A Lagrangian heuristic algorithm for the time-dependent combined network design and routing problem

Bernard Fortz Enrico Gorgone
Département d'Informatique,
Université Libre de Bruxelles,
Brussels, Belgium,
{bernard.fortz, egorgone}@ulb.ac.be

Dimitri Papadimitriou
Nokia Bell Labs,
Antwerp, Belgium,
dimitri.papadimitriou@nokia-bell-labs.com

November 9, 2016

Abstract

During the planning of communication networks, the routing decision process (distributed and online) often remains decoupled from the network design process, i.e., resource installation and allocation planning process (centralized and offline). To reconcile both processes and take into account demand variability, we generalize the capacitated multi-commodity fixed charge network design class of problems by including different types of fixed costs (installation and maintenance costs) and variable costs (routing costs) but also variable traffic demands over multiple periods. However, conventional integer programming methods can typically solve only small to medium size instances of this problem.

Two major difficulties are encountered when using commercial solvers to solve the associated mixed integer programs: (i) problems are large scale and even solving the linear relaxation of the problem can be challenging; and (ii) the solver hardly find good feasible solutions for medium to large scale instances. As an alternative, we propose a Lagrangian approach for computing a lower bound by relaxing the flow conservation constraints such that the Lagrangian subproblem itself decomposes by node. Though this approach yields one subproblem per network node, solving the Lagrangian dual by means of the bundle method remains a complex computational tasks. However, it always provides a lower bound on the optimal solution. Moreover, based on this relaxation, we propose a Lagrangian heuristic that makes the approach more robust than a black-box usage of a MIP solver.

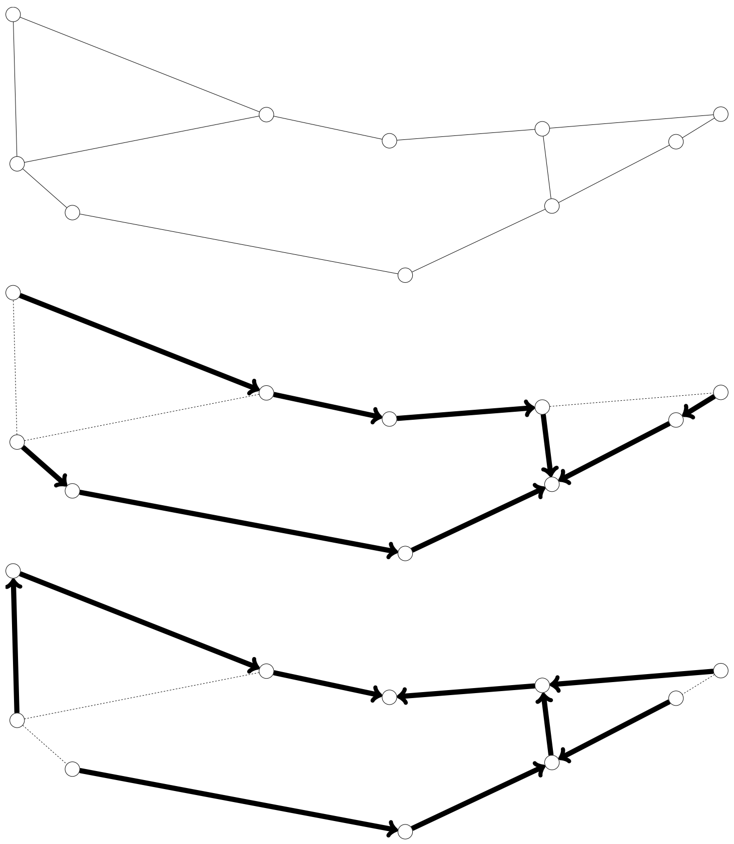


Figure 1: An example of a network for a single period and two destination-based routing trees

Keywords: Telecommunications networks, Network design, Routing, Lagrangian decomposition, Multi-commodity fixed-charge network design, Mixed Integer Programming.

1 Introduction

In today's communication networks, distributed control functions such as routing, are driven by path quality properties (such as cost and bandwidth delay product) but also their adaptation cost and convergence time. However, the design of the routing function (and associated routing protocol procedures) remains driven by their consumption of processing capacity and memory available locally at each node. Henceforth, the routing decision process (distributed and online) remains also decoupled from the network design process, i.e., resource installation and allocation planning process (centralized and offline).

The conventional model to formulate such problem assumes that routing decisions can be performed without informing the capacity optimization problem (link resource installation and (modular) allocation). These decisions are modeled in terms of capacity allocation per link but without accounting for the routing state creation and maintenance cost. This formulation is thus often extended by assuming that the routing optimization process can additionally inform the capacity installation and allocation process about its utility. The latter then adjusts the allocated capacity on each link and may decide to add new links (between node pairs not previously connected). This method has been applied for instance to various combined network design and traffic engineering problems including IP over Multi-Protocol Label Switching - Traffic Engineering (MPLS-TE) and IP over optical/wavelength switching layer. However, such formulation does not account for i) the cost associated to the creation of a routing adjacency once the corresponding link is added, ii) the cost of link maintenance during the lifetime of the corresponding routing adjacency, and iii) the routing cost function which remains independent of the link occupancy.

For these reasons, we study in this paper an extension of the multi-commodity capacitated fixed charge network design (MCND) problem introduced by [9, 10, 17] which deals with the simultaneous optimization of capacity installation cost and traffic flow routing cost. In the MCND problem, a fixed cost is incurred for opening a link and a linear routing cost is paid for sending traffic flow over an edge (or arc). The routing decision must be performed such that traffic flows remain bounded by the installed capacities. In [25], we generalized this problem over multiple time periods using an increasing convex routing cost function which takes into account congestion (number of routing paths per edge) and delay (routing path length). A compact Mixed Integer Linear Program (MILP) formulation for this problem is developed based on the aggregation of traffic flows following the per destination routing decision process underlying packet networks (as illustrated in Figure 1). However, the resolution with realistic topologies and traffic demands becomes rapidly intractable with state-of-the-art solvers due to the weak linear programming bound of the proposed MILP formulation. An extended formulation where traffic flows are disaggregated by source-destination pairs, while keeping the requirement of destination-based routing decisions has been studied in [26].

In general, direct formulations for determining optimal routing decisions obeying various protocol rules are complex to solve. Indeed, integer programming methods can typically solve only small to medium size instances as reported in [1]. To circumvent this problem, [10] among others have successfully applied the Lagrangian relaxation technique to compute efficient large-scale instances of the MCND problem. Indeed, by relaxing the linking constraints, the Lagrangian relaxation method can be applied to the base (aggregated) and extended (disaggregated) formulation in order to provide stronger lower bounds. Moreover, the suitable choice of the complicating constraints yields a Lagrangian subproblem decomposable by node, in line with the objective of obtaining a decomposition of the original optimization problem which preserves the distributed nature of the routing decision process.

Multi-commodity fixed charge network design problems are extremely challenging to solve. This complexity arises because even the simple continuous versions usually contain a huge number of variables and constraints, which makes them very hard to solve with standard approaches. Indeed, specialized algorithms are required [7, 16] and the use of parallel architectures could be necessary [6]. The complexity becomes even higher if integer variables are present in the models to represent logical decisions. The resulting mathematical model is a MILP with multi-commodity network flow structure.

The model considered in this paper extends the MCND problem by including different types of fixed costs (installation and maintenance costs) and variable costs (routing costs). In addition, time dependent demands are taken into account and the network is designed for more than one time period. We propose a Lagrangian relaxation approach for computing a lower bound, and a Lagrangian heuristic for obtaining good feasible solutions. For this purpose, we relax the flow conservation equations such that the Lagrangian subproblem can be decomposed by node. We remark that compared to what happens in the standard Fixed Charge Network Design problem, the Lagrangian subproblem is not a knapsack problem. Unfortunately, this yields a Lagrangian subproblem that is not so easy to solve. However, the approach is more robust than using CPLEX as a black-box MIP solver, as the Lagrangian relaxation always returns a lower bound and provides a feasible solution for many instances where CPLEX fails.

The remainder of this paper is structured as follows. Section 2 documents prior and related work together with motivating arguments. We describe our optimization model in Section 3. We present in the next section the numerical resolution method used for this application. In Section 5, we report the computational results and analyze them. Finally, Section 6 concludes this paper together with directions for future research work.

2 Prior and Related Work

In this section, we review the prior work related to computational methods for the combined capacitated FCND and routing optimization problem further referred hereafter to as the generalized MCND problem. Indeed, for the reasons explained in the introduction, the base MCND problem appears as a particular case of the model developed in this paper. However, compared to its generalized version, the conventional MCND problem is only defined over a single time period and the cost function does not take into account congestion (number of routing paths per edge) and delay (routing path length) phenomena.

