Shared Protection network design: valid inequalities and a decomposition approach

Pietro Belotti and Federico Malucelli
DEI – Politecnico di Milano, Italy

Abstract— **Shared Protection is a technique that can reduce the cost of backup resources in survivable networks. To fully take advantage of the potential savings, the Shared Protection has to be suitably taken into account at design level, providing effective optimization models and solution methods. We present an optimization model for this problem together with two families of strengthening valid inequalities.**

Since the resulting problem is computationally burdensome, we propose decomposition approaches based on Lagrangian Relaxation. We compare our approach with a commercial Mixed Integer Programming solver on a set of real-world network instances, and report the difference between the cost of a network considering the Shared Protection or a traditional protection methods.

Index Terms— **Network Planning, Survivability, Integer Programming, Cutting planes.**

I. Introduction

Optimization problems on Telecommunication networks have received much attention recently, due to the crisis in this business field that increased the need for efficiency.

One most desired features is survivability, i.e., the network must be redundant to equipment failure. Among the many techniques that allow network resiliency for several failure scenarios, we focus on *path protection*: every pair of nodes (s, t) communicates through a *working* path p from s to t, and a *backup* path p' that has no edge in common with p; p' is inactive in nominal conditions but actually carries the data flow when an edge of p goes out of order. Adopting this technique means assuming that only one link at a time may fail in order for the single backup path to be a sufficient replacement; the possibility of re-establishing a link soon after failure makes this *single link failure* hypothesis acceptable in most cases.

It is worth emphasizing that backup paths are only used upon failure of one link, but they in any case occupy part of the network capacity. In a *Dedicated Protection* network, each backup path has exclusive access to the assigned resources, while, as we will see in section II, in *Shared Protection* (SP) the resources are common to all backup paths provided they cannot be used simultaneously in the event of a failure on two intersecting working paths.

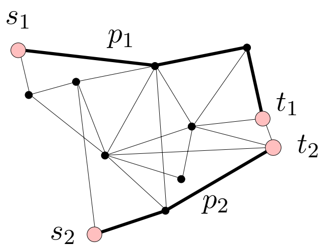
This problem falls in the class of multicommodity network design which has received much attention over the years. Several surveys are available in the literature, e.g. the PhD dissertation by Yuan [18], with a bibliography on many design and routing problems. Among the numerous attempts to tackle survivable network design are many algorithmic approaches; we cite the Analytic Center Cutting Plane method proposed by Gendron et al. [7], who compare the performance of several dual methods such as subgradient and bundle. Cutting planes have been applied to survivable network design in several papers [3], [12]. Stoer and Dahl [17] introduce classes of valid inequalities for a multi-facility problem, while for the same problem Kennington et al. [9] propose some valid inequalities and a decomposition approach.

The idea we express in this work is that Shared Protection can be fully exploited when embedded in the earliest steps of a design process. In Section III we give an optimization model and discuss some related computational issues. In Section IV we introduce a new family of valid inequalities for the model and outline a separation method based on max-flow algorithms. In Section V we decompose the problem through Lagrangian Relaxation and outline a bundle method used to solve the Lagrangian dual problem. Finally, we give in Section VI some results justifying the adoption of this technique as a main component of network optimization, and attempt some conclusion in Section VII.

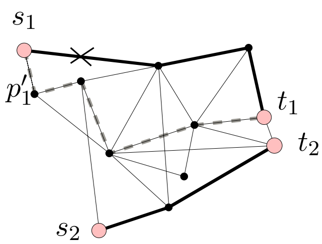
II. Sharing protection resources

We are given a topology $G = (V, E)$ and a set Q of traffic demands (we also use the term *commodity*) described by triplets (s_q, t_q, d_q), $q \in Q$, standing for the origin, destination, and traffic volume requested. Consider two traffic demands (s_1, t_1) and (s_2, t_2), as in Fig. 1a. In the absence of failure, two paths p_1 and p_2 are used for communication; note that they share no edge. If a failure occurs on an edge in p_1, network management routines keep the connection active by using a backup path p_1', as highlighted in Fig. 1b. Similarly, if a failure occurs on p_2 a backup path p_2' replaces the nominal one (see Fig. 1c).

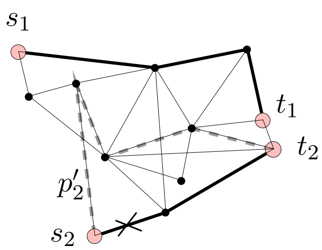
As a first observation, we notice that every backup path brings redundancy to the network but remarkably increases its cost up to more than its double, since to the capacity for working paths we must add a *backup capacity* which has an almost equal cost but is only used upon failure. Therefore, backup capacity has a high cost with respect to its actual utilization.



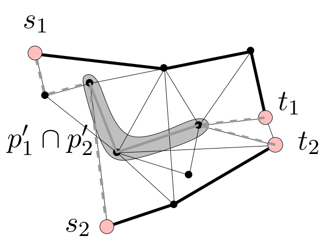
(a) Nominal condition



(b) Failure on path p_1



(c) Failure on path p_2



(d) Shareable resources

Fig. 1. Shared backup resources in a network with path protection.

