but stress even more the heuristic behavior of the approach.

The methodological approach proposed here, then, will be applied using the following reformulation of the problem:

$$\begin{aligned}
 \min \sum_{s \in S} p_s & \left(\sum_{r \in R} (f_r + \phi_{rs} + \frac{1}{2}\rho - \rho\bar{y}_r) y_{rs} + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ijs}^k + \right. \\
 \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)s} - e_{i(r)})_+ & + \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)s} - B)_+ + \\
 \sum_{k \in K} \lambda^k (\varepsilon_{ks} - b(k))_+ & + \left. \sum_{k \in K} \Lambda^k (\varepsilon_{ks} - B^k)_+ \right) \quad (28)
 \end{aligned}$$

$$\sum_{j \in N^+(i)} x_{ijs}^k - \sum_{j \in N^-(i)} x_{jis}^k = \begin{cases} w(k) & \text{if } i = o(k) \\ 0 & \text{if } i \neq o(k), i \neq d(k) \\ -w(k) & \text{if } i = d(k) \end{cases} \quad \forall i \in N, \forall k \in K, \forall s \in S \quad (29)$$

$$\sum_{k \in K} x_{i(r)s}^k \leq u_r y_{rs} \quad \forall i(r) \in L(r), \forall r \in R, \forall s \in S \quad (30)$$

$$x_{ijs}^k \geq 0 \quad \forall k \in K, \forall (i, j) \in A, \forall s \in S \quad (31)$$

$$y_{rs} \in \{0, 1\} \quad \forall r \in R, \forall s \in S \quad (32)$$

each sub-problem taking the form of:

$$\begin{aligned}
 \min \sum_{r \in R} (f_r + \phi_{rs} + \frac{1}{2}\rho - \rho\bar{y}_r) y_{rs} & + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ijs}^k + \\
 \sum_{r \in R} \sum_{i(r) \in L(r)} \lambda_{i(r)}^r (\eta_{i(r)s} - e_{i(r)})_+ & + \sum_{r \in R} \sum_{i(r) \in L(r)} \Lambda_{i(r)}^r (\eta_{i(r)s} - B)_+ + \\
 \sum_{k \in K} \lambda^k (\varepsilon_{ks} - b(k))_+ & + \sum_{k \in K} \Lambda^k (\varepsilon_{ks} - B^k)_+ \quad (33)
 \end{aligned}$$

$$\sum_{j \in N^+(i)} x_{ijs}^k - \sum_{j \in N^-(i)} x_{jis}^k = \begin{cases} w(k) & \text{if } i = o(k) \\ 0 & \text{if } i \neq o(k), i \neq d(k) \\ -w(k) & \text{if } i = d(k) \end{cases} \quad \forall i \in N, \forall k \in K, \forall s \in S \quad (34)$$

$$\sum_{k \in K} x_{i(r)s}^k \leq u_r y_{rs} \quad \forall i(r) \in L(r), \forall r \in R, \forall s \in S \quad (35)$$

$$x_{ijs}^k \geq 0 \quad \forall k \in K, \forall (i, j) \in A, \forall s \in S \quad (36)$$

$$y_{rs} \in \{0, 1\} \quad \forall r \in R, \forall s \in S \quad (37)$$

Note that now, activation cost of service r is composed by the expression $(f_r + \phi_{rs} + \frac{1}{2}\rho - \rho\bar{y}_r)$. Each sub-problem may be solved separately by applying efficient meta-heuristics approaches or even, when the size of sub-problems allows it, by an exact method. When sub-problems are solved,

from their solutions two kinds of information are simultaneously obtained: a vector which defines the activated service design \mathbf{y}^s on scenario s and a vector defining the related routing of flows \mathbf{x}^s . As classically done then, an aggregation operator brings together information about the variables for which consensus is looked for, from a sub-problem level to define what we called the overall design. As said, based on the “distance” of local solutions from it, costs are adjusted to guide toward consensus. We propose a first version of the algorithm only considering such scheme.

Nevertheless, in our case, we still do have additional available information, albeit scenario dependent: the routing of flows. Not considering such flow-information may disregard important insights related to the actual differences there may be in the utilization of the services selected in the different scenario sub-problem solutions. This amount of flow-information, which is given and do not require any additional effort when solving sub-problems, could be exploited as well in a search-of-consensus strategy. We propose, therefore, a second version of the algorithm in which such knowledge is considered in the seeking of consensus strategy.

The search of consensus is the objective of the first phase of the algorithm, at the end of which the second phase is performed. In this phase, a variables-reduced multi-scenario formulation is solved fixing those design arcs for which consensus has been reached, with the aim of finding the final solution of the original problem in terms of routing of freight and remaining design variables. Although some of the design variables are fixed, the remaining design and routing decisions are still taken before any stochastic information is revealed, still maintaining the structure of a two-stage formulation on those variables. The complete PHM procedure is described in the next session.

4.1 The Complete Progressive Hedging-Based Meta-heuristic

Let ν define the iteration index of the meta-heuristic. At the initialization step, $\nu = 0$, $|S|$ SSND-STT problems of the form (33) - (37), where the fixed costs of activating services equal the costs of the original problem, are solved. In the current implementation, sub-problems are solved optimally by using a linear solver. At the end of this step, $|S|$ solutions are found defining different designs and routings.

4.1.1 First Phase

The input of each iteration of Phase 1 is a set of $|S|$ solutions defining designs and routings. They may come directly from the initialization step or from a previous iteration of Phase 1. The operation to carry out at each iteration of Phase 1 are the following:

- extracting the overall design from input solutions;
- modify the augmented Lagrangean parameters and penalty term ρ ;
- adjust the fixed cost considering both the global and the local adjustments rules.

After having performed the latter operations, sub-problems are solved again. Each sub-problem at each iteration takes the form of a deterministic SND problem with possibly modified costs from one iteration to the others in accordance with the before mentioned updates.

Extracting the Overall Design The overall design summarizes into a single design the local information obtained by solving sub-problems and provided by the different scenario designs. It represents both an estimation of global trends and a referent point to “guide” iteration after iteration the sub-problems’ solutions toward consensus. In addition, its scope is also to define the constant values needed to allow separability during the running of the PHM. We use of an average function as aggregation operator (likewise as [Crainic et al., 2011b] and [Rockafellar and Wets, 1991]).

Given the scenario probabilities p_s , a weighted average is used to combine local information into the overall design as shown below:

$$\bar{y}_r^\nu = \sum_{s \in S} p_s y_{rs}^\nu \quad \forall r \in R \quad (38)$$

The values of \bar{y}_r^ν are between 0 and 1, namely $\bar{y}_r^\nu \in [0, 1]$. When all scenarios agree on the selection status of a service r , consensus is observed and \bar{y}_r^ν assumes an integer value. In particular, if $\bar{y}_r^\nu = 1$, all scenarios agree on activate service r . As opposed, if $\bar{y}_r^\nu = 0$, all scenarios agree on not activate service r . Most of the time, instead, one observes that $0 < \bar{y}_r^\nu < 1$. In this case, a low - close to zero - value for \bar{y}_r^ν indicates a trend toward not activate service r ; symmetrically, a high - close to one - value indicates a trend to activate that service.

Modifying the Augmented Lagrangean Parameters The update is directly inspired by the augmented Lagrangean method and modifies Lagrangean multipliers and parameter ρ considering the information related to design variables provided by the sub-problems resolution [Løkketangen and Woodruff, 1996; Rockafellar and Wets, 1991].

The objective of this adjustment is to give an incentive to the activation or not activation of a service when its status is different from that one in the current reference overall design. For any design variable y_{rs}^ν in a scenario sub-problem s , at iteration ν three different occurrences may be observed (note that the scenario sub-problem design variable y_{rs}^ν assumes only integer values, namely either 0 or 1):

- $y_{rs}^\nu < \bar{y}_r^\nu$. In this case, service r is not activated in scenario-design s , but the trend (the reference point \bar{y}_r^ν) suggests that not all the other scenarios agree on this decision. The idea is, then, to promote the activation of that service by reducing its cost in the scenario sub-problem s ;
- $y_{rs}^\nu > \bar{y}_r^\nu$. This is the opposite situation, in which service r is activated in scenario-design s , but not all other scenarios agree on this decision. The cost is adjusted so as to give a disincentive to activate that service within the scenario sub-problem s . The more sub-problems do not activate that service the stronger is the disincentive;
- $y_{rs}^\nu = \bar{y}_r^\nu$. Here universal consensus is observed among the scenario sub-problems. This may concern both the activation or not activation of a service. The fixed cost, in this case, remains unchanged.

