CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A MIP-Tabu Search Hybrid Framework for Multicommodity Capacitated Fixed-Charge Network Design

Mervat Chouman
Teodor Gabriel Crainic

June 2010

CIRRELT-2010-31

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

UNIVERSITÉ LAVAL UQÀM HEC MONTRÉAL ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

A MIP-Tabu Search Hybrid Framework for Multicommodity Capacitated Fixed-Charge Network Design

Mervat Chouman[1], Teodor Gabriel Crainic[2,*]

[1] College of Business, Effat University, P.O. Box 34689, Jeddah 21478, Saudi Arabia

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), and Department of Management and Technology, Université du Québec à Montréal, C.P. 8888, succursale Centre-ville, Montréal, Canada H3C 3P8

Abstract. This paper addresses the capacitated multicommodity fixed-charge network design problem, a particular case of the broad class network design problems. To identify good quality solutions, we propose a matheuristic combining an exact MIP method and a Tabu search meta-heuristic. A new arc-balanced cycle procedure for generating neighboring solutions is introduced. To assess the quality of the solutions, tight lower bounds are computed using a cutting-plane procedure. During the computation process, memories characterizing attributes of solutions are compiled. These memories are then used to find good quality starting solutions in short computational times. Computational results conducted on a large set of instances with various characteristics show an efficient behavior of the proposed method. The method provides high quality feasible solutions which outperformed those obtained by state-of-the-art optimization codes within the time allowed.

Keywords. Capacitated multicommodity fixed-charge network design, matheuristics, cutting planes, tabu search.

Acknowledgements. Partial funding for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs, and by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec.

---

* Corresponding author: Teodor-Gabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2010

1 Introduction

Network design formulations are used to model a wide variety of problems in several fields such as transportation, logistics, distribution-production, etc. For survey on network design, see Magnanti and Wong (1984), Minoux (1989), and Crainic (2000). Briefly, these formulations are characterized by a network where all or some of the links are optional and may be used providing a corresponding fixed cost is paid, and a set of known demands that must be routed between particular origins-destination pairs at given link unit flow costs. One then aims to design a network, that is, select the links to be used, to satisfy the demand, by determining the flow distribution, at minimum cost total cost computed as the total cost of the selected links and the total cost of the flow distribution.

We focus on the capacitated multicommodity fixed-charge network design problem, denoted CMND, defined on an oriented graph. The CMND is a particular case of the broad class of network design problems and is NP-hard (Magnanti and Wong (1984)). Other than the problem size, considerable algorithmic challenges are due also to the trade-offs between variable and fixed costs, fixed costs and capacity, the competition among commodities (the origin-destination pair demands) for the limited capacity of arcs, and the degeneracy of the linear relaxation of the formulation. The quest for efficient solution methods has thus aimed for both exact and approximate methods. Interestingly enough, most successful heuristics apply matheuristics principles, several having been proposed before the term was coined (e.g., Crainic et al. (2000), Ghamlouche et al. (2003), Hewitt et al. (2009)). Our objective is to continue this line of development and propose a new matheuristic for a family of CMND problems.

We propose a general framework methodology based on the integration of mathematical programming methods (MP), linear (LP) or mixed integer programming (MIP), with a neighbourhood-based meta-heuristic. The model formulation is at the core of the exact part of the methodology and also guides and is influenced by the meta-heuristic. The exact part of the method includes lower bound computation based on the cutting-plane method of Chouman et al. (2009) and a MP algorithm performed on restricted formulations of the problem. The restricted, but simpler, formulations are obtained through partial variable fixing. The upper bound method is a Tabu search (TS) meta-heuristic (Glover (1986), Glover and Laguna (1997)) based on the exploration of the space of the arc-design variables. The basic neighborhood structure is identified and explored using a new arc-balanced cycle procedure that computes shortest paths on partial graphs associated to candidate arcs. Then, the MP method aims to identify good-quality feasible solutions on restricted formulations, while the TS aims to improve them. Our basic strategy is to continuously perform a learning process to enhance the performance of the method. More precisely, we show how to take advantage of the time spent computing the lower and upper bounds to compile memories characterizing attributes of encountered solutions, memories that are then used to perform variable fixing.

The goal of this paper is to introduce the methodology and to present a proof-of-concept though a basic application to the CMND. Section 2 recalls the problem formulation and previous contributions. Section 3 describes the methodology approach, including the lower bound computation procedure, the first feasible solution, the tabu search algorithm, and the intensification mechanism. First computational results are summed up in Section 4. Perspectives on the development and applications of the methodology are presented in Section 5.

2 Problem Formulation

Given a directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs, and a set of commodities (or origin-destination pairs) $\mathcal{K}$ to be routed according to a known demand w^k for each commodity k, the problem is to satisfy the demand at minimum cost. The cost consists of the sum of transportation costs and fixed design costs, the latter being charged whenever an arc is included in the optimal design. The transportation cost per unit of commodity k on arc (i, j) is denoted $c^k_{ij} \geq 0$, while the fixed design cost for arc (i, j) is denoted $f_{ij} \geq 0$. A limited capacity, denoted u_{ij}, is associated to each arc (i, j). An origin $O(k)$ and a destination $D(k)$ are associated to each commodity k. We introduce continuous flow variables x^k_{ij}, which reflect the amount of flow on each arc (i, j) for each commodity k, and 0-1 design variables y_{ij}, which indicate if arc (i, j) is used or not. With this notation, the mathematical formulation of the CMND is as follows:

$$\min_{x,y} \quad \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} c^k_{ij} x^k_{ij} + \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij}, \tag{2.1}$$

