SOLVING THE TWO-CONNECTED NETWORK WITH BOUNDED MESHES PROBLEM

BERNARD FORTZ and MARTINE LABBÉ

Institut de Statistique et de Recherche Opérationnelle, SMG, Université Libre de Bruxelles, Belgium

FRANCESCO MAFFIOLI

DEI, Politecnico di Milano, Italy

(Received August 1996; revisions received April 1998, December 1998; accepted January 1999)

We study the problem of designing at minimum cost a two-connected network such that the shortest cycle to which each edge belongs (a "mesh") does not exceed a given length K. This problem arises in the design of fiber-optic-based backbone telecommunication networks. A Branch-and-Cut approach to this problem is presented for which we introduce several families of valid inequalities and discuss the corresponding separation algorithms. Because the size of the problems solvable to optimality by this approach is too small, we also develop some heuristics. The computational performances of these exact and approximate methods are then thoroughly assessed both on randomly generated instances as well as instances suggested by real applications.

We consider the problem of designing a minimum *cost* network N with the following constraints:

(a) N contains at least two vertex-disjoint paths between every pair of vertices (2-*connectivity constraints*), and

(b) each edge of N must belong to at least one cycle whose *length* is bounded by a given constant K (*mesh constraints*).

We call this problem the 2-*Connected Network with Bounded Meshes* problem (abbreviated as 2CNBM).

The 2CNBM problem finds its motivation in the rapidly developing field of telecommunication networks, in particular in the search for optimum topologies for these networks. In recent years the development of telecommunication networks is characterized by the introduction of fiber-optic technology: The very high capacity and reliability *per fiber cable* has made hierarchical routing and bundling of traffic economically attractive, resulting in sparse network topologies.

However, sparse networks are very sensitive to damage, and providing alternative communication paths to enhance the *survivability* of the network becomes mandatory (see, e.g., Stoer 1992, Ch. 1).

In the most common mathematical model, a graph $G=(V,E)$ is considered where V is the set of vertices that have to be connected and E is the set of edges that is the set of potential links between vertices. From E, the subset F constituting the network $N=(V,F)$ has to be chosen. Each edge e of E has a fixed nonnegative cost c_e, and the objective is to find the subset F of E of minimum total cost, such that the resulting network $N=(V,F)$ satisfies the survivability requirement. From this point of view a graph can be either *k-edge connected* or *k-vertex connected*, which means that the removal of any $(k-1)$ or fewer edges (respectively, vertices) leaves G connected.

Two-connected networks have been found to provide in most cases a sufficient level of survivability and a considerable amount of research has focused on so-called *low connectivity constrained* network design problems, i.e., problems for which each vertex j is characterized by a requirements $r_j \in \{0,1,2\}$ and $\min\{r_v, r_w\}$ vertex-disjoint paths between every distinct pair of vertices v, w are required. Work on this kind of problem goes from the early contributions of Steiglitz et al. (1969) to the more recent articles of Grötschel and Monma (1990), Boyd and Hao (1993), Monma and Shallcross (1989), Grötschel et al. (1992, 1995), and others. For in-depth surveys in this area the reader is referred to Stoer (1992) and Grötschel et al. (1992, 1995). The special case where two edge-disjoint (or vertex-disjoint) paths are required between each pair of vertices has been studied by Monma et al. (1990), where structural properties for optimal 2-connected spanning networks are given. There it is shown that—provided the graph is complete and costs satisfy the triangle inequality—the minimum cost of a two-edge connected spanning network is equal to the minimum cost of a two-vertex connected one. Moreover, every vertex in an optimal solution has degree 2 or 3.

Related to mesh constraints, Alevras et al. (1998) consider the installation of capacity in an existing network such that demand can be routed on length-constrained node-disjoint paths. However, to the authors' knowledge, mesh constraints have never been considered in the topological design of the network. This is a relevant extension: If a

Subject classifications: Communication: survivable network design/model for communication network design. Programming; integer: cutting plane and heuristics/ILP: branch-and-cut heuristics.
Area of review: TELECOMMUNICATIONS.

0030-364X/00/4806-0866 $05.00
1526-5463 electronic ISSN

connection is broken, the flow that was routed using such connection needs to be rerouted. The extra constraint limits the region of influence of the traffic which is necessary to reroute. For instance, the optimal solution of the 2-connected network problem without mesh constraints on the set of 52 zonal centers of Belgium turns out to be a shortest Hamiltonian cycle. This implies that any edge failure would require rerouting of the flow that passed through that edge, using all the edges of the network—an obviously undesirable feature. Furthermore, the emerging technology of *self-healing rings* can be used for rerouting only if the network satisfies such bounded mesh constraints. A self-healing ring is a cycle in the network equipped in such a way that any link failure in the ring is automatically detected by the link end nodes and the traffic rerouted along the alternative path in the cycle. When such a strategy is chosen, rings must cover the network, and their size must be limited, which are two properties provided by our model.

Two mathematical formulations are presented in Section 1. Section 2 describes a Branch-and-Cut approach. Section 3 presents several heuristic methods; one of them guarantees a 2-approximate solution. Both the exact as well as the heuristic methods presented in the paper are tested with randomly generated as well as instances coming from real applications. These results are summarized in Section 4.

1. MATHEMATICAL FORMULATIONS

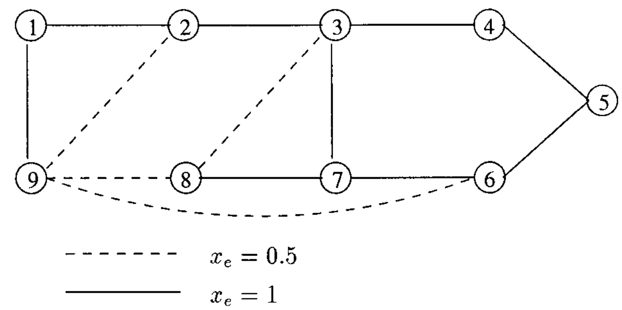
Let $G=(V,E)$ be an undirected graph where V is the set of vertices and E is the set of edges that represent the possible pairs of vertices between which a direct transmission link can be placed. Each edge $e:=\{i,j\}\in E$ has a nonnegative cost $c_e:=c_{ij}$ and a length $d_e:=d_{ij}:=d(i,j)$. It is assumed that these edge lengths satisfy the triangle inequality. The cost of a network (V,F) where $F\subseteq E$ is a subset of possible edges is denoted by $c(F):=\Sigma_{e\in F}\, c_e$. The distance between two vertices i and j in this network is denoted by $d_F(i,j)$ and is given by the length of a shortest path linking these two vertices.

Given the graph $G=(V,E)$ and $W\subset V$, the edge set $\delta(W):=\{\{i,j\}\in E\mid i\in W,\ \ j\in V\backslash W\}$ is called the *cut* induced by W. We write $\delta_G(W)$ to make clear—in case of possible ambiguities—with respect to which graph the cut induced by W is considered. We also denote by $V-z:=V\backslash\{z\}$ and $E-e:=E\backslash\{e\}$ the subsets obtained by removing one vertex or one edge from the set of vertices or edges. $G-z$ denotes the graph $(V-z, E\backslash\delta(\{z\})$, i.e., the graph obtained by removing a vertex z and its incident edges from G.

