Lagrangian relaxation for the time-dependent combined network design and routing problem

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

Dimitri Papadimitriou
Alcatel-Lucent Bell Labs
Antwerp, Belgium
Email: dimitri.papadimitriou@alcatel-lucent.com

***Abstract*—In communication networks, the routing decision process (distributed and online) remains decoupled from the network design process, i.e., resource installation and allocation planning process (centralized and offline). To reconcile both processes, we ambition to design a distributed optimization technique aware of distributed nature of the routing process by decomposing the optimization problem along same dimensions as (distributed) routing decision process. For this purpose, we generalize the capacitated multi-commodity capacitated fixed charge network design (MCND) 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. 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, the approach is more robust than any LP solvers and it always returns some solutions. Instead, we proved that `CPLEX`, which uses the Dual Simplex algorithm, is not able to provide a solution for large instances.**

I. 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 (re-)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 rely instead on the multi-commodity capacitated fixed charge network design (MCND) problem introduced by [1], [2], [3] which deals with the simultaneous optimization of capacity installation cost and traffic flow routing cost. In this 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 [4], we generalize 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. 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 [5].

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 [6]. To circumvent this problem, [1] 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 inline with the objective of obtaining a decomposition of the original optimization problem which preserves the distributed nature of the routing decision process.

The remainder of this paper is structured as follows. In section II, we present prior work in resolving the MCND problem and we detail our objectives and contributions. In Section III we formulate the model. In the next section, we present the numerical resolution method used for this application. In section V, we report the computational results and analyze them. Finally, section VI concludes this paper together with directions for future research work.

II. Prior Work, Objectives and Contribution

A. Prior Work

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, which makes them very hard to solve with standard approaches. Indeed, specialized algorithms are required [7], [8] and the use of parallel architectures could be necessary [9]. The complexity becomes even higher if integer variables are present in the models to represent logical decisions. The resulting mathematical model is that of a mixed integer linear program (MILP) problem with multi-commodity network flow structure.

1) Enumerative Approaches: The simplest approach for solving integer-programming problem consists in enumerate all finitely many possibilities and eliminate many possibilities by domination or by feasibility arguments. The Branch-and-Bound (B&B) technique before enumerating the possible solution (candidates) along a branch, checks them against a known upper and lower estimated bounds on the optimal solution value. Standard enumerative approaches for these problems require the repeated computation of lower bounds on the optimal value of the problem (assuming a minimization problem). For this purpose, the B&B approach uses the solution to some relaxation of the problem obtained by dropping the integrality conditions and solving the resulting continuous LP over the set of points satisfying all but the integrality restrictions. However, these can be computationally very costly. Furthermore, the bounds obtained by applying these methods can be rather weak, leading in turn to the enumeration of a very large number of sub-problems. Hence, only the smallest instances could usually be solved by such an approach. This fact explains the interest for innovative methods which efficiently compute accurate lower bounds for larger instances. One of the most promising techniques to reach this is that of Lagrangian relaxation [10], [11], [9].

2) Lagrangian Relaxation: The Lagrangian relaxation of the original (general mixed integer) problem consists in taking the set of complicating constraints of this problem into the objective function with 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 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 minimization problem. The latter obviously requires appropriate procedures [12], [13]. When coupled with appropriate heuristics and the use of valid inequalities to further strengthen the obtained bound, efficient "branch & cut & price" approaches can be obtained [14].

Often, there exist several ways by which a given original problem can be relaxed in a Lagrangian fashion, and it is not clear a priori which is the best one. Moreover, there are basically two families of algorithms to solve the Lagrangian dual: the bundle methods [15] and the sub-gradient methods [16]. Appropriate choice of the method is crucial both for the time efficiency of the lower bound computation and for the quality of the primal solution obtained. Sub-gradient methods require relatively low computation time per iteration but their convergence properties are often unpredictable and thus not adapted to our problem.