Let $\phi_r^{s\nu}$ be the value of the Lagrangean multiplier associated with the non-anticipativity constraint of design variable r of scenario sub-problem s at iteration ν , and let ρ^ν be the value of the penalty ratio at the same iteration. The parameters are then updated as follows:

$$\phi_{rs}^\nu \leftarrow \phi_{rs}^{\nu-1} + \rho^{\nu-1}(y_{rs}^\nu - \bar{y}_r^{\nu-1}) \quad \forall r \in R \quad (39)$$

$$\rho^\nu \leftarrow \gamma \rho^{\nu-1} \quad (40)$$

Update rule (39) represents the steepest ascent step in the space of the dual problem [Mulvey and Vladimirov, 1991] and depends on parameter ρ^ν . Initially, ρ^0 is set to an arbitrarily positive small value and is dynamically adjusted at each iteration through parameter γ , which is a constant $\gamma > 1$. Although dynamic adjustments of the penalty parameter are not covered by the convergence theory for the progressive hedging algorithm, in [Mulvey and Vladimirov, 1991] is found that this strategy can improve the overall convergence behavior.

Fixed Costs Adjustments Fixed cost adjustment modifies, globally at each iteration, the fixed costs of the design arcs with a status different from what a majority of the other arcs agree upon at the current iteration [Crainic et al., 2011b].

Starting from the reference point \bar{y}_r^ν at iteration ν , the global adjustment attempts to favor what appears as the current trend among scenarios to include or exclude service r in the final network design. As explained before, a low value of \bar{y}_r^ν indicates that the trend is to not activate service r , as it is activated only in a small number of sub-problems. As opposed, a high value of \bar{y}_r^ν indicates that the trend is to activate that service, meaning that it is activated within the majority of scenario designs.

As discussed in [Crainic et al., 2011b], one can conclude that when \bar{y}_r^ν is less than a given threshold $thres_{low}$, increasing the cost f_r of service r may drive the sub-problems to avoid activating it. On the other hand, when \bar{y}_r^ν is higher than a given threshold $thres_{high}$, lowering the cost f_r of service r should attract the scenario sub-problems to activate it. The thresholds may be fixed as less and more than half the scenarios.

When only design information are considered in costs adjustments, three occurrences may be observed and, following [Crainic et al., 2011b], cost modification operates in the following way:

$$f_r^\nu = \begin{cases} \beta f_r^\nu & \text{if } \bar{y}_r^{\nu-1} < thres_{low} \\ \frac{1}{\beta} f_r^\nu & \text{if } \bar{y}_r^{\nu-1} > thres_{high} \\ f_r^\nu & \text{otherwise} \end{cases} \quad (41)$$

with parameters $\beta > 1$, $0 < thres_{low} < 0.5$, $0.5 < thres_{high} < 1$ and f_r^ν representing the modified activation cost of service r at iteration ν .

Exploiting Local Flow Information We also propose a further cost adjustment at a local level, which exploits the information related to the level of utilization of each activated service at a locally. This information is given when sub-problems are solved and does not imply extra efforts to have it.

At iteration ν for a given service r and sub-problem s , two information, as said, are available: the trend among scenarios related to the status of that service reflected by the reference point \bar{y}_r^ν and its utilization rate in sub-problem s reflected by the sum of flows of the different commodities passing through the service leg $i(r)$ on which service r travels along, defined as $\sum_{k \in K} x_{i(r)}^{k\nu}$. In this case, four occurrences may be observed:

$$\bar{y}_r^\nu > \text{thres}_{high} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} > \text{thres}_{high}^k \quad (42)$$

$$\bar{y}_r^\nu < \text{thres}_{low} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} < \text{thres}_{low}^k \quad (43)$$

$$\bar{y}_r^\nu > \text{thres}_{high} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} < \text{thres}_{low}^k \quad (44)$$

$$\bar{y}_r^\nu < \text{thres}_{low} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} > \text{thres}_{high}^k \quad (45)$$

with parameters $0 < \text{thres}_{low} < u_r/2$, $u_r/2 < \text{thres}_{high} < u_r$.

Case (42) defines a situation in which the trend among scenarios is to activate service r operating on service leg $i(r)$ ($\bar{y}_r^\nu > \text{thres}_{high}$), which is also highly utilized at a flow level in sub-problem s ($\sum_{k \in K} x_{i(r)s}^{k\nu} > \text{thres}_{high}^k$). The local adjustment we propose attempts, then, to favor the utilization of this service, lowering its activation cost and trying to “push” consensus toward the activation of service r .

Case (43) defines a situation in which the trend among scenarios is to not activate service r operating on service leg $i(r)$ ($\bar{y}_r^\nu < \text{thres}_{high}$), which is also slightly utilized at a flow level in sub-problem s , if the service r is active ($\sum_{k \in K} x_{i(r)s}^{k\nu} < \text{thres}_{high}^k$). The local adjustment attempts, then, to discourage the utilization of this service, increasing its activation cost and trying to “push” consensus towards the not activation of service r .

Case (44) defines the occurrence in which the trend among scenarios is to activate service r ($\bar{y}_r^\nu > \text{thres}_{high}$), which is however slightly utilized at a flow level in sub-problem s ($\sum_{k \in K} x_{i(r)s}^{k\nu} < \text{thres}_{high}^k$). The service leg $i(r)$ operated by service r may be, thus, interpreted as a “safe” leg, which is activated in order to hedge against uncertainty, although it is not used at its maximum capacity. The local adjustment attempts, then, to favor the utilization of this service, lowering the activation cost of the service traveling through it. The cost, though, is not decreased with the same degree as in case (42). Incentive and disincentive for the latter occurrences are shown below.

$$c_{rs}^\nu = \begin{cases} \frac{1}{\beta^2} f_r^\nu & \text{if } \bar{y}_r^\nu > \text{thres}_{high} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} > \text{thres}_{high}^k \\ \frac{1}{\beta} f_r^\nu & \text{if } \bar{y}_r^\nu > \text{thres}_{high} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} < \text{thres}_{low}^k \\ \beta f_r^\nu & \text{if } \bar{y}_r^\nu < \text{thres}_{low} \quad \text{and} \quad \sum_{k \in K} x_{i(r)s}^{k\nu} < \text{thres}_{high}^k \\ f_r^\nu & \text{if } \bar{y}_r^{\nu-1} = 0.5 \end{cases} \quad (46)$$

Condition (45) appears as the most interesting. Here, the majority of the scenario sub-problems deactivate certain services, which are not only activated but also heavily used in a small subset of scenario sub-problems. In this occurrence, thus, service r operating on service leg $i(r)$ is activated and highly used in scenario sub-problem s ($\sum_{k \in K} x_{i(r)s}^{k\nu} > \text{thres}_{high}^k$). Service r , however, in the majority of the other scenario sub-problems is not activated ($\bar{y}_r^\nu < \text{thres}_{high}$).

The question raised here is how important these services could be with respect to the final design. That is, how a small subset of highly used but not agreed upon services should weight and influence a bigger set of services on which the majority of sub-problems agree on their (not activated) status. Should this minority of services be included or excluded from the final plan?

Should the methodology simply ignore this small set of services, considering also that each activated service involves costs?

The main problem is, therefore, how to consider the minority of services which satisfy condition (45). The strategy we propose, here, is described in the following. We try to limit at each iteration the number of services satisfying condition (45). When sub-problem s is solved, the latter four conditions are verified. The number of services satisfying condition (45) are counted. In the next iteration when the same sub-problem s is solved again, we try to solve it by limiting the number of those services. This goal is reached by adding to the linear reformulation of sub-problems, an additional constrain. Let $h_s^{\nu-1}$ be the number of services satisfying condition (45) at iteration $\nu-1$ for sub-problem s , at iteration ν sub-problem s is solved (after the usual costs and multipliers modifications) by constraining the number of those service at $h_s^\nu = h_s^{\nu-1} - 1$. If a solution is found trend and costs modifications allow to switch to other services. If a solution is not found, the latter are necessary services to find a feasible solution in sub-problems s and therefore are fixed at an overall level in next iterations.

In order to limit the number of those services, a new constrain, as said, is added to sub-problem s . We define the set $C_{4s}^\nu = \{r \in R : \bar{y}_r^\nu < thres_{low} \text{ and } \sum_{k \in K} x_{i(r)s}^{k\nu} > thres_{high}^k\}$ for each sub-problem s at iteration ν . In addition, we define the parameter $a_{rs}^\nu, \forall r \in R, \forall s \in S$ which assumes value 1 if service $r \in C_{4s}^\nu$ and 0 otherwise. Let $h_s^{\nu-1}$ define $|C_{4s}^{\nu-1}|$, the new constrain added at iteration ν to sub-problem s is then

$$\sum_{r \in R} a_{rs}^\nu y_{rs} \leq h_s^{\nu-1} - 1 \quad (47)$$

4.1.2 Second Phase

The second phase of the meta-heuristic is performed after a certain amount of iterations of the first phase, if consensus has only been reached for a subset of first stage design variables. The input of this second phase is, thus, the original multi-scenario problem reduced in terms of variables. That is, design variables for which consensus has been reached through the first phase are fixed and the multi-scenario problem is only solved considering the whole set of routing variables and the design variables for which consensus has not been reached. The output of the second phase gives the solution of the original problem.

Algorithm 1 sums up the entire procedures described above. We define the PHM considering only design cost adjustment PHM-D and the version considering design and routing costs adjustment PHM-DF.