Consider again Fig. 1: if the two nominal paths p_1 and p_2 have no edge in common and under the single failure hypothesis, utilization of the two backup paths p_1' and p_2' is mutually exclusive. Hence, these paths can be assigned the same resources on all the edges they share, thus allowing to at the most cut in half the capacity installed to these two node pairs on $p_1' \cap p_2'$, which is $\max(d_{q_1}, d_{q_2})$ rather than $d_{q_1} + d_{q_2}$ (see Fig. 1d). For instance, in Time, Frequency, and Wavelength Division Multiplexing networks these two backup paths would be assigned the same time slot, frequency band, and wavelength or waveband, respectively.

This property may apply to more than two traffic demands: in general, if a set of working paths $\{p_1, p_2, ..., p_m\}$ for demands $S = \{q_1, q_2, ..., q_m\} \subseteq Q$ are pairwise link disjoint and there is a non-empty intersection of their backup paths $P = \bigcap_{i=1}^{m} p_i'$, the resources on P may be shared among all of them. This means that on P a capacity equal to $\max_{q \in S} d_q$ instead of $\sum_{q \in S} d_q$ can be installed for routing these demands – resource savings thus increase with $|P|$. In other words, SP allows to share backup resources and therefore leads to a cheaper network when compared to Dedicated Protection, which gives a reserved capacity to every backup path.

Shared protection has been dealt with in Lisser et al. [11], where an Analytic Center Cutting Plane method (ACCPM) is presented for a multicommodity survivable network design problem; several failure scenarios which make the problem intractable are decomposed through Lagrangian Relaxation. The design problem is first solved on the whole topology $G = (V, E)$, obtaining the working capacity $c_e(G)$ to be installed on each edge $e \in E$. Then, for each edge $e \in E$, the same problem is solved on the $|E|$ reduced graphs $G_{e'}$ obtained by eliminating e' from E, thus finding a different value $c_e(G_{e'})$ of the capacity: this is the capacity needed to reroute all demands in the event that e' breaks. The capacity for edge e is given by the maximum capacity over the solved problems, $\max\{c_e(G), \max_{e' \in E} c_e(G_{e'})\}$. In the event of a failure, a re-routing might also be needed for some traffic demands whose working path does not contain any failed link.

Many multi-step heuristics have been proposed: Yuan and Jue propose a shortest-path based heuristic for a routing problem and test it over an NSFNet-like architecture; a comparison with dedicated protection shows a remarkable cost improvement brought by shared protection. Mauz [13] describes an algorithm for allocating shared capacity and solves a routing instance on the PanEuropean Cost 239 architecture. Datta et al. [4] study a pool-based channel reservation scheme, and Sengupta and Ramamurthy [16] present a distributed algorithm for path shareability. In these cases, the protection paths are created after all working paths are given, thus not guaranteeing optimal shared protection design and routing.

Some preliminary tests we have made suggested that embedding shared protection into an optimization model may lead to very efficient use of the backup resource. This encouraged us to study the problem of Shared Protection Network Design Problem (SPNDP) that we model in the next section.

III. An Integer Linear Programming model

We are given a topology $G = (V, E)$ and a set Q of traffic demands (we also use the term *commodity*) described by triplets (s_q, t_q, d_q), $q \in Q$, standing for the origin, destination, and traffic volume requested. The solution is a subset of E with associated capacity such that a routing of all demands is performed and the cost of the topology is minimum. We use notation e and $\{i, j\}$ when referring to edges, i and j being the endnodes of e. The set of edges E induces a set A of oriented arcs, defined as ordered node pairs: $A = \{(i, j) : \{i, j\} \in E\}$. An arc oriented from i to j is denoted by (i, j). Each edge e is associated a cable cost $c_d(e)$ and capacity cost $c_c(e)$.

We use two classes of variables: y_e is integer and defines the capacity of edge e, and x_e is binary and equal to 1 if $y_e > 0$. We assume that a capacity y_e allocated on edge $e = \{i, j\}$ is sufficient to carry a flow not greater than y_e from i to j and another flow, at most equal to y_e, in the opposite direction, i.e. the same capacity is installed on both directions. A constant λ represents the amount of traffic carried by a single unit of capacity. An upper bound M_e for the traffic across edge $e = \{i, j\}$ is given by the maximum between two terms: the total demand not ending in i and the total demand not ending in j, hence $M_{\{i,j\}} = \max\left(\sum_{q\in Q:t_q\neq i} d_q, \sum_{q\in Q:t_q\neq j} d_q\right)$.

We require that flow be unsplit by using binary variables φ^q_{ij} to describe the working flow on each oriented arc (i, j) for every demand q. The binary variable ψ^q_{ij} identifies instead the backup data flow on arc (i, j) for demand q. These classes of variables are subject to flow conservation constraints, imposing a unitary unsplit flow to follow a path from s_q to t_q. Moreover, link disjointness must hold between working and backup flows.