In solving the MCND problem, T.Crainic et al. [1] demonstrate that the bundle methods show 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 polyhedral function which corresponds to solve a linear problem, called master problem, in order to obtain the new iterate, following the proximal Bundle method [12]. For numerical reasons, the master problem must be "stabilized" [17] and, in addition, recently it has proved that it could be "structured". The latter thing means that it should exploit some properties of the specific problem [18]. To circumvent some these limits, new sub-gradient methods have also been proposed recently like deflected, projected [13] and primal-dual approaches [19]. In particular, the latter minimizes the gain in parameter adjustment provided by the bundle methods; however, they still converge much faster as they use much more detailed piecewise linear model of the objective function. These arguments justify (a priori) the choice of the bundle method that is well suited to our formulation of the Lagrangian dual.

B. Objectives and Contribution

The model considered in this paper extends the multi-commodity capacitated fixed charge network design (MCND) class of problems that include 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 refer to a multi-period problem. We propose a Lagrangian approach for computing a lower bound. For this purpose, we relax the flow conservation equations such that the Lagrangian relaxation is 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 any LP solver and it always returns some solution. In comparison, CPLEX is not able to provide a solution for all the instances considered in this study, on which the proposed Lagrangian formulation has been evaluated; in particular, those comprising the larger number of nodes.

III. Optimization Model

The problem can be formulated as follows. Given a directed network $G = (V, A)$, where N 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 origin-destination pair s, t has an associated demand $D^p(s, t)$ that must flow between s and t at a certain period p. Each arc (i, j) in the network can only be used whether the corresponding installation cost c_{ij} or the maintenance cost m_{ij} is paid, in this case it has an aggregated (mutual) arc capacity C^p_{ij}. Also, individual arc capacities C^{stp}_{ij} are defined for the maximum amount of flow of the pair s, t on arc (i, j) at the period p. Furthermore, a routing cost for each arc (i, j) and each period p is defined by an increasing convex function of its utilization, inspired by [20], [6].

Let $\mathcal{D} \subseteq V$ denote the set of reachable destinations. A routing function r determines $\forall t \in \mathcal{D}$ and $\forall u \in V$ the adjacent node (next-hop) w of u, $(u, w) \in A$, along a given trajectory from node u to destination t (reachable via node v). This trajectory is determined by the routing algorithm which computes $\forall t \in \mathcal{D}$ and for each node $u \in V$, a (routing) path $\pi(u, w, \ldots, v : t)$ to destination t. The application of the routing function r to the result of this computation enables any node $u \in V$ to forward its incoming traffic directed to destination t along a loop-free path to that destination. Finally, we refer to a distributed routing function r_u when the function r is executed locally at each node $u \in V$ and independently of all other nodes $v \in V \setminus \{u\}$.

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, (ii) maintenance costs and (iii) routing costs. The link installation cost c_{ij} accounts for the amount paid each time an arc (i, j) is used in a given period p that was not available in the preceding period(s). The link maintenance cost m_{ij} is introduced to account for the cost incurred each time an arc is maintained from one period p to the next. For each period $p \in \mathcal{P}$, we define the load on arc (i, j) during period p as $l^p_{ij} = \sum_{s,t \in V} f^{stp}_{ij}$. A cost function $\phi(\kappa_{ij}, l^p_{ij})$, depending on how close the load l^p_{ij} is to the capacity κ_{ij}, l^p_{ij}, is associated with each arc $(i, j) \in A$. In particular, we assume this function is piecewise linear. For each arc (i, j), defined from node i to node j, is provided a nominal maximum capacity κ_{ij}. However, to strengthen the model, the individual and mutual capacities are set, respectively, as follows: $C^{stp}_{ij} = \min(\kappa_{ij}, D^p(s, t))$ and $C^p_{ij} = \min\left(\kappa_{ij}, \sum_{s,t \in V} D^p(s, t)\right)$.

In order to model the combined optimization problem as a MILP, we define the binary variables y^p_{ij} (resp. z^p_{ij}) to indicate if arc (i, j) is newly opened (resp. maintained) at period p, x^{tp}_{ij} to indicate if node j is the next hop for node i to destination t at period p. We also define the continuous disaggragated flow variables f^{stp}_{ij} representing the amount of flow on arc (i, j) from source s to destination t at period p. The problem is formulated in Fig. 1. We refer to this as the *disaggregated formulation*, which corresponds to a simplification of the formulation developed in [5] as the variables indicating next hop change between periods for node i to destination t are not included.

