



DIPARTIMENTO DI INFORMATICA  
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA  
UNIVERSITÀ DI ROMA

*The Robust Network Loading Problem with  
Dynamic Routing*

Sara Mattia

Technical Report n. 3, 2010

# The Robust Network Loading Problem with Dynamic Routing

Sara Mattia

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"  
Sapienza, Università di Roma

## Abstract

The Robust Network Loading Problem (RNL) can be stated as follows. Given a graph and a set of traffic matrices, install minimum cost integer capacities on the edges such that all the matrices can be routed non simultaneously on the network. The routing scheme is said to be dynamic if we can choose a (possibly) different routing for every matrix, it is called static if the routing must be the same for all the matrices. The flows are unsplitable if each point-to-point demand (commodity) must use a single path, they are splittable if the flow for every commodity can be splitted along several paths. In this paper we present the first exact approach for solving the RNL problem with splittable flows and dynamic routing under polyhedral uncertainty for the demands. A branch-and-cut algorithm based on the capacity formulation of the problem defined by metric inequalities is developed, and polyhedral results are given. The separation problem is formulated as a bilevel programming problem and a corresponding single level problem is derived. Computational results are presented.

keywords: network design, robustness, branch-and-cut

# 1 Introduction

The Network Loading Problem (NL) is a classic network design problem. It can be defined as follows. Given a graph and set of point-to-point traffic demands (commodities), choose minimum cost integer capacities for the edges such that all the demands can be routed simultaneously on the network. This problem has received a lot of attention in the literature, see for example [25], [5], [10], [3], [4] and references therein. The assumption that there is only one traffic matrix to be considered, and that this matrix is known in advance or can be reliably estimated, can be too restrictive. In fact the demands can be difficult to predict and they can change over time. Therefore, in order to obtain a more flexible and robust network, it is necessary to take into account the demand uncertainty in the design process. A way to do that is to consider, instead of a single traffic matrix, a set of traffic matrices to be served non simultaneously. This version of the problem is known as the Robust Network Loading Problem.

The set of traffic matrices can either be explicitly given, that is it consists of a list of matrices to be served, or it can be implicitly defined, which means that we have to serve all the matrices belonging to a given polyhedron [6]. One possible polyhedral representation for the demands is the *hose model* [18], [20]. In this model upper bounds on node traffic are given and all the matrices satisfying the bounds belong to the polyhedron. The hose polyhedron is *asymmetric* if for every node two bounds are given: one for the incoming traffic and one for the outgoing traffic. It is called *symmetric* if only one bound on the sum of the incoming and outgoing traffic is given. In the following we say *symmetric (asymmetric) demands* to denote the demands belonging to the symmetric (asymmetric) hose polyhedron. The routing scheme is *dynamic* if we can choose a (possibly) different routing for every matrix, it is *static* (or *oblivious*) if the routing must be the same for all the matrices. The flows are *unsplittable* (or *non-bifurcated*) if each commodity must use a single path, they are *splittable* (or *bifurcated*) if the flow for every commodity can be splitted along several paths.

For general demands, the RNL problem is *NP-hard* both for static routing and for dynamic routing, as it includes the Steiner Tree Problem as a special case. The problem of finding a Steiner Tree in an undirected graph  $G$  given the set of terminals  $T$ , can be transformed into a RNL problem defining a symmetric hose polyhedron as follows. Choose  $r \in T$  and set  $b_r = 1$ ,  $b_i = 1/|T - 1|$  for  $i \in T, i \neq r$ . Solving the RNL problem with the given node bounds  $b$  solves the Steiner Tree Problem. The same result can be obtained for directed graphs and asymmetric demands. We also note that all the matrices in the polyhedron defined above are dominated by  $\bar{d}_{ri} = 1/|T - 1|$ , for  $i \in T, i \neq r$ ,  $\bar{d}_{ij} = 0$  otherwise. A traffic matrix  $d_1$  is dominated by a traffic matrix  $d_2$  if every capacity vector that supports  $d_2$ , also supports  $d_1$  [28]. Therefore solving the RNL problem with node bounds  $b$  reduces to solve the NL problem, which is also *NP-hard*, with demands  $\bar{d}$ .

The importance of robustness in practical applications has been analyzed in [32], [7], and [9].

Two well-known problems that are closely related to the Robust Network Loading Problem are the Robust Network Design Problem (RND) and the Virtual Private Network Problem (VPN). The RND problem is a RNL problem where capacities for the edges are not restricted to be integer numbers. The VPN problem is a RND problem with static routing and unsplittable flows.

Most of the contributions in the literature deal with approximation results, while there are very few papers presenting exact approaches for the RNL problem. For approximation results on the VPN problem see for example [19] and references therein. For the RND problem see [11], [22], [21] and references therein. As for exact approaches, in [1] a compact flow formulation and a branch-and-cut-and-price algorithm for the VPN problem using the hose model for the demands are presented. In [2] a similar approach is used to derive a branch-and-cut algorithm for the RNL problem with splittable flows and static routing under polyhedral uncertainty. They also derive a capacity formulation for the problem not based on metric inequalities and consider the case where multiple facilities can be chosen for the edges. Polyhedral results are given when the demands are defined using the hose model. In [6] a branch-and-cut-and-price algorithm based on the solution of two auxiliary problems is presented to solve the VPN problem under polyhedral uncertainty.

As far as we know, there is no exact algorithm in the literature for the RNL problem with splittable flows and dynamic routing. In this paper we present the first exact approach for this problem. Our approach is a branch-and-cut algorithm based on the capacity formulation of the problem defined by metric inequalities.

Metric inequalities have been used extensively to derive capacity formulations for different network design problems. See for example [10], [31],[4], [8], [26], [14] and references therein.

Being more flexible, the dynamic routing allows to realize cheaper networks. Consider for example the problem of Figure 1.

The graph in Figure 1(a) is an instance of the RNL problem with splittable flows on a directed graph with asymmetric demands. Edge labels are capacity costs and node labels are traffic bounds. In Figure 1(b) we report the optimal solution of the problem using dynamic routing, edge labels are capacity values. In Figure 1(c) we represent the optimal solution if we restrict to static routing, edge labels are capacity values, the capacity is zero for the missing edges. The optimal solution with dynamic routing has a cost of 16 but it is not static feasible. The optimal solution with static routing has a cost of 17. Similar examples can be given for undirected graphs and symmetric demands. For the RND

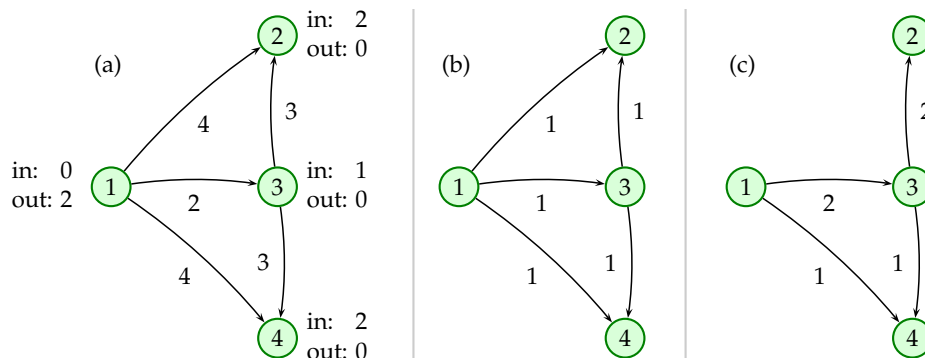


Figure 1: static vs dynamic

problem it has been proved that the gap between the optimal dynamic solution and the optimal static solution is  $O(\log n)$  where  $n$  is the number of nodes and that this bound is tight [21]. On the other hand the RNL problem with dynamic routing is, in a sense, more difficult to solve than the corresponding problem with static routing, see Section 2.

In the following, we refer to undirected graphs and to symmetric demands, but models, algorithms and results can be extended to directed graphs and to a general demand polyhedron. The rest of the paper is organized as follows. In Section 2 a mathematical model for the problem is presented. In Section 3 polyhedral results are given. In Section 4 a branch-and-cut approach based on the capacity formulation of the problem is described. The separation problem is formulated as a bilevel programming problem and a corresponding single level formulation is derived. The complexity of the separation problem is also discussed. In Section 5 computational results are presented. In Section 6 conclusions and open questions are discussed.