In dedicated protection, the capacity to be allocated on a certain edge $\{i, j\}$ is proportional to the overall flow from i to j, i.e. $\sum_{q\in Q} d_q(\varphi^q_{ij} + \psi^q_{ij})$, and can be discerned as working and backup capacity. When adopting Shared Protection, the working capacity $\sum_{q\in Q} d_q\varphi^q_{ij}$ does not change but the backup capacity, which we denote with ξ_{ij}, may decrease, as previously pointed out. Let us consider a failure on an edge $\{m, n\} \neq \{i, j\}$. The capacity required by commodity q on (i, j) due to this failure is equal to d_q if and only if the working path of q contains $\{m, n\}$ and its backup path contains (i, j), that is, formally:

$$(\varphi^q_{mn} = 1 \vee \varphi^q_{nm} = 1) \wedge \psi^q_{ij} = 1 \tag{1}$$

Notice that the two terms in parentheses are mutually exclusive. To deal with this expression, we introduce continuous variables $\vartheta^q_{(i,j),\{m,n\}}$ for each $q \in Q$, $\{m, n\} \in E$ and $(i, j) \in A$ such that $\{i, j\} \neq \{m, n\}$. $\vartheta^q_{(i,j),\{m,n\}}$ is 1 if and only if commodity q needs to route a backup flow on arc (i, j) in case of failure on edge $\{m, n\}$, therefore (1) becomes:

$$\vartheta^q_{(i,j),\{m,n\}} \geq \psi^q_{ij} + \varphi^q_{mn} + \varphi^q_{nm} - 1$$

A value for ξ_{ij} can be found by considering every condition of failure: the backup flow due to failure of edge $\{m, n\}$ is $\sum_{q\in Q} d_q\vartheta^q_{(i,j),\{m,n\}}$, and the capacity needed to cope with all possible failure situations is the maximum over all edges (i.e., failures):

$$\xi_{ij} = \max_{\{m,n\}\neq\{i,j\}} \sum_{q\in Q} d_q\vartheta^q_{(i,j),\{m,n\}} \tag{2}$$

Therefore, the capacity $y_{\{i,j\}}$ must suffice for the total traffic under all failure situations:

$$\sum_{q\in Q} d_q \left(\varphi^q_{ij} + \vartheta^q_{(i,j),\{m,n\}}\right) \leq \lambda y_{\{i,j\}} \; \forall\{m, n\} \neq \{i, j\}$$

We are now able to describe a model for SPNDP: the sum of deployment and capacity costs

$$z = \sum_{e\in E} (c_d(e)x_e + c_c(e)y_e)$$

is to be minimized subject to the constraints:

$$\sum_{j\in N(i)}(\varphi^q_{ij} - \varphi^q_{ji}) = b^q_i \qquad \forall q \in Q, \; \forall i \in V \tag{3}$$

$$\sum_{j\in N(i)}(\psi^q_{ij} - \psi^q_{ji}) = b^q_i \qquad \forall q \in Q, \; \forall i \in V \tag{4}$$

$$y_e \leq M_e x_e \qquad \forall e \in E \tag{5}$$

$$\varphi^q_{ij} + \varphi^q_{ji} + \psi^q_{ij} + \psi^q_{ji} \leq x_{\{i,j\}} \qquad \forall (i, j) \in A, \forall q \in Q \tag{6}$$

$$\psi^q_{ij} + \varphi^q_{mn} + \varphi^q_{nm} - 1 \leq \vartheta^q_{(i,j),\{m,n\}} \qquad \begin{array}{l}\forall (i, j) \in A, \forall q \in Q\\ \forall\{m, n\} \neq \{i, j\}\end{array} \tag{7}$$

$$\sum_{q\in Q} d_q \left(\varphi^q_{ij} + \vartheta^q_{(i,j),\{m,n\}}\right) \leq \lambda y_{\{i,j\}} \qquad \begin{array}{l}\forall (i, j) \in A\\ \forall\{m, n\} \neq \{i, j\}\end{array} \tag{8}$$

$$\begin{array}{rl} \varphi^q_{ij}, \psi^q_{ij} & \in \{0, 1\} \\ \vartheta^q_{(i,j),\{m,n\}} & \in [0, 1] \\ x_{\{i,j\}} & \in \{0, 1\} \\ y_{\{i,j\}} & \in \mathbb{Z}^+ \end{array} \qquad \begin{array}{l} \forall (i, j) \in A \\ \forall q \in Q \\ \forall\{m, n\} \neq \{i, j\}, \end{array} \tag{9}$$

where $b^q_i = \begin{cases} 1 & \text{if } i = s_q \\ -1 & \text{if } i = t_q \\ 0 & \text{otherwise.} \end{cases}$

We minimize z enforcing flow conservation for variables φ and ψ in constraints (3) and (4). Constraint (5) imposes that variable y_e be positive only when the corresponding variable x_e is 1. As we use a different class of variables for the backup flow, we need to ensure link disjointness between working and backup flows on all edges $\{i, j\} \in E$ and for all $q \in Q$:

$$\varphi^q_{ij} + \psi^q_{ij} \leq 1 \qquad \varphi^q_{ij} + \psi^q_{ji} \leq 1$$

$$\varphi^q_{ji} + \psi^q_{ij} \leq 1 \qquad \varphi^q_{ji} + \psi^q_{ji} \leq 1$$

This is also true for oppositely oriented flows, as the failure of edge $\{i, j\}$ interrupts the data flow in both directions. Intuition suggests us (but this can also be proven) that in an efficient solution the flow of a demand does not cross an edge in both directions, hence

$\varphi_{ij}^q + \varphi_{ji}^q \leq 1$ and $\psi_{ij}^q + \psi_{ji}^q \leq 1$. These six inequalities are clearly dominated by the clique inequality

$$\varphi_{ij}^q + \varphi_{ji}^q + \psi_{ij}^q + \psi_{ji}^q \leq 1 \tag{10}$$

Therefore, we obtain (6) by recalling that an edge e has a positive flow only if the related variable x_e is set to 1. Constraints (7) and (8) define shared protection, and (9) specifies the type of all variables involved.