Algorithm 1 The Hierarchic Progressive Hedging-Based Meta-heuristic

Initialization $\nu = 0$

- 1: $\phi_{rs}^\nu \leftarrow 0, \forall r \in R, \forall s \in S$;
- 2: $\rho^\nu \leftarrow \rho^0$;
- 3: $c_{rs}^\nu \leftarrow f_r, \forall r \in R, \forall s \in S$;
- 4: Solve the corresponding $|S|$ SSND-SDT sub-problems;

First Phase:

- 5: **while** stopping criterion is not met **do**
- 6: $\bar{y}_r^\nu \leftarrow \sum_{s \in S} p_s y_{rs}^\nu, \forall r \in R$;
- 7: adjust globally $f_r^\nu, \forall r \in R$ using equation (41);
- 8: **if** PHM-DF = TRUE **then**
- 9: adjust locally $c_{rs}^\nu, \forall r \in R, \forall s \in S$ using equation (46);
- 10: $h_s^\nu \leftarrow |C_{4s}^\nu|, \forall s \in S$;
- 11: add constrain (47) with $h_s^{\nu-1}$ to sub-problem s ;
- 12: **end if**
- 13: Solve the $|S|$ SSND-STT modified sub-problems;
- 14: **if** PHM-DF = TRUE **then**
- 15: **while** Solve == TRUE **do**
- 16: $h_s^{\nu-1} ++$;
- 17: **end while**
- 18: **end if**
- 19: $\nu \leftarrow \nu + 1$;
- 20: Update:

$$\phi_{rs}^\nu \leftarrow \phi_{rs}^{\nu-1} + \rho^{\nu-1}(y_{rs}^\nu - \bar{y}_r^{\nu-1}), \forall r \in R, \forall s \in S;$$

$$\rho^\nu \leftarrow \gamma \rho^{\nu-1};$$
- 21: **end while**

Second Phase:

- 22: Fix the design variables for which consensus is obtained;
- 23: Solve the restricted multi-scenario formulation.

5 Experimental Plan

We propose two sets of experiments. The first has the scope of quantifying the benefits of explicitly considering stochastic travel time in the decision model rather having a deterministic time assumption. Therefore, this experiment only involves the resolution of small and medium-sized problem instances exactly. A number of instances with different characteristics - in terms of level of variability, number of commodities, wideness of delivery time windows and penalty costs - are solved considering the stochastic and the deterministic formulations. A stochastic formulation may easily be transformed into its respective deterministic counterpart by replacing the stochastic parameter by a point forecast. The results are, then, evaluated through a Monte Carlo simulation and compared. In addition, we also investigate the impact that the penalty parameter has on stochastic solutions, in particular, on the reliability of the resulting service networks. Problem instances are, thus, solved for increasing values of penalties, keeping all the other parameters (mentioned above)

fixed.

The second part, instead, focuses on the algorithm and has the scope of qualifying the performance of the proposed meta-heuristic. A subset of instances used in the previous experimentations are addressed with the PHM (both PHM-D and PHM-DF approaches). Meta-heuristic results are, then, compared with exact solutions obtained by solving directly the multi-scenario formulations with *CPLEX*. In order to quantify the gain of using local flow information, solutions obtained by the PHM-D and PHM-DF are compared. Emphasis is also given to the impact of different parameter settings of the PHM.

Both mixed-integer linear programming models (deterministic and stochastic) were implemented in OPL language as well as the PHM. All experiments were conducted on an Intel Xeon X5675 with 3.07 GHz and 48 GB of RAM. Instances were solved by a standard linear programming solver, namely *CPLEX* 12.6 (IBM ILOG, 2016) with a branch and bound method.

In the following, problem instances, scenario generation procedures, results and analyses are presented.

5.1 Instances and Scenario Generation

The physical service network we consider in all our experimentations is inspired by that one used in [Crainic et al., 2014b], which consists of 5 physical nodes and 10 physical arcs (and shown in Figure 1).

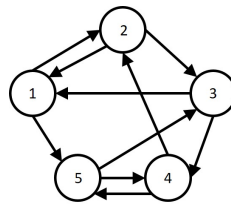


Figure 1: Physical Service Network

The service network is repeated for 15 periods and, as in [Crainic et al., 2014b], has a cyclic nature (see Figure 2).

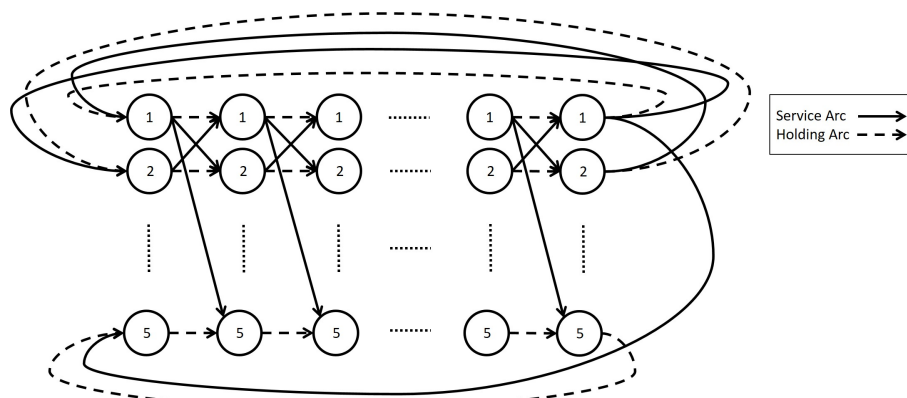


Figure 2: Time-Expanded Service Network

We consider 8 *problem classes* defining demand, differing in number of commodities and wide-

ness of delivery time windows. Each problem class (*Pclass*–) is identified by a couple of values. The first defines the *level of demand* 1, 2, 3 or 4 which refers, respectively, to a low (15 commodities), a medium (20 commodities and 25 commodities) and a relatively high number of commodities (50 commodities). The second value refers to the wideness of delivery time windows, l or t . The first is loose (l) and considers due dates after 11 – 14 periods after the availability dates of each commodity (over the total schedule length of 15 periods), the second is tight (t) and considers due dates after 9 – 12 periods after the availability dates (note that 7 is the minimum time period to use two consecutive services).

To answer demand a certain number of direct potential services are available (150 services) and a few number of potential services with one intermediate stop (7 services). The activation cost of a direct service is proportional to the distance that service covers (services need 3, 4 or 5 time periods to reach their destination). The activation cost of a service with an intermediate stop is 35% less than if for the same path two direct services would be activated. The set of services and their activation costs do not vary across instances.

The random event under study, namely the travel time between terminals in normal conditions, is represented by a random variable which must have specific characteristics. It should have a lower bound, since there is always a minimum time to cover the distance between two points defined by physical constraint (e.g., speed limits). After this minimum time, the probability should rapidly increase to a maximum representing the most usual or observed travel time realization (the mode) after which the probability should slowly decrease with a tail skewed to the right. In our case, the distribution also has an upper bound, since in normal condition infinite travel times are not ascertained (we do not consider a distribution with infinite tails). In our experimentation, thus, the random event is described by a *Truncated Gamma* (TG) probability distribution, which matches all the needed requirements (see Figure 3(a)).

The scenario generation process is, thus, performed by generating random values from a set of TG distributions (which differ by several characteristics, described in the following). A TG depends on two parameters alike a classical gamma distribution: a shape parameter and a scale parameter (for more details we refer to [Okasha and Alqanoo, 2014; Coffey and Muller, 2000; Chapman, 1956]). In our case, those parameters are estimated once the mode, the variance and the range (difference between upper and lower bounds of the truncation) are fixed. The mode is also used as travel time point estimation in the deterministic formulation of the problem.

To better demonstrate how uncertainty affects solutions, we assess 12 scenario classes. We considered 3 variability levels, measured in terms of standard deviation (low, medium and high) and three ranges. If on one side we always consider a same lower bound for the above mentioned distributions, on the other side three different upper bounds are chosen: the first is tight (t , mode –30% of a time period), the second is medium (m , mode +1 time period) and the third is loose (l , mode +130% of a time period). The looser the range is the wider the concept of “normal” travel time. Scenario classes (*Sclass*–) are thus identified by the couple *level of variability* (1, 2 and 3) and *range* (t , m and l).