We associate with every subset $F\subseteq E$ an *incidence vector* $\mathbf{x}=(x_e)_{e\in E}\in\{0,1\}^{|E|}$ by setting $x_e:=1$ if $e\in F$, and $x_e:=0$ otherwise. Conversely, each vector $\mathbf{x}\in\{0,1\}^{|E|}$ induces a subset $F:=\{e\in E\mid x_e=1\}$ of the edge set E. For any subset of edges $F\subseteq E$ we define $x(F):=\Sigma_{e\in F}\, x_e$.

For each edge $e\in E$, we define $\mathscr{C}_e$ as the set of cycles in $G=(V,E)$ that include edge e and whose length is less than or equal to K. The 2CNBM problem can be stated as follows:

Figure 1. A fractional solution satisfying equations (5) and (6).



$$\min \quad \sum_{e\in E} c_e x_e$$

s.t.

$$x(\delta(W))\geqslant 2, \quad W\subset V,\ \phi\neq W\neq V, \tag{1}$$

$$x(\delta_{G-z}(W))\geqslant 1, \quad z\in V,\ W\subset V\backslash\{z\},\ \phi\neq W\neq V\backslash\{z\}, \tag{2}$$

$$x_e=1\Rightarrow\sum_{f\in c} x_f=|c|, \quad \text{for some } c\in\mathscr{C}_e, \tag{3}$$

$$x_e\in\{0,1\}, \quad e\in E. \tag{4}$$

Inequalities (1) enforce that removing an edge preserves connectivity: They are called *cut inequalities*. Inequalities (2) ensure that there is no articulation vertex in the resulting graph: They are called *vertex* (or *node*) *cut inequalities*. Together with (4) we would then obtain the formulation of the minimum 2-connected network problem utilized by Grötschel et al. (1992) and Stoer (1992).

To formulate the 2CNBM problem as an integer linear program, we must express the mesh constraints (3) as linear inequalities. This can be done as in Fortz et al. (1997) by using new variables describing the presence of feasible cycles in the solution. More precisely, we associate a binary variable z^c to each cycle $c\in\bigcup_{e\in E}\mathscr{C}_e$, and constraints (3) can be expressed as:

$$\sum_{c\in\mathscr{C}_e} z^c\geqslant x_e, \quad e\in E, \tag{5}$$

$$z^c\leqslant x_f, \quad f\in c,\ c\in\bigcup_{e\in E}\mathscr{C}_e. \tag{6}$$

We propose here a strenghtened formulation. Consider a fractional solution given by the graph depicted in Figure 1. All edge lengths are equal to 1 and the bound $K=5$. The feasible cycles are:

$$c_1=\{\{1,2\},\{1,9\},\{2,9\}\},$$
$$c_2=\{\{1,2\},\{1,9\},\{2,3\},\{3,8\},\{8,9\}\},$$
$$c_3=\{\{2,3\},\{2,9\},\{3,8\},\{8,9\}\},$$
$$c_4=\{\{2,3\},\{2,9\},\{3,7\},\{7,8\},\{8,9\}\},$$
$$c_5=\{\{3,7\},\{3,8\},\{7,8\}\},$$
$$c_6=\{\{6,7\},\{6,9\},\{7,8\},\{8,9\}\},$$
$$c_7=\{\{3,7\},\{3,8\},\{6,7\},\{6,9\},\{8,9\}\},$$
$$c_8=\{\{3,4\},\{3,7\},\{4,5\},\{5,6\},\{6,7\}\}.$$

It is easy to verify that taking $z^{c_i}=0.5$, $i=1,\ldots,6$, $z^{c_7}=0$ and $z^{c_8}=1$ provide a feasible solution to (5)–(6). However, edge $\{2,3\}$ belongs only to cycles c_2, c_3, and c_4; and all these cycles contain edge $\{8,9\}$. Because $x_{\{8,9\}}=0.5$, we would like to restrict the contribution of these cycles to 0.5 when we impose that $\{2,3\}$ belongs to a feasible cycle.

To do so, we must distinguish which cycles are used to impose mesh constraints for a particular edge by introducing a variable for each feasible cycle containing a given edge, for all the edges in the network. Therefore, we define new binary variables y_e^c, $c\in\mathscr{C}_e$, $e\in E$, such that

$$y_e^c=\begin{cases}1 & \text{if cycle } c \text{ is present in the solution } F \text{ and covers edge } e,\\ 0 & \text{otherwise.}\end{cases}$$

With these variables, mesh constraints can be expressed as

$$\sum_{c\in\mathscr{C}_e} y_e^c \geqslant x_e, \quad e\in E, \tag{7}$$

$$\sum_{c\in\mathscr{C}_e:\, f\in c} y_e^c \leqslant x_f, \quad e\in E,\ f\in E\setminus\{e\}, \tag{8}$$

where constraints (8) restrict the contribution of cycles that share some edges. For the above fractional solution, we obtain the inconsistent subset of constraints

$$y_{\{2,3\}}^{c_2}+y_{\{2,3\}}^{c_3}+y_{\{2,3\}}^{c_4}\geqslant x_{\{2,3\}}=1, \tag{9}$$

$$y_{\{2,3\}}^{c_2}+y_{\{2,3\}}^{c_3}+y_{\{2,3\}}^{c_4}\leqslant x_{\{8,9\}}=0.5. \tag{10}$$

The new formulation is therefore stronger than the previous one. The complete formulation of the 2CNBM problem is given below:

$$\min \quad \sum_{e\in E} c_e x_e$$

s.t.

$$x(\delta(W))\geqslant 2, \quad W\subset V,\ \phi\neq W\neq V, \tag{11}$$

$$x(\delta_{G-z}(W))\geqslant 1, \quad z\in V,\ W\subset V\setminus\{z\},\ \phi\neq W\neq V\setminus\{z\}, \tag{12}$$

$$\sum_{c\in\mathscr{C}_e} y_e^c \geqslant x_e, \quad e\in E, \tag{13}$$

$$\sum_{c\in\mathscr{C}_e:\, f\in c} y_e^c \leqslant x_f, \quad e\in E,\ f\in E\setminus\{e\}, \tag{14}$$

$$x_e, y_e^c\in\{0,1\}, \quad c\in\mathscr{C}_e,\ e\in E. \tag{15}$$

This formulation involves $\Sigma_{e\in E}|\mathscr{C}_e|$ new variables and $m(m+1)$ new constraints with respect to the formulation of the 2-connected problem without mesh constraints. Because the number of variables and constraints are both exponential, we propose a Branch-and-Cut approach, using the decision variables $(x_e)_{e\in E}$ only, to solve the problem.

2. VALID INEQUALITIES AND A BRANCH-AND-CUT ALGORITHM

2.1. Cut and Vertex Partition Inequalities

The first inequalities to introduce are those that enforce two-connectivity. An obvious set of such inequalities is the set of cut inequalities (1) and vertex-cut inequalities (2). These inequalities can be generalized as in Grötschel et al. (1992), obtaining *partition inequalities* and *vertex partition inequalities*.