The problem is quite complex as its dimension is huge; for instance, the number of constraints is $O(|A||V|^2 P)$. Thus, we also provide an easier formulation reducing by a factor $|V|$ the number of constraints by aggregating the flows by destination t [4]. For this purpose, one replaces in the initial formulation the variables f^{stp}_{ij} by f^{tp}_{ij}, which represents the amount of flow on arc (i, j) destined to t at period p. We refer to this formulation as the *aggregated* one. For more details, we refer to [4].

IV. The Algorithm

There are several ways to relax the problem. We approached the Lagrangian relaxation by dualizing the group of flow constraints (2) in such a way that the corresponding Lagrangian subproblem (13) is decomposed by node $i \in V$.

A. Lagrangian Relaxation

Let ν^{stp}_i be the dual variable (multiplier) of one the flow constraints (2) relative to the node i, the pair origin-destination s, t and period p. It follows that the Lagrangian function has the form of:

$$\begin{aligned} f(\nu) = \min \sum_{p \in \mathcal{P}} \sum_{(i,j) \in A} & \left(c_{ij} y^p_{ij} + m_{ij} z^p_{ij} + \phi^p_{ij}\right) \\ & + \sum_{p \in \mathcal{P}} \sum_{s,t \in V} \sum_{(i,j) \in A} \left(\nu^{stp}_i - \nu^{stp}_j\right) f^{stp}_{ij} \\ & + \sum_{p \in \mathcal{P}} \sum_{s,t \in V} D^p(s,t)(\nu^{stp}_t - \nu^{stp}_s) \\ & \text{subject to:} (3) - (12) \end{aligned} \tag{13}$$

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

$$\max\left\{ f(\nu) \,:\, \nu \in \mathbb{R}^{|V|^3 \times P} \right\}. \tag{14}$$

Note that the term in (13) involving the demands $D^p(s, t)$ is linear, said $f^0(\nu) = \sum_{p \in \mathcal{P}} \sum_{s,t \in V} D^p(s, t)\,(\nu^{stp}_t - \nu^{stp}_s)$. Consequently, the Lagrangian subproblem itself can be decomposed in $|V|$ subproblems, said $f^i(\nu)$, $i \in V$., and Lagrangian

$$
\begin{aligned}
\min \quad & \sum_{p\in\mathcal{P}}\sum_{(i,j)\in A}\left(c_{ij}y^p_{ij}+m_{ij}z^p_{ij}+\phi\left(\kappa_{ij},\sum_{s,t\in V}f^{stp}_{ij}\right)\right) && (1)\\
\text{subject to:} \quad & \\
& \sum_{j:(i,j)\in A} f^{stp}_{ij}-\sum_{j:(j,i)\in A} f^{stp}_{ji}=\begin{cases} D^p(s,t) & \text{if } i=s\\ -D^p(s,t) & \text{if } i=t\\ 0 & \text{otherwise}\end{cases} \quad i,s,t\in V,p\in\mathcal{P} && (2)\\
& x^{tp}_{ij}\le y^p_{ij}+z^p_{ij} \quad (i,j)\in A,t\in V,p\in\mathcal{P} && (3)\\
& y^p_{ij}+z^p_{ij}\le 1 \quad (i,j)\in A,t\in V,p\in\mathcal{P} && (4)\\
& z^p_{ij}\le y^{p-1}_{ij}+z^{p-1}_{ij} \quad (i,j)\in A,p\in\mathcal{P},p\ge 2 && (5)\\
& z^1_{ij}=0 \quad (i,j)\in A && (6)\\
& f^{stp}_{ij}\le C^{stp}_{ij}x^{tp}_{ij} \quad (i,j)\in A,s,t\in V,p\in\mathcal{P} && (7)\\
& \sum_{s,t\in V} f^{stp}_{ij}\le C^p_{ij}(y^p_{ij}+z^p_{ij}) \quad (i,j)\in A,p\in\mathcal{P} && (8)\\
& \sum_{j:(i,j)\in A} x^{tp}_{ij}=1 \quad i,t\in V,i\neq t,p\in\mathcal{P} && (9)\\
& x^{tp}_{ij}\in\{0,1\} \quad (i,j)\in A,t\in V,p\in\mathcal{P} && (10)\\
& y^p_{ij},z^p_{ij}\in\{0,1\} \quad (i,j)\in A,p\in\mathcal{P} && (11)\\
& f^{stp}_{ij}\ge 0 \quad (i,j)\in A,s,t\in V,p\in\mathcal{P} && (12)
\end{aligned}
$$

