Decomposition Algorithms for Multicommodity Network Design Problems with Penalized Constraints

Tolga Bektaş, *Université du Québec à Montréal, Montréal, QC, Canada*
Teodor Gabriel Crainic, *Université du Québec à Montréal, Montréal, QC, Canada*

Keywords: nonlinear multicommodity network design, Lagrangean relaxation, decomposition.

1. Introduction

Network design models are extensively used to represent a wide range of planning and operation management issues in transportation, telecommunications, logistics and production-distribution. In a very general sense, the problem consists of designing a network by selecting links to connect a set of nodes and to determine the amount of flow on each link such that the demand of each node for a number of commodities is satisfied. The objective is to minimize the total cost of establishing the links and flows. This basic variant is usually referred to as the *uncapacitated network design problem*. The problem has extensions that arise when additional restrictions are incorporated, such as imposing capacity limits on the amount of demand that may be transported on the links (referred to as the *capacitated network design problem*). Interested readers on the problem may consult the surveys by Magnanti and Wong [7] and Minoux [8]. Network design formulation provide a good modelling framework for service network design problems, usually at the strategic or the tactical level, for which Crainic [3] gives an overview and a classification of formulations.

Consider the graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of links. For the links activated in the network, there is a fixed-charge cost vector denoted by $\mathbf{f} = [f_{ij}]$. There exists a set of commodities denoted by $\mathcal{P}$. Let $\mathbf{y} = \{y_{ij} | (i,j) \in \mathcal{A}\}$ denote the vector of design variables with $\mathbf{y} \in \mathcal{Y}$, where $\mathcal{Y} = \{0,1\}^{|\mathcal{A}|}$ and $\mathbf{x} = \{x_{ij}^p | (i,j) \in \mathcal{A}, p \in \mathcal{P}\}$ denote the vector of flow variables with $x \in \mathcal{X} = \mathbb{N}_+^{|\mathcal{A}||\mathcal{P}|}$. A generic formulation for the multicommodity network design problem can be given as follows:

$$
\begin{aligned}
\text{Minimize} \quad & \mathbf{cx} + \mathbf{fy} && (1)\\
\text{subject to} \quad & \mathbf{Nx} = \mathbf{d} && (2)\\
& \mathbf{Ax} \leq \mathbf{by} && (3)\\
& \mathbf{Dx} \leq \mathbf{ey} && (4)
\end{aligned}
$$

where $\mathbf{N}$ is an arc-node incidence matrix, $\mathbf{A}$ and $\mathbf{D}$ are matrices and $\mathbf{b}$ and $\mathbf{e}$ are column vectors of appropriate dimensions. Constraints (2) are network flow constraints and (3) and (4) are additional relations such as linking or capacity restrictions. This paper focuses on multicommodity network design formulations incorporating penalized constraints. In specific, let us assume that constraints (3) are allowed to be violated (penalized) at the expense of additional cost. Then, we are interested in the problems of the following form,

$$
\begin{aligned}
\text{Minimize} \quad & \mathbf{cx} + \mathbf{fy} + \mathbf{p}[max(\mathbf{0}, \mathbf{Ax} - \mathbf{by})]^n && (5)\\
\text{subject to} \quad & (2), (4)
\end{aligned}
$$

where the last component of (5) is the penalty term imposing an additional cost whenever the constraint set is violated, with $\mathbf{p}$ being the vector of penalty cost coefficients. The idea of penalizing various types of constraints, such as capacity, was discussed by Crainic [2] in the context of freight transportation. Considering capacity constraints as an example, the author argues that for a tactical model, "one is generally less concerned with the specific vehicle capacity, the emphasis rather being on determining the frequency of the service, which determines its capacity, and the distribution of the freight traffic, which determines how this capacity is to be used". Such constraints can be allowed to be violated, although at the expense of additional cost. It is therefore more appropriate to treat such relations as utilization targets as opposed to strict constraints, as this would provide a better modelling framework in terms of planning. However, these penalized structures give way to nonlinear integer multicommodity network design formulations. Notice the nonlinear structure of the last term in (5), which ensures that the penalty is increased in a nonlinear fashion as the violation of the constraint grows larger. It is clearly more appropriate to treat the penalties in such a manner, rather than assuming linear structures. In specific, in applications where it is not desirable to exceed constraint limits, such structures allow one to increase the penalties by using, say, quadratic ($n = 2$) or qubic ($n = 3$) functions. This class of problems falls under the category of nonlinear multicommodity network design problems, and no exact solution method, to the best of our knowledge, has been proposed for this specific variant. The goal of this paper is to provide a framework for an exact solution procedure for these problems based on relaxation and decomposition. A good example to a penalized structure would be the case where overcapacity assignment is permitted at the expense of additional cost and delays. In the following, we will focus on this case, although the algorithms described here are also applicable where other types of constraints are penalized.

