A Branch-and-Cut Algorithm for Multicommodity Capacitated Fixed Charge Network Design

Mervat Chouman 1
Mervat.Chouman@cirrelt.ca

Teodor Gabriel Crainic 1, 2, 3
Teodor.Crainic@cirrelt.ca

Bernard Gendron 1, 4
Bernard.Gendron@cirrelt.ca

1 CIRRELT
(Interuniversity Research Centre on
Enterprise Networks, Logistics and Transportation)
2 Département de management et technologie
École des sciences de la gestion
Université du Québec à Montréal
3 Chaire de recherche industrielle en management logistique
Université du Québec à Montréal
4 Département d'informatique et de recherche opérationnelle
Université de Montréal

November 2008

We present a branch-and-cut (B&C) algorithm for solving large-scale instances of the multicommodity capacitated fixed-charge network design problem (MCND). Our work is a follow-up on Chouman, Crainic and Gendron [1], which describes a cutting-plane method for improving the linear programming (LP) relaxation of a mixed-integer programming (MIP) formulation of the problem. To the best of our knowledge, this work is one of the few contributions on exact methods for the MCND, following Holmberg and Yuan [3], Sellmann, Kliewer and Koberstein [5], and Kliewer and Timajev [4].

1 Problem Formulation

Given a directed network, with V the set of nodes, and A the set of arcs, we let K be the set of commodities, each commodity k having one origin, $O(k)$, and one destination, $D(k)$. We associate to each arc (i,j) the per unit routing cost c_{ij}^k for each commodity k, the fixed cost f_{ij} and the capacity u_{ij}. Two types of variables are used to formulate the MCND: the continuous flow variable x_{ij}^k, which represents the flow of commodity k on arc (i,j), and the binary design variable y_{ij}, which equals 1 when arc (i,j) is used, and 0, otherwise. Given these definitions, the MCND can be formulated as follows:

$$Z = \min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{1}$$

$$\sum_{j \in V_i^+} x_{ij}^k - \sum_{j \in V_i^-} x_{ji}^k = \begin{cases} d^k, & \text{if } i = O(k), \\ -d^k, & \text{if } i = D(k), \\ 0, & \text{otherwise,} \end{cases} \quad \forall\, i \in V, \forall\, k \in K, \tag{2}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall\, (i,j) \in A, \tag{3}$$

$$x_{ij}^k \geq 0, \quad \forall\, (i,j) \in A,\ \forall\, k \in K, \tag{4}$$

$$y_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A, \tag{5}$$

where $V_i^+ = \{j \in V | (i,j) \in A\}$ and $V_i^- = \{j \in V | (j,i) \in A\}$. Constraints set (2) represent the flow conservation equations for each node and each commodity. Relations (3) are the weak forcing constraints, which ensure that no flow circulates on an arc unless the fixed cost for that arc is incurred, and that the flow on each arc does not exceed the arc capacity.

2 Outline of the Bounding Procedure

At each node of the B&C tree, the bounding procedure performs the following general steps:

1. *Fathom = False.*
2. *Preprocessing:* Perform connectivity-based variable fixing.
3. *Subproblem optimization:* Compute a lower bound Z^l and an upper bound Z^u by a cutting-plane method.
4. *Fathoming test:* If the subproblem is infeasible or $Z^l \geq Z^u$, then *Fathom = True* and stop.
5. *Postprocessing:* Use reduced cost information to generate cuts and fix variables; if some variables have been fixed, then go to 2.

At the end of this procedure, either *Fathom = True*, in which case the node is fathomed, or *Fathom = False*, in which case branching is performed and the generated nodes are inserted in the node pool. Next, if the node pool is empty, the B&C algorithm terminates; otherwise, a node is selected from the node pool and the bounding procedure is applied to that node.

The B&C algorithm manages two types of cuts: *global cuts*, generated at step 3, which are valid at any node of the B&C tree, and *local cuts*, generated at step 5, which are valid only at the current node and at all its descendants.

3 Cutting-Plane Method

At step 3 of the bounding procedure, we perform a cutting-plane method inspired by Chouman, Crainic and Gendron [1], which uses five families of valid inequalities to improve the LP relaxation of the MCND. To avoid overloading the paper, we will omit the references to the articles where these inequalities were introduced; see [1] for more details.

Strong Inequalities (SI)

The strong forcing constraints

$$x_{ij}^k \;\leq\; d^k y_{ij}, \;\; \forall (i,j) \in A, k \in K. \tag{6}$$

improve significantly the quality of the LP lower bounds. Although there is a polynomial number of them ($|A||K|$), adding all of them to the LP relaxation yields very large models that frequently exhibit degeneracy. Hence, we will add a small fraction of them within the cutting-plane algorithm.