Fig. 1. Mixed integer linear programming formulation of the design and routing problem

function $f(\nu)$ corresponds to the following sum of functions:

$$f(\nu)=f^0(\nu)+\sum_{i\in V} f^i(\nu)\,. \qquad (15)$$

The integrality property doesn't hold for the Lagrangian dual. From a theoretical standpoint this means that the Lagrangian approach can provide a better lower bound than the continuous relaxation could do. In practice, it doesn't happen so often and the Lagrangian relaxation (13) is still difficult because of the integer variables. Furthermore letting the problem (13) be as it is, for large scale instances the method should require an huge amount of time in order to obtain a solution with a sufficient accuracy. Hence, in our numerical experiments we assume that (13) is a linear programming (LP) problem, removing the integrality constraints.

B. Warm start

Initializing the computation is a fundamental issue which requires to find a suitable starting point. The objective is to look for a point $\bar{\nu}$ whose value $f(\bar{\nu})$ is not too far from the optimum of $f(\nu)$. In fact, the methods would require more iterations to reach the optimum if it started far away from the optimum. For both formulations, we solve a restricted problem of (1)-(12) i) involving only the flow variables (f^{stp}_{ij} for the aggregated formulation and f^{tp}_{ij} for the disaggregated formulation), ii) ignoring the design variables, and iii) considering all arcs as open. Therefore, we solve the restricted problem which consists of:

$$\min \sum_{p\in\mathcal{P}}\sum_{(i,j)\in A}\phi\left(\kappa_{ij},\sum_{s,t\in V} f^{stp}_{ij}\right)$$

subject to:

$$
\begin{aligned}
& \sum_{j:(i,j)\in A} f^{stp}_{ij}-\sum_{j:(j,i)\in A} f^{stp}_{ji}=b^{stp}_i \qquad i,s,t\in V,p\in\mathcal{P}\\
& f^{stp}_{ij}\le C^{stp}_{ij} \qquad (i,j)\in A,s,t\in V,p\in\mathcal{P}\\
& \sum_{s,t\in V} f^{stp}_{ij}\le C^p_{ij} \qquad (i,j)\in A,p\in\mathcal{P}\\
& f^{stp}_{ij}\ge 0 \qquad (i,j)\in A,s,t\in V,p\in\mathcal{P}
\end{aligned}
$$

where b^{stp}_i is the deficit of the node i, for the origin-destination pair s,t at period p, written in a compact form. Hopefully, the dual variables of the flow equations represent a "good" starting point for the Lagrangian approach.

For the disaggregated formulation we can do even better. Taking into account that the aggregated formulation is easier than the disaggregated one, after solving the aggregated formulation we could use its optimal solution to initialize the disaggregated formulation. The formula obtained by comparing the objective function of the two models is given by:

$$\nu^{tps}_i=\nu^{tp}_i\,, \quad i,s,t\in V,p\in\mathcal{P}\,. \qquad (16)$$

V. Numerical experiments

A. Implementation and Problem Instances

The Lagrangian relaxation (13) has been solved with `CPLEX 12.5.0`. The Lagrangian dual has been solved by

TABLE I
NUMERICAL RESULTS (BUNDLE)