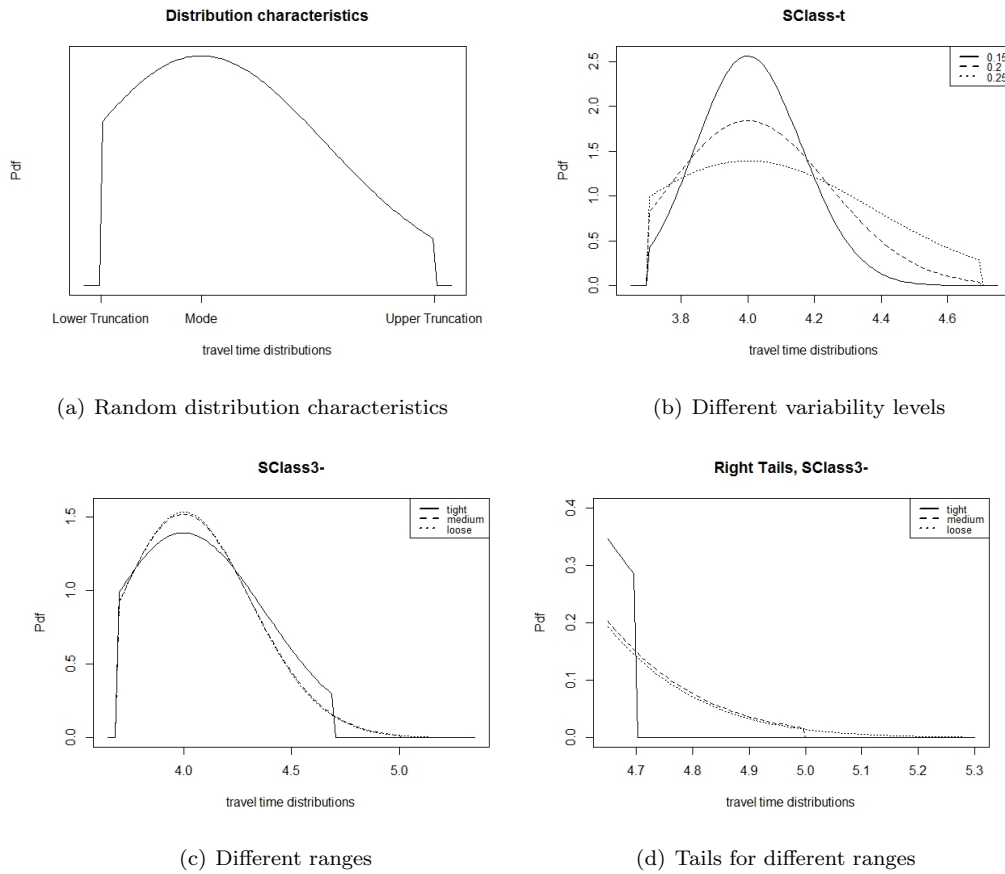


Figure 3: Travel Time Random Distributions

In Figure 3(b) distributions for a same tight range and different standard deviations are plotted, while in Figure 3(c) distributions for the same level of variability and different ranges are shown. Experimentations are performed under 3 different levels of increasing penalty costs. The complete formulation (SSND-SDT) and single target formulations, only the target of service (SSND-ST) and only the target of demand (SSND-DT), are considered. The latter are built by simply fixing at 0 the penalties of the not-considered target.

6 Results and Analysis

Before reporting the results of the above mentioned and described evaluation and comparative analyses, in-sample and out-of-sample stabilities are discussed.

6.1 In-Sample and Out-of-Sample Stability

When random scenario generation procedures (such as sampling from a distribution) are involved, stability requirements assume great importance in order to verify the correctness of the scenario generation procedure and the representativeness of the generated values in order to avoid some kind of bias on the results of the optimization model. Two stability conditions must normally be satisfied by a scenario-generation procedure: *in-sample* and *out-of-sample* stability.

In order to verify stability requirements, tests were conducted only considering the highest variability level (level 3), but varying their ranges (t , m and l), and the first 3 levels of demand. Each instance of each class was solved 10 times by generating new scenario sets.

In-sample stability achieving a difference between the highest and the lowest optimal values across scenario sets always less than 1% for each problem class is obtained by using sets of 30 scenarios. In Table 1(a) average results are shown for the third problem class and the third penalty level.

Out-of-sample stability was tested considering 30-scenario-sized sets to find solutions and 100-scenario-sized sets (generated from the same TG distributions used to construct the scenario sets for the optimization process) as the “true” stochastic phenomenon. A procedure similar to Monte-Carlo simulation is used to evaluate the solutions. The evaluation was performed by fixing the first-stage variables obtained as results of the stochastic programs on the 30-scenario-sized sets, and optimizing the temporal flow by solving again the the second stage on the 100-scenario-sized sets. In all cases, the difference between the highest and the lowest optimal values across scenario sets is less than 3%. To illustrate, we report in Table 1(b) average results for the same problem classes, scenario classes and penalty mentioned above.

Stochastic instances are, thus, solved considering a set size of 30 scenarios which is sufficiently large to ensure a good level of in-sample and out-of-sample stability, while being still easily solvable relatively fast to find optimal solutions. For a complete explanation and deeper details about in- and out-of sample stability, we refer to [Kali and Wallace, 1994] and [Kaut et al., 2007].

(a) In-Sample Stability Test			(b) Out-of-Sample Stability Test		
	PClass-3t	PClass-3l		PClass-3t	PClass-3l
SClass-3t	0.72	0.65	SClass-3t	2.24	0.61
SClass-3m	0.77	0.75	SClass-3m	1.08	1.44
SClass-3l	0.56	0.51	SClass-3l	2.33	0.79

Table 1: Stability conditions for the scenario-generation procedure

6.2 Formulation Evaluation

The evaluation analysis is performed considering 6 types of deterministic problem classes derived by the combined use of the first 3 levels of demand and the 2 different delivery time windows. For each class, we generated 10 instances for a total of 60 deterministic instances. For each deterministic instance, 27 stochastic instances are constructed, combining all the above described parameters between each others (3 levels of variability, 3 ranges and 3 penalty rates).

A solution, whether deterministic (SDM) or stochastic, consists of a set of activated services and the paths used by freight-flows to reach their destinations. Stochastic solutions are found considering the three formulations and are defined as (SSM-ST) when only service target, (SSM-DT) when only demand target and (SSM-SDT) when both are considered. A general comparison between deterministic and stochastic solutions is given related to the *set up cost* of the network: service activation costs plus routing costs. After that, a comparison among SDM, SSM-ST, SSM-DT and SSM-SDT is given considering their behavior when “plunged” in a stochastic environment, defined by a scenario set of 100 scenarios. The focus of this comparison are the costs raised as

penalties.

In general, SDM have the typical characteristics of consolidation-based transportation networks: different commodities share the capacity of single services for the vast majority of their journey, passing through several intermediary stops and often idle there before arriving at destination. In addition, just-in-time arrivals of freight at destination (with respect to due dates) seem to be a widespread feature of SDM. Furthermore, in order to lower fixed costs, one-stop services are always privileged when possible with respect to no-stop services (which are more expensive in our setting).

SSM-ST set up costs are not very dissimilar from the corresponding SDM (note that we are comparing set up costs and not objective values). However, the set up costs of the SSM-ST highlight an interesting characteristic, which also represents the substantial difference between SSM-ST and SDM. Comparing the number of activated services, in almost all cases less services operate in SSM-ST than in SDM, even though the solutions share part of them. This may be explained as follows. In the experimentation, we assumed that travel time perturbations are independent among service legs, even though delays propagate in the network. If a service has one intermediate stop before reaching its destination and experiences a delay in its first leg, most probably it will arrive at destination (its second stop) later than as scheduled, unless in the second leg the observed travel time is much lower than the forecast and absorbs the delay. In general, having one-stop services means having a higher risk of paying for delays. Stochastic solutions, therefore, habitually move from less expensive indirect services to set up the network, to more expensive direct connections which, however, lower the risk of extra costs when actually the services really operate. The trend of SSM-ST is, thus, to activate only the strictly necessary services to fulfill demand. A higher routing cost seems to be a feature of SSM-ST compared to the SDM. Routing in the stochastic case appears, thus, more tricky since demand, in the majority of the cases, is delivered from their origin to their destination using less services with respect to SDM, leading to freight-paths which are more tangled and with longer idle time at intermediary stops. The set up cost of the network in SSM-ST, therefore, is the result of two opposite effects: on one side fixed costs are lowered as well as the number of activated services, on the other side routing costs are gradually increased. This characteristic of SSM-ST appears as in contrast with the documented effect that stochastic demand has on the design of a service network. In this case, in fact, the number of services is, normally, increased (see [Wang et al., 2016]) in order to hedge the effects of the uncertain phenomenon, here instead is decreased.

To illustrate, average results for instances belonging to the third problem class ($SClass - 3t$ and $SClass - 3l$) solved considering increasing level of variability (scenario classes $SClass - 1m$, $SClass - 2m$ and $SClass - 3m$) under the highest penalty level (level 3) are shown in Table 2, where set up costs, fixed activation service costs, number of activated services and routing costs are reported.