2. Network design with penalized capacity constraints

To formally define the problem, we assume each commodity $p \in \mathcal{P}$ has one origin $o(p)$ and one destination $d(p)$, and the quantity of commodity p that is to be sent from $o(p)$ to $d(p)$ is denoted by w^p. If a commodity has either more than one origins or destination, this can be modelled by splitting the commodity into several commodities, each with a single origin and destination (see [6]). The binary variable y_{ij} is the design variable that takes the value 1 if link (i, j) is used, and 0 otherwise, and the flow variable x_{ij}^p denotes the amount of commodity p flowing on link (i, j). We denote by c_{ij}^p the unit cost of routing the demand for commodity p over link (i, j). Each link (i, j) in the network has a capacity u_{ij} where a penalty cost has to be paid for each unit exceeding this capacity. The network design problem with penalized capacity constraints (NDPC), in general, consists of establishing the links and determining the flows on the links in the network, in order to satisfy the demand of each node. The objective is to minimize the total cost of establishing links and flows, as well as the penalty that arises as a result of overcapacity usage. The general formulation for the NDPC is:

$$(\mathcal{F}) \qquad \text{Minimize} \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} + \sum_{(i,j)\in\mathcal{A}} \sum_{p\in\mathcal{P}} c_{ij}^p x_{ij}^p + \sum_{(i,j)\in\mathcal{A}} C_{ij} \left[\max(0, \sum_{p\in\mathcal{P}} x_{ij}^p - u_{ij} y_{ij}) \right]^n \tag{6}$$

subject to

$$\sum_{j\in\mathcal{N}} x_{ij}^p - \sum_{j\in\mathcal{N}} x_{ji}^p = d_i^p \qquad \forall i \in \mathcal{N}, p \in \mathcal{P} \tag{7}$$

$$x_{ij}^p \le w^p y_{ij} \qquad \forall (i,j) \in \mathcal{A}, p \in \mathcal{P} \tag{8}$$

$$\mathbf{y} \in \mathcal{Y}, \mathbf{x} \in \mathcal{X}. \tag{9}$$

where $d_i^p = \begin{cases} w^p, & \text{if node } i = o(p) \\ -w^p, & \text{if node } i = d(p) \\ 0, & \text{otherwise.} \end{cases}$ In this formulation, C_{ij} is the penalty cost coefficient for each link (i, j). Constraints (7) ensure that the demands are satisfied for each node, and (8) stipulate that the flow of any commodity on a link is zero when that link is not selected. In the following, we describe an algorithm to solve the network design problem with penalized constraints. The algorithm is based on relaxing some of the constraints in a Lagrangean manner and decomposition of the resulting model.

3. Lagrangean Relaxation and Decomposition

Multicommodity network design problems are huge formulations, especially when the number of arcs and the number of commodities are considerably big. The main difficulty of solving such formulations, or even their LP-relaxations, is heavily dependent upon the size of the problem. This is the reason why decomposition-based approaches are popular for the solution of such problems, enabling one to partition the formulation into a number of subproblems, which are smaller in size and typically easier to solve. To solve the NDPC, we will also consider decomposition-based approach. First, we introduce an extended formulation for the problem upon which we build the decomposition schemes. To this purpose, new variables q_{ij} are defined which show the excess flow on each link. Let $\mathbf{q} = \{q_{ij} | (i,j) \in \mathcal{A}\}$ with $\mathbf{q} \in \mathcal{Q} = \mathbb{N}_+^{|\mathcal{A}|}$. The extended formulation is given below:

$$(\mathcal{F}_1) \qquad \text{Minimize} \sum_{(i,j)\in\mathcal{A}} f_{ij}y_{ij} + \sum_{(i,j)\in\mathcal{A}}\sum_{p\in\mathcal{P}} c_{ij}^p x_{ij}^p + \sum_{(i,j)\in\mathcal{A}} C_{ij}(q_{ij})^n \tag{10}$$

subject to

$$\sum_{j\in\mathcal{N}} x_{ij}^p - \sum_{j\in\mathcal{N}} x_{ji}^p = d_i^p \qquad \forall i \in \mathcal{N}, p \in \mathcal{P} \tag{11}$$

$$x_{ij}^p \leq w^p y_{ij} \qquad \forall (i,j) \in \mathcal{A}, p \in \mathcal{P} \tag{12}$$

$$\sum_{p\in\mathcal{P}} x_{ij}^p \leq u_{ij}y_{ij} + q_{ij} \qquad \forall (i,j) \in \mathcal{A}, \tag{13}$$

$$q_{ij} \leq B_{ij}y_{ij} \qquad \forall (i,j) \in \mathcal{A}, \tag{14}$$

$$\mathbf{y} \in \mathcal{Y}, \mathbf{x} \in \mathcal{X}, \mathbf{q} \in \mathcal{Q}. \tag{15}$$

In this formulation, constraints (13) imply that the amount of flow on a link can be at most the capacity of the link plus the excess capacity q_{ij}. These constraints, together with (14) also ensure that no commodity can flow on a link unless it is established in the network. The value B_{ij} is simply an upper bound on the amount of excess flow, which can be set to $\sum_{p\in\mathcal{P}} w^p - u_{ij}$. Note that although constraint (14) is redundant, it will be useful in the decomposition algorithms described below. Let $S(\mathcal{F})$ and $S(\mathcal{F}_1)$ denote the set of feasible solutions to formulations $\mathcal{F}$ and $\mathcal{F}_1$, respectively. It is now easy to see that, for any solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \in S(\mathcal{F})$, there corresponds a solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \mathbf{q}) \in S(\mathcal{F}_1)$ with the same objective function value and vice versa. We illustrate below how formulation $\mathcal{F}_1$ can be decomposed with respect to paths and links.

Shortest-path relaxation

We now relax constraints (12) and (13) in a Lagrangean fashion by associating nonnegative variables γ_{ij}^p and β_{ij} to these constraints, respectively. The resulting formulation can be stated as follows:

$$\begin{aligned}(\mathcal{LR}_1) \quad & \text{Minimize} \quad \sum_{(i,j)\in\mathcal{A}} \hat{f}_{ij}y_{ij} + \sum_{(i,j)\in\mathcal{A}}\sum_{p\in\mathcal{P}} \hat{c}_{ij}x_{ij}^p + \sum_{(i,j)\in\mathcal{A}} [C_{ij}(q_{ij})^n - \beta_{ij}q_{ij}] \\ & \text{subject to} \quad (11), (14), (15),\end{aligned} \tag{16}$$

where $\hat{f}_{ij} = f_{ij} - \sum_{p\in\mathcal{P}} \gamma_{ij}^p w^p - \beta_{ij}u_{ij}$ and $\hat{c}_{ij}^p = c_{ij}^p + \gamma_{ij}^p + \beta_{ij}$. $\mathcal{SP}_2$. It is easy to see that $\mathcal{LR}_1$ decomposes into two subproblems, where the first subproblem is in the y and q variables and stated as follows,

$$\begin{aligned}(\mathcal{SP}_1) \quad & \underset{\mathbf{y}\in\mathcal{Y}}{\text{Minimize}} \quad \sum_{(i,j)\in\mathcal{A}} \left[(f_{ij} - \sum_{p\in\mathcal{P}} \gamma_{ij}^p w^p - \beta_{ij}u_{ij})y_{ij} + C_{ij}(q_{ij})^n - \beta_{ij}q_{ij} \right] \\ & \text{subject to} \quad (14), \mathbf{y} \in \mathcal{Y}, \mathbf{q} \in \mathcal{Q}.\end{aligned} \tag{17}$$

This problem can be solved by inspection as shown in the following proposition.