The problem just described is an integer capacitated multicommodity network flow problem, where survivability constraints bound the network structure to be two-connected. The NP-hardness of SPNDP can be hence proved by reduction from the minimum-weight two-connected spanning network problem [14].

Since in a solution to SPNDP the working and backup paths are fixed and do not depend on the failed link, the method proposed by Lisser et al. [11] gives a lower bound under the assumption that the routing of any demand may change even if the working path is not affected by the failure.

The optimization model presented above has $O(|E|^2|Q|)$ variables and constraints, or $O(|V|^6)$ if we assume the demand and network topology to be sufficiently dense. The impact of the model dimension on the solution times has been observed in some preliminary tests performed on small instances of SPNDP, where even the LP relaxation of the model required a large computational effort; for instance, on a 12-node, 37-link network, the best LP solver showed to be CPLEX 7.0 Barrier optimizer, which has taken about 1h:20' on a PC with a 600 MHz AMD processor.

The difficulty of SPNDP has also been pointed out by Fischetti and Lodi [6], who have tackled an instance of 12 nodes with a novel approach called *local branching*. The tough part of SPNDP seems to be SP constraints (7) and (8), as many as $O(|E|^2|Q|)$ and using all of the $O(|E|^2|Q|)$ continuous SP variables ϑ. However, very few of these variables are non-null, hence in an optimal solution few SP constraints are active.

IV. A NEW CLASS OF VALID INEQUALITIES

Several polyhedral approaches have been proposed for network design problems [1], [3], [12], [17]; some of them exploit some properties of multicommodity flow, such as the minimum cut, to introduce valid inequalities on the minimum capacity required on a subset of edges. After introducing some additional notation, we present two families of valid inequalities adopting this viewpoint.

Given a subset W of V, we call *cut* – and denote it with $\delta(W)$ – the set of edges with only one endpoint in W, i.e. $\delta(W) = \{\{i,j\} \in E : i \in W, j \notin W\}$. In the following discussion, we use $x(\delta(W)) = \sum_{e\in\delta(W)} x_e$ and $y(\delta(W)) = \sum_{e\in\delta(W)} y_e$, and for each $Q' \subseteq Q$, $d(Q') = \sum_{q\in Q'} d_q$. Moreover, we define the subset of demands with source in W and destination in $V \setminus W$ by $Q^+(W)$; similarly, the subset of demands originating outside W and with destination in W is denoted as $Q^-(W)$. Therefore, the capacity across a cut $\delta(W)$ must be at least the maximum of the traffic in the two directions: $d(\delta(W)) = \max(d(Q^+(W)), d(Q^-(W)))$.

A topological property needed in path protected networks is *2-connectivity*, i.e., if a traffic demand exists between two nodes s and t there must be at least two link-disjoint paths between them. This can be expressed as an inequality over variables x, that define the network structure. Namely, there must be at least two installed edges on a cut $\delta(W)$ dividing s from t, or:

$$x(\delta(W)) \geq 2 \tag{11}$$

This valid inequality has been already proposed for different survivable network design problems [17]. The separation procedure takes as input the value x^* of the linear relaxation of an instance of SPNDP and looks for the cut $\delta(W)$ which minimizes $x(\delta(W))$. This is clearly given by the minimum cut over the graph G with capacities given by x^*, provided that $\delta(W)$ divides at least one demand node pair. We employ the maximum flow algorithm proposed by Gusfield [8] to obtain a minimum cut x^*, hence solving the separation problem in polynomial time.

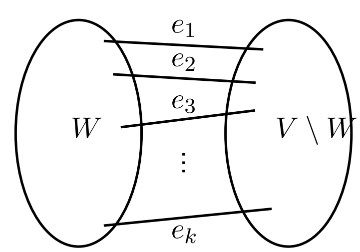


Fig. 2. Computing the capacity needed across a cut $\delta(W)$.

The inequality we introduce now is specific to this problem and has not been studied so far. Consider Fig. 2, where a cut $\delta(W)$ contains k edges $e_1, e_2, \ldots, e_k$. The capacity needed to carry working flow across the cut is at least $\lceil d(\delta(W))/\lambda \rceil$. In order to compute a lower bound on the backup capacity across $\delta(W)$, we consider the case where edges $e_1, e_2, \ldots e_{k-1}$ bear the working traffic between the two shores. Link e_k would then be used for shared backup traffic, and the required backup capacity would be no less than the maximum traffic borne on edges $e_1, e_2, \ldots e_{k-1}$; this, in turn, can be approximated from below by $d(\delta(W))/(|\delta(W)|-1)$. A better estimate is given considering the actual number of edges installed on the cut, i.e. $x(\delta(W))$. Hence, if the working traffic flows through all edges in the cut except one and the backup traffic occupies the remaining edge, the minimum capacity is given by (we use $x(\delta)$ and $d(\delta)$ instead of $x(\delta(W))$ and $d(\delta(W))$ for better readability)

$$\frac{1}{\lambda}\left(d(\delta)+\frac{d(\delta)}{x(\delta)-1}\right)=\frac{d(\delta)}{\lambda}\cdot\frac{x(\delta)}{x(\delta)-1} \tag{12}$$