## 2 The model

Formally, the RNL problem can be defined as follows. Let  $G(V, E)$  be an undirected graph, where  $V$  is the set of the nodes to be connected and  $E$  is the set of the edges. Let  $D$  be the polyhedron representing the traffic matrices whose routing must be ensured and let  $d \in D$  be a feasible vector of demands (traffic matrix) to be served. The entry  $d_{ij} \geq 0$  represents the amount to be sent from source node  $i$  to destination node  $j$  ( $d_{ii} = 0$ ). We can group the demands by source, therefore for every node  $k \in V$  we define a commodity  $k$  having a single source and multiple destinations. We denote by  $K$  the set of commodities. Let  $c : E \rightarrow \mathbb{R}_+$  and  $b : V \rightarrow \mathbb{R}_+$  be functions defining per-unit capacity installation costs for the edges and traffic bounds for the

nodes. With the above definitions, the symmetric hose polyhedron is the set  $D = \{d \in \mathbb{R}_+ : d_{ii} = 0, \sum_{k \in K} d_{ki} + \sum_{t \in V} d_{it} \leq b_i, i \in V\}$ .

Let  $x_e$  be an integer variable representing the capacity installed on edge  $e \in E$  and let  $f_{ij}^k(d)$  and  $f_{ji}^k(d)$  be continuous variables representing the flow for commodity  $k$  in traffic matrix  $d$  directed from  $i$  to  $j$  and viceversa on edge  $e = (i, j) \in E$ . We denote by  $N(i) = \{j \in V : (i, j) \in E\}$  the neighborhood of node  $i$ . The problem can be formulated as follows.

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

$$\sum_{j \in N(i)} (f_{ij}^k(d) - f_{ji}^k(d)) = -d_{ki} \quad i \in V, k \in K, d \in D \quad (1)$$

$$\max_{d \in D} \left\{ \sum_{k \in K} (f_{ij}^k(d) + f_{ji}^k(d)) \right\} \leq x_e \quad e = (i, j) \in E \quad (2)$$

$$f \geq 0$$

$$x_e \in \mathbb{Z}_+$$

Constraints (1) are flow conservation constraints. Constraints (2) are capacity constraints. Since there is a flow vector for every  $d \in D$ , formulation (RFF) is non-compact and, as far as we know, there is no compact formulation in the literature for this problem.

If we consider a list of matrices instead of a polyhedron for the demands, the above formulation is compact as long as we have a polynomial number of traffic matrices to be taken into account. This is the case, for example, when we have a single traffic matrix (NL problem). Using the polyhedral representation, instead of considering all feasible traffic matrices, we could restrict our attention to the vertices of the demand polyhedron, but they can be exponentially many.

On the other hand, there exists a compact flow formulation for the RNL problem with static routing [1], [2]. Therefore, in a sense, the problem with static routing is more tractable than the one with dynamic routing.

Since flows are splittable and there are no flow costs, we can project out flow variables obtaining, by duality, the following capacity formulation.

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

$$\sum_{d \in D} \sum_{e \in E} \bar{x}_e \mu_e(d) \geq \sum_{d \in D} \sum_{k \in K} \sum_{i \in V} \ell_{ki}^{\mu(d)} d_{ki} \quad \mu \geq 0 \quad (3)$$

$$x_e \in \mathbb{Z}_+$$

Where  $\ell^{\mu(d)}$  are shortest path lengths using  $\mu(d)$  as weights. Constraints (3) are dominated by constraints (4) where only one  $\mu(d)$  is not identically zero. In fact (3) can be obtained as the sum of constraints (4).

$$\sum_{e \in E} \bar{x}_e \mu_e \geq \sum_{k \in K} \sum_{i \in V} \ell_{ki}^{\mu} d_{ki} \quad d \in D \quad (4)$$

Therefore the formulation above can be changed into the following one.

$$\begin{aligned} (RCF) \quad & \min \sum_{e \in E} c_e x_e \\ & \sum_{e \in E} \mu_e x_e \geq \max_{d \in D} \left\{ \sum_{k \in K} \sum_{i \in V} \ell_{ki}^{\mu} d_{ki} \right\} \quad \mu \geq 0 \\ & x_e \in \mathbb{Z}_+ \end{aligned} \quad (5)$$

Constraints (4), that can be re-written in a compact way as (5), are metric inequalities [27], [23]. Due to metric inequalities, formulation (RCF) is non-compact, even if we consider a single traffic matrix.

For static routing and splittable flows there exists a capacity formulation not based on metric inequalities [2]. In fact, since the requirement to have a static routing is a routing constraint itself, metric inequalities are valid but they do not provide a formulation for the problem.

Solving (RCF) we get optimal  $x$  values, but not the corresponding routing, that must be computed in a second time. Note that, even if the capacities are fixed, (RFF) remains non-compact, therefore even computing the routing for a given  $x$  is not an easy task.

In practical applications, the approach that can be used is the following:

1. compute optimal capacities  $x$ .
2. select an initial traffic matrix  $d \in D$  (the current demands)
3. compute a routing  $f$  solving (RFF) with  $x$  fixed, considering only the given traffic matrix. Since only one traffic matrix is considered, (RFF) is compact (the problem reduces to the NL problem)
4. as the demands change, keep using  $f$  as long as all the commodities can be routed.
5. when the demands are no longer supported, let  $\bar{d}$  be the traffic matrix that can not be routed using  $f$ . Then go to back to step 3

If demands changes do not come quickly and they are not dramatic,  $f$  must be recomputed a limited number of times.

### 3 Polyhedral results

All the polyhedral properties that have been proved for the NL problem, can easily be extended to the RNL problem. Let  $RNL(G, D)$  be the convex hull of integer feasible solutions of (RCF).

**THEOREM 3.1**  $RNL(D, G)$  is full-dimensional

**PROOF:** Let  $d^* = \lceil \max_{d \in D} \{ \sum_{k \in K} \sum_{i \in V} d_{ki} \} \rceil$  be the maximum total demand. For a given  $f \in E$ , let  $v^f \in \mathbb{R}^{|E|}$  be the vector:

$$v_e^f = \begin{cases} d^* + 1 & e = f \\ d^* & \text{otherwise} \end{cases}$$

Let  $v^* \in \mathbb{R}^{|E|}$  be the vector having all entries equal to  $d^*$ . Vectors  $v^*$  and  $v^f$  for all  $f \in E$  are  $|E| + 1$  affinely independent vectors in  $RNL(G, D)$ .  $\square$

**DEFINITION 3.2** An edge  $e = (i, j) \in E$  is a bridge if its removal disconnects at least one origin-destination pair in at least one traffic matrix  $d \in D$

**THEOREM 3.3** Inequality  $x_f \geq 0$  is facet-defining if and only if  $f$  is not a bridge

**PROOF:** If  $f$  is a bridge, every solution must have  $x_f \geq \lceil \max_{d \in D} \{ \sum_{st \in K_{disc}} d_{st} \} \rceil$ , where  $K_{disc}$  is the set of node pairs disconnected by the removal of  $f$ . Therefore  $x_f \geq 0$  can not be facet-defining. If  $f$  is not a bridge, define  $d^*$  as above. Let  $v^* \in \mathbb{R}^{|E|}$  be the vector having  $v_f^* = 0$  and  $v_e^* = d^*$  for  $e \neq f$ . Let  $v^{fh}$  be the vector defined as:

$$v_e^{fh} = \begin{cases} 0 & e = f \\ d^* + 1 & e = h \\ d^* & \text{otherwise} \end{cases}$$

Vectors  $v_f^*$  and  $v_{fh}^*$  for  $h \neq f$  are  $|E|$  affinely independent vectors satisfying  $x_f \geq 0$  with equality.  $\square$

Since  $RNL(G, D)$  is a dominant, that is it has the non-negative orthant as recession cone, the following condition holds.

**THEOREM 3.4** [12] Let  $a^T x \geq b$  be any valid inequality for  $RNL(G, D)$  then  $a \geq 0$

**DEFINITION 3.5** Given a set of nodes  $V$ , a function  $\ell : V \times V \rightarrow \mathbb{R}_+$  is a semi-metric on  $V$  if and only if:

1.  $\ell_{ii} = 0 \quad i \in V$
2.  $\ell_{ij} \geq 0 \quad i, j \in V$



3.  $\ell_{ij} = \ell_{ji} \quad i, j \in V$
4.  $\ell_{ij} \leq \ell_{ik} + \ell_{kj} \quad i, j, k \in V$

More precisely, if condition 2. holds with strict inequality we have a *metric*, otherwise we have a *semi-metric*. If symmetry condition 3. does not hold we have an oriented distance function or *quasi-(semi)-metric* [16],[17]. Since we are working on undirected graphs and we allow  $\ell$  values to be zero, when we say *metric*, we are technically speaking of a semi-metric. Let  $Met_V$  the cone generated by all non zero metrics.