	PClass-3t				PClass-3l			
	Set up	Fixed	Tot.Serv	Routing	Set up	Fixed	Tot.Serv	Routing
SClass-1m	6326,5	136,2	31,3	6190,3	6684,8	167,2	37,4	6517,6
SClass-2m	6330,7	135,9	31,1	6194,8	6705,2	163,1	37	6542,1
SClass-3m	6333,6	131	30,5	6202,6	6713,7	158,4	36,6	6555,3
SDM	6340,8	139,8	31,4	6201	6694,8	177,1	39,5	6517,7

Table 2: Characteristic of SSM-ST

The opposite behavior, is observed for SSM-DT. SSM-DT set up costs are, in general, more expensive than the corresponding SDM. But differently from SSM-ST and similarly to the mentioned results related to stochastic demand, the increase of set up costs is directly related to the increase of the number of activated services. SSM-DT networks are built, in fact, in such a way to limit as much as possible the cases of just-in-time arrivals trying to deliver freight at least one period before due dates, if not even earlier. Sometimes, when an early arrival is not possible for a commodity entirely, it is split and part of it (the majority) is shipped in advance. To allow freight early arrivals, therefore, the number of activated services is increased with respect to SDM to deal with this purpose. Routing costs change as consequent depending on the activated services. If on one side tangle paths are a feature of SSM-ST, on the other side freight arrivals at least one period before due dates seem to be a feature of SSM-SD. Table 3 shows the same average results for the same problem classes, scenario classes and penalty level considered above.

	PClass-3t				PClass-3l			
	Set up	Fixed	Tot.Serv	Routing	Set up	Fixed	Tot.Serv	Routing
SClass-1m	6344,8	144,2	32,4	6200,6	6697	178,4	39,8	6518,6
SClass-2m	6350,6	144,4	32,9	6206,2	6715,2	176,9	40	6538,3
SClass-3m	6351,9	144,1	32,9	6207,8	6717,1	175,9	39,7	6541,2
SDM	6340,8	139,8	31,4	6201	6694,8	177,1	39,5	6517,7

Table 3: Characteristic of SSM-DT

We define the *full cost* of a network as the set up cost plus the penalties raised for delays when simulating the behavior of the network with our Monte Carlo procedure. SSM-ST and SSM-DT show a full cost always lower than the corresponding SDM, showing that considering the stochastic nature of travel time explicitly in the decision process may hedge or, at least, reduce the effects and consequence of its uncertainty, despite an initial higher set up cost. To illustrate, we show in Tables 4 and 5 average results of our Monte-Carlo simulation procedure for the same problem classes, scenario classes and penalty level considered above in order to compare SDM and, respectively, SSM-ST and SSM-DT. In the tables, average full costs and penalties are shown. It is also specified if the average amount of penalty belongs to short or long delay.

	PClass-3t				PClass-3l			
	Full Cost	Tot Penalty	Pen. short	Pen. long	Full Cost	Tot Penalty	Pen. short	Pen. long
SClass-1m	17871,2	11544,7	11118,2	426,4	20636,1	13951,2	13474,9	476,3
SDM	18221,3	11880,5	11437,4	443,0	21550,5	14855,7	14336,4	519,3
SClass-2m	36835,2	30504,48	13162,2	17342,2	42864	36158,8	16443,6	19715,2
SDM	39439,9	33099,1	13214,0	19885,1	51719,8	45025	16959,2	28065,8
SClass-3m	46531,2	40197,5	15873,4	24324,1	54481	47767,7	19589	28178,7
SDM	50362,1	44021,3	16532,2	27489,0	65478,2	58783,4	20940,6	37842,7

Table 4: Monte Carlo simulation analysis of SSM-ST

	PClass-3t				PClass-3l			
	Full Cost	Tot Penalty	Pen. short	Pen. long	Full Cost	Tot Penalty	Pen. short	Pen. long
SClass-1m	8647	2302,2	2222,9	79,3	10186,5	3489,5	3314,7	174,7
SDM	11068,5	4727,7	3069,3	1658,4	14803,6	8108,8	3826,6	4282,2
SClass-2m	13388,8	7038,1	2485,5	4552,6	17215,1	10499,9	3773,5	6726,3
SDM	21332	14991,2	3421,5	11569,6	28410,2	21715,4	4275,3	17440
SClass-3m	15615,6	9263,7	2877,3	6386,4	20735,9	14018,8	4537,2	9481,5
SDM	25199,9	18859,1	4327,9	14531,1	33396,2	26701,4	5335,5	21365,8

Table 5: Monte Carlo simulation analysis of SSM-DT

SSM-SDT appears as the compromise between SSM-ST and SSM-DT, showing a not-direct-services and early-freight-arrivals oriented trends. The SSM-ST component, however, influences SSM-SDT the most. The set up cost, in fact, has exactly the same behavior as in SSM-ST when penalties or variability levels increase: a decrease of the activated services and fixed costs and a consequent increase of routing costs. Nevertheless, its SSM-DT component tries to limit the just-in-time delivery cases as well as favoring deliveries one period before their due dates, if not earlier. Routing appears, thus, very tricky since demand, in the majority of the cases, is not only delivered in advance in order to lower the expenses related to the late arrivals of freight but also moved through the network with less services than in SDM. Freight-paths seem to be even more tangled than in SSM-ST and including longer idle time at some intermediary stops. Average set up costs and performance results are shown in Tables 6 and 7.

	PClass-3t				PClass-3l			
	Set up	Fixed	Tot.Serv	Routing	Set up	Fixed	Tot.Serv	Routing
SClass-1m	6347,2	136,9	31,6	6210,3	6703,5	170,2	38,6	6533,3
SClass-2m	6333,6	136,6	31,4	6219,4	6742,2	161,9	37,3	6580,3
SClass-3m	6365,9	135,1	31,2	6230,8	6765,5	156,4	36,3	6609,1
SDM	6340,8	139,8	31,4	6201	6694,8	177,1	39,5	6517,7

Table 6: Characteristic of SSM-SDT

	PClass-3t					
	Full Cost	Tot Penalty	Pen. short ST	Pen. long ST	Pen. short DT	Pen. long DT
SClass-1m	20245,6	13898,4	11227,4	419,8	2190,4	60,7
SDM	22949,04	16608,2	11437,4	443	3069,3	1658,3
SClass-2m	44871,2	38515,3	13173,7	18233,4	2541,6	4566,5
SDM	54431,2	48090,4	13214	19885,1	3421,5	11569,6
SClass-3m	57512	51146,1	16247,7	25483,8	2986,8	6427,7
SDM	69221,2	62880,4	16532,2	27489	4327,9	14531,1
	PClass-3l					
	Full Cost	Tot Penalty	Pen. short ST	Pen. long ST	Pen. short DT	Pen. long DT
SClass-1m	24389,8	17686,3	13792,7	469,6	3270,4	153,5
SDM	29659,4	22964,6	14336,4	519,3	3826,6	4282,2
SClass-2m	54048,8	47306,6	16332,5	19924,8	4032,1	7017,1
SDM	73435,3	66740,5	16959,2	28065,8	4275,3	17440,1
SClass-3m	67839,4	47306,6	16332,5	19924,8	4032,1	7017,1
SDM	92179,6	85484,8	20940,6	37842,8	5335,5	21365,8

Table 7: Monte Carlo simulation of SSM-SDT

6.2.1 Impact of Penalties

The comparative analysis is performed considering all the instances examined above.

With the increase of the penalty parameter in stochastic formulations, the optimization attempts to enhance reliability in order to avoid large penalty costs when the plan actually runs. The higher the penalty is, the more robust the shipment plan results against travel time perturbations.

We first consider SSM-ST. As foretold, by increasing the penalty parameter the amount of total service delay in the transportation network decreases. As said, a way is to avoid not direct services, which activations decrease as the penalty increases. The major delay decrease concerns the most expensive delay, that is delay over the chosen threshold B . The higher the penalty, the lower are such delays. By an increase of three times the penalties (that is by tripling our need of reliability), the fixed cost of the network increases of about the 0.03% with a decrease of the amount of total delay of about the 3% of short delays and 10% of long delays.

Similar results are obtained by considering SSM-DT. Here, the increase of the penalty parameter push the optimization to increase the number of activated services in order to deliver the highest number of commodities possible at least one period before due dates, to avoid additional costs

related to unexpected delays. The percentage of early freight arrivals is increased of the 8% when comparing solutions obtained considering the highest and the lowest level of penalty (consider that the number of commodities is 25 in total). The total amount of delay decreases of the 12%, which the vast majority relates to large delays (over the threshold B^k). This, at the expense of an initial additional set up cost of only the 0.05%.

In Table 8 the above mentioned increase of set up costs and decrease of amount of delay (expressed in percentage) of solutions obtained with penalty levels 2 and 3 are compared with solutions obtained with penalty level 1. Results still refer to the same problem and scenario classes considered in the previous section.

	Demand Target			Service Target		
	Fixed Cost (%)	Short delay (%)	Long delay (%)	Fixed Cost (%)	Short delay (%)	Long delay (%)
penalty 1	-	-	-	-	-	-
penalty 2	+0,01	-3,26	-9,99	+0,01	-1,06	-5,66
penalty 3	+0,05	-7,48	-18,36	+0,03	-3,38	-9,95

Table 8: Effects of the increase of penalties on SSM-DT and SSM-ST

6.3 PHM Performance Analyses

Experimentations are here conducted considering instances belonging to all problem classes (from 1 to 4), the loosest delivery time window (11 – 14), the higher variability level ($\sigma = 0.25$) and the loosest time distribution range (l).