In our case, partition inequalities are just the sum of the cut inequalities for the subsets inducing the partition, and so do not need to be considered. On the other hand, vertex partition inequalities are in general stronger than vertex cut inequalities. These inequalities are defined as

$$\frac{1}{2}\sum_{i=1}^{p} x(\delta_{G-z}(W_i))\geqslant p-1,$$

where $z\in V$ is a given vertex and $W_1,\ldots,W_p$ a partition of $V\setminus\{z\}$ into nonempty subsets. The separation problem is solved as suggested in Grötschel et al. (1992).

Other kinds of inequalities have been proposed for the two-connected network problem (Grötschel et al. 1992, Stoer 1992). However, we found cut and vertex partition inequalities sufficient to efficiently enforce the two-connectivity of all the instances we considered. The other inequalities are apparently useless in our case, because the constraint on the mesh lengths is a stronger requirement compared to the two-connectivity.

2.2. Mesh Cover Inequalities

Suppose we have a two-connected solution, with an (infeasible) edge e violating the mesh constraints (13) and (14), i.e., at least one edge in each feasible cycle $c\in\mathscr{C}_e$ is not in the current solution. A valid inequality can be generated by imposing that at least one of these edges must be in the solution, as proposed by Maffioli and Värbrand (1994). More precisely, if $S\subseteq E$ is a subset of edges such that $S\cap c\neq\phi$ forall $c\in\mathscr{C}_e$, then $x(S)\geqslant x_e$ is a valid inequality.

We derive here stronger inequalities by looking at the projection of our formulation of mesh constraints. Suppose $\tilde{\mathbf{x}}=(\tilde{x}_e)_{e\in E}$ is a vector not satisfying mesh constraints (14)–(13), i.e., such that the linear system

$$\sum_{c\in\mathscr{C}_e:\, f\in c} y_e^c \leqslant \tilde{x}_f, \quad e\in E,\ f\in E\backslash\{e\},$$

$$\sum_{c\in\mathscr{C}_e} y_e^c \geqslant \tilde{x}_e, \quad e\in E,$$

$$y_e^c\geqslant 0, \quad e\in E,\ c\in\mathscr{C}_e,$$

is infeasible. Since $\tilde{\mathbf{x}}$ is fixed, the system can be decomposed by edge. Therefore, there exists $e\in E$ such that

$$\sum_{c\in\mathscr{C}_e} y_e^c < \tilde{x}_e$$

for all $(y_e^c)_{c\in\mathscr{C}_e}$, satisfying

$$\sum_{c\in\mathscr{C}_e:\, f\in c} y_e^c \leqslant \tilde{x}_f, \quad f\in E\backslash\{e\}, \tag{16}$$

$$y_e^c\geqslant 0, \quad c\in\mathscr{C}_e. \tag{17}$$

By duality, the linear program

$$\min \quad \sum_{f\in E\backslash\{e\}} u_f\tilde{x}_f$$

$$\text{s.t.} \quad \sum_{f\in c\backslash\{e\}} u_f\geqslant 1, \quad c\in\mathscr{C}_e, \tag{18}$$

$$u_f\geqslant 0, \quad f\in E\backslash\{e\}, \tag{19}$$

has an optimal value smaller than $\tilde{x}_e$. Let $(\bar{u}_f)_{f\in E\backslash\{e\}}$ be the optimal solution of this linear program. A valid inequality for the 2CNBM problem is

$$\sum_{f\in E\backslash\{e\}} \bar{u}_f x_f\geqslant x_e. \tag{20}$$

We call inequalities (20) *mesh cover inequalities*.

Unfortunately, solving the separation problem exactly requires the knowledge of all the feasible cycles $c\in\mathscr{C}_e$.

2.3. Metric Inequalities

We now describe a new class of valid inequalities, obtained by projecting flow formulation of the mesh constraints. This approach is related to the work of Bienstock et al. (1998).

PROPOSITION 1. *Consider an edge $e:=\{i,j\}\in E$ and a set of node potentials $(\alpha_k)_{k\in V}$ satisfying*

$$\alpha_i-\alpha_j>K-d(i,j).$$

Then

$$\sum_{f\in E\backslash\{e\}} v_f x_f\geqslant x_e \tag{21}$$

is a valid inequality for 2CNBM where

$$v_f=\min\left(1,\max\left(0,\frac{|\alpha_l-\alpha_k|-d(k,l)}{\alpha_i-\alpha_j+d(i,j)-K}\right)\right) \tag{22}$$

for all $f:=\{k,l\}\in E\backslash\{e\}$.

PROOF. Let $F\subseteq E$ be a feasible solution defined by the $(x_e)_{e\in E}$ variables, and consider an edge $e:=\{i,j\}\in F$. This edge belongs to a feasible cycle of length less than or equal to K. It means that the shortest path between i and j in $F\backslash\{e\}$ has a length less than or equal to $K-d(i,j)$. In other words, the problem

$$\min \quad \bar{w}=\sum_{\{k,l\}\in E\backslash\{e\}} d(k,l)(u_{kl}+u_{lk})$$

s.t.

$$\sum_{l:\,\{k,l\}\in E\backslash\{e\}} (u_{kl}-u_{lk}) = \begin{cases} x_{\{i,j\}} & \text{if } k=i,\\ -x_{\{i,j\}} & \text{if } k=j,\\ 0 & \text{otherwise,}\end{cases} \quad k\in V, \tag{23}$$

$$u_{kl}+u_{lk}\leqslant x_{\{k,l\}}, \quad \{k,l\}\in E\backslash\{e\}, \tag{24}$$

$$u_{kl}\geqslant 0, \quad \{k,l\}\in E\backslash\{e\}, \tag{25}$$

$$u_{lk}\geqslant 0, \quad \{k,l\}\in E\backslash\{e\}, \tag{26}$$

has an optimal solution value w^* such that $w^*\leqslant K-d(i,j)$.

On the other hand, if $\{i,j\}\notin F$, $u_{kl}=0$, $k,l\in V$ is feasible, meaning that the optimal solution of the problem is $w^*=0$. We can conclude that the optimal solution value always satisfies

$$w^*\leqslant (K-d(i,j))x_{\{i,j\}}.$$

The dual of this minimum cost flow problem is

$$\max \quad w=(\alpha_i-\alpha_j)x_{\{i,j\}}-\sum_{\{k,l\}\in E\backslash\{e\}} \beta_{\{k,l\}}x_{\{k,l\}}$$

s.t.

$$\alpha_k-\alpha_l-\beta_{\{k,l\}}\leqslant d(k,l), \quad \{k,l\}\in E\backslash\{e\}, \tag{27}$$

$$\alpha_l-\alpha_k-\beta_{\{k,l\}}\leqslant d(k,l), \quad \{k,l\}\in E\backslash\{e\}, \tag{28}$$

$$\beta_{\{k,l\}}\geqslant 0, \quad \{k,l\}\in E\backslash\{e\}. \tag{29}$$

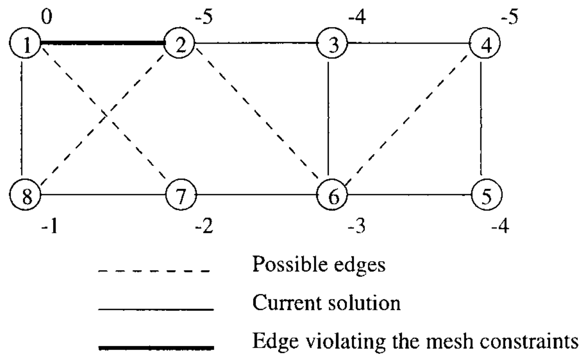
It is easy to see that the optimum of the dual is reached for

$$\beta_{\{k,l\}}=\max(0,|\alpha_l-\alpha_k|-d(k,l)), \quad \{k,l\}\in E\backslash\{e\},$$

and the dual problem becomes

$$\max w=(\alpha_i-\alpha_j)x_{\{i,j\}} - \sum_{\{k,l\}\in E\backslash\{e\}} \max(0,|\alpha_l-\alpha_k|-d(k,l))x_{\{k,l\}}.$$

Figure 2. Values of $\alpha_k = \pi_k$.