DEFINITION 3.6 Let  $G(V, E)$  be a graph, a function  $\mu : E \rightarrow \mathbb{R}_+$  defines a metric on  $G$  if and only if:

1.  $\mu_e \geq 0 \quad e \in E$
2.  $\mu_e \leq \mu(P_e) \quad e \in E$

where  $\mu(P_e)$  is the length of shortest path between the endpoints of edge  $e$  using  $\mu$  as weights.

Let  $Met_G$  be the cone of all non zero metrics defined on  $G(V, E)$ . Given  $u, v \in V$  we denote by  $\ell_{uv}^\mu$  the length of the shortest path in  $G$  between  $u$  and  $v$  using  $\mu$  as weights,  $\ell_{ii}^\mu = 0$  for all  $i$ . If  $(u, v) \in E$  then, by definition,  $\ell_{uv}^\mu = \mu_{uv}$ . In this way a metric  $\mu \in Met_G$  can be extended to a metric  $\ell^\mu \in Met_V$ . If  $G$  is directed we get a quasi-(semi)-metric, otherwise we get a (semi)-metric.

THEOREM 3.7 If  $a^T x \geq b$  is valid for  $RNL(G, D)$  then there exists a metric  $\mu \in Met_G$  such that  $\mu^T x \geq b$  is still valid and  $\mu_e \leq a_e$  for all  $e \in E$ .

PROOF: The same argument of the proof of theorem 2.1 in [4] can be used. Let  $a^T x \geq b$  be a valid inequality. Define  $\mu$  as the metric  $\mu_e = a(P_e)$  for all  $e \in E$ , where  $a(P_e)$  is the length of the shortest path between the endpoints of  $e$  using  $a$  as weights. If  $a$  is not a metric, let  $f \in E$  an edge with the property that  $a_f > \mu_f$  and assume, without loss of generality, that there is a unique edge with this property. Suppose that  $\mu^T x \geq b$  is not valid and let  $\bar{x}$  be a feasible solution with the property that  $\mu^T \bar{x} < b$ . Define a new solution  $\hat{x}$  as follows:

$$\hat{x}_e = \begin{cases} \bar{x}_e & e \neq f, e \notin P_f \\ \bar{x}_e + \bar{x}_f & e \in P_f \\ 0 & e = f \end{cases}$$

Solution  $\hat{x}$  is feasible and  $a^T \hat{x} = \mu^T \bar{x} < b$ , therefore  $a^T x \geq b$  is not valid and we get a contradiction.  $\square$

DEFINITION 3.8 Given a metric  $\mu \in Met_G$ , let  $R_\mu = \min \{ \mu^T x : x \in RNL(G, D) \}$

THEOREM 3.9 Every constraint  $\mu^T x \geq b$  is dominated by  $\mu^T x \geq R_\mu$ .

Inequalities  $\mu^T x \geq R_\mu$  are called *tight metric inequalities*. All facet-defining inequalities are tight metric inequalities.

**THEOREM 3.10** *If  $\mu$  is an integer valued extreme ray of the metric cone having greatest common divisor equal to one, then  $R_\mu = \lceil \max_{d \in D} \left\{ \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki} \right\} \rceil$ .*

**PROOF:** The same argument of the proof of theorem 2.4 in [4] can be used. Let  $P = \{x \in \mathbb{R}^n : Ax \geq b\}$  be a rational polyhedron and let  $\gamma^T x \geq \delta$  be a valid inequality, then  $\gamma^T x \geq \delta$  can be obtained by Chvatal-Gomory derivation from  $Ax \geq b$ . In our case  $P$  is described by metric inequalities therefore  $\gamma$  will be a metric too. Let  $\mu^T x \geq R_\mu$  be a valid inequality. If  $\mu$  is an extreme ray of the metric cone, it can not be obtained as conic combination of two or more metrics and, if the greatest common divisor is equal to one, the corresponding inequality can not be derived by Chvatal-Gomory procedure from a single inequality. Therefore the right-hand-side value can not improve over Chvatal-Gomory iterations and therefore  $R_\mu = \lceil \max_{d \in D} \left\{ \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki} \right\} \rceil$ .  $\square$

As for the NL problem, a special class of valid inequalities are the *cut inequalities*. A cut is a partition of the nodes  $V$  into two subsets  $S$  and  $V - S$ . Let  $\delta(S) \subseteq E$  be the set of edges having endpoints in different sets of the partition. A cut induces a metric  $\mu$  having  $\mu_e = 1$  for  $e \in \delta(S)$  and  $\mu_e = 0$  otherwise. Cut metrics are the only  $\{0, 1\}$  extreme metrics for undirected graphs [16].

**LEMMA 3.11** *Given a partition of the nodes into  $S$  and  $V - S$  and the corresponding cut metric  $\mu$ , then  $R_\mu = \max_{d \in D} \left\{ \sum_{i \in S, j \in V-S} (d_{ij} + d_{ji}) \right\}$*

**LEMMA 3.12** *If  $D$  is the symmetric hose polyhedron, then*

$$R_\mu = \min \left\{ \sum_{i \in S} b_i, \sum_{i \in V-S} b_i \right\}$$

**LEMMA 3.13** *If  $D$  is the asymmetric hose polyhedron, then*

$$R_\mu = \max \left\{ \min \left\{ \sum_{i \in S} b_i^{out}, \sum_{i \in V-S} b_i^{in} \right\}, \min \left\{ \sum_{i \in S} b_i^{in}, \sum_{i \in V-S} b_i^{out} \right\} \right\}$$

The conditions to be satisfied for a cut inequality to be facet-defining are the same of the standard NL problem [25], [5].

**THEOREM 3.14** *Let  $\mu$  be a cut metric derived by the partition into  $S$  and  $V - S$ , then the corresponding tight metric inequality is facet-defining if and only if the well-known following conditions are satisfied:*

1.  $S$  and  $V - S$  are connected
2.  $R_\mu > 0$

PROOF: If  $R_\mu = 0$ , cut inequality is dominated by non negativity constraints. Suppose that  $S$  is not connected, then it can be partitioned into connected components  $S_1, \dots, S_p$ . Cut inequalities corresponding to the connected components dominate the inequality corresponding to  $S$ . The same if  $V - S$  is not connected. Suppose now that the two conditions hold. Let  $d^*$  be the maximum total demand defined as before. For every edge  $f \in \delta(S)$  define the vector  $v^f \in \mathbb{R}^{|E|}$  as:

$$v_e^f = \begin{cases} R_\mu & e = f \\ 0 & e \in \delta(S), e \neq f \\ d^* & e \notin \delta(S) \end{cases}$$

Choose now an edge  $h \in \delta(S)$  and for every edge  $r \notin \delta(S)$ , let  $v^r$  be the vector:

$$v_e^r = \begin{cases} R_\mu & e = h \\ 0 & e \in \delta(S), e \neq h \\ d^* + 1 & e = r \\ d^* & e \notin \delta(S), e \neq r \end{cases}$$

Vectors  $v^f$  for  $f \in \delta(S)$  and  $v^r$  for  $r \notin \delta(S)$  are  $|E|$  affinely independent vectors satisfying  $\mu^T x \geq R_\mu$  with equality. □

## 4 The Algorithm

In this section a branch-and-cut approach for solving the problem is presented. Separation and heuristic techniques to be used within a branch-and-cut framework are given.

A branch-and-cut algorithm works as follows [30]. Starting from an initial formulation including only a reduced number of inequalities, the relaxed problem is solved. A separation routine is used to find a constraint that is violated by the current solution, if any. If a violated inequality is found, the inequality is added to the current formulation and the problem is solved again, until a feasible solution is found. If the solution is fractional, then a branching step is performed.

Given a polyhedron  $P$  and a vector  $\bar{x}$ , the separation problem can be stated as follows.