	Dimension			**Bundle Method**										
Network	$\|V\|$	$\|A\|$	P	iter	$time_a$	$time_d$	time$_f$	time$_{mp}$	time$_{LM}$	time$_{Lm}$	f_i	$\bar{f}$	$\hat{f}$	gap
dfn-bwin	10	90	10	50064	699	10001	5883	1922.19	17.2%	3.7%	8590e+3	8897e+3	1041e+4	5e-6
di-yuan	11	84	5	10268	79	403	338	10.17	11.3%	6.4%	2293e+3	2635e+3	2683e+3	7e-4
pdh	11	68	10	16709	276	1100	878	28.17	11.7%	4.0%	3138e+4	3606e+4	3652e+4	2e-4
abilene	12	30	5	12056	83	132	83	19.64	12.1%	2.7%	1021e+2	1181e+2	1310e+2	1e-6
polska-10p	12	36	10	100373	800	10000	6152	175.12	23.5%	3.3%	3400e+1	4019e+1	4922e+1	1e-4
nobel-us-5p	14	42	5	67149	641	2220	1504	178.96	12.7%	5.0%	3609e+3	4796e+3	4943e+3	5e-4
nobel-us-10p	14	42	10	63571	1000	10001	6842	89.97	10.5%	4.6%	3823e+3	5069e+3	5221e+3	1e-3
atlanta	15	44	5	79477	267	10001	7692	192.72	13.6%	2.5%	2927e+4	4120e+4	4634e+4	2e-3
newyork-5p	16	98	5	3430	101	146	115	10.21	10.7%	2.3%	1950e+3	2550e+3	2550e+3	9e-4
newyork-10p	16	98	10	3594	359	440	367	9.48	12.4%	1.8%	2058e+3	2690e+3	2692e+3	1e-3
geant	22	72	5	41903	381	10001	6065	225.16	16.8%	1.3%	5436e+2	5789e+2	6232e+2	4e-4
geant-1	22	72	5	40539	430	10001	7250	84.53	14.6%	1.2%	1032e+3	1070e+3	1121e+3	4e-4
norway	27	102	10	10532	1000	10020	7099	12.98	7.4%	1.5%	1291e+3	1860e+3	1992e+3	—
cost266	37	114	10	6407	1000	10023	7892	5.59	6.4%	1.0%	2596e+4	3496e+4	3764e+4	—
ta2	65	216	5	2950	1000	10099	9875	1.25	34.3%	0.1%	1933e+5	2752e+5	2868e+5	—
Avg				33935	**541**	**6306**	**4536**	**197.74**	**14.4%**	**2.8%**				**5e-4**

TABLE II
NUMERICAL RESULTS (CPLEX)

	Dimension			**Cplex**			
Network	$\|V\|$	$\|A\|$	P	time$_a$	$\bar{f}$	time$_d$	$\hat{f}$
dfn-bwin	10	90	10	0.51	8897e+3	3.85	1041e+4
di-yuan	11	84	5	5.03	2635e+3	23.93	2685e+3
pdh	11	68	10	9.04	3607e+4	56.09	3653e+4
abilene	12	30	5	0.04	1181e+2	0.57	1310e+2
polska-10p	12	36	10	0.62	4020e+1	8.96	4923e+1
nobel-us-5p	14	42	5	0.82	4805e+3	22.50	4945e+3
nobel-us-10p	14	42	10	2.98	5080e+3	94.82	5227e+3
atlanta	15	44	5	1.33	4122e+4	164.68	4643e+4
newyork-5p	16	98	5	2.07	2551e+3	120.65	2552e+3
newyork-10p	16	98	10	11.40	2693e+3	1170.17	2695e+3
geant	22	72	5	0.52	5793e+2	13.60	6235e+2
geant-1	22	72	5	0.56	1070e+3	13.68	1122e+3
norway	27	102	10	100.87	1866e+3	9974.17	—
cost266	37	114	10	95.82	3524e+4	9976.64	—
ta2	65	216	5	97.47	2846e+5	9984.19	—
Avg				**21.94**	**2850e+4**	**2108.57**	**7560e+3**

means of the *Bundle method*. We have implemented this method within a general-purpose C++ non-smooth code developed by A.Frangioni and already successfully used in solving several other applications [11], [18]. The structure of the code allows applying the *Bundle method* for evaluating different master problems. However, for our experiments, we have used a *quadratic stabilization* and we tested the specialized quadratic solver described in [21].