Proposition 1 *Let $(\mathbf{y}^*, \mathbf{q}^*)$ denote the optimal solution to $\mathcal{SP}_1$. Then, for each $(i,j) \in \mathcal{A}$, if $\frac{\beta_{ij}}{nC_{ij}} \leq B_{ij}$, then*

$$\begin{aligned} y_{ij}^* = 1 \text{ and } q_{ij}^* = \left(\frac{\beta_{ij}}{nC_{ij}}\right)^{\frac{1}{n-1}} \qquad & \text{if } f_{ij} - \sum_{p\in\mathcal{P}} \gamma_{ij}^p w^p - \beta_{ij}u_{ij} - (n-1)\left(\frac{\beta_{ij}}{n}\right)^{\frac{n}{n-1}} \left(\frac{1}{C_{ij}}\right)^{\frac{1}{n-1}} \leq 0 \\ y_{ij}^* = 0 \text{ and } q_{ij}^* = 0, \qquad & \text{otherwise}.\end{aligned}$$

and if $\left(\frac{\beta_{ij}}{nC_{ij}}\right)^{\frac{1}{n-1}} > B_{ij}$, *then*

$$y^*_{ij} = 1 \textit{ and } q^*_{ij} = B_{ij} \qquad \textit{if } f_{ij} - \sum_{p\in\mathcal{P}} \gamma^p_{ij} w^p - \beta_{ij}(u_{ij} + B_{ij}) + C_{ij} B^n_{ij} \leq 0$$
$$y^*_{ij} = 0 \textit{ and } q^*_{ij} = 0, \qquad \textit{otherwise}.$$

Proof $\mathcal{SP}_1$ decomposes into $|\mathcal{A}|$ problems, one for each link $(i,j) \in \mathcal{A}$. Therefore, in this proof, we will derive the optimal solution for a single subproblem. Let $\mathcal{SR}$ denote a subproblem for a link (i,j) and $v(\mathcal{SR}(\hat{y},\hat{q}))$ denote its solution value for a given solution $(\hat{y},\hat{q})$ (indices ij have been suppressed for notational convenience). We consider two cases: (*i*) If $\hat{y} = 0$, then $\hat{q} = 0$ by constraint (14). Hence, $v(\mathcal{SR}(\hat{y},\hat{q})) = 0$. (*ii*) If $\hat{y} = 1$, then the term $f - \sum_{p\in\mathcal{P}} \gamma^p w^p - \beta u$ is a constant and the resulting subproblem can be stated as $\underset{0\leq q\leq B}{\text{Minimize}}\ Cq^n - \beta q$. Since $f(q) = Cq^n - \beta q$ is a convex function on the interval $[0, B]$, from the first order necessary conditions, the value of the local (and thus global) minimizer is derived as $q^* = (\beta/nC)^{\frac{1}{n-1}}$ if $(\beta/nC)^{\frac{1}{n-1}} \leq B$, and $q^* = B$ otherwise. It is now easy to see that the optimal solution is as given in case (*i*) if $v(\mathcal{SR}(0,0)) = 0 < v(\mathcal{SR}(1,q^*)) = f - \sum_{p\in\mathcal{P}} \gamma^p w^p - \beta u - (n-1)\left(\frac{\beta}{n}\right)^{\frac{n}{n-1}} \left(\frac{1}{C}\right)^{\frac{1}{n-1}}$, and is as given in case (*ii*) otherwise. $\square$

The second subproblem defined over the x variables is given as,

$$(\mathcal{SP}_2) \qquad \text{Minimize} \qquad \sum_{(i,j)\in\mathcal{A}} \sum_{p\in\mathcal{P}} \hat{c}^p_{ij} x^p_{ij} \tag{18}$$
$$\text{subject to} \qquad (11), \mathbf{x} \in \mathcal{X},$$

which further decomposes into $|\mathcal{P}|$ single commodity minimum cost network flow problems, each of which can be solved by using a shortest path algorithm.