**DEFINITION 4.1** *Find, if any, an inequality  $a^T x \geq b$  which is valid for  $P$  but it is violated by  $\bar{x}$ , or conclude that  $x \in P$*

We start with an initial formulation including only single-node cut inequalities. The quality of the violated inequality to be found by the separation procedure is crucial to obtain good branch-and-cut algorithms. For this problem the best thing to do would be separating tight metric inequalities, but this is a very

difficult task. For the NL problem, finding a violated tight metric inequality is difficult, since even computing  $R_\mu$  for a given  $\mu$  is *NP-hard*. On the other hand separating metric inequalities is easy, since it can be done solving an LP problem. For the RNL problem even separating simple metric inequalities (5) is not easy.

#### 4.1 Separation of metric inequalities

Formally, given a solution  $\bar{x}$ , the separation problem for metric inequalities can be defined as follows.

DEFINITION 4.2 *Find either an inequality:*

$$\sum_{e \in E} \mu_e x_e \geq \max_{d \in D} \left\{ \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki} \right\} \quad \mu \in \text{Met}_G$$

*violated by  $\bar{x}$ , or conclude that none exists.*

The separation problems can be written as a bilevel programming problem. A bilevel programming problem models a hierarchical decision process [13] [15]. The variables of the problem are partitioned into a set of upper-level variables and a set of lower-level variables. We can suppose to have two players, one (leader) controlling the upper-level variables, that are fixed first solving the master problem, the other (follower) controlling the lower-level variables, that are fixed by solving a subproblem that depends on the value of the upper-level variables. Many separation problems can be reformulated using bilevel programming, see for example [24].

The separation problem for metric inequalities can be formulated as follows.

$$(sep_{max}) \quad \min \sum_{e \in E} \bar{x}_e \mu_e - \beta$$

$$\ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \quad (6)$$

$$\ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \quad (7)$$

$$\sum_{e \in E} \mu_e = 1 \quad (8)$$

$$\mu \geq 0, \ell \text{ free}$$

$$\beta = \max \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki}$$

$$(\varphi_i) \sum_{k \in K} d_{ki} + \sum_{t \in V} d_{it} \leq b_i \quad i \in V \quad (9)$$

$$d \geq 0, d_{ii} = 0$$

The master problem controls variables  $\mu$  and  $\ell$  (upper-level variables). Constraints (6)-(7) make sure that  $\ell$  are shortest-path distances using  $\mu$  as weights.

Since  $(sep_{max})$  is a minimization problem, it is not necessary to include additional constraints to guarantee that  $\mu$  is a metric. Constraint (8) is a normalization constraint to avoid unboundedness. The subproblem depends on the choice of  $\ell$  in the master problem and it controls variables  $d$  (lower-level variables). Given  $\mu$  (and the corresponding  $\ell$ ), the purpose of the subproblem is to select the traffic matrix  $d \in D$  corresponding to the maximum violation. Constraints (9) define the symmetric hose polyhedron.

Since we are minimizing, the max in the subproblem is redundant and  $(sep_{max})$  can be transformed into the single-level problem below.

$$\begin{aligned}
(sep_Q) \quad & \min \sum_{e \in E} \bar{x}_e \mu_e - \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_i^k \\
& \ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \sum_{e \in E} \mu_e = 1 \\
& \sum_{k \in K} d_{ki} + \sum_{t \in V} d_{it} \leq b_i \quad i \in V \\
& \mu, d \geq 0, d_{ii} = 0, \ell \text{ free}
\end{aligned}$$

We have a quadratic objective function to be optimized over a polyhedron. Unfortunately the quadratic matrix is neither positive- nor negative-semidefinite therefore the problem is non-convex. On the other hand, since the subproblem of  $(sep_{max})$  is feasible and bounded, it can be replaced by its dual, obtaining problem  $(sep_{min})$ , which is equivalent to  $(sep_{max})$ .

$$\begin{aligned}
(sep_{min}) \quad & \min \sum_{e \in E} \bar{x}_e \mu_e - \beta \\
& \ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \sum_{e \in E} \mu_e = 1 \\
& \mu \geq 0, \ell \text{ free} \\
& \beta = \min \sum_{i \in V} \varphi_i b_i \\
& (d_{ki}) \varphi_i + \varphi_k \geq \ell_{ki}^\mu \quad k \in K, i \in V \\
& \varphi \geq 0
\end{aligned}$$

From the duality theory, given a primal problem  $(P)$  and its dual  $(D)$ , two solutions that are feasible for  $(P)$  and  $(D)$  are optimal if and only if they respect the so-called complementarity conditions. Moreover the two solutions have

the same objective value. Therefore ( $sep_{min}$ ) can be rewritten as a single level problem in the following way.

$$\begin{aligned}
(sep_{metric}) \quad & \min \sum_{e \in E} \bar{x}_e \mu_e - \sum_{i \in V} \varphi_i b_i \\
& \ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \sum_{e \in E} \mu_e = 1 \\
& \sum_{k \in K} d_{ki} + \sum_{t \in V} d_{it} \leq b_i \quad i \in V \tag{10} \\
& \varphi_i + \varphi_k \geq \ell_{ki}^\mu \quad k \in K, i \in V \tag{11} \\
& d_{ki}(\varphi_i + \varphi_k - \ell_{ki}^\mu) = 0 \quad k \in K, i \in V \tag{12} \\
& \varphi_i(b_i - \sum_{k \in K} d_{ki} - \sum_{t \in V} d_{it}) = 0 \quad i \in V \tag{13} \\
& \mu, \varphi, d \geq 0, d_{ii} = 0, \ell \text{ free}
\end{aligned}$$

We replaced  $\beta$  by the dual objective function (which is linear), but in this case we are not allowed to simply eliminate the min in the subproblem. In order to guarantee the optimality of the subproblem it is necessary to ensure primal and dual feasibility (constraints (10)-(11)) and to enforce complementarity conditions (constraints (12)-(13)).

Complementarity conditions can be stated as follows: each non-zero primal variable must correspond to a dual constraint satisfied with equality, and each non-zero dual variable must correspond to a primal constraint satisfied with equality. Therefore the problem can be linearized adding additional variables and constraints. Let  $w_{ki}$  be a binary variable whose value is one if  $d_{ki} > 0$  and zero otherwise, and let  $y_i$  be a binary variable whose value is one if  $\varphi_i > 0$  and zero otherwise. Constraints (12) and (13) can be replaced by:

$$(CC) \quad w_{ki} \leq M d_{ki} \quad k \in K, i \in V \tag{14}$$

$$d_{ki} \leq M w_{ki} \quad k \in K, i \in V \tag{15}$$

$$y_i \leq M \varphi_i \quad i \in V \tag{16}$$

$$\varphi_i \leq M y_i \quad i \in V \tag{17}$$

$$\varphi_i + \varphi_k - \ell_{ki}^\mu \leq M(1 - w_{ki}) \quad k \in K, i \in V \tag{18}$$

$$b_i - \sum_{k \in K} d_{ki} - \sum_{t \in V} d_{it} \leq M(1 - y_i) \quad i \in V \tag{19}$$

Constraints (14)-(17) are variable upper bound constraints. Constraints (18) and (19) together with primal and dual feasibility, enforce complementarity

conditions.

Therefore the separation problem for metric inequalities becomes the MIP problem below.

$$\begin{aligned}
(sep_m) \quad & \min \sum_{e \in E} \bar{x}_e \mu_e - \sum_{i \in V} \varphi_i b_i \\
& \ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\
& \sum_{e \in E} \mu_e = 1 \\
& \varphi_i + \varphi_k \geq \ell_{ki}^\mu \quad k \in K, i \in V \\
& \sum_{k \in K} d_{ki} + \sum_{t \in V} d_{it} \leq b_i \quad i \in V \\
& w_{ki} \leq M d_{ki} \quad k \in K, i \in V \\
& d_{ki} \leq M w_{ki} \quad k \in K, i \in V \\
& y_i \leq M \varphi_i \quad i \in V \\
& \varphi_i \leq M y_i \quad i \in V \\
& \varphi_i + \varphi_k - \ell_{ki}^\mu \leq M(1 - w_{ki}) \quad k \in K, i \in V \\
& b_i - \sum_{k \in K} d_{ki} - \sum_{t \in V} d_{it} \leq M(1 - y_i) \quad i \in V \\
& \mu, \varphi, d \geq 0, d_{ii} = 0, \ell \text{ free}, w_{ki}, y_i \in \{0, 1\}
\end{aligned}$$

One could ask whether it is possible to write the separation problem for metric inequalities as an LP, as it happens when we have a single traffic matrix (NL problem). The existence of an LP separation problem for metric inequalities is closely related to the existence of a compact flow formulation for the problem. In fact, if a compact LP separation problem exists, by duality we can derive a compact flow formulation.