If the working traffic is carried through fewer edges of the cuts, the estimated backup capacity increases as $d(\delta)$ is divided by a smaller quantity. Finally, if the working traffic flows through all edges in the cut, the backup capacity must be installed on at least two edges e_h, e_j to ensure connectivity in the event that either of them fails; however, the capacity on each edge is bounded from below by $d(\delta)/x(\delta)$, thus giving a lower bound of $\frac{d(\delta)}{\lambda}(1+2/x(\delta)$, no smaller than the lower bound (12) if $x(\delta)\geq 2$. Therefore, (12) represents a lower bound for the capacity to be installed on δ and we may write the valid inequality

$$y(\delta)\geq\frac{d(\delta)}{\lambda}\cdot\frac{x(\delta)}{x(\delta)-1} \tag{13}$$

As the right hand side of this inequality is non-linear, we look for a linear approximation in the neighbourhood of a solution $x^*(\delta)$ to a linear relaxation of the problem. First, notice that the function $f(x_e, e\in\delta)=\frac{x(\delta)}{x(\delta)-1}$ is non-concave for all $x_e\in\mathbb{R}^+$ in that $\frac{\partial^2 f}{\partial x_{e_1}\partial x_{e_2}}=2/(x(\delta)-1)^3\ \forall e_1,e_2\in E$ and hence its Hessian is positive semidefinite. Thus we can approximate f from below with the linear function:

$$f(x_e,e\in\delta)\approx f(x_e^*,e\in\delta)+\sum_{e\in\delta}\frac{\partial f}{\partial x_e}(x_e-x_e^*)=$$
$$=\frac{x^*(\delta)}{x^*(\delta)-1}-\frac{x(\delta)-x^*(\delta)}{(x^*(\delta)-1)^2}=$$
$$\frac{(x^*(\delta))^2}{(x^*(\delta)-1)^2}-\frac{x(\delta)}{(x^*(\delta)-1)^2}$$

and write a lower bound to the capacity across the cut:

$$y(\delta)\geq\frac{d(\delta)}{\lambda}\left(\frac{(x^*(\delta))^2}{(x^*(\delta)-1)^2}-\frac{x(\delta)}{(x^*(\delta)-1)^2}\right)$$

Rounding the coefficients through the Chvátal-Gomory procedure yields the *shared cut-set inequality*:

$$y(\delta)\geq\left\lceil\frac{d(\delta)(x^*(\delta))^2}{\lambda(x^*(\delta)-1)^2}\right\rceil-\left\lceil\frac{d(\delta)}{\lambda(x^*(\delta)-1)^2}\right\rceil x(\delta) \tag{14}$$

As the right hand side in (14) is a linear approximation of that in (13), the set of inequalities (14) for a given cut $\delta(W)$ define a polyhedron containing the set of points satisfying (13). As a separation method, we take a solution (x^*,y^*) to a linear relaxation of the problem and look for a cut $\delta(W)$ maximizing the violation; to this purpose we solve a maximum-flow problem on a graph G' whose capacity on edge e is equal to $y_e x_e$, so as to privilege those cuts with low values of x_e.

As we have no guarantees on the value of $x(\delta(W))$, once we find this cut we insert the inequality (14) only if $x^*(\delta(W))\geq 2$, i.e. if (11) holds; otherwise, we insert (11) for $\delta(W)$ and the implied valid inequality (14)

$$y(\delta(W))\geq\lceil 4d(\delta(W))/\lambda\rceil-\lceil d(\delta(W))/\lambda\rceil\, x(\delta(W))$$

These inequalities do not include any of the commodity variables φ, ψ, ϑ, and can be used in a decomposition paradigm as the one we devise in the next section.

V. Bounding SPNDP with Lagrangian Relaxation

We have applied Lagrangian Relaxation (the reader may refer to [10] for a complete overview) to bound SPNDP from below. If the right-hand side of (6) changes to 1, the model is easily decomposed into $|Q|+1$ subproblems. Constraints (8) are relaxed and put into the objective function and the Lagrangian function is:

$$\check{z}(\pi)=\min\Bigg\{\sum_{\{i,j\}\in E}\left(c_d(\{i,j\})x_{\{i,j\}}+c_c(\{i,j\})y_{\{i,j\}}\right)+$$
$$+\sum_{(i,j)\in A}\ \sum_{\{m,n\}\neq\{i,j\}}\pi_{(i,j)}^{\{m,n\}}\left(\sum_{q\in Q}d_q\left(\varphi_{ij}^q+\vartheta_{(i,j),\{m,n\}}^q\right)+\right.$$
$$\left.-\lambda y_{\{i,j\}}\right):(3,4,5,10,7)\Bigg\}=$$
$$=\min\Bigg\{\sum_{\{i,j\}\in E}\Bigg(c_d(\{i,j\})x_{\{i,j\}}+$$
$$+\left(c_c(\{i,j\})-\lambda\textstyle\sum_{\{m,n\}\neq\{i,j\}}\left(\pi_{(i,j)}^{\{m,n\}}+\pi_{(j,i)}^{\{m,n\}}\right)\right)y_{\{i,j\}}\Bigg)+$$
$$+\sum_{(i,j)\in A}\ \sum_{\{m,n\}\neq\{i,j\}}\pi_{(i,j)}^{\{m,n\}}\sum_{q\in Q}d_q\left(\varphi_{ij}^q+\vartheta_{(i,j),\{m,n\}}^q\right):$$
$$(3,4,5,10,7)\Bigg\}=$$
$$=\min\left\{\mathcal{L}_0(\mathbf{x},\mathbf{y},\pi)+\sum_{q\in Q}d_q\mathcal{L}_q(\varphi,\vartheta,\pi):(3,4,5,10,7)\right\}.$$

