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

Mervat Chouman* Teodor Gabriel Crainic* Bernard Gendron*

*Centre de recherche sur les transports, Université de Montréal
C.P.6128, succursale Centre-ville
Montréal, CANADA H3C 3J7
{mervat,theo,bernard}@crt.umontreal.ca

In this paper, we present a branch-and-cut algorithm (B&C) to solve the multicommodity capacitated fixed charge network design problem (MCND). To the best of our knowledge, this work is one of the few attempts at solving optimally the MCND, following the contributions by Holmberg and Yuan (2000), and Sellmann, Kliewer and Koberstein (2002), who both propose branch-and-bound algorithms based on the same Lagrangian relaxation. The present paper is a follow-up on Chouman, Crainic and Gendron (2003), where the authors describe a cutting-plane method for improving the linear programming (LP) relaxation of a mixed-integer programming (MIP) formulation. This cutting-plane approach forms the basis of our B&C algorithm, but instead of performing it at every node of the B&C tree, which is computationally too heavy, we solve Lagrangian subproblems to perform variable fixing, generate local cuts, and derive branching rules.

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 is allowed unless the arc's fixed cost is payed, and that the flow on each arc does not exceed the arc's capacity.

The cutting-plane procedure presented in Chouman, Crainic and Gendron (2003) uses four families of VI to improve the LP relaxation of this MIP formulation.

Strong Inequalities (SI)

The *strong forcing inequalities*,

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

improve significantly the quality of the LP relaxation. However, solving the LP relaxation including all these inequalities is time consuming and not all such inequalities are necessary. Hence, in our cutting-plane procedure, we generate only SI that are violated by the current LP solution.

Cover Inequalities (CI)

Let $S \subset V$ be any non empty subset of V and its complement $\bar{S} = V \backslash S$, and let $K(S, \bar{S}) \subseteq K$ be the set of commodities having their origin in S and their destination in $\bar{S}$. Every feasible solution of the MCND satisfies the *cutset inequality*, $\sum_{(i,j)\in(S,\bar{S})} u_{ij} y_{ij} \geq d_{(S,\bar{S})}$, where $d_{(S,\bar{S})} = \sum_{k\in K(S,\bar{S})} d^k > 0$ and $(S, \bar{S})$ is the set of arcs that connect a node in S to a node in $\bar{S}$ (a *cutset*). The cutset inequality is redundant for the LP relaxation, but some VI derived from it might be useful. Chouman, Crainic and Gendron (2003) present a heuristic to generate cutsets $(S, \bar{S})$ having a "good" chance of yielding such violated VI. In the remainder, we will assume that some cutset $(S, \bar{S})$, generated by this heuristic, is given.

A set $C \subseteq (S, \bar{S})$ is called a minimal cover if the total capacity of the arcs in $(S, \bar{S}) \backslash C$ does not cover the demand ($\sum_{(i,j)\in(S,\bar{S})\backslash C} u_{ij} < d_{(S,\bar{S})}$), and it is sufficient to open any arc in C to cover the demand ($\sum_{(i,j)\in(S,\bar{S})\backslash C} u_{ij} + u_{pq} \geq d_{(S,\bar{S})}, \forall (p,q) \in C$) (see Chouman, Crainic, and Gendron 2003, for details on how such a minimal cover can be generated). For any minimal cover C, the *cover inequality* (CI), defined as

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

is valid for the MCND, and can be strengthened by a lifting procedure (Chouman, Crainic and Gendron 2003).

Minimum Cardinality Inequalities (MCI)

For any cutset $(S, \bar{S})$, we define the *minimum cardinality inequality*:

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

where $l_S = \max\{h : \sum_{t=1,\dots,h} u_{ij_{(t)}} < d_{(S,\bar{S})}\} + 1$ and $u_{ij_{(t)}} \geq u_{ij_{(t+1)}}, t = 1, ..., m-1$. In this inequality, l_S is the least number of arcs in $(S, \bar{S})$ that must be used in every feasible solution. The MCI can be strengthened by a lifting procedure (Chouman, Crainic and Gendron, 2003).

Single-Arc Network Cutset Inequalities (SNCI)