Given a vector  $\bar{x}$ , checking if a violated metric inequality exists, corresponds to verify if  $\bar{x}$  is feasible for the RNL problem when the integrality requirements are not considered. This means that we want to test if  $\bar{x}$  is feasible for the corresponding RND problem. The RND problem has been proved to be *coNP-hard* both for undirected graphs [11] and for directed graphs [22], if asymmetric demands are considered. It follows that it is (probably) not possible to write an LP separation problem for metric inequalities with asymmetric demands. For the same reason there is (probably) no compact flow formulation for the problem. On the other hand, the complexity of the RND problem for symmetric demands is still unknown.

## 4.2 Separation of rounded metric inequalities

If  $\mu$  is integral, the right-hand-side of the metric inequality can be rounded to the upper nearest integer obtaining a so-called rounded metric. If  $\mu$  and  $\ell$  are  $\{0, 1\}$ -vectors, it becomes a  $\{0, 1\}$ -rounded metric inequality [4].  $\{0, 1\}$ -rounded metric inequalities include cut inequalities, which are facet-defining. The separation problem for  $\{0, 1\}$ -rounded metric inequalities can be stated as follows.

DEFINITION 4.3 *Find either an inequality:*

$$\sum_{e \in E} \mu_e x_e \geq \max_{d \in D} \left\{ \left[ \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki} \right] \right\} \quad \mu_e, \ell_{ki} \in \{0, 1\}$$

violated by  $\bar{x}$ , or conclude that none exists.

From  $(sep_Q)$  we can derive the separation problem below.

$$\begin{aligned} (sep_{rmQ}) \quad & \min \sum_{e \in E} \bar{x}_e \mu_e - z \\ & \ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\ & \ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\ & \sum_{k \in K} d_{ki} + \sum_{t \in V} d_{it} \leq b_i \quad i \in V \\ & z \leq \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki} + 1 - \varepsilon \\ & \varphi, d \geq 0, d_{ii} = 0, \ell_{ki}^\mu, \mu_e \in \{0, 1\} \end{aligned} \tag{20}$$

Variable  $z$  represents the amount  $\left[ \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki} \right]$ . This is ensured by (20) and by  $z \geq \sum_{k \in K} \sum_{i \in V} \ell_{ki}^\mu d_{ki}$ , that has been omitted because we are solving a minimization problem. Since in this case  $\ell$  are binary variables, the problem can be linearized adding constraints  $d_{ki} \leq M \ell_{ki}^\mu$  and replacing constraint (20) by  $z \leq \sum_{k \in K} \sum_{i \in V} d_{ki} + 1 - \varepsilon$ , thus obtaining the problem below.

$$\begin{aligned} (sep_{rm}) \quad & \min \sum_{e \in E} \bar{x}_e \mu_e - z \\ & \ell_{kj}^\mu \leq \ell_{ki}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\ & \ell_{ki}^\mu \leq \ell_{kj}^\mu + \mu_e \quad k \in K, e = (i, j) \in E \\ & \sum_{k \in K} d_i^k + \sum_{t \in V} d_t^i \leq b_i \quad i \in V \\ & d_{ki} \leq M \ell_{ki}^\mu \quad k \in K, i \in V \end{aligned}$$



$$z \leq \sum_{k \in K} \sum_{i \in V} d_{ki} + 1 - \varepsilon$$

$$\varphi, d \geq 0, d_{ii} = 0, \ell_{ki}^\mu, \mu_e \in \{0, 1\}$$

### 4.3 Branch-and-bound and initial heuristics

A simple branch-and-bound heuristic is based on the following idea. Given a solution that is feasible but fractional, if each component is rounded to the upper-nearest integer the resulting capacity vector is integer and feasible for the problem. Therefore every time the heuristic is performed, we get the current fractional solution and we round each component obtaining an integer solution  $\hat{x}$ . We test  $\hat{x}$  for feasibility applying the separation routines of Section 4.1. If it is feasible we get a solution, otherwise no solution is returned.

An initial solution can be found as follows. The minimum spanning tree  $T$  is computed. Let  $\{S_e, V - S_e\}$  be the partition of the nodes when edge  $e \in T$  is removed. A feasible solution is  $x_e = 0$  for  $e \notin T$ ,  $x_e = \min \{\sum_{i \in S_e} b_i, \sum_{i \in V - S_e} b_i\}$  for  $e \in T$ . We call this algorithm *spanning tree heuristic*.

A first feasible solution can also be computed in the following way. Choose a traffic matrix  $d^*$ , not necessarily feasible, but such that it dominates all the matrices in  $D$  and solve the NL problem for  $d^*$ . Since  $d^*$  dominates all  $d \in D$ , every capacity vector that supports  $d^*$  also supports every  $d \in D$  [28]. If  $D$  is the symmetric hose polyhedron, a simple dominating matrix  $d^*$  can be computed setting  $d_{ij}^* = \min\{b_i, b_j\}$ . If  $d^* \in D$  then solving RNL reduces to solve NL with demands  $d^*$ . Unfortunately, most of the times matrix  $d^*$  is not feasible. The problem is solved using the flow formulation. We call this algorithm *dominating matrix heuristic*.

Since all solutions that are feasible for the problem with static routing are also feasible for the problem with dynamic routing, any heuristic (or exact algorithm) for solving the RNL with static routing can be used to find an initial feasible solution for the RNL with dynamic routing. In particular, the optimal solution of the problem with static routing obtained solving the compact flow formulation presented in [1][2] can be used. We call this algorithm *static routing heuristic*.

Using the compact formulation for the problem with static routing, it is also possible to obtain an initial feasible solution for the problem with dynamic routing solving the LP relaxation and then rounding each component of  $x$  to the upper-nearest integer. We call this algorithm *static routing LP heuristic*.

## 5 Computational results

The proposed approach has been implemented in C++, using CPLEX 12.1 as solver. Experiments have been made on a laptop having a 2.20 GHz Intel Core 2 Duo processor, with 4Gb RAM using 72 instances derived from SNDlib networks [29]. The size of the instances is reported in Table 1. Column *problem* is the instance name, columns  $|V|$  and  $|E|$  are the number of nodes and edges of the problem, column  $|d|$  is the number of demands that are non-zero in at least one traffic matrix. We tested our algorithm on undirected graphs using both symmetric (US instances) and asymmetric (UA instances) demands. The instances can be obtained sending request to the author.

In Table 2 we report the values of the feasible solutions found by the spanning tree heuristic (ST-H), the dominating matrix heuristic (DM-H), and the static routing LP heuristic (SR-LP-H) respectively. For all the algorithms a time limit of 600 seconds (10 minutes) has been considered. Column *problem* is the instance name. For the static routing LP heuristic, columns *obj* and *time* are objective values and solution times (in seconds). For the spanning tree heuristic, column *obj* reports the solution values. Solution times have been omitted since it runs in almost no computational time on the considered instances. For the dominating matrix heuristic, if the problem has been solved to optimality within the given time limit, *obj* is the optimal solution and *time/gap* is the solution time, if not, *obj* is the best UB found and *time/gap* is the corresponding gap. As one could expect, the static routing LP heuristic produces the best solutions. The performance of the dominating matrix heuristic is very poor, also because the choice of the traffic matrix to be used is not very accurate. The spanning tree heuristic is the fastest algorithm and it performs better than the dominating matrix heuristic.

In Tables 3 the results obtained solving the problem with static routing are reported. For all the problems a time limit of 5400 seconds (1h30min) is set. Column *problem* is the instance name, columns *UB* and *LB* are the best upper and lower bounds produced within the given time limit, and column *gap* is the corresponding gap. Column *time* is the solution time (in seconds).

In Table 4 we report computational results for the branch-and-cut algorithm.. Column *problem* is the instance name, columns *UB* and *LB* are the best upper and lower bounds produced within the given time limit and column *gap* is the corresponding gap. Columns *cuts* and *time* are the number of cuts added and the solution time (in seconds). Since solving to optimality ( $sep_m$ ) is expensive, the following strategy is used. First we use a simple heuristic to find violated cut inequalities. The nodes are randomly partitioned and the corresponding cut inequality is tested for violation. If it is violated, we add it, otherwise we look for a violated  $\{0, 1\}$ -rounded metric inequality, since ( $sep_{rm}$ ) is computationally easier to solve than ( $sep_m$ ). If there is a violated  $\{0, 1\}$ -rounded metric inequality we stop, otherwise we look for a violated metric inequality. As ini-

tial heuristic, we use the static routing heuristic with a time limit of 50 seconds. If no feasible solution is produced within the given time limit, the static routing LP heuristic is used. We set a time limit 5400 seconds (1h30min) for all the problems.