We notice that functions $\mathcal{L}_0$ and all $\mathcal{L}_q$'s depend on pairwise disjoint sets of variables, leading to a decomposition of the problem into a *topology subproblem* P_0, with objective function $\check{z}_0(\pi)=\mathcal{L}_0(\pi)$; and one *flow subproblem* P_q for each commodity q, with objective $\check{z}_q(\pi)=\mathcal{L}_q(\pi)$. The single topology subproblem depends on variables $x_{\{i,j\}}$, $y_{\{i,j\}}$ and on non-negative multipliers π, while each flow subproblem P_q has a single-commodity set of variables (φ,ψ,ϑ) that we can denote without index q. Recalling that each flow subproblem is uniquely identified by the node pair (s_q, t_q), we can reformulate the Lagrangian Relaxation of our problem as

$$\begin{aligned}\check{z}(\pi) = &\ \min\{\mathcal{L}_0(\mathbf{x}, \mathbf{y}, \pi) : (5)\} \\ + &\ \textstyle\sum_{q \in Q} d_q \min\{\mathcal{L}_q(\varphi, \vartheta, \pi) : (3, 4, 10, 7)\}\end{aligned}$$

Therefore, a lower bound to the solution of an instance of SPNDP is given by $\max_{\pi \geq 0} \check{z}(\pi)$.

A. The topology subproblem

Let us consider the single topology subproblem

$$\begin{aligned}\min\ \check{z}_0(\pi) = \Big\{ &\textstyle\sum_{\{i,j\} \in E} \big(c_d(\{i,j\}) x_{\{i,j\}} + \\ &+ \bar{c}_c(\{i,j\}) y_{\{i,j\}}\big) : y_e \leq M_e x_e \ \ \forall e \in E \Big\}\end{aligned} \tag{15}$$

As flow variables have been dropped with the relaxation of constraint (8), we are left with a trivial optimization problem only constrained by (5). Variables $y_{\{i,j\}}$ appear in the objective function with coefficient

$$\bar{c}_c(\{i,j\}) = c_c(\{i,j\}) - \lambda \sum_{\{m,n\} \neq \{i,j\}} \left(\pi^{\{m,n\}}_{(i,j)} + \pi^{\{m,n\}}_{(j,i)} \right)$$

hence for all edges $e = \{i,j\}$, in optimal solution one has $x_e = 1$ and $y_e = M_e$ if $c_d(e) + M_e \bar{c}(e) < 0$, and $x_e = y_e = 0$ otherwise. The valid inequalities introduced in the previous section are therefore of great help in improving the lower bound.

B. The flow subproblems

The Lagrangian Relaxation breaks the main problem into $|Q|$ single-commodity minimization subproblems with objective function (we drop indices q here)

$$\check{z}_q(\pi) = \sum_{(i,j) \in A} \sum_{\{m,n\} \neq \{i,j\}} \pi^{\{m,n\}}_{(i,j)} \left(\varphi_{ij} + \vartheta_{(i,j),\{m,n\}} \right)$$

to be minimized subject to

$$\textstyle\sum_{j \in N(i)} (\varphi_{ij} - \varphi_{ji}) \ = b^q_i \qquad \forall i \in V \tag{16}$$

$$\textstyle\sum_{j \in N(i)} (\psi_{ij} - \psi_{ji}) = b^q_i \qquad \forall i \in V \tag{17}$$

$$\varphi_{ij} + \varphi_{ji} + \psi_{ij} + \psi_{ji} \ \leq\ 1 \qquad \forall (i,j) \in A \tag{18}$$

$$\psi_{ij} + \varphi_{mn} + \varphi_{nm} - 1 \leq \vartheta_{(i,j),\{m,n\}} \quad \forall (i,j) \in A, \forall \{m,n\} \neq \{i,j\} \tag{19}$$

$$\varphi_{ij}, \psi_{ij}, \vartheta_{(i,j),\{m,n\}} \in [0,1] \quad \forall (i,j) \in A,\ \forall \{m,n\} \neq \{i,j\} \tag{20}$$

After relaxing integrality of variables φ and ψ, these subproblems are easily solved by an LP solver, even for large instances of SPNDP.

C. A bundle method for SPNDP

The decomposition approach we have exposed allows to obtain, given a set of non-negative multipliers π, a value of $\check{z}(\pi)$ in a reasonable time, and this gives us the possibility to use, for example, a subgradient technique to obtain a lower bound of SPNDP. Many approaches are available for maximizing $\check{z}(\pi)$, viz. subgradient, cutting planes, ACCPM; we have adopted a bundle method (Frangioni 1997) suited to this particular problem. We use a C++ library and link it with the implementation of an *oracle* method for computing the value of $\mathcal{L}_0(\pi) + \sum_{q \in Q} d_q \mathcal{L}_q(\pi)$ given a set of multiplicators π.