By duality for any values of $(\alpha_k)_{k\in V}$, we must have $w \leqslant w^* \leqslant (K-d(i,j))x_{\{i,j\}}$, which implies that

$$\sum_{\{k,l\}\in E\setminus\{e\}} \max(0, |\alpha_l-\alpha_k| - d(k,l))x_{\{k,l\}}$$

$$\geqslant (\alpha_i - \alpha_j + d(i,j) - K)x_{\{i,j\}}$$

is a valid inequality. Dividing by the positive coefficient of $x_{\{i,j\}}$, this leads to

$$\sum_{\{k,l\}\in E\setminus\{e\}} \max\left(0, \frac{|\alpha_l-\alpha_k| - d(k,l)}{\alpha_i-\alpha_j+d(i,j)-K}\right)x_{\{k,l\}} \geqslant x_{\{i,j\}}.$$

Because the variables are binary, a stronger valid inequality is obtained by changing the coefficient of a variable to 1 when

$$\frac{|\alpha_l-\alpha_k| - d(k,l)}{\alpha_i-\alpha_j+d(i,j)-K} > 1,$$

leading to (21). □

Inequalities (21) are called *metric inequalities*. Given a fractional point $(x_e)_{e\in E}$, an easy way to solve the separation problem is to compute the minimum cost flow defined by (23)–(26) for each $e\in E$ and to consider the inequality derived from α_k values equal to the corresponding node potentials π_k. However, it is possible to generate stronger inequalities, as illustrated below.

Consider the graph depicted in Figure 2, with edge lengths equal to 1 and a bound $K=5$. Edge $\{1,2\}$ violates the mesh constraints. Let π_k be the node potentials corresponding to the minimum cost flow between 1 and 2. These values are given in Figure 2. The inequality obtained for edge $\{1,2\}$ by taking $\alpha_k = \pi_k$ is

$$x_{17} + x_{28} + x_{26} + x_{46} \geqslant x_{12}. \tag{30}$$

The aim of introducing this inequality is to enforce that the distance between vertices 1 and 2 becomes less than or equal to 4. But if we add edge $\{4,6\}$ to the current solution in the graph of Figure 2, inequality (30) is satisfied, while the distance between vertices 1 and 2 remains equal to 5. Moreover, if we take $\alpha_4 = -4$ in place of $\alpha_4 = -5$, we get the new valid inequality

$$x_{17} + x_{28} + x_{26} \geqslant x_{12}, \tag{31}$$

which is stronger than (30) and involves only edges that contribute to shorten the distance between vertices 1 and 2. Note that this inequality is also a mesh cover inequality.

More precisely, we would like to find an inequality in which each edge with positive coefficient contributes to the decrease of the distance between i and j if added to the solution. Heuristic 1 provides $(\alpha_k)_{k\in V}$ with this property.

HEURISTIC 1 (SEPARATION OF METRIC INEQUALITIES). *Let $\tilde{\mathbf{x}} = (\tilde{x}_f)_{f\in E}$ be an infeasible (fractional) solution of the 2CNBM problem and $e := \{i,j\}\in E$ be an edge such that $\tilde{x}_e > 0$.*

1. *Compute a minimum cost flow of value $\tilde{x}_e$ between i and j with capacities $(\tilde{x}_f)_{f\in E\setminus\{e\}}$ in $G-e$. Let $(\pi_k)_{k\in V}$ be the corresponding node potentials.*
2. *If $\pi_j + K - d_e \geqslant 0$, all metric inequalities having x_e as right-hand side are satisfied. STOP.*
3. *Set*

$$\alpha_k := \begin{cases} \pi_j & \text{if } k=j, \\ 0 & \text{otherwise} \end{cases}$$

for all $k\in V$.
All vertices are unlabeled.

4. *While there exists an unlabeled vertex, do:*
   - *select a vertex $l\in V$ such that $\alpha_l = \min\{\alpha_k : k\in V, k \text{ unlabeled}\}$;*
   - *for all vertices $k\in V$ such that k is adjacent to l in $E\setminus\{e\}$, if $\alpha_k > \alpha_l + d(k,l)$ and if $\alpha_l + d(k,l) \geqslant \pi_k$, set $\alpha_k := \alpha_l + d(k,l)$;*
   - *label vertex l.*
5. *Set*

$$v_f = \min\left(1, \max\left(0, \frac{|\alpha_l-\alpha_k| - d(k,l)}{\alpha_i-\alpha_j+d(i,j)-K}\right)\right)$$

for all $f := \{k,l\}\in E\setminus\{e\}$.

6. *If $\sum_{f\in E\setminus\{e\}} v_f x_f < x_e$ is violated, add it to the current LP.*

Note that the computation can be restricted to the subgraph Ω_e induced by edges belonging to a feasible cycle including e, i.e., $\Omega_e = \bigcup_{C\in\mathscr{C}_e} C\setminus\{e\}$. Applied to the example of Figure 2, this heuristic leads to inequality (31).

Heuristic 1 is a polynomial time procedure to separate metric inequalities. The inequalities generated provide a weaker linear relaxation than the one obtained using mesh cover inequalities, as shown in Proposition 3. However, if the heuristic stops with node potentials $(\pi_k)_{k\in V}$ such that $\pi_j + K - d_e \geqslant 0$, then all mesh cover inequalities for edge e are satisfied (see Proposition 2), and the two linear relaxations are thus equivalent.

Applying this heuristic instead of the exact separation for mesh cover inequalities may fail if the heuristic stops with $\pi_j + K - d_e < 0$ and an inequality that is not violated. However, in practice, each time this case occurred, we remarked that all mesh cover inequalities were satisfied. We thus preferred this approach to the exact separation requiring computation of all the feasible cycles.

PROPOSITION 2. *If Heuristic* 1 *applied to edge* $e := \{i, j\} \in E$ *finishes with* $\pi_j + K - d_e \geqslant 0$, *then all mesh cover inequalities* (20) *are satisfied.*

PROOF. Because $-\pi_j \leqslant K - d_e$, the flow generated by the minimum cost flow algorithm uses only paths of length less than or equal to $K - d_e$. Each one of these paths, combined with edge e, forms a feasible cycle belonging to $\mathscr{C}_e$.