Even if we are not able to solve all the problems to optimality, the results are encouraging. The routine for separating  $\{0, 1\}$ -rounded metric inequalities is reasonably fast, even on large instances. The bottleneck of the algorithm is the separation of metric inequalities, that is very time consuming and when the size of the instance increases, it becomes quickly impractical. Therefore, in order to solve large instances, the separation of metric inequalities must be improved.

Comparing the results for static routing (Table 3) and for dynamic routing (Table 4) it is easy to see that the problem with static routing can be solved faster than the one with dynamic routing, but most of the times the solutions are more expensive.

<i>problem</i>	<i> V </i>	<i> E </i>	<i> d </i>	<i>problem</i>	<i> V </i>	<i> E </i>	<i> d </i>
atlanta.US50-1	15	22	72	atlanta.UA50-1	15	22	59
atlanta.US50-7	15	22	72	atlanta.UA50-7	15	22	75
atlanta.US50-12	15	22	72	atlanta.UA50-12	15	22	86
atlanta.US50-17	15	22	56	atlanta.UA50-17	15	22	75
dfn-bwin.US50-1	10	45	20	dfn-bwin.UA50-1	10	45	27
dfn-bwin.US50-7	10	45	30	dfn-bwin.UA50-7	10	45	33
dfn-bwin.US50-12	10	45	42	dfn-bwin.UA50-12	10	45	26
dfn-bwin.US50-17	10	45	30	dfn-bwin.UA50-17	10	45	27
dfn-gwin.US50-1	11	47	42	dfn-gwin.UA50-1	11	47	32
dfn-gwin.US50-7	11	47	42	dfn-gwin.UA50-7	11	47	33
dfn-gwin.US50-12	11	47	42	dfn-gwin.UA50-12	11	47	38
dfn-gwin.US50-17	11	47	42	dfn-gwin.UA50-17	11	47	39
di-yuan.US50-1	11	42	42	di-yuan.UA50-1	11	42	21
di-yuan.US50-7	11	42	42	di-yuan.UA50-7	11	42	28
di-yuan.US50-12	11	42	42	di-yuan.UA50-12	11	42	26
di-yuan.US50-17	11	42	42	di-yuan.UA50-17	11	42	18
nobel-us.US50-1	14	21	56	nobel-us.UA50-1	14	21	51
nobel-us.US50-7	14	21	56	nobel-us.UA50-7	14	21	53
nobel-us.US50-12	14	21	56	nobel-us.UA50-12	14	21	59
nobel-us.US50-17	14	21	42	nobel-us.UA50-17	14	21	60
pdh.US50-1	11	34	42	pdh.UA50-1	11	34	21
pdh.US50-7	11	34	42	pdh.UA50-7	11	34	27
pdh.US50-12	11	34	42	pdh.UA50-12	11	34	38
pdh.US50-17	11	34	42	pdh.UA50-17	11	34	27
polska.US50-1	12	18	56	polska.UA50-1	12	18	51
polska.US50-7	12	18	30	polska.UA50-7	12	18	27
polska.US50-12	12	18	42	polska.UA50-12	12	18	38
polska.US50-17	12	18	42	polska.UA50-17	12	18	27
nobel-germany.US50-1	17	26	100	nobel-germany.UA50-1	17	26	83
nobel-germany.US50-7	17	26	90	nobel-germany.UA50-7	17	26	85
nobel-germany.US50-12	17	26	90	nobel-germany.UA50-12	17	26	87
nobel-germany.US50-17	17	26	90	nobel-germany.UA50-17	17	26	74
norway.US50-1	27	51	182	norway.UA50-1	27	51	26
norway.US50-12	27	51	210	norway.UA50-7	27	51	45
norway.US50-7	27	51	156	norway.UA50-12	27	51	52
norway.US50-17	27	51	240	norway.UA50-17	27	51	76

Table 1: instances

## 6 Conclusions

In this paper we presented the first exact branch-and-cut approach for solving the RNL with splittable flows and dynamic routing under hose polyhedral uncertainty for the demands. We use a capacity formulation of the problem based on metric inequalities. Polyhedral results are given. The separation problem for finding violated metric inequalities is formulated as a bilevel programming problem and a corresponding single level problem is derived. The algorithm has been implemented in C++, using CPLEX 12.1 and computational results have been presented. A possible scheme for using such algorithm in real-life applications is also given.

Different questions are still open:

1. how difficult is the RND problem with symmetric demands? If the problem is discovered to be easy, then a better algorithm can be derived for such case
2. is it possible to improve the formulation of the separation problem for metric inequalities? If so, then larger instances could be solved using the proposed algorithm
3. is it possible to separate tight metric inequalities? The separation of inequalities stronger than rounded metrics, not necessarily tight metrics, is an open issue, both for the NL problem and for the RNL problem. To our knowledge, no algorithm, not even heuristic, is known for separating inequalities stronger than rounded metrics in the general case

problem	SR-LP-H		DM-H		ST-H		SR-LP-H		DM-H		ST-H		DM-H	
	obj	time	obj	time/gap	obj	time	obj	time	obj	time/gap	obj	time	obj	time/gap
atlanta.US50-1	5615	1.95	18861	0.858	8512	1.95	8296	2.184	8939	29331	1.82	8939	29331	1.82
atlanta.US50-7	5366	2.01	24533	10.904	11407	2.01	10148	2.418	12606	41596	7.39	12606	41596	7.39
atlanta.US50-12	6565	1.87	19751	5.928	9945	1.87	11520	2.465	13114	50464	278.60	13114	50464	278.60
atlanta.US50-17	3830	1.87	13814	0.14	8294	1.87	8474	2.309	10545	35622	9.36	10545	35622	9.36
dfn-bwin.US50-1	664	1.46	1377	0.577	1377	1.46	1387	1.716	3234	3115	2.07	3234	3115	2.07
dfn-bwin.US50-7	739	1.60	1799	1.794	1944	1.60	1551	1.841	3154	3594	199.68	3154	3594	199.68
dfn-bwin.US50-12	1059	1.59	3475	12.683	3475	1.59	1385	1.747	2669	2694	12.37	2669	2694	12.37
dfn-bwin.US50-17	897	1.49	2047	1.279	3881	1.49	1552	1.747	2910	3122	17.56	2910	3122	17.56
dfn-gwin.US50-1	222	2.21	611	10.811	445	2.21	294	2.496	456	682	14.72	456	682	14.72
dfn-gwin.US50-7	281	2.26	975	35.553	511	2.26	259	2.996	359	725	12.51	359	725	12.51
dfn-gwin.US50-12	152	2.26	338	8.112	338	2.26	204	2.558	269	750	49.77	269	750	49.77
dfn-gwin.US50-17	265	2.24	517	106.251	517	2.24	284	2.87	406	790	33.19	406	790	33.19
di-yuan.US50-1	1426	2.13	4307	235.382	1963	2.13	1180	2.012	1307	1855	0.48	1307	1855	0.48
di-yuan.US50-7	1371	1.90	4028	8.222	1788	1.90	1424	2.028	1086	2955	5.19	1086	2955	5.19
di-yuan.US50-12	1327	1.91	3729	5.709	1446	1.91	1405	2.075	1500	3158	38.95	1500	3158	38.95
di-yuan.US50-17	1676	1.93	3962	0.156	2404	1.93	835	1.934	708	1789	3.88	708	1789	3.88
nobel-us.US50-1	5395	1.43	23319	5.428	9622	1.43	9451	1.622	13260	24576	0.73	13260	24576	0.73
nobel-us.US50-7	6209	1.43	22164	4.805	7751	1.43	11300	1.669	10634	36957	2.05	10634	36957	2.05
nobel-us.US50-12	6521	1.40	28976	1.295	7815	1.40	10379	1.653	8624	41102	6.78	8624	41102	6.78
nobel-us.US50-17	23232	1.37	69890	7.581	31333	1.37	27958	1.7	36442	84348	5.70	36442	84348	5.70
pdh.US50-1	2585	1.45	7229	70.324	5127	1.45	2801	1.42	2734	5035	64.03	2734	5035	64.03
pdh.US50-7	3005	1.43	8006	2.996	5061	1.43	2713	1.498	2827	6444	2.54	2827	6444	2.54
pdh.US50-12	3144	1.45	7555	250.505	3945	1.45	3055	1.654	3582	8408	38.68	3582	8408	38.68
pdh.US50-17	3313	1.41	7602	6.24	4765	1.41	2554	1.482	3121	5859	84.61	3121	5859	84.61
polska.US50-1	16009	0.78	60119	16.661	20904	0.78	23339	0.936	28519	65159	1.35	28519	65159	1.35
polska.US50-7	6283	0.71	19307	2.87	7519	0.71	8690	0.796	10065	19966	1.10	10065	19966	1.10
polska.US50-12	11842	0.73	44377	6.365	14576	0.73	13522	0.843	10648	32335	1.37	10648	32335	1.37
polska.US50-17	9956	0.76	30918	15.91	17625	0.76	11547	0.84	11688	23819	190.52	11688	23819	190.52
nobel-germany.US50-1	55897	3.80	349761	2.84	131284	3.80	61422	4.181	78402	227883	28.48	78402	227883	28.48
nobel-germany.US50-7	71546	3.83	394953	0.452	131460	3.83	64274	5.07	115370	205386	192.79	115370	205386	192.79
nobel-germany.US50-12	40103	3.69	227013	22.932	81067	3.69	53119	4.727	98350	216289	0.14%	98350	216289	0.14%
nobel-germany.US50-17	45074	4.13	255446	13.073	117374	4.13	49497	4.118	91219	203394	38.98	91219	203394	38.98
norway.US50-1	1954.1	65.78	10452.9	0.194%	3731.6	65.78	1749.9	57.393	2440.8	3425	0.94%	2440.8	3425	0.94%
norway.US50-7	3081	65.05	15552.6	0.264%	5242.4	65.05	2977.4	62.278	2948.7	5794.8	0.10%	2948.7	5794.8	0.10%
norway.US50-12	2800.6	61.77	12562.5	0.017%	4157.2	61.77	2979.8	70.418	3479.2	6516.5	0.27%	3479.2	6516.5	0.27%
norway.US50-17	2921.1	66.09	17385.4	0.139%	5360	66.09	3668.4	62.993	4338	9981.9	0.26%	4338	9981.9	0.26%