2.1 Spatial Decomposition

One successful method to overcome the computational limits when solving large instances of the MCND problem consists in solving the formulation with a decomposition method. The driving concept behind such methods is to decompose the master (combined) problem into subproblems such that each subproblem involves only local decisions to be performed by the node computing online the routing decisions. The Benders decomposition [3] method developed for linear optimization problems relies on the ideas of partition and delayed constraint generation. This method decomposes the initial problem into two simpler problems: the first problem, called *master problem*, solves a relaxed version of the original problem and obtains values for a subset of the original variables and the associated constraints. The second problem, called *auxiliary problem* (or subproblem), obtains the values for the remaining variables while keeping the variables obtained in the master problem fixed, and uses these to generate cuts for the master problem. The master and auxiliary problems are solved iteratively until no more cuts can be generated. The conjunction of the variables found in the last master and subproblem iteration is the solution to the original formulation. The structure of MCND problems presents a logical decomposition for the Benders method: the values of integer variables representing the installation of the links are given by the solution of the master problem while the continuous variables representing the actual flows are kept in the subproblem for the tentative network obtained in the master problem. In other terms, at each iteration the master solution gives a tentative network for which the subproblem finds the optimal solution. Due to the need to solve the master and the auxiliary problems several times, the Benders decomposition method is applicable if these problems can be solved efficiently, i.e., when it is much easier to solve the decomposed problems than the original one. Moreover, especially for the subproblem, it is sometimes possible to proceed with further decomposition (by flow, by period, etc.), resulting in even more efficient methods. Costa [8] provides a detailed survey on Benders decomposition methods applied to problems that can be formulated as network design problems.

Hence, the Benders decomposition [3] could be thought as a competitive method for the optimization problem at hand. The structure of MCND problems presents a priori a logical decomposition for the Benders method: the integer variables representing the installation of the links are solved in the master problem while the continuous variables representing the actual flows are kept in the subproblem for the tentative network obtained in the master problem. In other terms, at each iteration the master solution gives a tentative network for which the subproblem finds the optimal solution. The caveat being that decomposing the Benders master problem per individual node (to take into account the distributed nature of the routing process) is not directly possible for the generalized MCND problem considered in this paper. In [25], we anticipated to decompose the original problem by keeping the link installation, link maintenance, and routing decision change variables in the master problem while projecting out the routing decisions. However, this approach does not allow to perform a canonical Benders decomposition as the capacity constraints link the different subproblems altogether. In other terms, the nodal decomposition dimension of the routing decision process prevents the effective use of canonical Benders decomposition.

2.2 Temporal Decomposition

Several computational methods rely on primal decomposition. Temporal decomposition methods such as the rolling horizon technique [21] have been successfully applied to solve combined planning and distribution problems over multiple periods (both short- and long-term) in transportation and logistics [30, 5], energy supply chain, production planning, etc. All these problems involve execution scenarios where the future is not known in advance. This method is thus worthwhile considering for the problem at hand since we aim at solving our optimization problem over a large number of time periods. It consists of dividing the generalized MCND problem into multiple subproblems to be solved over a limited number of periods and reuse the solution produced at one (set of) periods to the next. The goal is to decrease the computational time while limiting the degradation of the solution compared to the one produced by the exact method performing over the entire set of periods. The basic idea is indeed to iteratively consider only a small part of the time steps, called the planning horizon of p_H periods, and find an optimal solution for this reduced planning problem. With this solution at hand, a specified number of time steps are fixed, and then the planning horizon is shifted equally by p_H periods. Moreover, this method shows properties that are well suited for combined capacity installation, maintenance and routing decisions that must be designed by accounting for the uncertainty introduced by varying traffic demands (thus varying link capacity availability) and link ageing effects that impose periodic updates.

In [27] we propose to resolve this computationally challenging problem by means of the rolling horizon heuristic with the objective to decrease the computational time while degrading as less as possible the quality of the solution. The resulting improvements enable to progressively overcome the computational limits encountered when solving such problem, in particular, when the network size and number of periods increase. The improvements provided by the rolling horizon heuristic can be further exploited to account for different patterns of failures that may affect installed arcs over time.

2.3 Lagrangian Decomposition and Related Objectives

As stated in the introduction, one of the most promising techniques to solve the base MCND problem is the Lagrangian relaxation [22, 10, 6]. By relaxing the linking constraints, the Lagrangian relaxation method can be applied to the MNCD problem in order to provide stronger lower bounds. Moreover, with a suitable choice of the complicating constraints (by dualizing the flow constraints), this method offers the possibility to decompose the Lagrangian subproblem per network node inline with the objective of obtaining a decomposition of the original optimization problem which preserves the distributed nature of the routing decisions. For instance, [10] have successfully applied the Lagrangian relaxation technique to efficiently solve large-scale instances of the MCND problem.

In solving the MCND problem, Crainic et al. [10] showed that the bundle methods have two main advantages compared to subgradient approaches: i) their increased complexity is most often compensated by faster convergence (than subgradient methods) towards optimal value of the Lagrangian dual, ii) they require usually fewer parameters to adjust and are less sensitive to these parameters than subgradient methods; hence, more robust. On the other hand, the bundle method requires, at each iteration, the minimization of a polyhedral function which corresponds to solve a linear problem, called master problem, in order to obtain the new iterate, following the proximal Bundle method [19, 2]. For numerical reasons, the master problem must be "stabilized" [15]. Moreover, exploiting the properties of the specific problem [18] leads to a "structured" master that usually also provides a better numerical behaviour. To circumvent some of these limits, new sub-gradient methods have also been proposed recently like deflected, projected [11] and primal-dual approaches [23]. In particular, the latter minimizes the gain in parameter adjustment provided by the bundle methods; however, the bundle methods still converge much faster as they use much more detailed piece-wise linear model of the objective function. These arguments justify (a priori) the choice of the bundle method following in this paper that is well suited to our formulation of the Lagrangian dual.

The fundamental objective of this is paper is to extend the Lagrangian decomposition method to the generalized MCND problem. Reaching this objective requires however to address several challenges because the Lagrangian subproblem is not structured as a knapsack problem; hence, it becomes much harder to solve. First, we propose a Lagrangian relaxation approach for computing a lower bound. Next as obtaining good feasible solutions for medium to large scale instances remains indeed challenging; we propose a Lagrangian heuristic with computationally provable performance. Finally, though we compute a lower bound by relaxing the flow conservation constraints such that the Lagrangian approach yields one subproblem per network node, solving the Lagrangian

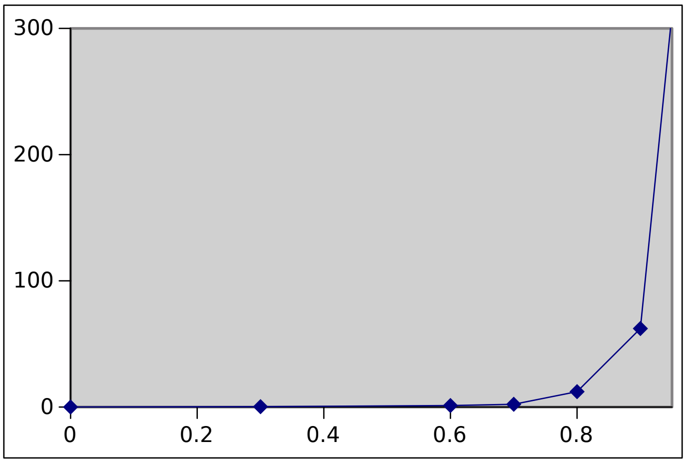


Figure 2: Piecewise linear cost function

dual by means of the bundle method remains a complex computational tasks. From this perspective our objective is to show that the proposed Lagrangian heuristic makes the approach more robust compared to out-of-shelve MIP solvers.

3 The Optimization Model

The problem we study in this paper can be described as follows. Given a directed network $G = (V, A)$, where V is the set of nodes and A is the set of links, we must satisfy the demands of each pair of nodes $s, t \in V$ over a set of periods $p \in \mathcal{P} = \{1, \ldots, P\}$. Each triplet origin-destination-period (s, t, p) has an associated demand $D^p(s, t)$ that must flow between s and t at period p. To simplify the notation, we also assume $D^p(s, s) = 0$. Each arc (i, j) in the network can only be used in period p if it is newly installed at period p, in which case an installation cost c_{ij} is paid, or if it was already used in period $p - 1$, in which case a maintenance cost m_{ij} (with $m_{ij} < c_{ij}$) is paid. Each arc (i, j) has also a nominal capacity κ_{ij}.

The routing of demands is performed as follows: for each potential destination, a next-hop selection is performed at a given node. This allows to maintain a reasonably sized routing table in each router, in the line of what shortest path routing protocols do [1, 4, 29], but with some additional flexibility as we don't impose that the path created here are compatible with a (given or computed) distance matrix.

Furthermore, a routing cost for each arc (i, j) and each period p is defined as an increasing piecewise linear convex function of its utilization, inspired by [13]. Given the load l^p_{ij} on arc (i, j) resulting from the routing decisions in a given period p, the routing cost associated to (i, j) at period p has the form

$$\phi(\kappa_{ij}, l^p_{ij}) = \max_{r=1\ldots,R} \{\theta_r l^p_{ij} - \tau_r \kappa_{ij}\},$$

where θ_r and τ_r ($r \in \{1, \ldots, R\}$) define the slopes and independent terms of the segments composing ϕ. The cost function used in our experiments is illustrated in Figure 2.

The problem consists in minimizing the sum of all costs, while satisfying demand requirements and capacity constraints. The cost of a solution to the optimization problem combines the sum of (i) installation costs c_{ij}, (ii) maintenance costs m_{ij}, and (iii) routing costs $\phi(\kappa_{ij}, l^p_{ij})$.

This problem is a relaxation of the problem presented in [25] as we do not consider here the additional constraint limiting the number of allowed routing changes over the time horizon.