Setting y_e^c equal to the flow sent on the corresponding path leads to a feasible solution of the system (16)–(17) such that $\sum_{c\in\mathscr{C}_e} y_e^c \geqslant \tilde{x}_e$. We can conclude that all mesh cover inequalities are satisfied. □

PROPOSITION 3. *Any metric inequality generated by Heuristic* 1 *is a mesh cover inequality.*

PROOF. To prove this result, it is sufficient to show that $(v_f)_{f\in E}$ is a feasible solution to the separation problem (18)–(19) for mesh cover inequalities. The nonnegativity constraints are obviously satisfied.

Let $c \in \mathscr{C}_e$ be a feasible cycle. If there exists $f \in c$ such that $v_f = 1$, obviously $\sum_{f\in c\setminus\{e\}} v_f \geqslant 1$ and (18) is satisfied. Otherwise, for all $f \in c$,

$$v_f = \max\left(0, \frac{|\alpha_l - \alpha_k| - d(k,l)}{\alpha_i - \alpha_j + d(i,j) - K}\right),$$

and

$$\begin{aligned}\sum_{f\in c\setminus\{e\}} v_f &\geqslant \frac{\sum_{f:=\{k,l\}\in c\setminus\{e\}} (|\alpha_l - \alpha_k| - d(k,l))}{\alpha_i - \alpha_j + d(i,j) - K}\\ &\geqslant \frac{\sum_{f:=\{k,l\}\in c\setminus\{e\}} (\alpha_l - \alpha_k - d(k,l))}{\alpha_i - \alpha_j + d(i,j) - K}\\ &= \frac{\alpha_i - \alpha_j - \sum_{f\in c\setminus\{e\}} d_f}{\alpha_i - \alpha_j + d(i,j) - K}\\ &\geqslant 1,\end{aligned}$$

because c is feasible and thus $\sum_{f\in c\setminus\{e\}} d_f \leqslant K - d(i,j)$.

Constraint (18) is therefore satisfied. □

This result implies that if Heuristic 1 generates a violated metric inequality, then there exists a mesh cover inequality which is at least as violated as the metric inequality.

2.4. Weighted Partition Inequalities

We now describe inequalities related to a partition of V into subsets $W_1, \ldots, W_p$.

PROPOSITION 4. *Consider a graph* $G = (V, E)$ *and let* $W_1, \ldots, W_p$ *be a partition of* V. *Then*

$$\frac{1}{2}\sum_{i=1}^{p} \sum_{e\in\delta(W_i)} (K - d_e)x_e \geqslant (p-1)K \tag{32}$$

is a valid inequality for 2CNBM.

PROOF. Let $(x_e)_{e\in E}$ be variables defining a feasible solution of 2CNBM.

We will first prove the result for $p = 2$, i.e., the partition defines a cut. Note that, by the triangle inequality, an edge $e \in E$ such that $d_e \geqslant K/2$ does not belong to any cycle of length less than or equal to K, and so $x_e = 0$. This means that

$$d_e x_e \leqslant \frac{K}{2} x_e, \quad e \in E. \tag{33}$$

Thus we have

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{2} \sum_{e\in\delta(W_i)} (K - d_e)x_e &= \sum_{e\in\delta(W_1)} (K - d_e)x_e \quad \text{because } \delta(W_1) = \delta(W_2),\\ &\geqslant \sum_{e\in\delta(W_1)} \frac{K}{2}x_e \quad \text{by (33)},\\ &= \frac{K}{2}x(\delta(W_1))\\ &\geqslant K,\end{aligned}$$

because the solution network must be two-connected.

Now suppose that $p > 2$ and that the result holds for all q, $2 \leqslant q \leqslant p-1$. Suppose also that there exist a partition $W_1, \ldots, W_p$ of V such that (32) is violated, i.e.,

$$\frac{1}{2}\sum_{i=1}^{p} \sum_{e\in\delta(W_i)} (K - d_e)x_e < (p-1)K. \tag{34}$$

Consider a feasible cycle using at least one edge in $\bigcup_{i=1}^{p} \delta(W_i)$, and denote by C the subset of edges defining this cycle. C is such that

$$\sum_{e\in C} d_e x_e \leqslant K.$$

Let $\overline{C} = C \cap (\bigcup_{i=1}^{p} \delta(W_i))$. Obviously,

$$\sum_{e\in\overline{C}} d_e x_e \leqslant K. \tag{35}$$

Moreover, the number of elements in $\overline{C}$ is given by $\sum_{e\in\overline{C}} x_e$, implying

$$\sum_{e\in\overline{C}} K x_e = |\overline{C}|K. \tag{36}$$

Combining (35) and (36), we obtain

$$\sum_{e\in\overline{C}} (K - d_e)x_e \geqslant (|\overline{C}| - 1)K. \tag{37}$$

Without loss of generality, we may assume that the first $q-1$ subsets of the partition—$W_1, \ldots, W_{q-1}$—are not incident to $\overline{C}$, while all the others—$W_q, W_{q+1}, \ldots, W_p$—are. We define a new partition of V in q subsets as follows:

$$\overline{W_i} = \begin{cases} W_i & \text{if } i<q, \\ \bigcup_{j=q}^{p} W_j & \text{if } i=q. \end{cases}$$

This partition is such that

$$\frac{1}{2}\sum_{i=1}^{q} \sum_{e\in\delta(\overline{W}_i)} (K-d_e)x_e$$

$$= \frac{1}{2}\sum_{i=1}^{p} \sum_{e\in\delta(W_i)} (K-d_e)x_e - \sum_{e\in T}(K-d_e)x_e, \qquad (38)$$

where $T = (\bigcup_{i=q}^{p} \delta(W_i)) - \delta(\overline{W_q})$.

By the definition of $\overline{C}$ and $\overline{W_q}$, it is clear that $\overline{C} \subseteq \bigcup_{i=q}^{p} \delta(W_i)$ and $\overline{C} \cap \delta(\overline{W_q}) = \phi$, so $\overline{C} \subseteq T$. Using (37), this leads to:

$$\sum_{e\in T}(K-d_e)x_e \geqslant \sum_{e\in\overline{C}}(K-d_e)x_e \geqslant (|\overline{C}|-1)K. \qquad (39)$$

Combining (34), (38), and (39), we have:

$$\frac{1}{2}\sum_{i=1}^{q} \sum_{e\in\delta(\overline{W}_i)} (K-d_e)x_e$$

$$<(p-1)K-(|\overline{C}|-1)K=(p-|\overline{C}|)K.$$

Moreover, $\overline{C}$ is adjacent to at most $|\overline{C}|$ subsets in the original partition, meaning that $p-q+1 \leqslant |\overline{C}|$, or $p-|\overline{C}| \leqslant q-1$.

We conclude that

$$\frac{1}{2}\sum_{i=1}^{q} \sum_{e\in\delta(\overline{W}_i)} (K-d_e)x_e <(q-1)K,$$

which leads to a contradiction since $q<p$, and (32) is thus a valid inequality for 2CNBM. □

We call this new class of inequalities *weighted partition inequalities*. A polynomial method for separating the weighted partition inequalities is available only when $p=2$ —*weighted cut inequalities*—using the Gomory-Hu algorithm (Gomory and Hu 1961). Following ideas of Grötschel et al. (1992), we try to transform the minimum weighted cut into a partition leading to a violated inequality.