Table 2: heuristics

<i>problem</i>	<i>UB</i>	<i>LB</i>	<i>gap</i>	<i>time</i>	<i>problem</i>	<i>UB</i>	<i>LB</i>	<i>gap</i>	<i>time</i>
atlanta.US50-1	5526	5526	0%	26.177	atlanta.UA50-1	8162	8162	0%	56.347
atlanta.US50-7	5197	5197	0%	7.769	atlanta.UA50-7	10123	10123	0%	246.09
atlanta.US50-12	6290	6290	0%	23.852	atlanta.UA50-12	10979	10979	0%	991.567
atlanta.US50-17	3769	3769	0%	2.636	atlanta.UA50-17	8318	8318	0%	236.808
dfn-bwin.US50-1	577	577	0%	6.177	dfn-bwin.UA50-1	1386	1386	0%	32.947
dfn-bwin.US50-7	733	733	0%	67.033	dfn-bwin.UA50-7	1548	1429.82	7.63405%	1h30
dfn-bwin.US50-12	1058	1058	0%	147.092	dfn-bwin.UA50-12	1225	1225	0%	1319.45
dfn-bwin.US50-17	896	896	0%	16.723	dfn-bwin.UA50-17	1471	1471	0%	569.073
dfn-gwin.US50-1	214	214	0%	79.794	dfn-gwin.UA50-1	291	291	0%	1075.49
dfn-gwin.US50-7	281	281	0%	57.736	dfn-gwin.UA50-7	253	253	0%	84.583
dfn-gwin.US50-12	148	148	0%	18.673	dfn-gwin.UA50-12	196	196	0%	108.794
dfn-gwin.US50-17	258	258	0%	39.421	dfn-gwin.UA50-17	280	280	0%	655.668
di-yuan.US50-1	1395	1395	0%	8.939	di-yuan.UA50-1	1063	1063	0%	34.617
di-yuan.US50-7	1371	1371	0%	6.271	di-yuan.UA50-7	1165	1165	0%	87.282
di-yuan.US50-12	1327	1327	0%	37.471	di-yuan.UA50-12	1246	1246	0%	396.583
di-yuan.US50-17	1648	1648	0%	3.666	di-yuan.UA50-17	726	726	0%	48.594
nobel-us.US50-1	5395	5395	0%	18.579	nobel-us.UA50-1	9394	9394	0%	19.531
nobel-us.US50-7	6209	6209	0%	29.828	nobel-us.UA50-7	11265	11265	0%	187.949
nobel-us.US50-12	6521	6521	0%	14.617	nobel-us.UA50-12	10253	10253	0%	250.708
nobel-us.US50-17	23183	23183	0%	6.146	nobel-us.UA50-17	27518	27518	0%	19.219
pdh.US50-1	2458	2458	0%	41.558	pdh.UA50-1	2381	2381	0%	231.255
pdh.US50-7	2988	2988	0%	9.516	pdh.UA50-7	2589	2589	0%	26.458
pdh.US50-12	3014	3014	0%	122.569	pdh.UA50-12	2889	2889	0%	159.728
pdh.US50-17	3153	3153	0%	26.021	pdh.UA50-17	2294	2294	0%	35.412
polka.US50-1	15909	15909	0%	397.94	polka.UA50-1	22448	22448	0%	380.094
polka.US50-7	6159	6159	0%	1.747	polka.UA50-7	8440	8440	0%	3.572
polka.US50-12	11718	11718	0%	2.137	polka.UA50-12	12378	12378	0%	36.317
polka.US50-17	9360	9360	0%	37.238	polka.UA50-17	10147	10147	0%	17.535
nobel-germany.US50-1	55897	55897	0%	12.855	nobel-germany.UA50-1	61107	61107	0%	959.151
nobel-germany.US50-7	71360	71360	0%	83.881	nobel-germany.UA50-7	63014	63014	0%	126.438
nobel-germany.US50-12	40017	40017	0%	580.275	nobel-germany.UA50-12	52446	52446	0%	667.524
nobel-germany.US50-17	44759	44759	0%	87.048	nobel-germany.UA50-17	49497	49497	0%	5.835
norway.US50-1	1927.1	1881.8	2.3509%	1h30	norway.UA50-1	1672.9	1652.03	1.24731%	1h30
norway.US50-7	3005.3	2899.44	3.52257%	1h30	norway.UA50-7	2768.2	2707.54	2.19138%	1h30
norway.US50-12	2766.7	2732.03	1.25295%	1h30	norway.UA50-12	2926.5	2891.44	1.19817%	1h30
norway.US50-17	2858.3	2752.69	3.69502%	1h30	norway.UA50-17	3570.3	3510.58	1.67266%	1h30