A first formulation for the problem (similar to [25]) uses the following set of variables:

- y^p_{ij}: binary variable indicating if arc (i, j) should be newly opened or re-opened at period p;
- z^p_{ij}: binary variable indicating if arc (i, j) should be maintained at period p;
- x^{tp}_{ij}: binary variable indicating if node j is the next hop for node i to destination t at period p.
- f^{tp}_{ij}: continuous variable indicating the amount of flow on arc (i, j) destined to t at period p.
- ϕ^p_{ij}: continuous variable representing the piecewise linear routing cost on arc (i, j) at period t.

The problem can then be formulated as:

$$\textbf{(AF)}\ \min \sum_{p\in\mathcal{P}} \sum_{(i,j)\in A} \left(c_{ij} y^p_{ij} + m_{ij} z^p_{ij} + \phi^p_{ij}\right) \tag{1a}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} f^{tp}_{ij} - \sum_{j:(j,i)\in A} f^{tp}_{ji} = D^p(i,t) \qquad i,t \in V, i \neq t, p \in \mathcal{P} \tag{1b}$$

$$\sum_{j:(j,i)\in A} f^{tp}_{jt} = \sum_{s\in V} D^p(s,t) \qquad t \in V, p \in \mathcal{P} \tag{1c}$$

$$x^{tp}_{ij} \le y^p_{ij} + z^p_{ij} \qquad (i,j) \in A, t \in V, p \in \mathcal{P} \tag{1d}$$

$$y^p_{ij} + z^p_{ij} \le 1 \qquad (i,j) \in A, t \in V, p \in \mathcal{P} \tag{1e}$$

$$z^p_{ij} \le y^{p-1}_{ij} + z^{p-1}_{ij} \qquad (i,j) \in A, p \in \mathcal{P}, p \ge 2 \tag{1f}$$

$$z^1_{ij} = 0 \qquad (i,j) \in A \tag{1g}$$

$$f^{tp}_{ij} \le C^{tp}_{ij} x^{tp}_{ij} \qquad (i,j) \in A, t \in V, p \in \mathcal{P} \tag{1h}$$

$$\sum_{t\in V} f^{tp}_{ij} \le C^p_{ij}(y^p_{ij} + z^p_{ij}) \qquad (i,j) \in A, p \in \mathcal{P} \tag{1i}$$

$$\sum_{j:(i,j)\in A} x^{tp}_{ij} = 1 \qquad i,t \in V, i \neq t, p \in \mathcal{P} \tag{1j}$$

$$\phi^p_{ij} \ge \theta_r \sum_{s,t\in V} f^{stp}_{ij} - \tau_r \kappa_{ij} \qquad r \in R, (i,j) \in A \tag{1k}$$

$$x^{tp}_{ij} \in \{0,1\} \qquad (i,j) \in A, t \in V, p \in \mathcal{P} \tag{1l}$$

$$y^p_{ij}, z^p_{ij} \in \{0,1\} \qquad (i,j) \in A, p \in \mathcal{P} \tag{1m}$$

$$f^{tp}_{ij} \ge 0 \qquad (i,j) \in A, t \in V, p \in \mathcal{P} \tag{1n}$$

The first term of the objective function (1a) corresponds to the installation cost, the second one to the maintenance cost and the last one to the piecewise linear routing cost. Constraints (1b) and (1c) are flow conservation constraints that ensure that the demand flow requirement given by matrix D^p is routed at period p. Constraints (1d) ensure that an arc can be used for routing at period p only if it is installed or maintained open at period p, while (1e) impose that an arc is not both installed and maintained during the same period. Next, (1f) ensure that an arc can be maintained only if it was open during the previous period, and we assume no arc was open before the first period starts (1g). Constraints (1h) state that flow can be sent on an arc for a given destination at a given period only if the arc is in the routing table, where $C_{ij}^{tp} = \min\left(\kappa_{ij}, \sum_{s \in V} D^p(s,t)\right)$ is a tight upper bound on the flow on arc (i,j) destined to t at period p. Similarly, constraints (1i) ensure that flow is sent only on opened arcs, and that the capacity of the arcs are not exceeded, where $C_{ij}^{p} = \min\left(\kappa_{ij}, \sum_{s,t \in V} D^p(s,t)\right)$ is a tight upper bound on the total load on arc (i,j) at period p. Finally, equalities (1j) impose that exactly one next-hop is chosen for each node, towards each destination, at each period, and (1k) model the piecewise linear routing costs.

Note that in this formulation, flow variables and flow conservation constraints are aggregated per destination. We call this model the *Aggregated Formulation (AF)*. As shown in [25], a stronger lower bound can be obtained by disaggregating the flows by source, i.e. defining new flow variables f_{ij}^{stp} representing the amount of flow on arc (i,j) from source s to destination t at period p. To simplify the notation, we define a deficit vector d_i^{stp}, $i \in V$, indicating the net amount of flow required by the node: nodes with negative deficit are sources, nodes with positive deficits are sinks, and nodes with zero deficits are transshipment, i.e. $d_i^{stp} = D^p(s,t)$ if $i = s$, $d_i^{stp} = -D^p(s,t)$ if $i = t$, and $d_i^{stp} = 0$ otherwise. Moreover, we implicitly consider that $f_{ij}^{ssp} = 0$, and we can tighten the capacity bounds by defining $C_{ij}^{stp} = \min\left(\kappa_{ij}, D^p(s,t)\right)$.

The *Disaggregated formulation (DF)* can then be written as:

$$
\begin{aligned}
\textbf{(DF)}\ \min \sum_{p\in\mathcal{P}} \sum_{(i,j)\in A} & \left(c_{ij}y_{ij}^p + m_{ij}z_{ij}^p + \phi_{ij}^p\right) && && \text{(2a)}\\
\text{s.t.} \sum_{j:(i,j)\in A} f_{ij}^{stp} - \sum_{j:(j,i)\in A} f_{ji}^{stp} &= d_i^{stp} && i,s,t\in V, p\in\mathcal{P} && \text{(2b)}\\
x_{ij}^{tp} &\le y_{ij}^p + z_{ij}^p && (i,j)\in A, t\in V, p\in\mathcal{P} && \text{(2c)}\\
y_{ij}^p + z_{ij}^p &\le 1 && (i,j)\in A, t\in V, p\in\mathcal{P} && \text{(2d)}\\
z_{ij}^p &\le y_{ij}^{p-1} + z_{ij}^{p-1} && (i,j)\in A, p\in\mathcal{P}, p\ge 2 && \text{(2e)}\\
z_{ij}^1 &= 0 && (i,j)\in A && \text{(2f)}\\
f_{ij}^{stp} &\le C_{ij}^{stp} x_{ij}^{tp} && (i,j)\in A, s,t\in V, p\in\mathcal{P} && \text{(2g)}\\
\sum_{s,t\in V} f_{ij}^{stp} &\le C_{ij}^p (y_{ij}^p + z_{ij}^p) && (i,j)\in A, p\in\mathcal{P} && \text{(2h)}\\
\sum_{j:(i,j)\in A} x_{ij}^{tp} &= 1 && i,t\in V, i\ne t, p\in\mathcal{P} && \text{(2i)}\\
\phi_{ij}^p &\ge \theta_r \sum_{s,t\in V} f_{ij}^{stp} - \tau_r \kappa_{ij} && r\in R, (i,j)\in A && \text{(2j)}\\
x_{ij}^{tp} &\in \{0,1\} && (i,j)\in A, t\in V, p\in\mathcal{P} && \text{(2k)}\\
y_{ij}^p, z_{ij}^p &\in \{0,1\} && (i,j)\in A, p\in\mathcal{P} && \text{(2l)}\\
f_{ij}^{stp} &\ge 0 && (i,j)\in A, s,t\in V, p\in\mathcal{P} && \text{(2m)}
\end{aligned}
$$

Despite the better lower bounds provided by (DF), the drawback of this formulation is its very large dimension, as the number of variables and the number of constraints are multiplied by $O(|V|)$ when compared to (AF). As reported in [25], (DF) becomes quickly untractable for state-of-the-art MIP solvers. This motivates us for the development of a Lagrangian decomposition algorithm.

4 The Algorithm

The main novelty of our approach relies in the design of a Lagrangian heuristic suitable for both (DF) and (AF) within a non-smooth algorithm together with an iterative resolution procedure. The Lagrangian relaxation problem is solved by means of the bundle method (BM) [19] which aims at overcoming the slow convergence and instability of the Cutting Plane algorithm [20]. A preliminary version of the resolution procedure is documented in [28]. The bundle method is extended in this paper with a Lagrangian heuristic to produce a feasible solution. In our numerical experiments in Section 5, we use this extended method to solve the Lagrangian dual of both (DF) and (AF) described in Section 4.1.

4.1 Lagrangian Relaxation

The Lagrangian relaxation [22, 10, 6] of a general mixed integer problem consists in taking the set of complicating constraints of this problem into the objective function multiplied by a vector of weights (the Lagrange multipliers). The corresponding Lagrangian dual problem is solved iteratively by seeking for the optimal multipliers of the relaxed constraints. Lagrangian relaxation requires the reformulation of the model as a much larger problem, which sometimes provides a better bound. Because of its huge size, dynamic generation of the model is required. This technique leads to the solution of a non-smooth convex problem.