The algorithm has been compiled with GNU `g++ 4.0.1` (with `-O3` optimization option) and executed on an Opteron 246 (2 GHz) computer with 2 GB of RAM, under Linux Fedora Core 3. The formulations has been evaluated on a set of network topologies extracted from the SNDlib library [22]. 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): *abilene*, *atlanta*, *cost266*, *dfn-bwin*, *di-yuan*, *geant*, *newyork*, *nobel-us*, *norway*, *pdh*, *polska*, and *ta2*. The characteristics of these 12 networks are summarized in Table I. The cardinality of the set V and A are reported, respectively, in columns "$|V|$" and "$|A|$". The number of periods P, listed in the column "$|P|$", is either 5 or 10. The last row "Avg" reports for each column the average of all the instances.

B. Simulation Results and Analysis

The results obtained by means of the *Bundle method* are reported in Table I and those obtained by means of the best technique that `CPLEX` provides for solving the continuous relaxation of (1)-(12), in Table II.

1) Bundle method: A maximum running time threshold of 10000s has been set. The aggregated formulation executes during at most 1000s, while the disaggregated formulation runs the rest of the time, i.e., during 9000s at maximum. To switch to the disaggregated formulation we use the trick presented in Section IV-B. The stopping criterion was set to a *relative* accuracy of `1e-6`. To account for the case where such an accuracy is not reached within the allowed time or the stopping criterion is not sufficiently rigorous, the final relative gap with respect to the "exact" lower bound computed with CPLEX is reported in the column "gap". No value is not reported if CPLEX is not able to provide any solution.

Column "iter" reports the number of iterations (computations of the Lagrangian function and master problem solutions). The total running time (in seconds) is reported in column "time$_a$" for the aggregated formulation and in column "time$_d$" for the disaggregated formulation. Column "time$_f$" and "time$_{mp}$" report respectively, the total running time spent for the computation of the Lagrangian function (f) and for the master problem (mp). The column "time$_{LM}$" and "time$_{Lm}$" report respectively, the maximum and minimum computation time per node in percent. The column f_i provides the starting value for the aggregated formulation. Finally, "$\bar{f}$" and "$\hat{f}$" represent the final value provided by the aggregated and the disaggregated formulation, respectively.

2) CPLEX: The meaning of the columns "$\bar{f}$" and "$\hat{f}$" is the same as for the bundle case. The total running time for the aggregated formulation is reported in column "time$_a$", while the disaggregated one is in column "time$_d$". The CPLEX results are obtained by using the *Dual Simplex algorithm* which also corresponds to the one automatically chosen by this solver method when the problem has linear constraints.

C. Analysis

The gain in execution time obtained with the *Bundle method* is not significant; this observation results from the fact that even the continuous relaxation of the subproblem (i.e., the subproblem without the integrality conditions) is not so easy to achieve.

On the other hand, CPLEX is not able to produce any solution for the disaggregated formulation when applied to the networks *cost266*, *norway* and *ta2* whereas the *Bundle method* reaches a solution whose value is better than the one obtained for the aggregated formulation. For instance, CPLEX provides for *cost266* the value $3524e^{+4}$ while our approach using the hybrid formulation finds $3764e^{+4}$. This result corroborates the initial assumption that for the disaggregated formulation, the Lagrangian relaxation is able to resolve larger instances and provides stronger lower bounds.

In our numerical experiments we have adopted the *disaggregated master problem* which is the former master problem designed for the bundle methods [12]. It has been showed in [18] that the *disaggregated master problem* should work better whenever the Lagrangian dual can be formulated as a sum of functions. As it is the case in this application, $f(\nu)$ is the sum of $|V|$ functions. Unfortunately, the number $|V|$ for all of the above instances is relatively small. Consequently, the disaggregated master is rich in information which in turn should improve the global rate of convergence. However, because of the small size of the set of nodes, the information accumulated in the disaggregated master problem 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 justifies the fact that we have used a standard quadratic master problem.

VI. Conclusion

We presented a Lagrangian relaxation approach to solve a multi-commodity capacitated fixed charge network design problem with variable traffic demands over multiple time periods and including routing decisions. This practical problem is computationally challenging since it aims at considering several tasks at once. In fact, the MIP problem is even difficult testing reasonable instances as already reported in [5].