Knapsack relaxation

An alternative relaxation to $\mathcal{F}_1$ can be obtained when constraints (11) are relaxed in a Lagrangean fashion with the multipliers α^p_i associated to each constraint. The relaxed problem then takes the following form:

$$(\mathcal{LR}_2) \qquad \text{Minimize} \qquad \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} + \sum_{(i,j)\in\mathcal{A}} \sum_{p\in\mathcal{P}} \bar{c}^p_{ij} x^p_{ij} + \sum_{(i,j)\in\mathcal{A}} C_{ij}(q_{ij})^n + \sum_{i\in\mathcal{N}} \sum_{p\in\mathcal{P}} \alpha^p_i d^p_i \tag{19}$$
$$\text{subject to} \qquad (12) - (15),$$

where $\bar{c}^p_{ij} = c^p_{ij} + \alpha^p_j - \alpha^p_i$. Notice that, for a given set of multipliers α^p_i, the last term on the objective is a constant. In this case, problem $\mathcal{LR}_2$ decomposes into $|\mathcal{A}|$ problems, one for each link $(i,j) \in \mathcal{A}$, each being in the following form (indices ij have again been suppressed for convenience):

$$(\mathcal{KR}) \qquad \text{Minimize} \qquad fy + \sum_{p\in\mathcal{P}} \hat{c}^p x^p + C(q)^n \tag{20}$$
$$\text{subject to} \qquad x^p \leq w^p y, \qquad p \in \mathcal{P}$$
$$\sum_{p\in\mathcal{P}} x^p \leq uy + q \tag{21}$$
$$q \leq By \tag{22}$$
$$\mathbf{y} \in \mathcal{Y}, \mathbf{x} \in \mathcal{X}, \mathbf{q} \in \mathcal{Q}. \tag{23}$$

$\mathcal{KR}$ is a mixed integer nonlinear programming problem with a single binary variable y. However, making use of the binary property of this variable, one can further reduce to solving $\mathcal{KR}$ to solving a continuous global optimization problem. When $y = 1$ in $\mathcal{KR}$, we will denote the resulting problem by $\mathcal{KR}^y$, its solution by $(\tilde{x},\tilde{q})$, and the value of its optimal solution by $v(\mathcal{KR}^y)$. Then,

Proposition 2 *Let $(\hat{y}, \hat{x}, \hat{q})$ denote the optimal solution to $\mathcal{KR}$. Then, $\mathcal{KR}$ has an optimal solution $(\hat{y}, \hat{x}, \hat{q}) = (1, \tilde{x}, \tilde{q})$ if $f + v(\mathcal{KR}^y) < 0$ and $(\hat{y}, \hat{x}, \hat{q}) = (0, 0, 0)$, otherwise.*

Observation 1 *When $y = 1$, the problem $\mathcal{KR}^y$ can be solved by inspection for the two special cases given below:*

- *If $\hat{c}^p > 0$ for all $p \in \mathcal{P}$, then $v(\mathcal{KR}^y) = 0$ with all the variables equal to 0.*
- *If $\sum_{p \in \mathcal{P}} w^p \leq u$, then $\tilde{x}^p = \begin{cases} 0, & \text{if } \hat{c}^p \geq 0 \\ w^p, & \text{if } \hat{c}^p < 0 \end{cases}$ and $\tilde{q} = 0$.*

To solve $\mathcal{KR}$, we first check if the conditions stated in Observation 1 are satisfied. If not, we solve the continuous subproblem using IPOPT, an interior point algorithm described in [9] and is publicly available in COIN-OR (`http://www.coin-or.org/`).

Obtaining feasible solutions

Given any solution to either the shortest-path or knapsack relaxations that specifies a set of y^*_{ij} variables, we describe here a procedure to generate feasible solutions to problem $\mathcal{F}_1$. The need for such a procedure comes from the fact that, although there are no strict capacity constraints, there might not be enough arcs opened ($y^*_{ij} = 1$) to find a feasible path to route the demands. Feasible solutions are then constructed using a modified formulation $\mathcal{F}_1$ (denoted by $\mathcal{F}_r$), where the original objective function is replaced with a linear objective function as $\sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij} + \sum_{(i,j) \in \mathcal{A}} \sum_{p \in \mathcal{P}} c^p_{ij} x^p_{ij} + \sum_{(i,j) \in \mathcal{A}} M(q_{ij})$, with M being a sufficiently large value. We then add to $\mathcal{F}_r$ a set of constraints implying $y^*_{ij} = 1$. CPLEX 9.0 is invoked to obtain a feasible solution to $\mathcal{F}_r$, which is also a feasible solution for $\mathcal{F}_1$. Note that $\mathcal{F}_r$ need not be solved to optimality, as it would suffice to use the first feasible solution encountered during the solution procedure.