Table 3: static routing

<i>problem</i>	<i>UJB</i>	<i>LB</i>	<i>gap</i>	<i>cuts</i>	<i>time</i>	<i>problem</i>	<i>UJB</i>	<i>LB</i>	<i>gap</i>	<i>cuts</i>	<i>time</i>
atlanta.US50-1	5526	5526	0%	50	28.095	atlanta.UA50-1	8162	7980.5	2.22372%	76	1h30
atlanta.US50-7	5197	5197	0%	46	1368.67	atlanta.UA50-7	10150	9922	2.24631%	89	1h30
atlanta.US50-12	6205	6205	0%	51	2143.49	atlanta.UA50-12	11028	10512.5	4.67446%	67	1h30
atlanta.US50-17	3769	3769	0%	51	82.508	atlanta.UA50-17	8318	8067.5	3.01154%	64	1h30
dfn-bwin.US50-1	577	577	0%	37	17.207	dfn-bwin.UA50-1	1386	1386	0%	163	1230.86
dfn-bwin.US50-7	732	732	0%	77	460.34	dfn-bwin.UA50-7	1551	1378.75	11.1057%	172	1h30
dfn-bwin.US50-12	1058	1058	0%	65	437.268	dfn-bwin.UA50-12	1141	1101	3.5057%	216	1h30
dfn-bwin.US50-17	896	896	0%	28	20.171	dfn-bwin.UA50-17	1473	1271.5	13.6796%	298	1h30
dfn-gwin.US50-1	211	211	0%	28	273.078	dfn-gwin.UA50-1	257	257	0%	68	400.234
dfn-gwin.US50-7	281	281	0%	58	1042.38	dfn-gwin.UA50-7	250	250	0%	97	1457.31
dfn-gwin.US50-12	148	148	0%	26	72.54	dfn-gwin.UA50-12	196	196	0%	93	2443.57
dfn-gwin.US50-17	255	255	0%	61	815.677	dfn-gwin.UA50-17	268	268	0%	152	99.559
di-yuan.US50-1	1367	1367	0%	30	77.5	di-yuan.UA50-1	1034	1034	0%	60	41.48
di-yuan.US50-7	1351	1351	0%	28	21.59	di-yuan.UA50-7	1151	1151	0%	62	1010.66
di-yuan.US50-12	1327	1327	0%	23	39.031	di-yuan.UA50-12	1232	1232	0%	54	186.342
di-yuan.US50-17	1603	1603	0%	35	42.76	di-yuan.UA50-17	717	717	0%	40	62.978
nobel-us.US50-1	5395	5395	0%	63	155	nobel-us.UA50-1	9394	8722	7.1535%	70	1h30
nobel-us.US50-7	6113	6113	0%	85	4154.31	nobel-us.UA50-7	11300	11195.5	0.924779%	75	1h30
nobel-us.US50-12	6521	6521	0%	53	353.854	nobel-us.UA50-12	10331	10149.3	1.75927%	91	1h30
nobel-us.US50-17	23014	23014	0%	61	88.093	nobel-us.UA50-17	27518	26558.5	3.48681%	43	1h30
pdh.US50-1	2348	2348	0%	98	448.484	pdh.UA50-1	2300	2300	0%	90	61.105
pdh.US50-7	2988	2988	0%	34	37.549	pdh.UA50-7	2481	2481	0%	108	249.896
pdh.US50-12	2850	2850	0%	27	243.188	pdh.UA50-12	2934	2738	6.6803%	98	5735.88
pdh.US50-17	3143	3143	0%	28	98.124	pdh.UA50-17	2177	2177	0%	63	39.437
poliska.US50-1	15553	15553	0%	57	1391.51	poliska.UA50-1	22522	20740.5	7.91004%	48	1h30
poliska.US50-7	6007	6007	0%	41	10.405	poliska.UA50-7	8428	8428	0%	70	337.1
poliska.US50-12	11718	11718	0%	46	23.712	poliska.UA50-12	11757	11757	0%	43	348.925
poliska.US50-17	9060	9060	0%	42	113.209	poliska.UA50-17	9484	9484	0%	46	119.964
nobel-germany.US50-1	55897	55761	0.243305%	79	1h30	nobel-germany.UA50-1	61193	60354.5	1.37025%	110	1h30
nobel-germany.US50-7	71360	69141	3.10959%	71	1h30	nobel-germany.UA50-7	63014	60141.5	4.55851%	82	1h30
nobel-germany.US50-12	40146	39279	2.15962%	85	1h30	nobel-germany.UA50-12	52489	50018	4.70765%	90	1h30
nobel-germany.US50-17	44759	44379.5	0.847874%	78	1h30	nobel-germany.UA50-17	49497	49497	0%	80	24.96
norway.US50-1	1954.1	1837.5	5.96694%	226	1h30	norway.UA50-1	1569.3	1569.3	0%	417	3438.93
norway.US50-7	3081	2898.8	5.91366%	215	1h30	norway.UA50-7	2977.4	2657.25	10.7527%	415	1h30
norway.US50-12	2800.6	2659.85	5.02571%	240	1h30	norway.UA50-12	2979.8	2709.8	9.06101%	383	1h30
norway.US50-17	2921.1	2793.35	4.37335%	197	1h30	norway.UA50-17	3668.4	3350.9	8.6555%	402	1h30

Table 4: branch-and-cut

## References

- [1] A. Altın, E. Amaldi, P. Belotti, and M. Ç. Pınar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1):100–115, 2007.
- [2] A. Altın, H. Yaman, and M. Ç. Pınar. The robust network loading problem under hose demand uncertainty: formulation, polyhedral analysis, and computations. In *Proceedings of the International Network Optimization Conference (INOC) 2009*, 2009.
- [3] A. Atamtürk. On capacitated network design cut-set polyhedra. *Mathematical Programming*, 92:425–437, 2002.
- [4] P. Avella, S. Mattia, and A. Sassano. Metric inequalities and the network loading problem. *Disc. Opt.*, 4:103–114, 2007.
- [5] F. Barahona. Network design using cut inequalities. *SIAM J. Optim.*, 6:823–834, 1996.
- [6] W. Ben-Ameur and H. Kerivin. Routing of uncertain traffic demands. *Optimization and Engineering*, 6:283–313, 2005.
- [7] A. Ben-Tal and A. Nemirowski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.
- [8] B. Bernard and M. Poss. An improved benders decomposition applied to a multi-layer network design problem. *Oper. Res. Lett.*, 37(5):777–795, 2009.
- [9] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- [10] D. Bienstock, S. Chopra, O. Günlük, and C.Y. Tsai. Minimum cost capacity installation for multicommodity flows. *Mathematical Programming*, 81:177–199, 1998.
- [11] C. Chekuri, G. Oriolo, M.G. Scutellà, and F.B. Shepherd. Hardness of robust network design. *Networks*, 50(1):50–154, 2007.
- [12] S. Chopra and M. R. Rao. The steiner tree problem i: Formulations, compositions and extensions of facets. *Mathematical Programming*, 64:209–229, 1994.
- [13] B. Colson, P. Marcotte, and G. Savard. Bilevel programming: A survey. *4OR: A Quarterly Journal of Operations Research*, 3:87–107, 2005.
- [14] A. M. Costa, JF Cordeau, and B. Gendron. Metric inequalities, cutset inequalities and benders feasibility cuts for multicommodity capacitated network design. *Computational Optimization and Applications*, 42(3):371–392.



- [15] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52:333–359, 2003.
- [16] M. Deza. and M. Laurent. *Geometry of Cuts and Metrics*. Springer-Verlag, Berlin, 1997.
- [17] M. Deza and E. Pantaleeva. Quasi-semi-metrics, oriented multi-cuts and related polyhedra. *Europ. J. Combinatorics*, 21:777–795, 2000.
- [18] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. MishraRamakrishnan, and J. E. van der Merwe. A flexible model for resource management in virtual private networks. In *Proceedings of the ACM SIGCOMM Computer Communication Review*, volume 29, pages 95–109, 1999.
- [19] F. Eisenbrand, F. Grandoni, G. Oriolo, and M. Skutella. New approaches for virtual private network design. In *Proceedings of ICALP*, volume 86, pages 1151–1162, 2005.
- [20] J. Andrew Fingerhut, S. Suri, and J. Turner. Designing least-cost nonblocking broadband networks. *J Algorithms*, 24:287–309, 1997.
- [21] N. Goyal, N. Olver, and B. Shepherd. Dynamic vs oblivious routing in network design. In *17th Annual European Symposium on Algorithms (ESA 2009)*, 2009.
- [22] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flows. In *Proceedings of ACMSTOC 2001*, pages 389–398, 2001.
- [23] M. Iri. On an extension of the max-flow min-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13:129–135, 1971.
- [24] A. Lodi and T.K. Ralphs. Bilevel programming and maximally violated valid inequalities. In *Proceedings of the Cologne Twente Workshop on Graphs and Combinatorial Optimization*, 2009.
- [25] T. L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the two- facility capacitated network loading problem. *Operations Research*, 43:142–157, 1995.
- [26] S. Mattia. Solving survivable two-layer network design problems by metric inequalities. Technical report, DIS, Dipartimento di Informatica e Sistemistica “Antonio Ruberti”. Sapienza, Università di Roma, 2010.
- [27] K. Onaga and O. Kakusho. On feasibility conditions of multicommodity flows in network. *IEEE Trans. Circuit Theory*, 18(4):425–429, 1971.
- [28] G. Oriolo. Domination between traffic matrices. *Mathematics of Operations Research*, 33:91–96, 2008.

- [29] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessály. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, April 2007*. <http://sndlib.zib.de>.
- [30] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large scale symmetric traveling salesman problems. *SIAM review*, 33:60–100, 1991.
- [31] D. Rajan. *Designing capacitated survivable networks: polyhedral analysis and algorithms*. PhD thesis, UC Berkeley, 2004.
- [32] A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.