Unfortunately, from the proof of Proposition 4 when $p=2$, we see that the weighted cut inequalities are weaker than cut inequalities, implying that all weighted cut inequalities are satisfied when the solution is two-connected. This implies that we cannot rely on "almost violated" weighted cut inequalities but that we have to try all cuts coming from the Gomory-Hu tree if we want to generate violated weighted partition inequalities. Nevertheless, weighted partition inequalities are the only valid inequalities that combine the two-connectivity constraints with the mesh constraints, and our numerical experiments showed that these inequalities are useful for improving the lower bounds computed in the Branch-and-Cut algorithm.

2.5. The Branch-and-Cut Algorithm

We now briefly describe how we implement a Branch-and-Cut algorithm for 2CNBM.

- Preprocessing: For each edge, compute the length of the shortest cycle containing it. If this length is greater than the bound imposed on meshes, the edge will never appear in a feasible solution and can be removed. If the remaining network is not two-vertex connected, stop: The problem is infeasible.
- Start with an empty pool of valid inequalities.
- Solve the LP formed by the degree constraints $x(\delta(v)) \geqslant 2,\ v\in V$.
- At each node of the Branch-and-Bound tree:
  1. Look for violated inequalities in the pool. If none is found, generate cut, weighted partition and metric inequalities. If no new inequality is found, generate vertex partition inequalities.
  2. If some inequalities are found:
     - Solve the new LP,
     - Put all the inactive inequalities in the pool, and remove them from the LP,
     - Go to Step 1.

     Otherwise,
     - Apply the dual greedy heuristic of §3.4 on the graph corresponding to nonzero variables to get a valid upper bound,
     - Branch on the fractional variable maximizing $\min(x_e, 1-x_e)L$, where L is the length of the shortest cycle including e in the graph defined by the edge set $\{f : x_f > 0\}$. The Branch-and-Bound tree is examined using a best first search strategy.

3. HEURISTICS

We describe briefly five heuristics used to generate feasible solutions; the first four are constructive. In each case, we assume that the problem is feasible, i.e., the set of potential edges is two-vertex connected and satisfies the mesh constraints. Furthermore, we suppose that all redundant edges always have been removed.

3.1. Ear-Inserting Method

The basic idea of this heuristic is derived from the ear-inserting procedure proposed by Monma and Shallcross (1989) for designing minimum-weight two-connected networks. We adapt this procedure in order to satisfy mesh constraints. It is developed for the particular case where the edge costs equal to their lengths.

Our aim is to construct cycles with length at most K, so that the total cost does not increase too much: We first construct the minimum length cycle; we then add at each step the vertex which minimizes the total length increase. When no new vertex can be added to the starting cycle without violating the mesh constraints, we create a new cycle including a pair of vertices already in the solution in order to preserve the two-connectivity requirement. We add as many vertices as possible to this cycle until we reach the bound. Then we repeat the procedure by initiating a new cycle until all vertices are in the solution. Moreover, we also start a new cycle when it is better than augmenting the current one. This procedure is improved by local 2-opt exchange in the cycle being constructed and by allowing to re-use a vertex already in the solution if it leads to a better solution.

3.2. Cutting Cycles in Two Equal Parts

Our numerical experiments show that the optimal two-connected network with no further constraints often turns out to be a Hamiltonian cycle. By adding edges to this cycle, we can satisfy the mesh constraints. In this heuristic, edges are added to cut the cycle in two almost equal parts, and the cutting is repeated for all subcycles until the mesh constraints are satisfied.

3.3. Path Following Method

This method consists of determining an Hamiltonian tour and then, beginning with any vertex, following the tour in, say, the clockwise direction until we reach the most distant vertex of the tour, such that connecting it with the starting vertex yields a cycle of length not greater than K. We insert that edge and start again, taking as first vertex the one that was considered last in the previous iteration, until the solution graph becomes feasible. The procedure is repeated starting from every vertex, and the best solution is kept.

Although rarely attractive in practice, this simple constructive heuristic has the nice feature of suggesting a polynomial algorithm with a worst-case performance bound with respect to the cost of an optimal solution, in the important particular case that the costs satisfy the triangle inequality (and hence G is a complete graph).

Assume we apply the polynomial heuristic starting with the Hamiltonian tour obtained by Christofides heuristic (Christofides 1976). Let $\hat{c}_H$ be the cost of this tour and $\hat{c}_{TCBM}$ the cost of the solution to the 2CNBM problem found by the path following heuristic. Because of the triangle inequality, we have $\hat{c}_{TCBM} \leqslant 2\hat{c}_H$. It was already observed by Frederickson and Já Já (1982) that $\hat{c}_H \leqslant \frac{3}{2} c^*_{TC}$ where c^*_{TC} is the cost of an optimal two-connected subgraph of G. Obviously, $c^*_{TC} \leqslant c^*_{TCBM}$, the cost of an optimal solution to the 2CNBM problem. Hence, this version of the path following heuristic gives a solution with a cost never worse than three times the optimal cost.

Note that each inequality used to get that bound is tight. However, examples for which the bounds are reached do not coincide, and the question of the tightness of our worst-case bound remains open.

3.4. Dual Greedy Method

The fourth heuristic is based on a greedy removal of edges. It starts with all edges and tries to remove the most costly one while preserving two-connectivity and keeping cycles of bounded lengths.

3.5. Randomly Iterated Dual Greedy

Each dual greedy solution can be seen as a local minimum for our problem. In order to find better solutions, we would like to iterate the process to generate different solutions. Suppose we have a dual greedy solution S1: If a better solution S2 exists, at least one edge from S1 is not in S2. In other words, by removing an S1-edge from the set of all possible edges and applying dual greedy again, we find a new solution. We can iterate this procedure by keeping a list of all solutions found and removing from the initial set of edges a randomly selected "covering" of the generated solutions. Applying dual greedy again leads to a new solution.

Even if this heuristic is time and memory consuming, it leads to better results than the others. This is because of the fact that this heuristic permits us to explore different regions of the solution space, avoiding being trapped in local minima. However, choices for new solutions are made at random, while it would be advisable to guide the search. Meta-heuristics such as Simulated Annealing or Tabu Search can be useful tools to achieve this goal, but mesh constraints make the identification of a neighborhood that can be effectively searched a difficult task.

4. COMPUTATIONAL RESULTS

We have tested the Branch-and-Cut algorithm and the heuristics on planar problems in which lengths and costs are equal to the Euclidean distances. Tests were made for different values of the bound, for instances coming from real applications, with 12, 17, 30, and 52 vertices; and for random problems, with vertices uniformly generated in a square of size 250×250. Random problems with 10 to 50 vertices were generated, and we tested five instances of each size.

4.1. Branch-and-Cut Method

The Branch-and-Cut algorithm was implemented in C++ on a SUN Sparc 10 workstation, using CPLEX 4.0 as LP solver. The CPU time limit was fixed to 20 000 seconds.

Table 1 reports results obtained for problems coming from real applications, while Table 2 reports a summary of results obtained for randomly generated problems. The gaps are relative to the best known upper bound. Data on the randomly generated test problems are available at the Web page http://smg.ulb.ac.be/~bfortz/2cnbm/data.html.

Table 1. Results of the Branch-and-Cut algorithm for instances coming from real applications.