4. Preliminary computational results

The algorithms described above were implemented within a traditional subgradient optimization scheme [5]. We denote by $A1$ and $A2$ the algorithms based on knapsack and shortest path relaxations, respectively. We compare the proposed algorithms with the package BONMIN [1], a state-of-the-art solver for mixed integer nonlinear programming problems (available at `https://projects.coin-or.org/Bonmin` or through the NEOS server `http://www-neos.mcs.anl.gov/`). All the comparisons have been made on a set of problems that are described in [4]. The capacities specified in these instances have been modified as $u'_{ij} = u_{ij}/2$ to obtain a tighter capacity structure. In these instances, costs on each link are the same for all commodities. The penalty cost C_{ij} has been set twice the flow cost of each link and $n = 3$. Algorithms $A2$ and BONMIN have been limited to a running time of one hour, and algorithm $A1$ to 1000 iterations. The results are presented in Table 1, where columns `col` and `con` respectively correspond to the number of variables and constraints in the formulation. Columns v_1 and v_2 present respectively the value of the best feasible solution obtained by algorithms $A1$ and $A2$, and the numbers in the parentheses correspond to the final optimality gap produced by each algorithm (calculated by $100(v_u - v_l)/v_u$, where v_u is the best upper bound and v_l is the best lower bound) at the end of the one-hour time limit. The last column presents the value of the best solution obtained by BONMIN, where a "*" indicates that the solution is optimal and "N/A" shows BONMIN was not able to produce a feasible solution within the given time limit.

Results given in Table 1 show that algorithm $A1$ is able to output good quality solutions, typically around 5% of the optimal solution. Algorithm $A2$ has a similar performance in terms of the upper bound, but it yields weak lower bounds. Finally, we observe that BONMIN can solve to optimality rather small-sized problems, but it has difficulty in obtaining even a feasible solution for larger problems. These preliminary results give

Table 1: Comparison results on a set of network design instances

$\|\mathcal{V}\|$	$\|\mathcal{A}\|$	$\|\mathcal{P}\|$	`col`	`con`	v_1	v_2	v_3
10	60	50	3120	3620	290289 (4.27)	291607 (40.27)	286486 *
10	83	50	4316	4816	193146 (4.62)	195244 (36.98)	192720 *
20	120	40	5040	5840	233627 (5.85)	233552 (34.22)	N/A
20	120	100	12240	14240	881167 (4.33)	884463 (47.71)	969289
20	120	200	24240	28240	2334084 (8.10)	2337403 (61.45)	N/A
20	220	40	9240	10040	153554 (4.07)	155874 (38.85)	N/A
20	220	100	22440	24440	443142 (3.62)	453524 (45.76)	531856
20	220	200	44440	48440	1179184 (6.12)	1189229 (56.93)	N/A

an implication that algorithm $A1$ can be used as a stand-alone efficient procedure to obtain near-optimal solutions to the problem. On the other hand, in order to close the gaps, one can also devise an efficient exact algorithm by embeedding the decompositions described here into a branch-and-bound framework. This is currently under consideration by the authors of this paper.

Acknowledgement

The first author wishes to thank Pietro Belotti for kindly providing help with the AMPL files. Detailed comments and corrections provided by an anonymous reviewer are gratefully acknowledged.

References

[1] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuejols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Waechter. An algorithmic framework for convex mixed integer nonlinear programs. Technical Report RC23771, IBM Research Report, October 2005.

[2] T.G. Crainic. Rail tactical planning: issues, models and tools. Technical Report CRT-520, Center for research on transportation, 1987.

[3] T.G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122:272–288, 2000.

[4] T.G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112:73–99, 2001.

[5] M. Held, P. Wolfe, and H.P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.

[6] K. Holmberg and D. Yuan. A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48:461–481, 2000.

[7] T.L. Magnanti and R.T. Wong. Network design and transportation planning: models and algorithms. *Transportation Science*, 1:1–55, 1986.

[8] M. Minoux. Network synthesis and optimum network design problems: models, solution methods and applications. *Networks*, 19:313–360, 1986.

[9] A. Wächter and L.T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.