In this paper, we showed how its continuous relaxation remains a difficult problem. Then, we applied some decomposition techniques to construct the Lagrangian dual. The latter has been solved by means of a bundle method. It is well known that the Lagrangian dual is equivalent to the convexified relaxation. Hopefully, the lower bound is better than the one provided by the LP relaxation but we don't have any information about the integer solution. This fact motivates the need to construct a feasible primal integer solution. The existence of some good integer solutions starting from the solution of (13) may be a direction to explore.

Acknowledgment

The work of B. Fortz is supported by the Interuniversity Attraction Poles Program of the Belgian Science Policy Office.

References

[1] T. Crainic, A. Frangioni, and B. Gendron, "Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems," *Discrete Appl. Math.*, vol. 112, pp. 73–99, 2001.

[2] A. M. Costa, J.-F. Cordeau, and B. Gendron, "Benders, metric and cutset inequalities for multicommodity capacitated network design," *Comp. Opt. and Appl.*, vol. 42, no. 3, pp. 371–392, 2009.

[3] A. Frangioni and B. Gendron, "0-1 reformulations of the multicommodity capacitated network design problem," *Discrete Applied Mathematics*, vol. 157, no. 6, pp. 1229–1241, 2009.

[4] B. Fortz and D. Papadimitriou, "Methods for time-dependent combined network design and routing optimization," Université Libre de Bruxelles, Tech. Rep., 2014, http://hdl.handle.net/2013/ULB-DIPOT:oai:dipot.ulb.ac.be:2013/163697.

[5] 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.

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

[7] A. Frangioni and G. Gallo, "A bundle type dual-ascent approach to linear multicommodity min cost flow problems," *INFORMS Journal On Computing*, vol. 11, no. 4, pp. 370 – 393, 1999.

[8] J. Castro, "A specialized interior-point algorithm for multicommodity network flows," *SIAM Journal on Optimization*, vol. 10, no. 852- 877, 2000.

[9] P. Cappanera and A. Frangioni, "Symmetric and asymmetric parallelization of a cost-decomposition algorithm for multi-commodity flow problems," *INFORMS Journal On Computing*, vol. 15, no. 4, pp. 369–384, 2003.

[10] 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]*, 2001, pp. 112–156.
[11] T. Crainic, A. Frangioni, and B. Gendron, "Bundle-based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems," *Discrete Applied Mathematics*, vol. 112, pp. 73–99, 2001.
[12] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* II—*Advanced Theory and Bundle Methods*, ser. Grundlehren Math. Wiss. New York: Springer-Verlag, 1993, vol. 306.
[13] G. d'Antonio and A. Frangioni, "Convergence analysis of deflected conditional approximate subgradient methods," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 357–386, 2009.
[14] C. Barnhart, C. Hane, and P. Vance, "Using branch-and-price-and-cut to solve origin-ddestination integer multicommodity flow problems," *Operations Research*, vol. 48, no. 2, pp. 318–326, 2000.
[15] C.Lemarechal, "An algorithm for minimizing convex functions," in *Proceedings of IFIP'74 Congress*. North Holland, 1974, pp. 552–556.
[16] N. Ladson, *Minimization Methods for Non-differentiable Functions*, ser. Springer Series in Computational Mathematics. New York: Springer Berlin Heidelberg, 1985.
[17] A. Frangioni, "Generalized bundle methods," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 117–156, 2002.
[18] A. Frangioni and E. Gorgone, "Generalized Bundle Methods for Sum-Functions with "Easy" Components: Applications to Multicommodity Network Design"," *Mathematical Programming*, vol. 145, pp. 133–161, 2014.
[19] Y.Nesterov, "Primal-dual subgradient methods for convex problems," *Math. Program., Ser. B*, vol. 120, pp. 221–259, 2009.
[20] B. Fortz and M. Thorup, "Increasing internet capacity using local search," *Computational Optimization and Applications*, vol. 29, no. 1, pp. 13–48, 2004.
[21] A. Frangioni, "Solving Semidefinite Quadratic Problems Within Nonsmooth Optimization Algorithms," *Computers & Operations Research*, vol. 21, pp. 1099–1118, 1996.
[22] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.