V	E	K	C	WP	M	VP	BBN	T	LB	UB	G	GR
12	31	150		Infeasible			—	1	—	—	—	—
12	49	200	8	23	112	29	78	22	622	622	0	9.5
12	58	250	6	17	42	3	10	3	541	541	0	2.6
12	63	300	6	17	71	35	50	11	541	541	0	5.0
12	65	350	7	14	37	28	10	2	521	521	0	2.7
17	55	150		Infeasible			—	1	—	—	—	—
17	88	200	19	44	332	43	147	166	834	834	0	7.6
17	109	250	26	40	304	52	310	280	789	789	0	6.6
17	119	300	21	33	124	22	66	38	726	726	0	2.1
17	126	350	22	40	164	37	170	95	725	725	0	3.3
30	174	150	115	275	811	207	2327	20000	1164	1268	8.2	18.1
30	266	200	79	190	1301	170	2227	20000	1034	1095	5.6	12.2
30	328	250	49	110	654	84	1520	6925	935	935	0	4.3
30	372	300	47	78	764	80	1085	6188	898	898	0	2.7
30	406	350	52	71	314	42	282	940	874	874	0	2.6
52	559	150	115	155	683	66	427	20000	1212	1343	9.8	13.3
52	821	200	103	122	565	69	367	20000	1143	1246	8.3	11.2
52	1040	250	90	122	654	50	308	20000	1107	1181	6.3	8.9
52	1149	300	102	115	858	47	279	20000	1084	1117	3.0	4.7
52	1230	350	137	147	799	53	256	20000	1067	1085	1.7	3.1

V = Number of vertices
E = Number of edges after preprocessing
K = Bound on the mesh lengths
C = Number of cut inequalities
WP = Number of weighted partition inequalities
M = Number of metric inequalities
VP = Number of vertex partition inequalities
BBN = Number of nodes explored in the Branch-and-Bound tree
T = Time (in seconds)
LB = Best lower bound
UB = Best upper bound
G = Remaining gap (in %)
GR = Gap at root node (in %)

For 20 vertices or less, most problems could be solved to optimality. Then, from 30 vertices on, the problem becomes intractable within the time limit. Also, the gap at the root node GR and thus the difficulty of the problem decrease when the bound K on the mesh lengths increases. This is because when this value increases, the problem becomes closer to the two-connected network problem (without mesh constraints), which can be solved efficiently using cut and vertex-partition inequalities. Furthermore, the fact that the gap GR increases when K decreases suggests the need for new valid inequalities exploiting the mesh constraints.

In order to compare the efficiency of the different classes of inequalities, we present in Table 3 the evolution of the lower bound on the 52 node real network for different values of the bound K when using different subsets of inequalities. We can observe that weighted partition inequalities provide the best increase of the lower bound when added to constraints imposing two-connectivity, while the conjunction of all inequalities is the most efficient.

4.2. Heuristic Methods

Because the Branch-and-Cut approach is not able to solve instances of reasonable size (e.g., the Belgian network contains 52 vertices), we implemented the five heuristics proposed in Section 3. They all were coded in C++ and tested on a Pentium P90 under Linux. The randomly iterated dual greedy heuristic was run five times on each problem and stopped when 100 different solutions were generated.

Table 4 reports results obtained for the problems coming from real applications, while Table 5 reports a summary of results obtained for the randomly generated problems. Optimal solutions—when known—are denoted in bold.

The randomly iterated dual greedy heuristic outperforms the other ones, but is also very time consuming. It seems that this heuristic has a quite stable behaviour, because the costs of the solutions found over five runs are not too different.

Among the four constructive heuristics, the dual greedy one performs the best, while sometimes beaten by the ear-

Table 2. Results of the Branch-and-Cut algorithm for randomly generated instances (average over five instances).

			Solved to Optimality				Time Limit Reached			
V	E	K	INST	BBN	T	GR	INST	BBN	G	GR
10	34.6	400	5	70.00	13.20	12.7	0	—	—	—
10	40.0	450	5	75.20	14.80	12.3	0	—	—	—
10	43.2	500	5	71.00	15.80	11.8	0	—	—	—
10	44.4	550	5	30.60	6.80	8.5	0	—	—	—
20	96.5	250	4	2182.50	3078.50	14.7	1	6379.00	2.1	19.2
20	121.8	300	4	548.25	579.00	10.6	1	5293.00	4.6	19.3
20	149.6	350	5	874.00	2102.00	11.0	0	—	—	—
20	170.2	400	5	702.80	1356.80	10.0	0	—	—	—
30	226.2	250	0	—	—	—	5	2261.50	8.4	19.1
30	288.8	300	1	2461.00	16093.00	7.4	4	1581.00	3.9	12.9
30	345.8	350	2	1706.00	10726.00	6.1	3	1481.33	1.9	8.7
30	381.6	400	3	236.67	1801.33	4.8	2	1417.00	2.0	8.5
40	368.8	250	0	—	—	—	5	915.00	10.8	17.6
40	466.6	300	0	—	—	—	5	757.80	6.9	13.2
40	561.6	350	0	—	—	—	5	629.00	3.3	8.3
40	645.6	400	0	—	—	—	5	576.00	2.1	6.3
50	607.0	250	0	—	—	—	5	370.80	11.4	16.1
50	775.4	300	0	—	—	—	5	345.20	5.6	9.0
50	926.0	350	0	—	—	—	5	300.40	4.5	7.5
50	—	400	0	—	—	—	5	267.40	3.9	6.5

V = Number of vertices
E = Number of edges after preprocessing (average)
K = Bound on the mesh lengths
INST = Number of instances
BBN = Number of nodes explored in the Branch-and-Bound tree (average)
T = Time (in seconds) (average)
G = Remaining gap (in %)
GR = Gap at root node (in %)
Time limit = 20 000 seconds

Table 3. Lower bounds obtained with different subsets of inequalities.

Bound	C + VP	C+VP+M	C+VP+WP	C+VP+M+WP
150	1025	1092	1116	1165
200	1025	1060	1077	1106
250	1025	1048	1059	1076
300	1025	1041	1050	1064
350	1025	1033	1044	1051

C: Cut inequalities
VP: Vertex partition inequalities
M: Metric inequalities
WP: Weighted partition inequalities

inserting or the cut-in-two heuristic when the value of the bound on the mesh lengths is small. However, the cut-in-two heuristic is not always able to find a feasible solution when the bound is small. The path-following heuristic has only a theoretical interest—its worst case bound—but performs poorly in practice.

5. CONCLUSION

In this paper, we studied a problem arising from the need to design survivable telecommunication networks. Designing a two-connected network at minimum cost is a problem that was widely studied in the literature, and efficient methods for solving it were already available.

Unfortunately, it turned out that the optimal solution of this problem on the set of 52 zonal centers of the Belgian backbone network is a Hamiltonian cycle. Hence, any edge failure would require to reroute the flow that passed through that edge, using all the edges of the network—an obviously undesirable feature. This led us to examine a new model for limiting the region of influence of the traffic that is necessary to reroute: In addition to two-connectivity, we required that each edge belongs to at least one cycle whose length is bounded by a given constant. This problem is called the *Two-Connected Network with Bounded Meshes Problem.*

The problem was formulated as an integer linear program, and several classes of valid inequalities were presented, leading to a Branch-and-Cut Algorithm. Because we could solve

Table 4. Results of heuristics for instances coming from real applications.