There are a plenty of strategies to decompose the problem by relaxing some groups of constraints. Following the idea coming from Network Design problems (see, e.g., [10]), (AF) and (DF) are relaxed respectively by dualizing the group of flow constraints (1b) and (2b) in such a way that the corresponding Lagrangian subproblem can be decomposed by node $i \in V$. In principle, we could decompose more than that but worsening the complexity of the Master problem.

We first describe the Lagrangian relaxation of (DF). Let ν_i^{stp} be the dual variable (multiplier) of the flow constraint (2b) associated to node i, source s, destination t and period p. It follows that the Lagrangian function associated to ν has the form:

$$\xi_{DF}(\nu) = \min \sum_{p\in\mathcal{P}} \sum_{(i,j)\in A} \left(c_{ij} y_{ij}^p + m_{ij} z_{ij}^p + \phi_{ij}^p\right) \tag{3a}$$

$$+ \sum_{p\in\mathcal{P}} \sum_{s,t\in V} \sum_{(i,j)\in A} \left(\nu_i^{stp} - \nu_j^{stp}\right) f_{ij}^{stp} \tag{3b}$$

$$+ \sum_{p\in\mathcal{P}} \sum_{s,t\in V} (\nu_t^{stp} - \nu_s^{stp}) D^p(s,t) \tag{3c}$$

$$\text{s.t.} \quad \text{(2c) - (2m)} \tag{3d}$$

where the objective function is linearized by introducing the variables ϕ_{ij}^p, $(i,j) \in A$, $p \in P$.

The Lagrangian dual of the original problem consists in maximizing the function $\xi_{DF}(\nu)$ over the whole space $\mathbb{R}^{|V|^3 \times P}$, that is:

$$\max \left\{ \xi_{DF}(\nu) \; : \; \nu \in \mathbb{R}^{|V|^3 \times P} \right\} . \tag{4}$$

Note that the term (3c) involving the demands $D^p(s,t)$ is linear in ν. We define

$$\xi_{DF}^0(\nu) = \sum_{p\in\mathcal{P}} \sum_{s,t\in V} (\nu_t^{stp} - \nu_s^{stp}) D^p(s,t),$$

and the remaining Lagrangian subproblem can be decomposed in $|V|$ subproblems $\xi_{DF}^i(\nu)$, $i \in V$, defined as:

$$\xi^i_{DF}(\nu) =$$

$$\min \sum_{p\in\mathcal{P}} \sum_{j:(i,j)\in A} \left(c_{ij} y^p_{ij} + m_{ij} z^p_{ij} + \phi^p_{ij} + \sum_{s,t\in V} \left(\nu^{stp}_i - \nu^{stp}_j\right) f^{stp}_{ij} \right) \tag{5a}$$

$$\text{s.t. } \phi^p_{ij} \geq \theta_r \sum_{s,t\in V} f^{stp}_{ij} + \tau_r \kappa_{ij} \quad j:(i,j)\in A, p\in\mathcal{P}, r\in R \tag{5b}$$

$$x^{tp}_{ij} \leq y^p_{ij} + z^p_{ij} \quad j:(i,j)\in A, t\in V, p\in\mathcal{P} \tag{5c}$$

$$y^p_{ij} + z^p_{ij} \leq 1 \quad j:(i,j)\in A, p\in\mathcal{P} \tag{5d}$$

$$z^p_{ij} \leq y^{p-1}_{ij} + z^{p-1}_{ij} \quad j:(i,j)\in A, p\in\mathcal{P}, p\geq 2 \tag{5e}$$

$$z^1_{ij} = 0 \quad j:(i,j)\in A \tag{5f}$$

$$f^{stp}_{ij} \leq C^{stp}_{ij} x^{tp}_{ij} \quad j:(i,j)\in A, s,t\in V, p\in\mathcal{P} \tag{5g}$$

$$\sum_{t,s\in V} f^{stp}_{ij} \leq C^p_{ij}(y^p_{ij} + z^p_{ij}) \quad j:(i,j)\in A, p\in\mathcal{P} \tag{5h}$$

$$\sum_{j:(i,j)\in A} x^{tp}_{ij} = 1 \quad t\in V, i\neq t, p\in\mathcal{P} \tag{5i}$$

$$x^{tp}_{ij} \in \{0,1\} \quad j:(i,j)\in A, t\in V, p\in\mathcal{P} \tag{5j}$$

$$y^p_{ij}, z^p_{ij} \in \{0,1\} \quad j:(i,j)\in A, p\in\mathcal{P} \tag{5k}$$

$$f^{stp}_{ij} \geq 0 \quad j:(i,j)\in A, s,t\in V, p\in\mathcal{P} \tag{5l}$$

Consequently, the Lagrangian function $\xi_{DF}(\nu)$ can be rewritten as the following sum of functions:

$$\xi_{DF}(\nu) = \xi^0_{DF}(\nu) + \sum_{i\in V} \xi^i_{DF}(\nu). \tag{6}$$

A similar approach leads to the Lagrangian relaxation of (AF). Let ν^{tp}_i be the dual variable of flow constraints (1b) with respect to node i, destination t and period p for $i \neq t$ and ν^{tp}_t the dual variable of (1c). The Lagrangian function has the following form:

$$\xi_{AF}(\nu) = \min \sum_{p\in\mathcal{P}} \sum_{(i,j)\in A} \left(c_{ij} y^p_{ij} + m_{ij} z^p_{ij} + \phi^p_{ij} \right) \tag{7a}$$

$$+ \sum_{p\in\mathcal{P}} \sum_{t\in V} \sum_{(i,j)\in A} \left(\nu^{tp}_i - \nu^{tp}_j\right) f^{tp}_{ij} \tag{7b}$$

$$+ \sum_{p\in\mathcal{P}} \sum_{s,t\in V} (\nu^{tp}_t - \nu^{tp}_s) D^p(s,t) \tag{7c}$$

$$\text{s.t. (2c) - (2m)} \tag{7d}$$

$$\phi^p_{ij} \geq \theta_r \sum_{t\in V} f^{tp}_{ij} - \tau_r \kappa_{ij} \quad r\in R, (i,j)\in A \tag{7e}$$

The Lagrangian dual of (AF) is then

$$\max \left\{ \xi_{AF}(\nu) \, : \, \nu \in \mathbb{R}^{|V|^2 \times P} \right\}. \tag{8}$$

and can then be decomposed along the same lines as the one of (DF).

Observe that the integrality property does not hold for these Lagrangian duals. From a theoretical standpoint, this means that the Lagrangian approach can provide a better lower bound than the continuous relaxation could do. In practice however, our empirical experiments show that the gain is usually not significant, and solving exactly (3) and (7) as MILPs is computationally challenging. Hence, in our numerical experiments, we assume that (3) and (7) are linear programs by relaxing the integrality constraints.

One of the major challenges in solving the Lagrangian duals (4) and (8) is their non-smooth nature. For such problems, Bundle methods [15, 19] have been shown to be very effective.

Let us assume we want to maximize a dual function $\xi(\nu)$, ξ being either ξ_{AF} or ξ_{DF}. Bundle methods work by constructing a polyhedral model $\xi_{\mathcal{B}}$ approximating ξ, and solve the stabilized master problem

$$\psi_t(\bar{\nu}) = \sup \left\{ \xi_{\mathcal{B}}(\bar{\nu} + d) + \Delta_t(d) \, : \, d \in \mathbb{R}^n \right\} \tag{9}$$

where $\bar{\nu}$ is the current dual solution and Δ_t is a stabilizing term. The polyhedral model is iteratively refined by adding cutting planes based on the current solution of (9). We refer to [15] for a complete description of the method.

4.2 Lagrangian Heuristic

Both (DF) and (AF) are hard to solve with general-purpose MILP solvers, and the difficulty increases significantly with the number of network nodes and time periods. One of the main challenges is that MILP solvers have difficulties in finding good primal solutions. In this subsection, we present a Lagrangian Heuristic (LH) based on the ideas that (i) removing linking design constraints (1f) and (2e) make (AF) and (DF) decomposable by period; and (ii) using the Lagrangian dual values obtained from the relaxations presented above could guide the search towards a better solution by recovering some part of the connection between periods.

The heuristic works by replacing the variables y_{ij}^p and z_{ij}^p by "unified" design variables h_{ij}^p, representing if the link (i, j) is opened at period p, but ignoring if it is newly opened or just maintained from the previous period. The problem can then be solved for each period independently with an adequate cost d_{ij}^p associated to h_{ij}^p variables.

The problem for period p, $H(p)$, derived from (AF) can then be formulated as follows:

$$\textbf{(H(p))} \quad \min \sum_{(i,j)\in A} \left(d_{ij}^p h_{ij}^p + \phi_{ij}^p + \sum_{t\in V} \left(\nu_i^{tp} - \nu_j^{tp}\right) f_{ij}^{tp} \right) \tag{10a}$$

$$\text{s.t. } \phi_{ij}^p \geq \theta_r \sum_{t\in V} f_{ij}^{tp} - \tau_r \kappa_{ij} \qquad r \in R, (ij) \in A \tag{10b}$$

$$\sum_{j:(i,j)\in A} f_{ij}^{tp} - \sum_{j:(j,i)\in A} f_{ji}^{tp} = D^p(i,t) \qquad i,t \in V, i \neq t \tag{10c}$$

$$\sum_{j:(j,i)\in A} f_{jt}^{tp} = \sum_{\in V} D^p(s,t) \qquad t \in V \tag{10d}$$

$$x_{ij}^{tp} \leq h_{ij}^p \qquad (i,j) \in A, t \in V \tag{10e}$$

$$f_{ij}^{tp} \leq C_{ij}^{tp} x_{ij}^{tp} \qquad (i,j) \in A, t \in V \tag{10f}$$

$$\sum_{t\in V} f_{ij}^{tp} \leq C_{ij}^p h_{ij}^p \qquad (i,j) \in A \tag{10g}$$

$$\sum_{j:(i,j)\in A} x_{ij}^{tp} = 1 \qquad i,t \in V, i \neq t \tag{10h}$$

$$x_{ij}^{tp} \in \{0,1\} \qquad (i,j) \in A, t \in V \tag{10i}$$

$$h_{ij}^p \in \{0,1\} \qquad (i,j) \in A \tag{10j}$$

$$f_{ij}^{tp} \geq 0 \qquad (i,j) \in A, s,t \in V \tag{10k}$$

where ν_i^{tp}, $i,t \in V$ are the Lagrangian multipliers obtained from the Lagrangian relaxation.

The heuristic constructs a solution $(\bar{x}, \bar{y}, \bar{z}, \bar{f})$ to (AP) as follows. The decomposed problem is first solved for period $p = 1$ with $d_{ij}^p = c_{ij}$, as a consequence of (1g). Given an optimal solution (h^*, x^*, f^*) to $H(1)$, the heuristic solution is built for the first period by setting $\bar{x}_{ij}^{t1} = x_{ij}^{*t1}$, $\bar{y}_{ij}^1 = h_{ij}^{*1}$, $\bar{z}_{ij}^1 = 0$ and $\bar{f}_{ij}^{t1} = f_{ij}^{*t1}$ for all $(i,j) \in A$ and $t \in V$.

The construction then iterates over the next periods. Given a partial solution computed up to period $p-1$, $H(p)$ is solved by setting, for each $(i,j) \in A$, $d_{ij}^p = m_{ij}$ if $\bar{y}_{ij}^{p-1} + \bar{z}_{ij}^{p-1} = 1$ (i.e. the arc was opened at the previous period) and $d_{ij}^p = c_{ij}$ otherwise. Given an optimal solution (h^*, x^*, f^*) to $H(p)$, the heuristic solution is built for period p by setting $\bar{x}_{ij}^{tp} = x_{ij}^{*tp}$, $\bar{y}_{ij}^p = 0$ and $\bar{z}_{ij}^p = h_{ij}^{*p}$ if $\bar{y}_{ij}^{p-1} + \bar{z}_{ij}^{p-1} = 1$, $\bar{y}_{ij}^p = h_{ij}^{*p}$ and $\bar{z}_{ij}^p = 0$ otherwise, and $\bar{f}_{ij}^{tp} = f_{ij}^{*tp}$ for all $(i,j) \in A$ and $t \in V$.

A similar approach can be used with the linear relaxation of (DF), replacing the objective function in H(p) by:

$$\sum_{(i,j)\in A} \left(d_{ij}^p h_{ij}^p + \phi_{ij}^p + \sum_{s,t\in V} \left(\nu_i^{stp} - \nu_j^{stp}\right) f_{ij}^{tp}/|V| \right)$$

The problem $H(p)$ has a structure similar to the standard Fixed-Charge Multicommodity Capacitated Network Design Problem [17] but the problem remains challenging because of routing constraints (10h) and the piecewise objective function ϕ_{ij}^p.

4.3 Computational aspects

The performance of our algorithm depends on several strategic issues. We first review how to initialize the dual multipliers, then we describe how the Lagrangian heuristic interacts with the Bundle method. Finally, we point out how the primal solutions can help strengthening the polyhedral approximation of the Lagrangian dual.

4.3.1 Warm start

The initialization of the dual multipliers is a fundamental issue. Let us first consider (AF). The objective is to find a point $\bar{\nu}$ whose value $\xi_{AF}(\bar{\nu})$ is not too far from the optimum of ξ_{AF}, in order to limit the number of iterations required to converge.

Given that (AF) contains a classical linear multi-commodity flow as a subproblem, it is logical to consider that the dual variables of this subproblem could be a good starting point. To this aim, we solve a restricted version of (AF) that involves only the flow variables f_{ij}^{tp}, ignoring the design variables and considering all arcs as open, i.e.

$$\textbf{(WS)} \min \sum_{p\in\mathcal{P}} \sum_{(i,j)\in A} \phi\left(\kappa_{ij}, \sum_{t\in V} f_{ij}^{tp}\right) \tag{11a}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} f_{ij}^{tp} - \sum_{j:(j,i)\in A} f_{ji}^{tp} = D^p(i,t) \qquad i,t\in V, i\neq t, p\in\mathcal{P} \tag{11b}$$

$$\sum_{j:(j,i)\in A} f_{jt}^{tp} = \sum_{\in V} D^p(s,t) \qquad t\in V, p\in\mathcal{P} \tag{11c}$$

$$f_{ij}^{tp} \leq C_{ij}^{tp} \qquad (i,j)\in A, t\in V, p\in\mathcal{P} \tag{11d}$$

$$\sum_{t\in V} f_{ij}^{tp} \leq C_{ij}^{p} \qquad (i,j)\in A, p\in\mathcal{P} \tag{11e}$$

$$f_{ij}^{tp} \geq 0 \qquad (i,j)\in A, s,t\in V, p\in\mathcal{P} \tag{11f}$$

The dual variables of flow equations (11b) represent a “good” starting point for the Lagrangian approach.

For (DF), instead of solving a problem similar to (11), we reuse the best solution $\nu_i^{*tp}, i,t\in V, p\in P$, found for the Lagrangian dual of the (AF). To initialize BM when applied to ξ_{DF}, we adopt the following formula:

$$\bar{\nu}_i^{tps} = \nu_i^{*tp}, \quad i,s,t\in V, p\in\mathcal{P}\,. \tag{12}$$

The numerical experiments confirmed that the objective functions has the same value, i.e. $\xi_{DF}(\bar{\nu}) = \xi_{AF}(\nu^*)$.

4.3.2 Stabilization and stopping criteria

The stabilization term employed in (9) is the quadratic one, i.e. $\Delta_t(d) = -\frac{1}{2t}\| \, d \, \|^2$, because it leads to a fast convergence rate. Morever, a specialized quadratic solver is available [14] in the framework we use (see Section 5), that can be directly applied to solve the stabilized master.

Given a scaling factor t^* and the final (relative) accuracy $\epsilon > 0$, BM stops when

$$\frac{t^*}{2}\|\zeta^*\|_2^2 + \alpha^* \leq \epsilon \max\{1, \xi(\bar{\nu})\} \tag{13}$$

holds, where ζ^* and α^* are, respectively, the aggregated supergradient and the aggregated linearization error (see, e.g., [15]).

4.3.3 Interaction between BM and LH

The solution obtained by the Lagrangian heuristic depends on the dual multipliers obtained by the Bundle method. A good trade-off must be found between time spent for finding a good lower bound (i.e. time used in iterations of BM) and time spent for finding primal solutions with LH. At the extremities of the spectrum, we could choose to run LH only once with the best dual multipliers obtained with BM, or to run LH at each iteration of BM. Both options are quite ineffective: the first approach only produces a single feasible solution that can be of very bad quality, while the second approach only allows to perform a very small number of BM iterations and spends a large part of the computational time to solve (10), leading to a very poor lower bound.

As a compromise, we call LH at a decreasing rate, very often at the beginning to quickly find feasible solutions, then less frequently to give more time to BM to improve the lower bound. We first set a maximum total running time t_M. Lagrangian dual values for (AF) are computed using the warm start procedure described above. Then, BM is applied to ξ_{AF} for a predefined time t_{AF} ($\ll t_M$). The initial phase stops when either (13) is satisfied or the time limit t_{AF} is reached.

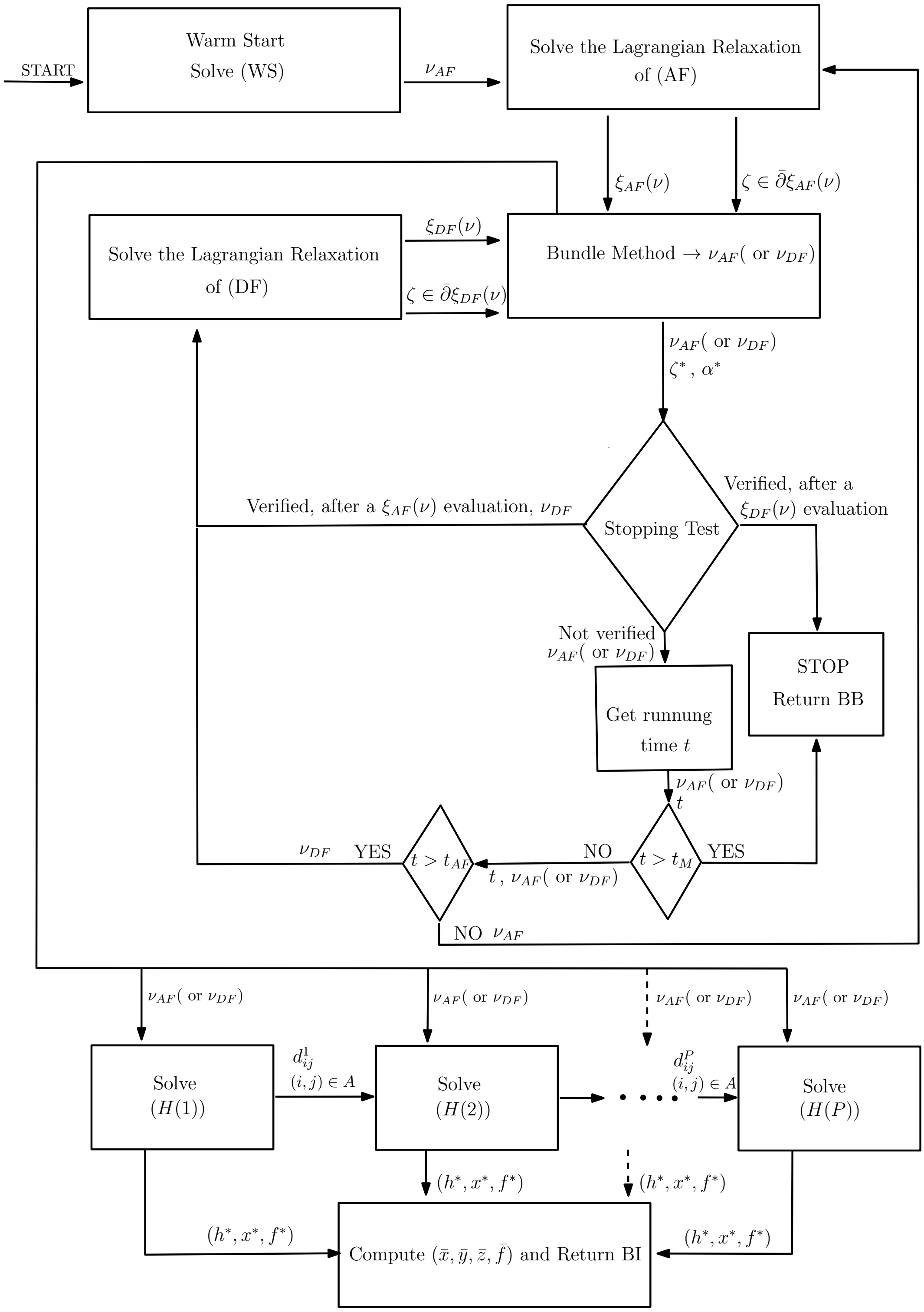
In the remaining time, the algorithm attempts to improve the lower bound, i.e., BM is applied to ξ_{DF}. Dual values are initialized with (12), then BM is executed for a given time t_B. After this phase, LH is applied, with a maximum running time of t_H for each period. Then, the procedure iterates between BM and LH, increasing by t_B the time allowed for BM at each iteration. The procedure stops when (13) is satisfied or t_M is reached.

Figure 3 describes the work flow of our method, where ν_{AF} and ν_{DF} denote the Lagrangian multipliers ν_i^{tps} and ν_i^{tp}, for $i, s, t \in V, p \in \mathcal{P}$, respectively, of the Lagrangian relaxation of (DF) and (AF). BM is summarized in the upper part of the figure, while LH is shown in the lower part. Note that LH needs the Lagrangian multipliers ν_{AF}(or ν_{DF}) from BM to set up the costs of the objective function of $(H(p))$, $p \in \mathcal{P}$. The best (lower) bound provided by BM is denoted BB, and the upper bound given by the best integer solution provided by LH is denoted BI.

4.3.4 Exploiting Feasible Solutions

The information gained from primal solutions obtained with LH can be used to improve the convergence of BM. The best solution found by LH provides an upper bound u on the value of the Lagrangian dual. The polyhedral approximation of the Lagrangian dual can be improved by adding the cut

$$\xi_{\mathcal{B}}(\bar{\nu} + d) \leq u$$



$\bar{\partial}\xi_{AF}(\nu)$ is the superdifferential of ξ_{AF} at ν / $\bar{\partial}\xi_{DF}(\nu)$ is the superdifferential of ξ_{DF} at ν

Figure 3: Work flow of the Algorithm

to the master problem (9), as suggested in [18]. We have observed that adding such a cut to the master problem reduces the execution time by an order of magnitude for our specific problem.

4.3.5 Aggregated vs Disaggregated master

We solve the aggregated master problem with the Bundle method. It has been shown in [18] that the disaggregated master problem should work better when the Lagrangian dual can be formulated as a sum of functions, as it is in this application. Unfortunately, the number $|V|$ of subproblems for all our instances is relatively small. Because of the small size of the set of nodes, the information accumulated in the disaggregated master problem is insufficient to yield the optimality in few iterations. On the other hand, all this information makes the disaggregated master problem difficult to solve. Consequently, the bundle method presents a large number of iterations and each iteration requires a big computational effort. This observation was computationally confirmed by our experiments reported in [28], which justify the choice of using the aggregated master problem.

5 Numerical experiments

In all our experiments, CPLEX 12.5.1 was used to solve the MILPs involved in the algorithm, i.e., the Lagrangian subproblems (3) and (7), the heuristic problems LH(p) (10) and the restricted problem used for the initialization of the dual multipliers (11). We must stress that we did not solve (3) and (7) but their linear relaxations. We have tried to switch to their integer versions in the last iterations of BM but without any significant improvement of the lower bound.

The Lagrangian dual was solved using an implementation of BM available in a general-purpose C++ non-smooth optimization code developed by A. Frangioni and already successfully used in other applications [10, 18].

The algorithm has been compiled with GNU g++ 4.4.5 (with the -O3 optimization option) and ran single-threaded on a server with multiple Opteron 6174 processors (12 cores, 2.2 GHz), each with with 32 GB of RAM, under a i686 GNU/Linux operating system.

Our algorithm has been evaluated on a set of network topologies extracted from the SNDlib library [24]. This library provides a repository of several topologies together with their link capacities, link costs, and traffic demands. The following topologies have been considered (in alphabetical order): *cost-266*, *france*, *germany50*, *giul39*, *india35*, *nobel-eu*, *norway*, *pioro40*, *polska*, *sun*, *zib54*. We focused on medium to large size networks that make the problem very challenging for MILP solvers.

For each of these topologies, instances with 5, 10 and 15 periods were generated. Demand matrices were generated as described in [26]. For all the experiments, we used the following time limits: $t_M = 36000s$, $t_{AF} = 100s$ and $t_H = \frac{|V|.|A|}{5}s$. The required accuracy ϵ of BM was set to $1e-4$ when applied to ξ_{AF} and to $1e-6$ when applied to ξ_{DF}. Only in a few cases the accuracy of the solution of ξ_{DF} is reached because increasing the time t_H penalizes the execution time of BM in favor of LH. However, we are more interested in finding a better primal solution. The performance of LH strongly depends

			AF-CPLEX		DF-CPLEX		BM / LH	
Network	$\|V\|$	$\|A\|$	BB	BI	BB	BI	BB	BI
polska	12	36	**8.97%**	**8.97%**	**8.97%**	**8.97%**	0.00%	11.86%
france	25	90	0.65%	3.12%	**1.39%**	**2.74%**	0.00%	5.35%
norway	27	102	0.28%	**7.99%**	**0.58%**	97.29%	0.00%	10.13%
sun	27	204	-1.17%	25.74%	**1.91%**	**6.95%**	-0.02%	10.35%
nobel-eu	28	82	**4.63%**	18.34%	3.36%	—	-0.09%	**14.60%**
india35	35	160	-4.80%	32.42%	**0.15%**	—	-0.01%	**20.05%**
cost-266	37	114	-0.26%	6.28%	**0.00%**	7.11%	-0.04%	**1.75%**
giul39 (*)	39	334	-1.27%	—	—	—	**0.00%**	**1.44%**
pioro40	40	178	**12.78%**	—	0.00%	—	-0.03%	**23.65%**
germany50	50	176	**5.64%**	—	0.00%	—	-1.76%	**52.96%**
zib54	54	162	-0.37%	14.77%	**0.00%**	—	-12.34%	**3.23%**

Table 1: Numerical results for instances with 5 periods

on appropriate tuning of t_H. On the one hand, a small value of t_H may yield a primal solution of low quality because the execution of the solver of (10) could be stopped too soon. On the other hand, a fine-tuned value of t_H may allow to find several good primal solutions. We noted that the chosen value of t_H must take in account the dimension of the network. We observed that $t_H = \frac{|V|.|A|}{5}s$ allows to obtain good enough primal solutions for small instances and relatively good solutions for the largest ones.

To position the results obtained with the methods proposed for solving (DF) or (AF), we compare our algorithm with CPLEX. The optimality accuracy of CPLEX was set to 1e-6 as well. We have tested all the available methods in CPLEX (Automatic, Primal, Dual, Barrier), and ultimately we have selected the Barrier algorithm as the one offering the best performance for solving the MILP problems (AF) and (DF) and their linear relaxation. We observed that the Barrier algorithm significantly outperformed the default Automatic setting in which CPLEX chooses the algorithm.

Results are reported for $P = 5, 10$ and 15, respectively, in Tables 1 to 3. The tables report the gaps with respect to the Linear Relaxation (LR) solution of (DF) found with the Barrier algorithm of CPLEX. Columns *AF-CPLEX* and *DF-CPLEX* report the results obtained by solving (AF) and (DF) with the Barrier algorithm of CPLEX, also with a maximum running time of $36000s$. The results of our method are reported in column *BM / LH*. For each approach, we report the gap between the Best Bound (BB) and LR as well as the gap between the Best Integer (BI) and LR. The best lower bounds and primal solutions found are emphasized in bold. Unfortunately, CPLEX is not able to solve LR for *giul39* with 5, 10 and 15 periods, and *pioro40, germany50* and *zib54* with 15 periods. For those instances (marked with a star in the tables), the gap is computed with respect to the best bound obtained by BM. Finally, we do not report the execution time because the methods did not stop before $36000s$, except for *polska* with 5 periods, using CPLEX and for *polska* and *france* with 5 periods, using BM.

Several observations can be drawn out of these numerical results:

- Only one instance (*polska* for 5 periods) could be solved to optimality (both with AF-CPLEX and DF-CPLEX). This confirms how challenging these instances are.

			AF-CPLEX		DF-CPLEX		BM / LH	
Network	$\|V\|$	$\|A\|$	BB	BI	BB	BI	BB	BI
polska	12	36	**3.91%**	4.86%	3.85%	**4.76%**	0.00%	8.22%
france	25	90	0.35%	1.00%	**0.86%**	**0.86%**	-0.01%	10.28%
norway	27	102	-0.97%	—	**0.00%**	—	-0.02%	**14.51%**
sun	27	204	-3.17%	—	**0.01%**	—	-0.02%	**14.09%**
nobel-eu	28	82	**4.68%**	49.16%	2.99%	—	-0.22%	**16.01%**
india35	35	160	-3.99%	—	**0.01%**	—	-4.35%	**20.04%**
cost-266	37	114	-1.51%	—	**0.00%**	—	-0.07%	**3.89%**
giul39(*)	39	334	**3.37%**	—	—	—	0.00%	—
pioro40	40	178	**5.69%**	—	0.00%	—	-2.14%	**19.84%**
germany50	50	176	**0.03%**	—	0.00%	—	-6.42%	—
zib54	54	162	-1.13%	—	**0.00%**	—	-24.03%	**3.12%**

Table 2: Numerical results for instances with 10 periods

			AF-CPLEX		DF-CPLEX		BM / LH	
Network	$\|V\|$	$\|A\|$	BB	BI	BB	BI	BB	BI
polska	12	36	2.33%	4.12%	**2.55%**	**4.09%**	0.00%	6.86%
france	25	90	**0.28%**	**0.28%**	**0.28%**	**0.28%**	-0.01%	7.52%
norway	27	102	-2.49%	—	**0.00%**	—	-0.03%	**33.89%**
sun	27	204	-2.79%	—	**0.00%**	42.90%	-0.03%	**18.59%**
nobel-eu	28	82	2.05%	41.36%	**3.79%**	—	-1.19%	**20.21%**
india35	35	160	-3.40%	—	**0.00%**	—	-5.81%	**20.78%**
cost-266	37	114	-2.29%	—	**0.00%**	—	-0.14%	**8.76%**
giul39(*)	39	334	**3.06%**	—	—	—	0.00%	—
pioro40(*)	40	178	**3.25%**	—	—	—	0.00%	—
germany50(*)	50	176	**6.57%**	—	—	—	0.00%	—
zib54(*)	54	162	**37.78%**	—	—	—	0.00%	**45.49%**

Table 3: Numerical results for instances with 15 periods

- As the number of periods increases, the quality of the results decreases for both CPLEX and BM/LH but the latter is still able to find a feasible solution except for *giul39* and *germany50* when $P = 10$ and *giul39*, *pioro40* and *germany50* when $P = 15$.

- As the number of nodes increases (and for any number of periods), the quality of the best feasible solution produced by the CPLEX solver decreases compared to BM/LH until reaching the situation where only BM/LH is able to produce a solution (though their quality deteriorates as the number of periods increases, see below).

- Concerning the lower bounds, the results are more contrasted. The best bound produced by DF-CPLEX partially confirm the better lower bounds provided by (DF) in particular for $P = 5$ but as the size of the instances increases (due to the number of nodes and/or periods), its performance compared to AF-CPLEX decreases, because linear programs become much larger and time consuming to solve; this trend is striking for $P = 15$, where even the linear relaxation of (DF) could not be computed within the time limit. For this number of periods, a global trend nevertheless appears, for small-size instances (1,2) AF-CPLEX is the best method though close to the two others, for medium-size instances (3,4,5,6,7) DF-CPLEX yields the best lower bound and BM/LH the best feasible solution, and for large-scale instances (8,9,10), AF-CPLEX is still able to provide the best lower bound (compared to the two others) whereas left without any feasible solution. The latter observation shows the possibility to further improve the BM/LH method to obtain the same trend as the one observed when $P = 10$ for which the BM/LH is clearly the winner (beside for small instances 1 and 2). A possible explanation stems from the deviation induced by $H(p)$ which performs per-period given a partial solution computed up to period $p - 1$.

In the above experiments, we have used a particular piecewise function whose derivative has the following form:

$$\phi'(\kappa_{ij}, l_{ij}^p) = \begin{cases} 1 & \text{if } 0 \leq l_{ij}^p/\kappa_{ij} \leq 0.3 \\ 3 & \text{if } 0.3 \leq l_{ij}^p/\kappa_{ij} \leq 0.6 \\ 10 & \text{if } 0.6 \leq l_{ij}^p/\kappa_{ij} \leq 0.7 \\ 100 & \text{if } 0.7 \leq l_{ij}^p/\kappa_{ij} \leq 0.8 \\ 500 & \text{if } 0.8 \leq l_{ij}^p/\kappa_{ij} \leq 0.9 \\ 5000 & \text{if } 0.9 \leq l_{ij}^p/\kappa_{ij} \end{cases}$$

In order to measure the impact of the number of pieces on the difficulty of the problem, we also performed additional experiments described below. We present below the results obtained for instances *france*, *germany50*, *nobel-eu*, *norway*, *polska*, *sun* with 5 periods for three different forms of the piecewise function ϕ: ϕ_1, ϕ_2 and ϕ_3. For function ϕ_1, we keep the number of breakpoints unmodified while changing the values; its derivative is

$$\phi_1'(\kappa_{ij}, l_{ij}^p) = \begin{cases} 5 & \text{if } 0 \le l_{ij}^p/\kappa_{ij} \le 0.1 \\ 10 & \text{if } 0.1 \le l_{ij}^p/\kappa_{ij} \le 0.3 \\ 50 & \text{if } 0.3 \le l_{ij}^p/\kappa_{ij} \le 0.5 \\ 100 & \text{if } 0.5 \le l_{ij}^p/\kappa_{ij} \le 0.7 \\ 500 & \text{if } 0.7 \le l_{ij}^p/\kappa_{ij} \le 0.9 \\ 1000 & \text{if } 0.9 \le l_{ij}^p/\kappa_{ij} \end{cases}$$

For ϕ_2, we consider a higher number of breakpoints. The derivative of the function ϕ_2 is given below:

$$\phi_2'(\kappa_{ij}, l_{ij}^p) = \begin{cases} 0.1 & \text{if } 0 \le l_{ij}^p/\kappa_{ij} \le 0.20 \\ 0.2 & \text{if } 0.20 \le l_{ij}^p/\kappa_{ij} \le 0.30 \\ 0.5 & \text{if } 0.30 \le l_{ij}^p/\kappa_{ij} \le 0.35 \\ 1 & \text{if } 0.35 \le l_{ij}^p/\kappa_{ij} \le 0.40 \\ 2 & \text{if } 0.40 \le l_{ij}^p/\kappa_{ij} \le 0.45 \\ 5 & \text{if } 0.45 \le l_{ij}^p/\kappa_{ij} \le 0.50 \\ 10 & \text{if } 0.50 \le l_{ij}^p/\kappa_{ij} \le 0.55 \\ 20 & \text{if } 0.55 \le l_{ij}^p/\kappa_{ij} \le 0.60 \\ 50 & \text{if } 0.60 \le l_{ij}^p/\kappa_{ij} \le 0.65 \\ 100 & \text{if } 0.65 \le l_{ij}^p/\kappa_{ij} \le 0.70 \\ 200 & \text{if } 0.70 \le l_{ij}^p/\kappa_{ij} \le 0.75 \\ 500 & \text{if } 0.75 \le l_{ij}^p/\kappa_{ij} \le 0.80 \\ 1000 & \text{if } 0.80 \le l_{ij}^p/\kappa_{ij} \le 0.85 \\ 2000 & \text{if } 0.85 \le l_{ij}^p/\kappa_{ij} \le 0.90 \\ 5000 & \text{if } 0.90 \le l_{ij}^p/\kappa_{ij} \end{cases}$$

For the last one, ϕ_3, we approximate the routing cost by means of the linear function:

$$\phi_3(\kappa_{ij}, l_{ij}^p) = 100\, l_{ij}^p\,.$$

We report the results with CPLEX for solving the MILP problems (AF) and (DF), respectively in Tables 4 and 5, and with BM / LH for solving (DF) in Tables 6. For each ϕ_i, $i = 1, \ldots, 4$, we give the gap between the Best Bound (BB) and LR as well as the gap between the Best Integer (BI) and LR. For ϕ_3 we report also the execution time in the column *Time*.