Cover Inequalities (CI)

If we let $S \subset N$ be any non empty subset of N and $\bar{S} = N \backslash S$ its complement, we denote the corresponding cutset by $(S, \bar{S})$, i.e., the set of arcs that connect a node in S to a node in $\bar{S}$. The following cutset inequality is valid:

$$\sum_{(i,j) \in (S,\bar{S})} u_{ij} y_{ij} \;\geq\; d_{(S,\bar{S})},$$

where $d_{(S,\bar{S})}$ is a lower bound on the amount of flow that must circulate across the cutset in any feasible solution. A set $C \subseteq (S, \bar{S})$ is a cover if the total capacity of arcs in $(S, \bar{S}) \backslash C$ does not cover the demand:

$$\sum_{(i,j) \in (S,\bar{S}) \backslash C} u_{ij} \;<\; d_{(S,\bar{S})}.$$

For every cover $C \subseteq (S, \bar{S})$, the cover inequality (CI)

$$\sum_{(i,j) \in C} y_{ij} \;\geq\; 1 \tag{7}$$

is valid for the MCND. The basic idea behind this inequality is that one has to open at least one arc from the set C in order to meet the demand.

Minimum Cardinality Inequalities (MCI)

Let us assume the capacities of the arcs in $(S, \bar{S})$ are represented in non-increasing order: $u_{ij_{(t)}} \geq u_{ij_{(t+1)}}, t = 1, ..., |(S, \bar{S})| - 1$. We can then define the least number of arcs in $(S, \bar{S})$ that must be used in every feasible solution: $l_S = \max \{h | \sum_{t=1,...,h} u_{ij_{(t)}} < d_{(S,\bar{S})}\} + 1$. From this number, we can derive the minimum cardinality inequality (MCI), defined as:

$$\sum_{(i,j)\in(S,\bar{S})} y_{ij} \geq l_S. \tag{8}$$

Flow Cover Inequalities (FCI)

For any $L \subseteq K$, let $x^L_{ij} = \sum_{k\in L} x^k_{ij}$, $b^L_{ij} = \min\{u_{ij}, \sum_{k\in L} d^k\}$ and $d^L_{(S,\bar{S})} = \sum_{k\in K(S,\bar{S})\cap L} d^k$, for a given cutset $(S, \bar{S})$. A flow cover (C_1, C_2) is defined by two sets $C_1 \subseteq (S, \bar{S})$ and $C_2 \subset (\bar{S}, S)$ such that $\mu = \sum_{(i,j)\in C_1} b^L_{ij} - \sum_{(j,i)\in C_2} b^L_{ji} - d^L_{(S,\bar{S})} > 0$. The flow cover inequality (FCI) is then defined as follows, where $a^+ = \max\{0, a\}$ and $D_2 \subset (\bar{S}, S)\backslash C_2$:

$$\begin{aligned}\sum_{(i,j)\in C_1} (x^L_{ij} + (b^L_{ij} - \mu)^+(1 - y_{ij})) \leq & \sum_{(j,i)\in D_2} \min\{b^L_{ji}, \mu\} y_{ji} + \sum_{(j,i)\in C_2} b^L_{ji} \\ & + d^L_{(S,\bar{S})} + \sum_{(j,i)\in(\bar{S},S)\backslash C_2\cup D_2} x^L_{ji}.\end{aligned} \tag{9}$$

Flow Pack Inequalities (FPI)

Using the same notation as above, a flow pack (C_1, C_2) is defined by two sets $C_1 \subseteq (S, \bar{S})$ and $C_2 \subset (\bar{S}, S)$ such that $\mu = \sum_{(i,j)\in C_1} b^L_{ij} - \sum_{(j,i)\in C_2} b^L_{ji} - d^L_{(S,\bar{S})} < 0$. The flow pack inequality (FPI) is then defined as follows, where $D_1 \subset (S, \bar{S})\backslash C_1$:

$$\begin{aligned}\sum_{(i,j)\in C_1} x^L_{ij} + \sum_{(i,j)\in D_1} (x^L_{ij} - \min\{b^L_{ij}, -\mu\} y_{ij})) \leq & - \sum_{(j,i)\in C_2} (b^L_{ji} + \mu)^+(1 - y_{ji}) + \\ & \sum_{(j,i)\in(\bar{S},S)\backslash C_2} x^L_{ji} + \sum_{(i,j)\in C_1} b^L_{ij}.\end{aligned} \tag{10}$$

By adding these five families of valid inequalities, we can reformulate the LP relaxation of the MCND as follows:

$$\min cx + fy \tag{11}$$

$$Nx^k = d^k, \quad k \in K, \quad (\pi_k) \tag{12}$$

$$\sum_{k\in K} x^k \leq uy, \quad (\alpha \geq 0) \tag{13}$$

$$x^k \leq d^k y, \quad k \in K, \quad (\beta_k \geq 0) \tag{14}$$

$$Ex - Gy \leq v, \quad (\omega \geq 0) \tag{15}$$

$$Hy \geq t, \quad (\theta \geq 0) \tag{16}$$

$$x \geq 0, \quad 0 \leq y \leq 1. \tag{17}$$

In this formulation, x is a vector of size $|A| \times |K|$ representing the flow variables, x^k and y are vectors of size $|A|$ representing the flow variables for each commodity and the design variables; N is the node-arc incidence matrix of the network, hence constraints (12) are the flow conservation equations; constraints (13) and (14) are, respectively, the weak and the strong, forcing inequalities; constraints (15) correspond to the SCI and the FPI, while (16) refer to the other inequalities, CI and MCI, which involve only the y variables.