V	K	Ear-Insert	Cut in Two	Path Follow	Dual Greedy	Randomly Iterated D.G.			Branch-and-cut	
						Best	Worst	Avg time	LB	UB
12	200	680	675	783	684	629	629	7	**622**	**622**
12	250	**541**	558	650	**541**	**541**	**541**	7	**541**	**541**
12	300	558	558	576	**541**	**541**	**541**	7	**541**	**541**
12	350	558	**521**	**521**	**521**	**521**	**521**	7	**521**	**521**
17	200	923	933	985	920	860	871	19	**834**	**834**
17	250	845	849	990	**789**	**789**	**789**	22	**789**	**789**
17	300	823	785	865	789	**726**	727	25	**726**	**726**
17	350	784	746	768	743	**725**	726	27	**725**	**725**
30	150	1304	1579(*)	1390	1323	1287	1302	114	1164	1268
30	200	1194	1169	1309	1157	1108	1132	206	1034	1095
30	250	1068	1003	1231	977	**935**	954	286	**935**	**935**
30	300	1042	979	1218	951	921	925	345	**898**	**898**
30	350	1014	950	1057	910	**874**	909	400	**874**	**874**
52	150	1522	1600	1618	1367	1367	1367	1727	1212	1343
52	200	1380	1368	1526	1274	1266	1266	3015	1143	1246
52	250	1307	1291	1429	1185	1185	1185	4426	1107	1181
52	300	1345	1221	1331	1164	1147	1162	5135	1084	1117
52	350	1328	1195	1342	1232	1182	1197	5737	1067	1085

(*) Infeasible solution.
CPU times are:
⩽1 second for all applications of split-in-two and path-following;
⩽4 seconds for ear-inserting;
⩽1 minute for dual greedy.

Table 5. Results of heuristics for randomly generated instances (average over five instances).

V	K	Ear-Insert	Cut in Two	Path Follow	Dual Greedy	Randomly Iterated D.G.			Branch-and-Cut	
						Best	Worst	Avg Time	LB	UB
10	400	1083	1257 (a)	1224	1159	**1069**	1071	3	**1069**	**1069**
10	450	1094	1156 (b)	1195	1097	**1007**	1047	3	**1007**	**1007**
10	500	1046	1162 (b)	1134	1061	**978**	979	3	**978**	**978**
10	550	1000	1061 (b)	1054	1006	932	948	4	**931**	**931**
20	250	1371	1557 (a)	1507	1366	1326	1330	27	1296	1303
20	300	1339	1521 (a)	1412	1324	1251	1263	36	1210	1222
20	350	1261	1320	1393	1278	1180	1193	47	**1158**	**1158**
20	400	1259	1207	1315	1238	1131	1151	57	**1125**	**1125**
30	250	1706	2079 (a)	1823	1798	1642	1668	169	1463	1600
30	300	1602	1730	1519	1674	1505	1521	241	1402	1451
30	350	1504	1752	1697	1494	1384	1408	313	1322	1338
30	400	1443	1436	1608	1406	1309	1337	363	1267	1278
40	250	1999	2313 (a)	2096	2018	1924	1936	577	1661	1864
40	300	1920	1988	2013	1847	1748	1769	807	1581	1700
40	350	1815	1796	1918	1695	1625	1636	1071	1521	1573
40	400	1730	1760	1880	1643	1571	1583	1332	1478	1510
50	250	2109	2170	2327	2061	1985	1997	1805	1712	1935
50	300	1976	1976	2261	1900	1796	1817	2602	1641	1738
50	350	1829	1849	2264	1822	1735	1756	3431	1599	1675
50	400	1888	1821	1997	1814	1706	1729	4199	1565	1628

(a) Infeasible solution found for two instances out of five.
(b) Infeasible solution found for one instance out of five.
CPU times are:
⩽1 second for all applications of split-in-two and path-following;
⩽4 seconds for ear-inserting;
⩽1 minute for dual greedy.

only small problems to optimality with the Branch-and-Cut algorithm, we developed a series of heuristics, among which the randomly iterated dual greedy one performs the best. This heuristic is very time-consuming and is only a first step in the development of more elaborate and effective local search methods. However, on a practical point of view, it was sufficient for Belgacom's instances.

ACKNOWLEDGMENTS

The authors thank A. Segalini for coding some of the heuristics, P. Värbrand for fruitful discussions at an early stage of this work, and three referees for their constructive comments and suggestions. This work was partially sponsored by contract ERB CHRXCT930087 of the Human Capital and Mobility Program of the EU. The work of the first author was sponsored by Belgacom. The work of the third author was sponsored by a grant "Research in Brussels" No. B6/12-93-028 and by a grant of the program Short-Term Mobility of the Italian National Research Council (CNR).

REFERENCES

Alevras, D., M. Grötschel, R. Wessäly. 1998. Cost-efficient network synthesis from leased lines. *Ann. Oper. Res.* **76** 1–20.

Bienstock, D., S. Chopra, O. Günlük, C-Y. Tsai. 1998. Minimum cost capacity installation for multicommodity network flows. *Math. Programming* **81**(2) 177–199.

Boyd, S., T. Hao. 1993. An integer polytope related to the design of survivable communication networks. *SIAM J. Discrete Math.* **6**(4) 612–630.

Christofides, N. 1976. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388. Grad. School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA.

Fortz, B., M. Labbé, F. Maffioli. 1997. Methods for designing reliable networks with bounded meshes. V. Ramaswami, P. Wirth, eds. *Teletraffic Contributions for the Information Age*, Volume 2a of *Teletraffic Science and Engineering*. Elsevier, Amsterdam, The Netherlands, 341–350.

Frederickson, G., J. Já Já. 1982. On the relationship between the biconnectivity augmentation and traveling salesman problems. *Theoretical Comput. Sci.* **19** 189–201.

Gomory, R., T. Hu 1961. Multi-terminal network flows. *SIAM J. Appl. Math.* **9** 551–570.

Grötschel, M., C. Monma. 1990. Integer polyhedra arising from certain design problems with connectivity constraints. *SIAM J. Discrete Math.* **3** 502–523.

——,——, M. Stoer. 1992. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Oper. Res.* **40**(2) 309–330.

——,——,——. 1995. *Design of Survivable Networks*. Volume 7 of *Handbooks in OR/MS*, Chapter 10, North-Holland, Amsterdam, The Netherlands, 617–672.

Maffioli, F., P. Värbrand. 1994. The constrained two-connected sub-graph problem. Technical Report 94.013, Politecnico di Milano, Italy.

Monma, C., B. Munson, W. Pulleyblank. 1990. Minimum-weight two-connected spanning networks. *Math. Programming* **46** 153–171.

——, D. Shallcross. 1989. Methods for designing communications networks with certain two-connected survivability constraints. *Oper. Res.* **37**(4) 531–541.

Steiglitz, K., P. Weiner, D. Kleitman. 1969. The design of minimum-cost survivable networks. *IEEE Trans. Circuit Theory* **CT-16** 455–460.

Stoer, M. 1992. *Design of Survivable Networks.* Volume 1531 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin.