CANADA EXCELLENCE RESEARCH CHAIR

DATA SCIENCE FOR REAL-TIME DECISION-MAKING

EXPERIMENTS ON VIRTUAL PRIVATE NETWORK DESIGN WITH CONCAVE CAPACITY COSTS

Andrea Lodi
Ahmad Moradi

December 2016

DS4DM-2016-004

POLYTECHNIQUE MONTRÉAL
DÉPARTEMENT DE MATHÉMATIQUES ET GÉNIE INDUSTRIEL
Pavillon André-Aisenstadt
Succursale Centre-Ville C.P. 6079
Montréal - Québec
H3C 3A7 - Canada
Téléphone: 514-340-5121 # 3314

Experiments on Virtual Private Network Design with Concave Capacity Costs

Andrea Lodi * Ahmad Moradi †

December 27, 2016

Abstract

For the first time in the literature, the paper considers computational aspects of concave cost virtual private network design problems. It introduces careful bound tightening mechanisms and computationally demonstrate how such bound tightening could impressively improve convex relaxations of the problem. It turns out that, incorporating such bound tightening with a general solution approach could significantly enhance the behavior of the solution approach over the problem.

1 Introduction

In the basic (asymmetric) Virtual Private Network (VPN) design problem (see, e.g., [6]) we are

Given: a communication network represented as an undirected graph $G = (V, E)$ within which there is a set of terminal nodes, $T \subset V$, needing to communicate with each other. In this network, each edge, $e \in E$, has an associated per-unit capacity reservation cost $c_e \geq 0$ and each terminal s has associated *hose* thresholds (upper bounds), b_s^+ and b_s^-, specifying the amount of traffic the terminal can send to or receive from the network, respectively.

Let $S = \{(s,t) : s,t \in T, s \neq t\}$ be the set of all ordered pairs of distinct terminal nodes. Also let d_{st} denote a nonnegative expected demand assigned to a terminal pair $(s,t) \in S$ and define $d = \{d_{st} \geq 0 : \forall (s,t) \in S\}$ as to be a *traffic assignment* over the underlying network. Such a traffic assignment is called *valid* (see, e.g., [6]) if it respects hose thresholds, i.e.,

$$\sum_{(s,t)\in S} d_{st} \leq b_s^+, \sum_{(t,s)\in S} d_{ts} \leq b_s^-, \quad \forall s \in T. \tag{1}$$

Constraint (1) together with demand non-negativity constraints define a polytope usually referred to as *hose polytope*. Indeed, a valid traffic assignment is nothing but a point feasible to the hose polytope. For every $(s,t) \in S$, the traffic d_{st} is allowed to be routed only on a simple path from s to t, say P_{st}. The set of such paths, $P = \{P_{st} : \forall (s,t) \in S\}$, is called a *path assignment* over the network. Also let $x = \{x_{ij} \geq 0 : \forall \{i,j\} \in E\}$ be a *capacity reservation* over the underlying network. With the above setting, the goal of the VPN design problem is to accomplish the following

Task: find a proper capacity reservation x, and a proper path assignment P in such a way that any valid traffic assignment could be routed along the corresponding paths specified by P without exceeding the capacities reserved by x and the total reservation cost is minimized.

The problem, as defined above, comes also with a restricting assumption as each link has an associated cost proportional to the amount of capacity reserved on the link. Such simplification usually comes at a price as it could not address an essential feature of link capacity pricing in telecommunication networks, the so-called economy of scale phenomenon [9]: on each link, the larger the amount of reserved capacity, the smaller the cost per unit capacity reserved. It then announces a natural extension of the VPN design

*École Polytechnique de Montréal, Québec, Canada. andrea.lodi@polymtl.ca
†University of Mazandaran, Babolsar, Iran. a.moradi@umz.ac.ir

problem with a more realistic cost model to allow for economies of scales. In the extended problem, called *concave cost* VPN (ccVPN) design problem, the contribution of an edge $e \in E$ to the total cost is equal to some arbitrary fixed concave, non-decreasing function f of the capacity x_e reserved on e. The ccVPN design problem was first considered in [12] where complexity status of the problem is investigated and an approximation algorithm is designed. The result is then extended and elaborated in [5].

The Model. As an adaptation to the linear case (see [1]), the problem could be represented by a compact formulation. Having the network graph $G = (V, E)$, terminal set $T \subset V$ and the hose nonnegative threshold vectors $\boldsymbol{b}^+ = (b_i^+)_{i\in T}$, $\boldsymbol{b}^- = (b_i^-)_{i\in T}$, one could express the (ccVPN) problem through the following mixed-integer nonlinear programming (MINLP) model:

$$(\text{ccVPN}) \quad \min \sum_{\{i,j\}\in E} f(x_{ij}) \tag{2}$$

$$\sum_{(i,j)\in A} y_{ij}^{st} - \sum_{(j,i)\in A} y_{ji}^{st} = \begin{cases} +1 & i = s \\ -1 & i = t \\ 0 & \text{otheriwse} \end{cases} \quad \forall i \in V, \forall (s,t) \in S \tag{3}$$

$$\sum_{s\in T}(b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{s-}) \le x_{ij} \quad \forall \{i,j\} \in E \tag{4}$$

$$y_{ij}^{st} + y_{ji}^{st} - \omega_{ij}^{s+} - \omega_{ij}^{t-} \le 0 \quad \forall \{i,j\} \in E,\ \forall (s,t) \in S \tag{5}$$

$$y_{ij}^{st} \in \{0,1\} \quad \forall (i,j) \in A,\ \forall (s,t) \in S \tag{6}$$

$$\omega_{ij}^{s+}, \omega_{ij}^{s-} \ge 0 \quad \forall \{i,j\} \in E,\ \forall s \in T \tag{7}$$

$$x_{\{i,j\}} \ge 0 \quad \forall \{i,j\} \in E. \tag{8}$$

where $A = \{(i,j),(j,i)|\{i,j\} \in E\}$ and f is a concave non-decreasing real function with $f(0) = 0$. For each arc $(i,j) \in A$ and each $(s,t) \in S$, the binary variable y_{ij}^{st} takes value 1 if the arc (i,j) is used to route traffic demand from s to t, Otherwise, it takes value 0.

In the above formulation, constraints (3) and (6) define the set of all possible path assignments. Then, having fixed a path assignment, one could dualize hose polytope via omega variables to obtain the amount of capacity needed to support any valid traffic demand over the path assignment. This will simply be translated into constraints (4), (5) and (7). We say that non-negative ω variables in (ccVPN) could be reduced to be binary. This is an immediate consequence of our previous results for the linear case (see, Theorem 3.2 of [11]) and the fact that f is non decreasing. Thus, constraints (7) in the model are replaced with

$$\omega_{ij}^{s+}, \omega_{ij}^{s-} \in \{0,1\} \quad \forall \{i,j\} \in E,\ \forall s \in T. \tag{9}$$

(ccVPN) is a non-convex MINLP, thus much more difficult to solve than its linear-cost counterpart because its continuous relaxation is in general NP-hard (see, e.g., [3]).

To the best of our knowledge, the problem has not received any computational attention yet. In this short paper, we give a first attempt to that. In Section 2, we initially take a straightforward step and we solve the (ccVPN) MINLP formulation presented in this section through the SCIP solver [14] as it is, by quickly realizing that bound tightening is a necessary condition to succeed (Section 2.1). Such a bound tightening is presented in Section 2.2 and the final set of experiments is reported in Section 2.3. Finally, in Section 3 we draw some conclusions and we outline some possible avenues for further improvement.

2 Computational Experiments

In this section, we use the SCIP optimizer to solve the (ccVPN) MINLP formulation by more and more sophisticated attempts.

Test instances considered in this section are the challenging instances studied in [10, 11] and the instances derived from SNDlib networks [15] used in [13] for a related network design problem. For each problem instance, capacity reservation cost on an edge is modeled by the following concave function:

$$f(x) = (ax + b)^r + c, \quad a > 0,\ r \in (0,1). \tag{10}$$

Using this popular class of concave functions [9], one could easily adjust degree of economy of scale on an edge by simply tuning parameter r. The two sets of instances are denoted by I_1 and I_2, respectively.

We performed all experiments on a single core of an Intel Core i5 with 2.53 GHz processor, under Linux with 4 GB of RAM. We coded cut-and-branch algorithms in C++ by using the SCIP framework, where Cplex 12.5 [7] has been used for solving linear programming (LP) relaxations. We set a time limit of 3,600 CPU seconds for solving each benchmark instance. For all instances that cannot be solved to optimality within the time limit, we report the gap between the best known upper bound and the lower bound obtained.

2.1 Solving the (ccVPN) MINLP formulation

Directly solving (ccVPN) within a branch-and-bound framework intrinsically encounters some difficulties. Recall that the continuous relaxation of the model is a non-convex (nonlinear) programming problem for which one can only, currently, aim at finding in polynomial time local optima. As a result, one could not find a valid lower bound just by relaxing integrality constraints. To obtain a lower bound, we have to first build a convex relaxation of (ccVPN). Such a convex relaxation could be either linear or nonlinear. However, for computational reasons, most of the existing solvers use linear ones [17].

In order to build a convex relaxation of (ccVPN), one should replace cost function (10) with a convex function, say f^c, underestimating f and at the same time close enough to it. Let us again draw our attention to a fixed edge. More precisely, a convex under-estimator, f^c, of f over an interval $[a, b]$ ($\subseteq [0, UB]$) is a convex function over $[a, b]$, for which $f^c(x) \leq f(x)$ for every $x \in [a, b]$. The tightest possible convex under-estimator of f is its *convex envelope* denoted by f^{ce}. Building the convex envelope of a general function is hard (see, e.g., [16, 8]). However, because (10) is concave, its convex envelope exists and is given by the linear secant connecting the points $(a, f(a))$ and $(b, f(b))$

$$f^{ce}(t) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \quad \forall t \in [a, b]. \tag{11}$$

Computing the above secant is precisely what the non-convex MINLP solver SCIP does to construct the first convex continous relaxation of (ccVPN) and such a relaxation is iteratively strengthened by branching in the so-called *spatial branch and bound* (see, e.g., [2] for details).

In the following, we first, we investigate MINLPs given in I_1 and I_2 by applying SCIP as a global optimization solver. Here, any instance is evaluated before imposing an upper bound on capacity variables, i.e., by using $UB = +\infty$. The results are reported in the left-most part of Table 1. Note that, when no bound on capacity variables is imposed, SCIP finds such bound with the aim of presolving. For each instance in this case, the upper bound on capacity variables (averaged over all capacity variables) by presolving is reported (presolve bound) together with the percentage gap of the initial continuous relaxation computed using the presolved bounds (gap_i).

It is easy to see that relying only on such presolving bounds generally leaves a big gap on the MINLP instances and no instance can be solved to optimality with a time limit of 3,600 CPU seconds (no detailed results on the SCIP spatial branch and bound are reported because of the lack of practical significance). Thus, the first order of business is to improve the formulation by computing and installing better upper bounds on the capacity variables and a simple bound can be obtained by observing that, for the edge $\{i, j\}$, (ccVPN) could only increase x_{ij} through constraint (4). As a result, x_{ij} could not take a value more that $\sum_{s \in T}(b_s^+ + b_s^-)$. This defines an upper bound, say ub_0, on x_{ij}.

2.2 Improving by Bound tightening

As noted in the previous section, tightening the bound on the capacity variables x_{ij} is crucial for solving the (ccVPN) MINLP formulation, and, in the following, we carefully consider problem structure and install some tight bounds on capacity variables.

Let us consider again an edge $\{i, j\} \in E$ and suppose that the traffic d_{st} is going to be routed through $\{i, j\}$. The amount of capacity on the edge needed to support this communication is $\min(b_s^+, b_t^-)$. The first idea to bound the amount of capacity, x_{ij}, needed on the edge is to see the edge in the extreme situation where all ordered terminal pairs in S use $\{i, j\}$ in their routing. Then, we have

$$x_{ij} \leq \sum_{(s,t) \in S} \min(b_s^+, b_t^-) \quad \forall \{i, j\} \in E. \tag{12}$$

Let us define $ub_1 = \sum_{(s,t)\in S} \min(b_s^+, b_t^-)$. As we will see, ub_1 is not the best possible bound on capacity variables. In fact, while computing ub_1 on an edge, traffic associated with terminal pairs are considered separately. In this way, demand relation through the hose polytope is ignored, and the computed maximum capacity on the edge is highly overestimated.

Using ub_1 significantly improves initial dual bound of the (ccVPN) and in general the overal computation. However, this is not the best possible bound on capacity variables. Consider a feasible solution of (ccVPN) and let $\dot{y}$ be its path assignment given by (3) and (6). Knowing $\dot{y}$, one could compute maximum capacity needed on the edge $\{i, j\}$ by means of the following:

$$x_{ij} = \max\{ \sum_{(s,t)\in S} d_{st}(\dot{y}_{ij}^{st} + \dot{y}_{ji}^{st}) : (1) \quad \text{and} \quad d_{st} \geq 0 \quad \forall (s,t) \in S\}, \tag{13}$$

where the term $d_{st}(\dot{y}_{ij}^{st} + \dot{y}_{ji}^{st})$ in the objective function is the amount of capacity needed on the edge $\{i, j\}$ (in both directions) imposed by the terminal pair (s, t). Furthermore, since $\dot{y}$ is a binary vector representing a collection of simple paths, then $\forall \{i, j\} \in E,\ \forall (s, t) \in S$ we have $\dot{y}_{ij}^{st} + \dot{y}_{ji}^{st} \leq 1$. As a result,

$$x_{ij} \leq \max\{ \sum_{(s,t)\in S} d_{st} : (1) \quad \text{and} \quad d_{st} \geq 0 \quad \forall (s,t) \in S\}. \tag{14}$$

Now define ub_2 as the value of the the right hand side linear program in (14). We have,

Lemma 2.1 *Upper bound ub_2 dominates upper bound ub_0 i.e. $ub_2 \leq ub_0$.*

Proof Taking the dual of the right hand side LP in (14), we have

$$ub_2 = \min\{\sum_{s\in T}(b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{s-}) : \omega_{ij}^{s+} + \omega_{ij}^{t-} \geq 1,\ \forall (s,t) \in S \text{ and } \omega_{ij}^{s+}, \omega_{ij}^{s-} \geq 0,\ \forall s \in T\}$$

$$\leq \sum_{s\in T}(b_s^+ + b_s^-) = ub_0 \quad \blacksquare$$

Lemma 2.2 *Upper bound ub_2 dominates upper bound ub_1 i.e. $ub_2 \leq ub_1$.*

Proof For any valid traffic vector d and any terminal pair $(s, t) \in S$, we always have $d_{st} \leq b_s^+, d_{st} \leq b_t^-$. Then, $d_{st} \leq \min(b_s^+, b_t^-)$ and

$$ub_2 \leq \sum_{(s,t)\in S} d_{st} \leq \sum_{(s,t)\in S} \min(b_s^+, b_t^-) = ub_1 \quad \blacksquare$$

We have to note that, ub_0 and ub_1 are *not* in general comparable, as it is clarified by Example 1 below. Lemma 2.2 shows that exploiting demand relations through hose polytope always provide a better bound on capacity variables. Our experiments below also show that ub_2 is far superior to ub_0 and ub_1 in practice. However, in theory, the inequality in Lemma 2.2 holds with equality as shown by simple computations for an instance of the ccVPN problem with only two distinct terminal nodes. Then, a natural question is if a further improvement on ub_2 can be achieved. In fact, Example 2.3 below shows that it is not possible.

Example 2.3 *Consider a ccVPN instance over a three-node complete graph in which any node is a terminal one and hose thresholds are given by $\boldsymbol{b}^- = (4, 5, 5)$ and $\boldsymbol{b}^+ = (1, 2, 2)$. Also let the capacity installation cost function be defined by the identity function. Then, an optimal solution to the ccVPN instance will install capacities $ub_2 = x_{12} = x_{13} = 5, x_{23} = 0$. Then further improvement on ub_2 is not possible. For the instance, simple computations also show that $ub_0 > ub_1$. However, for most of the instances given in I_1 or I_2, $ub_0 < ub_1$ (see Tables 1).* $\blacksquare$

The results reported in the central part of Table 1 evaluates the above mentioned upper bounds. Namely, for each instance, columns four to seven report the $min\{ub_0, ub_1\}$ value, its percentage gap, the ub_2 value and its percentage gap, respectively. A "*" appears to the right of a bound value in the forth column to indicate that $\min\{ub_0, ub_1\} = ub_1$.