As the bundle optimization begins, the topology subproblem contains only constraints (5); for each call to the oracle, the quantity $\mathcal{L}(\pi) = \mathcal{L}_0(\pi) + \sum_{q \in Q} d_q \mathcal{L}_q(\pi)$ is calculated by solving the topology subproblem and all flow subproblems. After solving the former, we look for inequalities (11) and (14) violated by the current solution and insert them. The next call to the oracle has a strengthened subproblem and can therefore obtain a better lower bound of SPNDP.

VI. Computational results

Optimization methods often find their toughest test bed on real-world problems; we have chosen to apply our algorithms to some real topologies and traffic demands, so as to obtain a complete overview of their efficiency. A computer equipped with a single Athlon 600 Mhz processor and 768 MB of memory has been used for all tests. The operating system is Linux, kernel 2.4.7-10; we have written our code in ANSI C and compiled it with Gnu `gcc` 2.95.3 with compile option `-O`. We have used Cplex 7.0 callable library; the C implementation of Hao-Orlin and Gusfield methods used in the separation procedures described in section IV is available on line.

All the instances solved in our computational tests are real world networks found in the literature (e.g. Barnhart, Hane, and Vance (2000), Mauz (2001)) and can be downloaded from the ftp site of the Department of Electronics and Information of Politecnico di Milano:

ftp://ftp.elet.polimi.it/users/Pietro.Belotti/mcf/data

All files are in a DIMACS-like format (see reference), containing for each network problem the topology and the traffic demands. We report here the results obtained by applying the bundle algorithm described in section V. For each instance we have let the bundle perform 100 iterations and posed a time limit of 3600 s. We have not attempted to solve the whole ILP model; as we know, in most cases it requires a huge computational effort and for all but two instances it has not been possible to compute even the linear relaxation of the model.

The results are shown in table I; for each instance, we report its size and three lower bounds, obtained respectively through CPLEX MIP solver ($z(mip)$), our method without cut-set inequalities ($z(lr)$) and our method with separation of cut-set inequalities ($z(lr+vi)$). Each of these algorithms has been allowed a time limit of 3600 s. We also report in column $z(heur)$ an upper bound to the network cost obtained by a simple rounding-and-rerouting heuristic, so as to evaluate the accuracy of our method. Moreover, we have solved the design problem with dedicated protection (the network cost is displayed in column $z(dp)$) in order to compare the cost savings when adopting Shared Protection. Finally, the last two columns contain the number of valid inequalities (11) and (14) inserted between each call to the oracle.

It is apparent that our Lagrangian approach with the insertion of cut-set inequalities outperforms the other methods in almost all cases. It is worth to insert valid inequalities to improve the lower bound as their separation routines are very simple and fast. In most cases, with the given time limit CPLEX could not even solve the LP relaxation; as a lower bound we have therefore used the best solution from the dual simplex method.

The upper bound computed by our simple rounding heuristic gives an idea of the substantial savings in the cost of a network that adopts Shared Protection. It is not our aim to present an efficient heuristic for SP network design, yet we feel that a stronger upper bounding approach would obtain very cheap networks and give a good measure of accuracy of our Lagrangian approach, whose main feature is to decompose an ILP problem with a huge number of variables and constraints.

VII. Conclusions

Shared Protection is definitely of some help in reducing the cost of a topology. Our tests on real-world instances show that in some cases the overall cost can be greatly reduced if shared protection is considered as part of the optimization process. However, modelling even simple instances leads to a huge ILP model which is hardly tractable even when considering its LP relaxation. We have obtained good lower bounds by means of a technique that decomposes an instance to handle it more efficiently and finds tighter bounds of the problem.

Although we have restricted our attention to single-fault protection, the more general case of 1:k SP, where the resulting network is tolerant to k simultaneous link failures, is also of interest from a modelling point of view, given the complexity of its simplest version. Some research is currently under way for the quadratic, non-separable min-cost flow problem that builds up part of the decomposed problem shown in section V.

Acknowledgements

We are grateful to Antonio Frangioni from Università di Pisa for providing us the C++ code for the bundle method used in all tests.

References

[1] A. Atamtürk. (2002). “On capacitated network design cut-set polyhedra,” *Mathematical Programming B* 92, 425-437.

[2] P. Belotti. (2003). *Multicommodity network design with survivability constraints: some models and algorithms,* PhD Thesis, Politecnico di Milano.

[3] D. Bienstock, Oktay Günlük. (1996). “Capacitated network design – polyhedral structure and computation,” *INFORMS Journal on Computing* 8, 243-259.

[4] S. Datta, S. Sengupta, S. Biswas, S. Datta, “Efficient channel reservation for backup paths in optical mesh networks,” *IEEE Globecom*, San Antonio (TX), November 2001.

[5] DIMACS network file format. Specification available on line at ftp://dimacs.rutgers.edu/pub/netflow/general-info/specs.tex

[6] M. Fischetti, A. Lodi, “Local Branching,” Technical Report OR/02/6, DEIS – Università di Bologna, 2002.

[7] B. Gendron, T. G. Crainic, A. Frangioni. (2001). “Bundle-based relaxation methods for multicommodity capacitated fixed charge network design,” *Discrete Applied Mathematics* 112, 73-99.

[8] D. Gusfield. (1990). “Very simple methods for all pairs network flow analysis”, *SIAM Journal on Computing* 19, 143-155.

[9] J. Kennington, S. Nair, G. Spiride. (1998). “Optimal Spare Capacity Assignment for Path Restorable Mesh Networks,” Technical report 98-CSE-11, Southern Methodist University, Dallas, Texas.