The main conclusion from these experiments is that CPLEX struggles a lot to solve the problem when the number of pieces increases. On the contrary, our heuristic is able to provide a feasible solution in each case and lower bounds close to the linear relaxation of AF.

6 Conclusion

This paper presents a Lagrangian relaxation approach to solve a multi-commodity capacitated fixed charge network design problem with variable traffic demands over multiple

			AF-CPLEX ϕ_1		AF-CPLEX ϕ_2		AF-CPLEX ϕ_3			AF-CPLEX ϕ_0	
Network	$\|V\|$	$\|A\|$	BB	BI	BB	BI	Time	BB	BI	BB	BI
polska	12	36	6.03%	6.48%	28.61%	29.60%	1	0.01%	0.01%	8.97%	8.97%
france	25	90	0.00%	0.00%	1.55%	4.58%	7	0.00%	0.00%	0.65%	3.12%
norway	27	102	21.21%	88.62%	3.34%	3.34%	36000	-0.83%	12.27%	0.28%	7.99%
sun	27	204	0.26%	8.93%	0.12%	7.28%	3134	0.00%	0.00%	-1.17%	25.74%
nobel-eu	28	82	15.19%	37.34%	7.07%	33.52%	36000	0.21%	9.06%	4.63%	18.34%
germany50	50	176	25.52%	—	16.98%	—	36000	-4.28%	28.26%	5.64%	—

Table 4: Numerical results of (AF), changing ϕ

			DF-CPLEX ϕ_1		DF-CPLEX ϕ_2		DF-CPLEX ϕ_3			DF-CPLEX ϕ_0	
Network	$\|V\|$	$\|A\|$	BB	BI	BB	BI	Time	BB	BI	BB	BI
polska	12	36	5.88%	6.48%	29.27%	29.60%	1	0.01%	0.01%	8.97%	8.97%
france	25	90	0.00%	0.00%	2.22%	4.10%	12	0.00%	0.00%	1.39%	2.74%
norway	27	102	—	—	0.06%	—.	36000	0.80%	8.75%	0.58%	97.29%
sun	27	204	1.21%	2.97%	1.31%	6.00%	39	0.00%	0.00%	1.91%	6.95%
nobel-eu	28	82	5.56%	—	4.90%	—	36000	1.35%	6.22%	3.36%	—
germany50	50	176	—	—	—	—	36000	0.01%	—	0.00%	—

Table 5: Numerical results of (DF), changing ϕ

			BM / LH ϕ_1		BM / LH ϕ_2		BM / LH ϕ_3			BM / LH ϕ_0	
Network	$\|V\|$	$\|A\|$	BB	BI	BB	BI	Time	BB	BI	BB	BI
polska	12	36	0.00%	7.54%	0.00%	31.01%	16	0.00%	0.02%	0.00%	11.86%
france	25	90	0.00%	0.12%	0.00%	4.78%	98	0.00%	0.05%	0.00%	5.35%
norway	27	102	0.00%	99.02%	-0.01%	12.22%.	36000	0.00%	8.91%	0.00%	10.13%
sun	27	204	-0.03%	3.68%	-0.02%	7.94%	140	0.00%	0.30%	-0.02%	10.35%
nobel-eu	28	82	-0.12%	22.45%	-0.07%	26.45%	36000	-0.02%	5.12%	-0.09%	14.60%
germany50	50	176	-1.12%	33.45%	-2.01%	48.96%	36000	-0.01%	19.11%	-1.76%	52.96%

Table 6: Numerical results of BM / LH, changing ϕ

time periods and including routing decisions. This practical problem is computationally challenging since it aims at considering several tasks at once. In fact, the MILP problem is even difficult testing reasonable instances as already reported in [26].

We show how to construct a feasible solution to the problem by exploiting the Lagrangian multipliers. The method is able to find feasible solutions faster than CPLEX, and in many cases CPLEX is not even able to find a single feasible solution. For some very large instances, the Lagrangian heuristic also fails at finding a primal solution.

One major difficulty in solving large instances is that the heuristic currently relies on solving the problem for one period, which is already challenging by itself. Improvements could be obtained by focusing on fast and efficient heuristics for solving the single period problem. Another difficulty arises from the combination of routing constraints and the piecewise objective function. Recent results by Fortz et al. [12] on problems with similar constraints could be integrated in our formulations to strengthen the lower bounds.

Acknowledgment

This work is supported by the Interuniversity Attraction Poles Programme P7/36 "COMEX" initiated by the Belgian Science Policy Office.

References

[1] A. Altin, B. Fortz, M. Thorup, and H. Ümit. Intra-domain traffic engineering with shortest path routing protocols. *Annals of Operations Research*, 204:65–95, 2013.

[2] A. Astorino, A. Frangioni, A. Fuduli, and E. Gorgone. A nonmonotone proximal bundle method with (potentially) continuous step decisions. *SIAM Journal on Optimization*, 23(3):1784–1809, 2013.

[3] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[4] A. Bley, B. Fortz, E. Gourdin, K. Holmberg, O. Klopfenstein, M. Pióro, A. Tomaszewski, and H. Ümit. Optimization of OSPF routing in IP networks. In A. M. C. A. Koster and X. Muñoz, editors, *Graphs and Algorithms in Communication Networks: Studies in Broadband, Optical, Wireless and Ad Hoc Networks*, chapter 8, pages 199–240. Springer, 2010.

[5] N. Bostel, P. Dejax, P. Guez, and F. Tricoire. Multiperiod planning and routing on a rolling horizon for field force optimization logistics. *Operations Research/Computer Science Interfaces*, 43:503–525, 2008.

[6] P. Cappanera and A. Frangioni. Symmetric and asymmetric parallelization of a cost-decomposition algorithm for multi-commodity flow problems. *INFORMS Journal On Computing*, 15(4):369–384, 2003.

[7] J. Castro. A specialized interior-point algorithm for multicommodity network flows. *SIAM Journal on Optimization*, 10(852- 877), 2000.

[8] A. Costa. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429 – 1450, 2005.

[9] A. M. Costa, J.-F. Cordeau, and B. Gendron. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Comp. Opt. and Appl.*, 42(3):371–392, 2009.

[10] T. Crainic, A. Frangioni, and B. Gendron. Bundle-based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems. *Discrete Applied Mathematics*, 112:73–99, 2001.

[11] G. d'Antonio and A. Frangioni. Convergence analysis of deflected conditional approximate subgradient methods. *SIAM Journal on Optimization*, 20(1):357–386, 2009.

[12] B. Fortz, L. Gouveia, and M. Joyce-Moniz. On the convex piecewise linear unsplittable multicommodity flow problem. In *12th International Conference on the Design of Reliable Communication Networks, DRCN 2016, Paris, France, March 15-17, 2016*, pages 9–13, 2016.

[13] B. Fortz and M. Thorup. Increasing internet capacity using local search. *Computational Optimization and Applications*, 29(1):13–48, 2004.

[14] A. Frangioni. Solving Semidefinite Quadratic Problems Within Nonsmooth Optimization Algorithms. *Computers & Operations Research*, 21:1099–1118, 1996.

[15] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.

[16] A. Frangioni and G. Gallo. A bundle type dual-ascent approach to linear multicommodity min cost flow problems. *INFORMS Journal On Computing*, 11(4):370 – 393, 1999.

[17] A. Frangioni and B. Gendron. 0-1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, 157(6):1229–1241, 2009.

[18] A. Frangioni and E. Gorgone. Generalized Bundle Methods for Sum-Functions with "Easy" Components: Applications to Multicommodity Network Design". *Mathematical Programming*, 145:133–161, 2014.

[19] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms* II—*Advanced Theory and Bundle Methods*, volume 306 of *Grundlehren Math. Wiss.* Springer-Verlag, New York, 1993.

[20] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of SIAM*, 8(4):703–712, 1960.

[21] K. Le and J. Day. Rolling horizon method: A new optimization technique for generation expansion studies. *Power Apparatus and Systems, IEEE Transactions on*, 101(9):3112–3116, 1982.

[22] C. Lemaréchal. Lagrangian relaxation. In *Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions [based on a Spring School, Schloß Dagstuhl, Germany, 15-19 May 2000]*, pages 112–156, 2001.

[23] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Math. Program., Ser. B*, 120:221–259, 2009.

[24] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3):276–286, 2010.

[25] D. Papadimitriou and B. Fortz. Methods for time-dependent combined network design and routing optimization. In *IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, December 8-12, 2014*, pages 1303–1309, 2014.

[26] D. Papadimitriou and B. Fortz. Time-dependent combined network design and routing optimization. In *Proceedings of 2014 IEEE International Conference on Communications (ICC)*, Sidney, Australia, June 2014.

[27] D. Papadimitriou and B. Fortz. A rolling horizon heuristic for the multiperiod network design and routing problem. *Networks*, 66(4):364–379, 2015.

[28] D. Papadimitriou, B. Fortz, and E. Gorgone. Lagrangian relaxation for the time-dependent combined network design and routing problem. In *2015 IEEE International Conference on Communications, ICC 2015, London, United Kingdom, June 8-12, 2015*, pages 6030–6036, 2015.

[29] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[30] F. Tricoire, N. Bostel, P. Dejax, and P. Guez. Exact and hybrid methods for the multiperiod field service routing problem. *Central European Journal of Operations Research*, 21(2):359–377, 2013.