Let $L \subseteq K$, $x^L_{ij} = \sum_{k \in L} x^k_{ij}$, $b^L_{ij} = \min\{u_{ij}, \sum_{k \in L} d^k\}$ and $d^L_{(S,\bar{S})} = \sum_{k \in K(S,\bar{S}) \cap L} d^k$. Further assume that for any arc $(i,j) \in A$, any set $L \subseteq K$ is partitioned into two subsets L^1_{ij} and L^0_{ij}. Then, any feasible solution of the MCND must satisfy the *single-arc network cutset inequality*:

$$\sum_{(i,j) \in C_1} x^{L^1_{ij}}_{ij} + x^L_{rt} \leq (\sum_{(j,i) \in C_2} b^{L^1_{ji}}_{ji} + d^L_{(S,\bar{S})}) y_{rt} + \sum_{(j,i) \in C_2} x^{L^0_{ji}}_{ji} + \sum_{(j,i) \in (\bar{S},S) \setminus C_2} x^L_{ji} + (1 - y_{rt}) \sum_{(i,j) \in C_1} b^{L^1_{ij}}_{ij}, \quad (9)$$

where $(r,t) \in (S,\bar{S})$, $C_1 \subseteq (S,\bar{S}) \setminus \{(r,t)\}$ and $C_2 \subseteq (\bar{S},S)$. Given a set $S \subset V$ and an arc $(r,t) \in (S,\bar{S})$, it is easy to solve the separation problem for the SNCI (Chouman, Crainic and Gendron, 2003).

By adding the four families of VI, we can reformulate the LP relaxation of the MCND as follows:

$$\min cx + fy \quad (10)$$

$$Nx^k = d^k, \quad k \in K, \quad (\pi_k) \quad (11)$$

$$\sum_{k \in K} x^k \leq uy, \quad (\alpha \geq 0) \quad (12)$$

$$x^k \leq d^k y, \quad k \in K, \quad (\beta_k \geq 0) \quad (13)$$

$$Ex - Gy \leq v, \quad (\omega \geq 0) \quad (14)$$

$$Hy \geq t, \quad (\theta \geq 0) \quad (15)$$

$$x \geq 0, \quad 0 \leq y \leq 1. \quad (16)$$

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 (11) are the flow conservation equations; constraints (12) and (13) are, respectively, the weak and the strong, forcing inequalities; constraints (14) correspond to the SNCI, while (15) refer to the other cutset-based inequalities (CI and MCI), which involve only the y variables.

Cutset Subproblem (CS)

By relaxing the flow conservation constraints (11) and the SNCI constraints (14) in a Lagrangian way, we obtain the following cutset subproblem:

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

subject to constraints (12), (13), (15), and (16). This problem can be solved by first considering for each arc (i,j) the following continuous knapsack problem:

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

where $c^k_{ij}(\pi,\omega)$ 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 $f(\omega)$ the vector $(f - \omega G)$, the cutset subproblem can be reformulated as follows:

$$Z(CS) = \pi d - \omega v + \min(Z(KS) + f(\omega))y$$

$$Hy \geq t,$$
$$0 \leq y \leq 1.$$

To stregthen this bound, we will add to the constraints $Hy \geq t$ (CI and MCI generated when solving the LP relaxation), the cutset inequalities for one and two-node cutsets (along with CI and MCI generated from them) and some local cuts (cuts valid only for the current node and its descendants). The resulting constraints will be denoted $H'y \geq t'$ and their corresponding Lagrangian multipliers θ'. From the solution of this strengthened cutset subproblem, it is easy to derive variable fixing rules and local cuts by using $\overline{f}(CS)$, the vector of reduced costs of the y variables:

$$\overline{f}(CS) = f(\omega) + Z(KS) - (\theta' H').$$

This relaxation (without the addition of cutset-based inequalities) has been used by Holmberg and Yuan (2000), and Sellmann, Kliewer and Koberstein (2002) in their branch-and-bound algorithms, where at each node a subgradient algorithm is used to optimize the Lagrangian dual. Here, we propose to exploit this relaxation by fixing the Lagrangian multipliers to the values provided by the cutting-plane procedure.

Multicommodity Flow Subproblem (MF)

The multicommodity flow subproblem is obtained by dropping all constraints except the flow conservation equations, and by introducing the valid inequalities: $\sum_{k \in K} x_{ij}^k \leq u_{ij}, \quad \forall\, (i,j) \in A$. The resulting subproblem is a multicommodity minimum cost network flow problem, and we denote its optimal solution $\widetilde{x}$ and its optimal value $Z(MF)$. Clearly, $Z(MF)$ is a lower bound on Z; by adding the fixed costs of the arcs fixed to 1 by branching and variable fixing, we obtain an improved lower bound (Holmberg and Yuan, 2000). An upper bound can also be computed from the solution of MF by fixing to 1 the y variables corresponding to the arcs that are used: let $\widetilde{y}_{ij} = 1$, if $\sum_{k \in K} \widetilde{x}_{ij}^k > 0$, and 0, otherwise. The upper bound is then equal to $c\widetilde{x} + f\widetilde{y}$. Subproblem MF will be solved first at every node to determine if the node is feasible and to derive quick fathoming tests.