[10] C. Lémarechal, “Lagrangian Relaxation”. Available on line at http://www.optimization-online.org/DB_HTML/2001/03/298.html

[11] A. Lisser, R. Sarkissian, J.P. Vial. (1995). “Survivability in telecommunication networks,” Technical Report 1995.3, Department of Management Studies, University of Geneva, Switzerland.

[12] T. L. Magnanti, P. Mirchandani, R. Vachani. (1993). “The convex hull of two core capacitated network design problems,” *Mathematical Programming* 60, 233-250.

[13] C. Mauz. (2001). “Allocation of spare capacity for shared protection of optical paths in transport networks,” DRCN Conference, Budapest, Hungary, 4-8 October 2001.

[14] C. L. Monma, B.S. Munson, W.R. Pulleyblank. (1990). “Minimum-weight two-connected spanning networks,” *Mathematical Programming* 46, 153-171.

[15] S. Sen, R. Pillai, S. Joshi, A.K. Rathi. (2001). “A mean-variance model for route guidance in advanced traveler information systems,” Transportation Science 35 (1), 37-49.

[16] S. Sengupta, R. Ramamurthy. (2001). “Capacity efficient distributed routing of mesh-restored lightpaths in optical networks,” *IEEE Globecom*, San Antonio (TX).

[17] M. Stoer, G. Dahl. (1994). “A polyhedral approach to multicommodity survivable network design,” *Numerische Mathematik* 68, 149-167.

[18] D. Yuan. (2001). “An annotated bibliography in communication network design and routing”, in *Optimization models and methods for communication network design and routing*, PhD dissertation no. 682, Institute of Technology, Linköpings Universitet (Sweden).

[19] S. Yuan, Jason P. Jue. (2001). “Shared protection routing algorithm for optical network,” *OptiComm 2001*, Denver, USA.

Name	$\|V\|$	$\|E\|$	$\|Q\|$	$z(mip)$	$z(lr)$	$z(lr+vi)$	$z(heur)$	$z(dp)$	n_{11}	n_{14}
bhv1	14	19	70	54869.2	53027.6	59129.1	72169	74560	0	59
bhv2	24	28	136	144160.1	137671.1	165798.3	213461	267236	0	136
bhv3	29	62	140	23072.7	116471.1	120492.8	263080	349838	16	105
bhv4	18	30	116	40644.2	76342.1	84909.5	118618	118618	5	226
bhv5	19	27	94	47096.1	85759.6	90144.4	145342	146324	6	189
bhv6	27	39	186	123410.7	169288.1	191273.4	241218	255329	0	100
bhv7	23	33	186	71233.6	131913.6	156885.8	224725	233196	1	71
bhv8	28	35	82	93086.0	129881.1	146702.9	192015	207789	0	168
bhv9	24	43	174	37146.5	109442.6	119044.2	204631	230441	29	149
bhva	19	23	82	83223.7	79884.0	86361.5	101760	114482	0	96
arpa2_5	21	26	19	7877.0	4824.0	4824.0	10987	11632	3	62
arpa2_25	21	26	94	4926.9	3672.3	3715.3	7755	8559	3	207
arpa2_99	21	26	417	2592.7	3649.9	3649.9	6702	7888	3	131
arpanet_5	24	50	21	3431.4	6094.1	10632.2	20661	20747	30	320
arpanet_25	24	50	127	4503.1	6176.1	14628.6	21650	22654	26	367
arpanet_99	24	50	544	–	6604.3	7494.2	19840	21493	37	204
cost239	11	22	110	6545.0	5060.5	5517.3	9999	13107	9	116
eon_5	19	37	16	7562.5	4292.2	7819.3	13103	13786	31	193
eon_25	19	37	82	2521.1	4359.8	5504.3	9635	9635	31	179
eon_99	19	37	339	1585.1	4045.6	5182.1	8025	10635	30	152
metro	11	42	25	195.6	309.8	341.8	567	567	12	17
njlata_5	11	23	5	10947.3	9896.1	8208.8	11000	16849	11	44
njlata_25	11	23	29	6962.3	7655.4	7988.4	10761	10761	10	35
njlata_99	11	23	110	5094.9	7403.5	8097.5	11235	11235	16	48
nsf1a	14	21	174	63972.7	160512.9	160512.9	457540	488382	7	18
nsf1b	14	21	150	104210.0	235069.3	359526.5	676490	746423	6	38
nsf2	14	22	108	19278.3	22898.1	29612.6	52250	56440	17	74
pacbell_5	15	21	16	11187.0	7054.2	9234.9	14088	15451	0	42
pacbell_25	15	21	44	4534.0	3460.1	4377.7	6223	6223	0	78
pacbell_99	15	21	206	3008.8	3249.1	3791.5	5466	5466	0	80
toronto_5	25	55	32	388.9	2232.1	2248.8	5126	6060	25	54
toronto_25	25	55	151	254.0	1671.0	1866.1	4577	6845	21	47
usld_5	28	45	30	–	12329.0	22401.9	40019	40019	27	193
usld_25	28	45	165	–	16309.9	18690.2	41387	49163	29	50

TABLE I

COMPARISON OF LOWER BOUNDING METHODS AND OF NETWORK COSTS FOR DIFFERENT PROTECTION SCHEMES ON REAL-WORLD INSTANCES.