It is not hard to see that installing a better and better upper bound on capacity variables significantly improves the percentage gap of the initial dual bound and in the next section we will evaluate the impact of such an improvement in the capacity of SCIP to compute optimal solutions.

2.3 Computing optimal (ccVPN) solutions by using ub_2

Before going into the details of the computation on the MINLP model by using ub_2 as a bound for the capacity variables, one needs to observe that another straightforward strategy for solving (ccVPN) is to apply a piecewise linear approximation to the concave cost function (10), see Figure 1. (For a complete

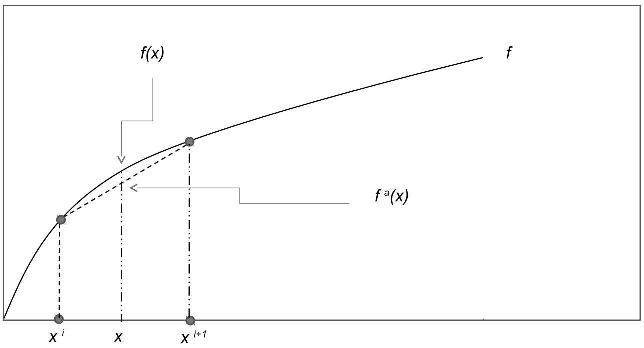

Figure 1: For an x located in the i-th consecutive interval, $f(x)$ is approximated by $f^a(x)$; value of x on the linear secant connecting $(x^i, f(x^i))$ and $(x^{i+1}, f(x^{i+1}))$.

discussion on how to efficiently build the associated Mixed-Integer Linear Programming (MILP), which requires a binary variable for each of the intervals, the reader is referred to [4].) It is not hard to see that installing the same upper bound ub_2 on capacity variables makes the valid lower bound computed by the resulting MILP approximation identical to the one obtained by solving the secant convex relaxation of the MINLP model: essentially, the first and last samples of the piecewise approximation/relaxation are used precisely as in the secant approach. In other words, solving the MILP provides a static *a priori* convexification, although the actual objective value of the optimal solution to the MILP has to be recomputed on the original concave curve f, thus potentially being only an approximation, i.e., heuristic. Such a static approximation requires to decide the number of breakpoints the concave function is sampled (the higher this number the better the approximation but the larger the resulting MILP) and in the experiments described below we used 21 breakpoints, i.e., 20 equal length intervals.

Thus, the right-most part of Table 1 evaluates the use of bound ub_2 in both the MILP and MINLP formulations to solve (ccVPN). More precisely, the last six columns are devoted to compare the two solution approaches when ub_2 is installed on capacity variables in terms of final gap at the time limit (gap_f), computing time (t_f) and number of branch-and-bound nodes (n_{BB}).

The results clearly reveal that having installed ub_2 as the best possible bound on capacity variables in (ccVPN), both of the solution methods are able to close a large fraction of initial integrality gap and a non-negligible number of instances, especially in the I_2 set can be solved within the time limit. In addition, solving (ccVPN) instances as a MILP often provides slightly better dual bound (on average) within the given time limit and solves more instances to optimality.

3 Conclusions

The paper studied how bound tightening could improve convex relaxations of VPN design problem in the presence of concave capacity reservation costs. It introduced the best possible upper bound on the capacity of an edge in a solution and clarified that such an upper bound could only be obtained when demand relations through the hose polytope are carefully considered. It also computationally demonstrated that imposing such a bound could impressively improve quality of the relaxations in terms of the initial dual bound. The computational results showed that the SCIP solver either by using the piecewise linear approximation model or the MINLP one both enhanced by those tight variable bounds is a viable way of computing (ccVPN) optimal solutions, although further improvements either on the formulations or on the algorithmic techniques are required to tackle the hardest instances.

Table 1: Experiments with various versions of both MINLP and MILP formulations and impact of tightening bounds on capacity variables.

	MINLP		MINLP				MILP ($UB = ub_2$)			MINLP ($UB = ub_2$)		
I_1 instances	presolve bound (on average)	gap_i %	$\min\{ub_0, ub_1\}$	gap_i %	ub_2	gap_i %	gap_f %	t_f (sec.)	n_{BB} #	gap_f %	t_f (sec.)	n_{BB} #
ring-20-10	9.64E+004	1,815.30	5,206	194.28	806	30.26	0.00	17.11	119	0.00	12.81	396
ring-20-15	2.35E+005	2,571.70	7,809	244.13	1,209	49.32	0.00	195.28	4,987	3.18	T.L.	88,580
ring-20-20	1.31E+005	3,614.56	10,711	284.32	1,811	70.52	12.78	T.L.	49,011	12.32	T.L.	62,095
ring-50-10	4.05E+005	2,299.54	5,206	268.68	806	63.19	0.14	T.L.	123,272	4.79	T.L.	222,664
ring-50-15	8.92E+004	2,714.18	7,809	262.48	1,209	57.28	11.06	T.L.	27,700	15.62	T.L.	48,466
ring-50-20	4.70E+005	2,980.24	10,412	249.16	1,612	49.66	19.53	T.L.	13,290	25.21	T.L.	10,398
ring-50-25	1.29E+006	3,515.82	13,513	220.27	2,413	44.25	20.02	T.L.	1,941	33.59	T.L.	1
ring-50-30	5.66E+006	3,598.71	15,817	226.69	2,617	41.80	23.96	T.L.	11	29.34	T.L.	1
ring-80-10	1.45E+006	1,913.46	5,206	209.36	806	36.93	1.77	T.L.	40,060	11.17	T.L.	73,312
ring-80-15	6.64E+006	2,661.45	7,809	255.69	1,209	54.34	20.55	T.L.	4,559	24.19	T.L.	4,451
ring-80-20	1.31E+008	2,424.61	10,412	186.18	1,612	22.67	4.76	T.L.	268	14.11	T.L.	1
ring-80-25	1.11E+007	3,396.26	13,015	258.44	2,015	52.34	37.39	T.L.	7	45.15	T.L.	1
ring-80-30	1.94E+008	3,500.66	15,319	263.99	2,219	49.66	37.34	T.L.	3	49.66	T.L.	1
ring-100-7	9.99E+005	1,869.32	3,624 (*)	231.87	604	53.87	1.85	T.L.	78,611	5.03	T.L.	203,687
ring-100-14	8.78E+006	2,233.13	7,408	193.97	1,208	30.45	9.35	T.L.	8,784	20.59	T.L.	1
ring-100-20	1.84E+007	3,790.56	11,010	270.99	2,010	69.73	50.53	T.L.	1	59.86	T.L.	1
ring-100-25	2.67E+008	3,374.91	13,513	207.79	2,413	38.63	38.63	T.L.	1	29.34	T.L.	1
ring-100-30	3.13E+008	3,518.70	15,817	219.62	2,617	38.74	38.74	T.L.	1	26.00	T.L.	1
Average		2,877.39		235.99		47.42	18.24			22.73		
I_2 instances												
Atalanta 3	473.4149	103.78	58.5	36.90	22.8	18.19	6.58	T.L.	110,803	10.92	T.L.	114,079
Dfn-gwin 1	42.6631	14.04	22.8	6.42	8.5	3.38	0.00	1,911.02	72,333	0.00	500.59	18,379
Dfn-gwin 4	20.7062	17.64	35.0	8.87	8.4	3.32	0.83	T.L.	126,897	0.00	1,392.41	50,216
Di-yuan 1	29.7342	11.90	23.8	5.23	8.8	2.50	0.00	72.77	3,787	0.00	32.66	2,269
Di-yuan 2	26.5537	21.91	44.0	12.95	9.0	6.29	0.00	2,832.07	167,236	0.49	T.L.	314,684
Di-yuan 3	71.3975	23.75	48.7	14.23	9.3	6.73	0.00	1,739.66	108,583	0.00	3,540.1	392,377
Di-yuan 4	16.8881	13.71	25.4	6.25	7.8	2.56	0.00	192.70	1,2016	0.00	46.23	5,405
Nobel-germany 1	1.57E+005	135.51	130.6	46.11	37.3	14.24	0.00	386.65	12,926	10.04	T.L.	269,676
Nobel-germany 2	1.46E+003	143.25	147.4	54.58	44.3	22.97	6.12	T.L.	122,882	13.04	T.L.	172,420
Nobel-germany 3	248.7384	135.10	123.7	53.95	35.6	23.10	6.65	T.L.	155,388	15.78	T.L.	107,657
Nobel-us 4	2.98E+003	249.82	338.3	96.40	60.0	28.53	0.00	1,260.87	104,787	8.45	T.L.	233,317
Pdh 1	39.6273	17.81	24.9 (*)	8.94	6.1	3.77	0.00	71.84	5174	0.00	46.16	4,802
pdh2	40.4517	20.09	32.2	10.39	9.0	5.09	0.00	514.04	38,612	0.00	294.04	28,311
Pdh 3	17.9617	25.35	28.4	11.33	10.2	5.69	1.42	T.L.	239,468	2.19	T.L.	155,841
Pdh 4	19.7725	21.35	31.4	11.70	6.4	5.03	1.26	T.L.	426,102	0.00	2,729.53	333,791
Polska 1	434.3869	128.78	77.3	43.32	25.5	17.01	0.00	669.20	71,546	6.21	T.L.	496,120
Polska 3	173.6125	84.14	41.9	30.91	10.6	8.53	0.00	89.45	10,027	0.00	216.45	41,733
Polska 4	91.6994	60.62	31.4	26.57	6.4	8.25	0.00	36.53	9,690	0.00	37.86	12,064
Norway 1	515.4485	46.82	40.5	20.17	12.4	8.42	4.06	T.L.	142,828	3.57	T.L.	248,418
Norway 2	673.6783	49.70	56.6	19.04	24.2	9.47	5.88	T.L.	78,455	7.74	T.L.	86,052
Average		66.25		26.21		10.15	1.64			3.92		