Branch-and-Cut Algorithm

The B&C algorithm is based on a depth-first strategy, where the cutting-plane procedure is called only at the root node and when the search backtracks. At all other nodes, the cutset subproblem is solved instead. The bounding procedure at each node starts by solving the multicommodity flow subproblem, then either the LP relaxation or the cutset subproblem. The bounding procedure accepts as input the best known upper bound, Z^*, and a Boolean variable *Backtrack*, which is set to *True* at the root node. It returns *Backtrack* = *True*, if the node is fathomed, or *Backtrack* = *False*, otherwise (then the branching operation will be performed). The steps of the bounding procedure are as follows:

1. *SolveMF* = *Backtrack*; *SolveLP* = *Backtrack*; *SolveCS* = *Backtrack*.
2. If *SolveMF*, solve subproblem MF; if MF is infeasible, *Backtrack* = *True* and stop; otherwise, let $(\widetilde{x}, \widetilde{y})$ its optimal solution and $Z(MF)$ its optimal value; if $Z(MF) + f\widetilde{y} < Z^*$, $Z^* = Z(MF) + f\widetilde{y}$.
3. If $Z(MF) + fy^* \geq Z^*$, *Backtrack* = *True* and stop ($y_{ij}^* = 1$, if arc (i,j) is fixed to 1, by branching and variable fixing, and 0, otherwise).

4. If *SolveLP*, apply the cutting-plane procedure; let $(\overline{x}, \overline{y})$ the optimal solution and $Z(LP)$ it optimal value; if $\overline{y}$ is integral: 1) if $Z(LP) < Z^*$, $Z^* = Z(LP)$; 2) *Backtrack* = *True* and stop.

5. If *SolveCS*, solve subproblem CS; let $\overline{y}$ its solution, $Z(CS)$ its optimal value, and $Z(LP) = Z(CS)$.

6. If $Z(LP) \geq Z^*$, *Backtrack* = *True* and stop.

7. Perform reduced cost fixing; if some variables are fixed, let *Fix* = *True*.

8. Generate local cuts; if some local cuts are generated, let *Local* = *True* and add these cuts to subproblem CS.

9. If *Fix*: if there is no variable y_{ij} fixed to δ (0 or 1) such that $\widetilde{y}_{ij} = 1 - \delta$, let *SolveMF* = *False*, otherwise *SolveMF* = *True*.

10. If *Local* or *Fix*, *SolveLP* = *False*, *SolveCS* = *True* and go to 2.

11. Perform upper bound improvement; if Z^* has improved, *SolveMF* = *False*, *SolveLP* = *False*, *SolveCS* = *False*, and go to 2; otherwise, *Backtrack* = *False*.

At the end of this procedure, if *Backtrack* = *False*, branching is performed. As remarked by Sellmann, Kliewer and Koberstein (2002), it is advantageous to branch on an arc (i, j) such that $\widetilde{y}_{ij} = 1$ and to explore first the branch where this arc is fixed to 1, since this favors the chance of fathoming in step 3. This way also, it is not necessary to solve the multicommodity flow subproblem; this explains why *SolveMF* is initialized to the value of *Backtrack*. Also, it is promising to branch on a variable y_{ij} such that $\widetilde{y}_{ij} \neq \overline{y}_{ij}$, if there is one, since then the solutions of the two subproblems (LP or CS, and MF) are different. Among these variables, we propose to branch on the variable that is the most uncertain, i.e., the one with the reduced cost closer to 0, since the arcs with large, positive or negative, reduced costs, are likely to be fixed by variable fixing, as remarked by Sellmann, Kliewer and Koberstein (2002). To summarize, the branching rule is the following: Determine the arc $(i, j)^*$ in $A_1 = \{(i, j) \in A | \widetilde{y}_{ij} = 1,\ \overline{y}_{ij} < 1\}$ with the smallest $|\overline{f}_{ij}(CS)|$ and explore first the branch where arc $(i, j)^*$ is fixed to 1; if A_1 is empty, determine the arc $(i, j)^*$ in $A_0 = \{(i, j) \in A | \widetilde{y}_{ij} = 0,\ \overline{y}_{ij} > 0\}$ with the smallest $|\overline{f}_{ij}(CS)|$ and explore first the branch where arc $(i, j)^*$ is fixed to 0; if $A_1 \cup A_0 = \emptyset$ (i.e., $\widetilde{y} = \overline{y}$), determine the arc $(i, j)^*$ with the smallest $|\overline{f}_{ij}(CS)|$ and explore first the branch where arc $(i, j)^*$ is fixed to $1 - \overline{y}_{ij}$. Without entering into the details of the upper bound improvement method, step 11, we note that it is based on solving multicommodity flow problems derived from the solution of the cutset subproblem.

References

[1] Chouman M., Crainic T.G. and Gendron B. (2003), "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.

[2] 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.

[3] Sellmann M., Kliewer G. and Koberstein A. 2002, “Capacitated Network Design, Cardinality Cuts and Coupled Variable Fixing Algorithms based on Lagrangian Relaxations”, Technical report tr-ri-02-234, Department of Mathematics and Computer Science, University of Paderborn.