The parameters we fixed before running the algorithm are parameter γ , used in expression (40); $thres_{low}$ and $thres_{high}$ used in expression (41) as well as parameter β ; lastly and only for version PHM-DF, $thres_{low}^k$ and $thres_{high}^k$. Same parameters' values are used in both versions of the algorithm (PHM-D and PHM-DF). Fixed (and local) cost adjustments were performed with $\gamma = 1.1$ and $\beta = 1.1$; thresholds were set to $thres_{high} = 0.8$, $thres_{low} = 0.2$ for global adjustments and $thres_{high}^k = 1 + u_r/2$ and $thres_{low}^k = (u_r/2) - 1$ for local adjustments. Parameters are chosen based on the ones have been used in [Crainic et al., 2011b], after not having observed any significant changes in the meta-heuristic behavior for small changes of those values during some preliminary experiments. We consider, instead, different values of parameter ρ^0 , ranging from 0.1 to 50, and different thresholds of reached consensus on services to stop phase one, ranging from 80% to 95%.

Table 9 displays the performance results of the meta-heuristic considering the PHM-D and PHM-DF approaches when applied to the selected set of instances. The reported values refer, respectively, to exact solutions (*CPLEX*) and the best solutions found using the PHM-D and the PHM-DF with $\rho^0 = 5$. First phase is either stopped when consensus is obtained on the 90% of design arcs or when a classical meta-heuristic criteria is satisfied, namely, reaching a total of 20 (first phase) iterations or after 4 hours (first phase) running time. Total computation time (expressed in seconds), number of iterations and relative gaps (from the solution obtained with *CPLEX*) are also reported.

CPLEX solves in a very short time almost all instances, requiring few seconds for the easiest ones and about 2 minutes for the most complicated. Both approaches find in general very good quality solutions. The 45% of the instances are solved achieving the optimum. On the remaining instances, the relative gaps between the best found solutions and the optimum is always less than

1%.

How does the flow-information intervene? The PHM-DF approach seems to perform slightly better than the PHM-D approach with respect to the closeness to optimum. It succeeds, in fact, in lowering some relative gaps with respect to the PHM-D approach (relative gap of PHM-DF is on average 0,022% and the relative gap of PHM-D is on average 0,054%). The latter improvement, however, is reached at the expense of a higher number of iterations (on average 4,63 instead of 3,58) and, most important, a much longer running time, which sometimes even doubled PHM-D performances (the PHM-DF needs on average 58% more time than the PHM-D to solve instances). This additional effort is certainly due to the higher number of operations characterizing the PHM-DF with respect to PHM-D, and make the improvement in accuracy negligible compared to the additional time required to find solutions. In all cases the PHM-D algorithm stops phase one because the chosen consensus threshold is reached, while in the 25% of the cases the PHM-DF stops because the time limit was reached.

K	CPLEX		PHM-D				PHM-DF			
	Time (sec.)	Obj	Time (sec.)	Obj	Iterations	Gap (%)	Time (sec.)	Obj	Iterations	Gap (%)
15	13,11	3964,38	61,25	3964,38	2	0	82,66	3964,38	2	0
15	13,86	4538,18	114,24	4538,18	2	0	151,33	4538,18	2	0
15	12,95	3701,94	70,42	3712,86	2	0,29	104,88	3701,94	2	0
15	13,87	3834,58	101,16	3841,76	3	0,19	185,51	3841,76	3	0,19
15	14,21	4562,23	128,22	4565,61	2	0,07	395,12	4565,61	5	0,07
15	13,74	4679,08	120,91	4679,08	2	0	324,04	4679,08	4	0
20	32,53	6007,60	189,53	6007,60	2	0	230,42	6007,60	2	0
20	18,23	5061,42	172,41	5061,42	3	0	312,03	5061,42	4	0
20	23,14	6178,43	249,02	6181,83	3	0,06	544,75	6180,20	4	0,03
20	30,71	6532,49	350,09	6565,06	2	0,50	715,83	6532,49	4	0
20	16,15	4830,19	87,80	4830,19	3	0	151,25	4830,19	3	0
20	17,83	5216,48	116,97	5216,48	3	0	338,74	5216,48	5	0
25	24,90	7151,51	207,49	7153,12	2	0,02	313,53	7153,12	2	0,02
25	21,57	6245,17	104,80	6245,17	2	0	222,61	6250,66	3	0,09
25	27,10	6729,70	244,82	6729,70	4	0	828,99	6729,70	7	0
25	23,45	6968,70	176,93	6968,70	2	0	257,61	6968,70	2	0
25	29,54	7295,06	374,55	7300,43	4	0,07	887,82	7302,14	5	0,10
25	22,34	6638,79	89,24	6638,79	2	0	147,18	6639,35	3	0,01
50	48,26	14494,40	1682,03	14494,40	6	0	9313,16	14494,40	8	0
50	51,19	15562,70	1806,54	15575,10	6	0,08	9574,86	15564,30	8	0,01
50	90,65	15068,40	2514,37	15068,40	8	0	11129,99	15068,40	7	0
50	49,37	15317,40	1871,77	15317,40	7	0	8927,28	15317,40	8	0
50	123,89	14776,00	2024,23	14779,40	7	0,02	8850,26	14779,40	8	0,02
50	50,38	15035,60	2177,47	15035,60	7	0	9184,05	15035,60	10	0
Average	32,62		626,51		3,58	0,05	2632,25		4,63	0,02

Table 9: Performances of CPLEX, PHM-D and PHM-DF with $\rho^0 = 5$

6.3.1 Impact of parameter ρ^0

The performance of the method is generally sensitive to the choice of the penalty parameter ρ^0 , which scales the penalty term [Bertsekas, 2014]. Theory suggests that high values of the penalty parameter should induce faster, but often prematurely, convergence leading to ill-conditioned solutions. Conversely, small values of ρ^0 yield to weaker enforcement of the non-anticipativity constraints resulting in a more gradual convergence to, typically, better solutions after, however, many iterations [Wallace and Helgason, 1991; Mulvey and Vladimirou, 1991]. This is indeed supported

by empirical evidence also in our case. We solve the chosen instance set considering different values of ρ^0 , ranging from 0.1 to 50. First phase is stopped when consensus is obtained on the 80% of service design arcs.

In Table 10, average performance results for both the PHM-D and PHM-DF are shown. They relates to average number of iterations, percentage of instances solved reaching the optimum as well as average relative gaps (with respect to CPLEX solutions). In line with the results known from the literature, when ρ^0 is small, convergence of the first phase is slow, but at the same time the mechanism has enough time to “absorb” all the information from the scenarios. For a big value of ρ^0 , as opposed, convergence seems to be too fast and the methodology stops just reaching a local optimum in some cases. As shown in table, in fact, both approaches are able to solve some of the instances achieving the optimum, when ρ^0 is small. Increasing its value, however, induces a reduction of this number of instances from the 70% – 79% to the 58% – 62%. The accuracy of solutions decreases as well, as shown by the relative gaps (which are in all cases always very low and less than 1%). As already observed earlier, the PHM-DF always outperforms the PHM-D approach in relative gaps and number of instances solved reaching the optimum. Note that, in fact, the PHM-DF approach is able to solve at least the 75% of instances at optimum maintaining this standard till $\rho^0 = 5$, while the PHM-D not only is not able to reach this threshold but this percentage decreases as soon as ρ^0 increases. Regarding iterations, it seems that if ρ^0 is very low, the iterations needed by the PHM-DF are less than the PHM-D, as the additional flow information not only influences the accuracy of the solutions but allows the approach to converge sooner. Such a characteristic is lost already for $\rho^0 = 1$.

ρ^0	PHM-D			PHM-DF		
	Iterations	(%) of Optimum	Gap (%)	Iterations	(%) of Optimum	Gap (%)
0,1	3,17	70,83	0,0498	2,54	79,17	0,02215
0,5	2,5	66,67	0,0504	2,33	75	0,02239
1	2,17	66,67	0,0522	2,29	75	0,02471
5	1,88	62,5	0,0542	2,25	75	0,02617
10	1,83	62,5	0,0558	2,17	70,83	0,02863
20	1,71	62,5	0,0598	2,13	70,83	0,02938
50	1,63	58,33	0,0675	1,63	62,5	0,04406

Table 10: Average performances of PHM-D and PHM-DF with different values of ρ^0

6.3.2 Impact of consensus threshold parameter

We solve the chosen instance set considering different values of consensus threshold to stop phase one, ranging from 80% to 95%. The value of ρ^0 is, instead, fixed to 5. Intuitively, the value of the threshold directly influences the number of iterations the algorithm needs to perform before stopping the first phase: the higher this value is, the more iterations are needed in phase one to reach the requested consensus threshold and pass to phase two. At the same time, however, it allows to start phase two with a higher number of fixed arcs, decreasing its complexity and increasing the accuracy of final solutions.