References

[1] A. Altın, E. Amaldi, P. Belotti, M.C. Pınar, Provisioning virtual private networks under traffic uncertainty. Networks, 49 (2007), 100–115.

[2] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan, Mixed-integer nonlinear optimization. Acta Numerica, 22, (2013), 1–131.

[3] S. Burer, A. N. Letchford, Non-convex mixed-integer nonlinear programming: A survey. Surveys in Operations Research and Management Science, 17(2), (2012), 97–106.

[4] C. D'Ambrosio, A. Lodi, S. Martello. Piecewise linear approximation of functions of two variables in MILP models. Operations Research Letters 38(1), (2010), 39–46.

[5] F. Grandoni, T. Rothvoß, L. Sanità, From Uncertainty to NonLinearity: Solving Virtual Private Network via Single-Sink Buy-at-Bulk, Mathematics of Operations Research 36(2), (2011), 185–204.

[6] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, Provisioning a virtual private network: A network design problem for multicommodity flow, Proc 33th Ann ACM Symposium on Theory of Computing (STOC), Crete, Greece, 2001, pp. 389–398.

[7] IBM: IBM ILOG CPLEX Optimization Studio. `http://www.cplex.com`

[8] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. Mathematical Programming, 10(1), (1976), 147–175.

[9] M. Minoux. Multicommodity network flow models and algorithms in telecommunications. In Handbook of optimization in telecommunications (2006), 163–184. Springer US.

[10] A. Moradi, A. Lodi, S.M. Hashemi, On the Difficulty of Virtual Private Network Instances. Networks 63 (2014), 327–333.

[11] A. Moradi, A. Lodi and S. Mehdi Hashemi. Virtual private network design over the first Chvátal closure. RAIRO-Oper. Res. 49(3) (2015), 569–588.

[12] T. Rothvoß and L. Sanitá, On the complexity of the asymmetric VPN problem, International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Springer, 2009, 326–338.

[13] S. Mattia, The robust network loading problem with dynamic routing, Computational Optimization and Applications 54 (2013), 619–643.

[14] SCIP – Solving Constraint Integer Programs. `http://scip.zib.de`

[15] S. Orlowski, M. Pioro, A. Tomaszewski, and R. Wessaly. SNDlib 1.0 – Survivable Network Design Library. In Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, 2007.

[16] M. Tawarmalani, N. V. Sahinidis. Convex extensions and envelopes of lower semicontinuous functions. Mathematical Programming, 93(2), (2002), 247–263.

[17] S. Vigerske. Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming. PhD diss., Humboldt-Universität Berlin, (2012).