At each node of the B&C tree, a subset of the inequalities (14)-(16) are generated through the separation and lifting procedures described in [1], combined with an efficient management of the list of global cuts.

To compute upper bounds, two approaches are used. First, any solution to the LP relaxation provides a feasible solution, obtained by simply rounding up the values of the y variables. Second, any solution to the LP relaxation can serve as a starting point to a *slope-scaling* procedure [2] that solves a sequence of linear multicommodity flow subproblems with costs modified to reflect the contribution of the fixed costs.

4 Variable Fixing, Local Cut Generation and Branching

At step 2 of the bounding procedure, we fix variables based on simple connectivity tests. Basically, an arc (i, j) can be closed if it does not belong to any path between $O(k)$ and $D(k)$ for all commodities k. Conversely, an arc (i, j) can be opened if it belongs to all paths between $O(k)$ and $D(k)$ for at least one commodity k. These tests can be performed efficiently using graph traversal algorithms. Moreover, these tests are not performed if the last variables that were fixed (by branching or reduced cost fixing) are all at value 1.

At step 5 of the bounding procedure, we use the reduced costs derived from the Lagrangian relaxation obtained by dualizing constraints (12), (15) and (16), with multipliers equal to the optimal LP dual values:

$$Z(LR) = \pi d - \omega v + \theta t + \min(c - \pi N + \omega E)x + (f - \omega G - \theta H)y$$

subject to constraints (13), (14), and (17). This problem can be solved by first considering for each arc (i, j) the following continuous knapsack problem:

$$Z_{ij}(KS) = \min\{\sum_{k \in K} c_{ij}^k(\pi, \omega) | \sum_{k \in K} x_{ij}^k \leq u_{ij};\ 0 \leq x_{ij}^k \leq d^k,\ k \in K\},$$

where $c_{ij}^k(\pi, \omega)$ corresponds to component $((i, j), k)$ of the vector $(c - \pi N + \omega E)$. Then, by denoting $Z(KS)$ the vector of continuous knapsack problems optimal values and $f(\omega, \theta)$ the vector $(f - \omega G - \theta H)$, the Lagrangian subproblem can be reformulated as follows:

$$Z(LR) = \pi d - \omega v + \theta t + \min_{y \in [0,1]} (Z(KS) + f(\omega, \theta))y.$$

We define the vector of reduced costs of the y variables as:

$$\overline{f} = Z(KS) + f(\omega, \theta).$$

An optimal solution to the Lagrangian subproblem is then given by $\overline{y}_{ij} = 1$, if $\overline{f}_{ij} < 0$, and $\overline{y}_{ij} = 0$, otherwise. It is easy to derive variable fixing rules and local cuts by using $\overline{f}$. For instance, if $Z(LR) + |\overline{f}_{ij}| \geq Z^u$, then we can fix $y_{ij} = 1 - \overline{y}_{ij}$.

If no variables are fixed using these tests, we use these reduced costs again to determine two sets of candidate arcs, the arcs to be closed and the arcs to be opened. For each arc candidate to be closed (opened), we compute an approximation of the increase in the LP lower bound incurred by closing (opening) the arc. We use this approximation to try to further fix variables. If no variables are fixed, we use the approximation to determine the branching variable, in a way similar to strong branching.

References

[1] Chouman M., Crainic T.G. and Gendron B. (2008), "A Cutting-Plane Algorithm Based on Cutset Inequalities for Multicommodity Capacitated Fixed Charge Network Design", Publication CRT-03, Centre de recherche sur les transports, Université de Montréal, Montréal (revised).

[2] Crainic T.G., Gendron B. and Hernu G. (2008), "A Slope Scaling/Lagrangean Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design", *Journal of Heuristics* 10, 525-545.

[3] Holmberg K. and Yuan D. (2000), "A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem", *Operations Research* 48, 461-481.

[4] Kliewer G. and Timajev L. (2005), "Relax-and-Cut for Capacitated Network Design", *Proceedings of Algorithms-ESA 2005: 13th Annual European Symposium on algorithms* in *Lecture Notes in Computer Science* 3369, 47-58.

[5] Sellmann M., Kliewer G. and Koberstein A. 2002, "Lagrangian Cardinality Cuts and Variable Fixing for Capacitated Network Design", *Proceedings of Algorithms-ESA 2002: 10th Annual European Symposium on Algorithms* in *Lecture Notes in Computer Science* 2461, 845-858.