Results obtained and presented in Table 11 are congruent with the above described behaviors. Results concern the average number of iterations performed by the PHM-D and PHM-DF

approaches to find solutions, the percentage of instances solved reaching the optimum and the average relative gaps with respect to the optimum (suggested by CPLEX).

Both approaches have the same increasing-iterations and increasing-accuracy behaviors by changing the consensus threshold. PHM-DF keeps outperforming PHM-D for accuracy. Unfortunately - and as already observed before - this is at the expenses of a higher number of iterations (on average the 27% more) with respect to the PHM-D. Thresholds higher than 85% seem to only improve the accuracy level of the solutions of both approaches. The percentage of instances solved reaching the optimum does not change after it, while the relative gaps decrease, even though just at a (third for PHM-D and second for PHM-DF) decimal level.

Threshold	PHM-D			PHM-DF		
	Iterations	(%) of Optimum	Gap (%)	Iterations	(%) of Optimum	Gap (%)
80%	1,88	62,5	0,0542	2,25	66,67	0,0263
85%	2,63	66,67	0,0338	3,21	75	0,0255
90%	3,58	66,67	0,0337	4,63	75	0,0173
95%	8,73	66,67	0,0331	11,21	75	0,0136

Table 11: Average performances of PHM-D and PHM-DF with different values of consensus threshold

It should be also noted, however, that the highest threshold (95%) seems to be too strict and first phases of both approaches stopped in most of the hardest cases not because the consensus threshold is actually attained, but just because the first phase time limit (4 hours) is reached. In Figure 4 the percentage of times first phase of PHM-D and PHM-DF stopped by reaching a chosen consensus threshold is reported.

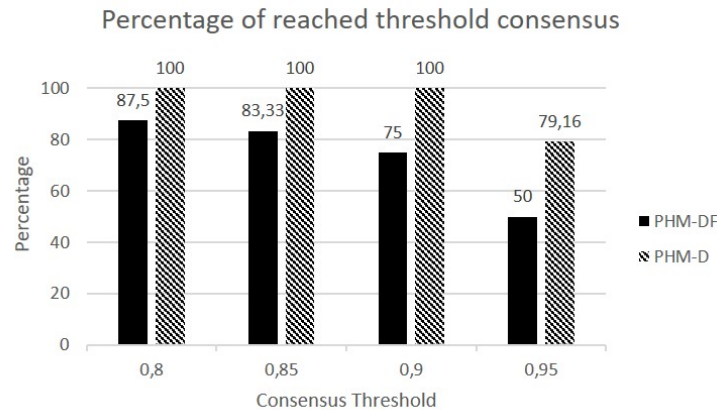


Figure 4: Percentage of instances Consensus

We also solved some of those instances not considering the time limit as a stop rule. Solutions and relatives gaps do not change significantly (only after the third decimal place). What consistently changes is the number of iterations needed to reach that threshold and stop phase one (on average 20). Consensus thresholds and stop time limits should be appropriately chosen.

6.3.3 Final remarks on PHM-D and PHM-DF

In general, although both approaches find very good quality solutions (as the relative gaps obtained from the experiments suggest), they show slight different behaviors and characteristics in their performances. PHM-DF seems to extremely benefit of small ρ^0 parameters (note that in case of $\rho^0 = 0.1$ and $\rho^0 = 0.5$, PHM-DF even outperforms PHM-D in number of iterations) and high consensus thresholds (which allow to reach a 0.02% gap). The PHM-D approach seems to have a more stable behavior, for what concerns relative gaps, but a well-defined decreasing number of iterations behavior when the parameters are changed. In Figure 5(a) and Figure 5(b) relative gaps are shown for increasing values of ρ^0 and consensus threshold.

The PHM-DF approach seems to enable the algorithm to reach better performances with respect to the PHM-D approach if only solution accuracy and precision are of interest. It should also be considered that even though flow-information is obtained just by solving sub-problems not requiring any extra operation, managing this additional information requires higher computational efforts. As mentioned before, the benefits of PHM-DF are negligible when compared to the considerable increase of time required to solve instances. It, thus, seems that the information of service trends (the PHM-D approach) is sufficient to find good solutions, making of the PHM-D approach a more suitable methodology to solve problem instances efficiently and quickly.

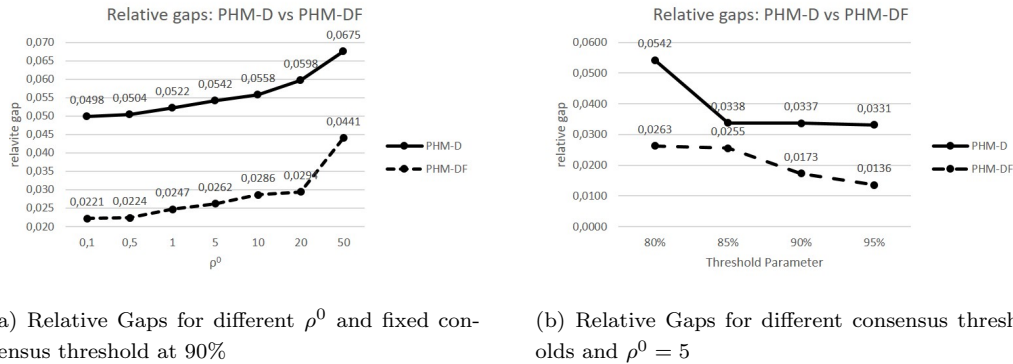


Figure 5: Relative Gaps for different ρ^0 and thresholds

All the above reported results are based on relatively small size instances and the application of the PHM approaches do not seem comparable with *CPLEX*, especially in terms of computation time. Nevertheless, through a simulation process a set of type-instances have been identified, for which *CPLEX* can not find a solution whereas our proposed algorithm is, instead, still able. The dimension of this set of instances is not far from the dimensions we used in the above described experimentations, highlighting the limitation of *CPLEX* to solve this kind of problems. In particular, the number of scenarios is increased up to 35, the number of services is increased up to 200 (with 20 not direct services) and the commodities are increased up to 85. The PHM-D approach is here only considered, with $\rho^0 = 5$ and a consensus threshold of 90%. Running times are high, on average about 4 hours. It should be noted, however, that at each iteration the PHM-D solves 35 big-sized SND problems. Nevertheless, it finds solutions, whereas *CPLEX* crashes after a short running time (on average after 10 minutes). In Table 12, performance results of 4 instances of that kind are shown. Objective, running time of phase one and two in minutes and number of iterations

are reported.

Instance			<i>CPLEX</i>		PHM-D			
—R—	—K—	—S—	Obj	Time	Obj	Phase 1 (min.)	Phase 2 (min.)	Iterations
200	85	35	-	-	23832,8	197,32	1,93	6
200	85	35	-	-	22018,4	339,5	2,10	9
200	85	35	-	-	23643,8	285,99	2,02	8
200	85	35	-	-	23867,8	252,86	2,09	6

Table 12: Results of *CPLEX* and PHM-D algorithm with $\rho^0 = 5$ for big instances

7 Conclusion

SND formulations in which time is explicitly taken as a stochastic parameter have been neglected in favor of settings in which other stochastic parameters were taken into account (in particular demand). Nevertheless, considering such a random parameter explicitly in the decision process may be beneficial, in particular to reduce the impact that the uncertainty in travel time may have on service performances. The novelty of the topic considered here is supported by a review of the very few published contributions on SND problems with uncertain time, pointing out the lack of contributions dealing with this specific problem and highlighting, in addition, the need of further research in other directions.

In this paper, a formulation which scope is to build an economically-efficient freight transportation network respecting given quality targets consistently as close as possible hedging the uncertainty in travel time is proposed. The uncertainty considered here belongs to the so-called randomness class. A two-stage stochastic linear mixed-integer programming original formulation is proposed, where design and routing make up the first stage and the given targets are accounted in the second stage through a set of penalties. Furthermore, given the NP-hardness of SND problems and the inability of traditional solvers to find solutions for large instances in reasonable time, a progressive hedging-based meta-heuristic able to provide good quality solutions is, also, proposed. Results are presented showing the attractiveness of the presented formulation and the effectiveness of the proposed resolution methodology.

This work opens a number of interesting research avenues related to the problem setting itself and to the proposed meta-heuristic approach. Let us first start with the problem setting.

One aspect that could be considered involves the relaxation of the independence assumption among time distributions. On real application, in fact, the assumption of independence is not always true and a certain degree of correlation is always observed (even though negligible in some cases). In addition, extensions could be done considering both travel and operation times as stochastic parameters. Future research will certainly consider the possibility of a different recourse, upon observing a long delay. A common way is to activate *ad hoc* services to deliver the tardy shipments till destination. The second stage, then, will consider not only the already defined costs, but also the additional cost required to adjust the start-up shipment plan as well as the determination of new freight routes. Lastly, in this work we assumed that all resources to perform services and terminal operations (like power units, carrying units, loading/unloading units and crews) are given and we did not assume any particular restriction on them. Extensions could

concern the introduction of specific constraints about resources or, given the importance that idle time plays, capacity restrictions at terminals.

Considering the meta-heuristic here proposed, several issues need further investigation and a number of research avenues appear promising to improve its performances. In general, the exploration of alternative mechanisms to modify costs to guide consensus towards a unique overall solution is an avenue to further explore. For instance, some kinds of *hybrid* approach may be considered in which fixed cost and variable cost adjustments may alternate, when given cost-thresholds are satisfied. Perhaps, the most impacting improvement can be obtained by bundling scenarios and creating multi-scenario sub-problems instead of single-scenario sub-problems, as done in here. One research issue may then concern the impact of several criteria to divide scenarios. Should scenario-groups partition or cover the original set of scenarios? Should the scenarios in each group be (dis)similar? Intuitively, aggregation of scenarios should yield more rapid agreement in solutions, involving however a higher complexity of sub-problems.

All the above mentioned extensions define new characteristics for the formulation and the algorithm we considered in this work and interesting new problems to study and explore.

Acknowledgements

Computations were made on the supercomputer Briarée from University of Montreal, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), the ministère de l'Économie, de la science et de l'innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

References

- Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.
- Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
- Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1-3):49–71, 2003.
- Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.
- John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
- P.-L. Carpentier, M Gendreau, and F Bastin. Long-term management of a hydroelectric multi-reservoir system under uncertainty using the progressive hedging algorithm. *Water Resources Research*, 49(5):2812–2827, 2013.
- Douglas G Chapman. Estimating the parameters of a truncated gamma distribution. *The Annals of Mathematical Statistics*, pages 498–506, 1956.

- Christopher S Coffey and Keith E Muller. Properties of doubly-truncated gamma variables. *Communications in Statistics-Theory and Methods*, 29(4):851–857, 2000.
- Teodor Crainic, Fausto Errico, Walter Rei, and Nicoletta Ricciardi. Modeling demand uncertainty in two-tiered city logistics planning. *Technical report CIRRELT-2012-65, Université de Montréal, Canada*, 2011a.
- Teodor Gabriel Crainic. *A survey of optimization models for long-haul freight transportation*. Citeseer, 1998.
- Teodor Gabriel Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.
- Teodor Gabriel Crainic. Long-haul freight transportation. In *Handbook of transportation science*, pages 451–516. Springer, 2003.
- Teodor Gabriel Crainic and Kap Hwan Kim. Intermodal transportation. *Transportation*, 14: 467–537, 2006.
- Teodor Gabriel Crainic and Gilbert Laporte. Planning models for freight transportation. *European journal of operational research*, 97(3):409–438, 1997.
- Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. Models for evaluating and planning city logistics systems. *Transportation science*, 43(4):432–454, 2009.
- Teodor Gabriel Crainic, Xiaorui Fu, Michel Gendreau, Walter Rei, and Stein W Wallace. Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124, 2011b.
- Teodor Gabriel Crainic, Mike Hewitt, and Walter Rei. Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research*, 43:90–99, 2014a.
- Teodor Gabriel Crainic, Mike Hewitt, Michel Toulouse, and Duc Minh Vu. Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393, 2014b.
- Teodor Gabriel Crainic, Fausto Errico, Walter Rei, and Nicoletta Ricciardi. Modeling demand uncertainty in two-tier city logistics tactical planning. *Transportation Science*, 50(2):559–578, 2015.
- Emrah Demir, Wolfgang Burgholzer, Martin Hrušovský, Emel Arıkan, Werner Jammerneegg, and Tom Van Woensel. A green intermodal service network design problem with travel time uncertainty. *Transportation Research Part B: Methodological*, 93:789–807, 2016.
- Jan Fabian Ehmke, Ann Melissa Campbell, and Timothy L Urban. Ensuring service levels in routing problems with time windows and stochastic travel times. *European Journal of Operational Research*, 240(2):539–550, 2015.
- Yueyue Fan and Changzheng Liu. Solving stochastic transportation network protection problems using the progressive hedging-based method. *Networks and Spatial Economics*, 10(2):193–208, 2010.

- Dinakar Gade, Gabriel Hackebeil, Sarah M Ryan, Jean-Paul Watson, Roger J-B Wets, and David L Woodruff. Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming*, 157(1):47–67, 2016.
- Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.
- Gianpaolo Ghiani, Gilbert Laporte, and Roberto Musmanno. *Introduction to logistics systems planning and control*. John Wiley & Sons, 2004.
- Raphael EC Gonçalves, Erlon Cristian Finardi, and Edson Luiz da Silva. Applying different decomposition schemes using the progressive hedging algorithm to the operation planning problem of a hydrothermal system. *Electric Power Systems Research*, 83(1):19–27, 2012.
- Ge Guo, Gabriel Hackebeil, Sarah M Ryan, Jean-Paul Watson, and David L Woodruff. Integration of progressive hedging and dual decomposition in stochastic integer programs. *Operations Research Letters*, 43(3):311–316, 2015.
- Kjetil K Haugen, Arne Løkketangen, and David L Woodruff. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research*, 132(1):116–122, 2001.
- Thorkell Helgason and Stein W Wallace. Approximate scenario solutions in the progressive hedging algorithm. *Annals of Operations Research*, 31(1):425–444, 1991.
- Lars Magnus Hvattum and Arne Løkketangen. Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics*, 15(6):527–557, 2009.
- Peter Kali and Stein W Wallace. *Stochastic programming*. Springer, 1994.
- Michal Kaut, Hercules Vladimirov, Stein W Wallace, and Stavros A Zenios. Stability analysis of portfolio management with conditional value-at-risk. *Quantitative Finance*, 7(4):397–409, 2007.
- Walid Klibi and Alain Martel. Scenario-based supply chain network risk modeling. *European Journal of Operational Research*, 223(3):644–658, 2012.
- Walid Klibi, Alain Martel, and Adel Guitouni. The design of robust value-creating supply chain networks: a critical review. *European Journal of Operational Research*, 203(2):283–293, 2010.
- Arne Løkketangen and David L Woodruff. Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming. *Journal of Heuristics*, 2(2):111–128, 1996.
- Elise D Miller-Hooks and Hani S Mahmassani. Least possible time paths in stochastic, time-varying networks. *Computers & operations research*, 25(12):1107–1125, 1998.
- John M Mulvey and Hercules Vladimirov. Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, 31(1):399–424, 1991.
- Mahmoud K Okasha and I.M. Alqanoo. Inference on the doubly truncated gamma distribution for lifetime data. *Int. J. Math. Stat. Invent*, 2:1–17, 2014.

- Olafur P Palsson and Hans F Ravn. Stochastic heat storage problemsolved by the progressive hedging algorithm. *Energy conversion and management*, 35(12):1157–1171, 1994.
- Warren B Powell and Huseyin Topaloglu. Stochastic programming in transportation and logistics. *Handbooks in operations research and management science*, 10:555–635, 2003.
- András Prékopa. *Stochastic programming*, volume 324. Springer Science & Business Media, 2013.
- R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.
- RT Rockafellar. Lagrange multipliers and subderivatives of optimal value functions in nonlinear programming. *Nondifferential and variational techniques in optimization*, pages 28–66, 1982.
- Andrzej P Ruszczyński and Alexander Shapiro. *Stochastic programming*, volume 10. Elsevier Amsterdam, 2003.
- Sarah M Ryan, Roger J-B Wets, David L Woodruff, César Silva-Monroy, and Jean-Paul Watson. Toward scalable, parallel progressive hedging for stochastic unit commitment. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.
- Dong-Ping Song, Dong Li, and Paul Drake. Multi-objective optimization for planning liner shipping service with uncertain port times. *Transportation Research Part E: Logistics and Transportation Review*, 84:1–22, 2015.
- Yer Van Hui, Jia Gao, Lawrence Leung, and Stein Wallace. Airfreight forwarders shipment planning under uncertainty: A two-stage stochastic programming approach. *Transportation Research Part E: Logistics and Transportation Review*, 66:83–102, 2014.
- Fernando Badilla Veliz, Jean-Paul Watson, Andres Weintraub, Roger J-B Wets, and David L Woodruff. Stochastic optimization models in forest planning: A progressive hedging solution approach. *Annals of Operations Research*, 232(1):259–274, 2015.
- Stein W Wallace and Thorkell Helgason. Structural properties of the progressive hedging algorithm. *Annals of Operations Research*, 31(1):445–455, 1991.
- Shuaian Wang and Qiang Meng. Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633, 2012a.
- Shuaian Wang and Qiang Meng. Robust schedule design for liner shipping services. *Transportation Research Part E: Logistics and Transportation Review*, 48(6):1093–1106, 2012b.
- Xin Wang, Teodor Gabriel Crainic, and Stein W Wallace. Stochastic scheduled service network design: the value of deterministic solutions. *Technical report CIRRELT-2016-14, Université de Montréal, Canada*, 2016.
- Jean-Paul Watson and David L Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, 2011.
- Nicole Wieberneit. Service network design for freight transportation: a review. *OR spectrum*, 30(1):77–112, 2008.