$$\sum_{j\in\mathcal{N}^+_i} x^k_{ij} - \sum_{j\in\mathcal{N}^-_i} x^k_{ji} = d^k, \qquad \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}, \tag{2.2}$$

$$\sum_{k\in\mathcal{K}} x^k_{ij} \leq u_{ij} y_{ij}, \quad \forall\, (i, j) \in \mathcal{A}, \tag{2.3}$$

$$x^k_{ij} \geq 0, \qquad \forall\, (i, j) \in \mathcal{A},\ \forall\, k \in \mathcal{K}, \tag{2.4}$$

$$y_{ij} \in \{0, 1\}, \quad \forall\, (i, j) \in \mathcal{A}, \tag{2.5}$$

where

$$d^k = \begin{cases} w^k, & \text{if } i = O(k), \\ -w^k, & \text{if } i = D(k), \\ 0, & \text{otherwise,} \end{cases} \qquad \begin{array}{l} \mathcal{N}^-_i = \{j \in \mathcal{N} : (j, i) \in \mathcal{A}\}, \\ \mathcal{N}^+_i = \{j \in \mathcal{N} : (i, j) \in \mathcal{A}\}. \end{array}$$

The objective function (2.1) minimizes the total cost computed as the sum of the total transportation cost for commodities and the total fixed cost for arcs included in the optimal design (denoted as open). Constraints (2.2) correspond to flow conservation equations for each node and each commodity. Relations (2.3) represent capacity constraints for each arc. These constraints also link together flow and design variables by forbidding any flow to pass through an arc that is not chosen as part of the design. Note that, the linear relaxation (LP) is obtained by replacing the integrality constraints (2.5) by $0 \leq y_{ij} \leq 1, \ \forall (i, j) \in \mathcal{A}$.

Several bounding procedures based on classical relaxation approaches have been developed for the CMND. In particular, Lagrangian-based methods have been proposed in Crainic et al. (1999), Crainic et al. (2001), Gendron and Crainic (1996), Holmberg and Yuan (2000), Kliewer and Timajev (2005), and Sellmann et al. (2002), while Benders-based decomposition procedures have been proposed in Costa et al. (2009). Chouman et al. (2009) investigated new separation and lifting procedures of well known valid inequalities applied to the CMND and proposed a cutting planes procedure to improve its formulation and hence the quality of its LP-bounds. Tabu search-based meta-heuristics were proposed in Crainic et al. (2000), Crainic and Gendreau (2002), Crainic et al. (2004), Ghamlouche et al. (2003), Ghamlouche et al. (2004), Crainic et al. (2006).

3 Methodological Approach

The basic version of the proposed math-heuristic is illustrated in Figure 1. Roughly speaking, the MP method identifies feasible solutions, while the TS meta-heuristic aims to improve these solutions by an exploration of a large-scale neighboring space.

An initial phase executes the cutting-plane method, gathers statistical information on the solutions generated, and uses this information to fix a number of variables . The resulting restricted problem is then solved by a LP or a MIP method, depending on the characteristics of the problem. The resulting feasible solution constitutes the starting solution of the meta-heuristic. Statistical information is also gathered during this phase and is used to guide the meta-heuristic through variable fixing and the resolution of the corresponding restricted formulation. Obviously, appropriate memories are used to keep track of the evolution of the method and avoid repeatedly fixing the same set of variables. In the following, we briefly present the building blocks of the method for the CMND problem.

3.1 Lower Bound Computation and Memories

We aim for a good trade-off between quality and time. We therefore use an efficient version of the cutting-plane method proposed in Chouman et al. (2009), which includes

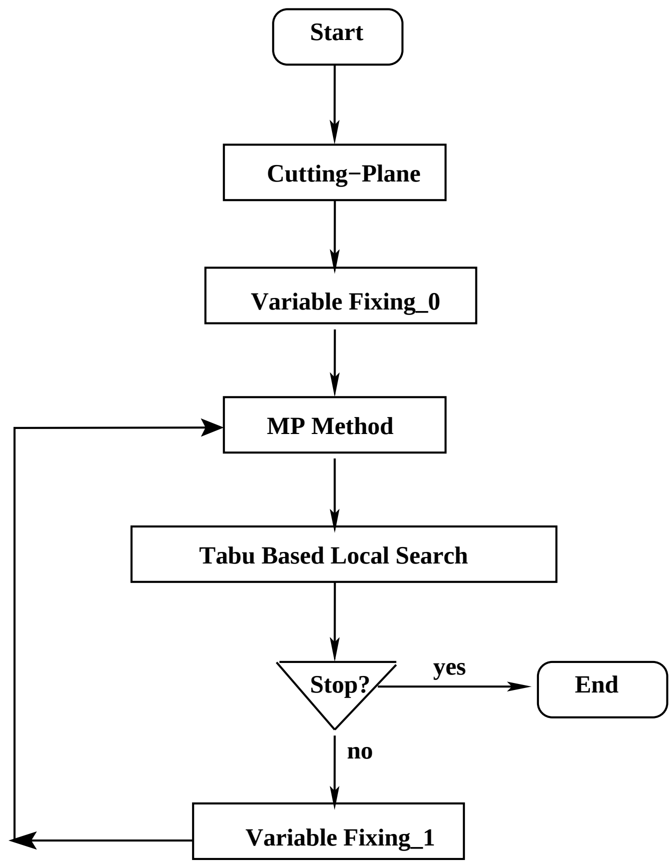

Figure 1: Schematics of the Basic Version of the Proposed Matheuristic

only three families of valid inequalities: the strong (SI), the cover(CI), and the flow pack (FPI). Details, including the respective separation problems and implementation issues may be found in Chouman et al. (2009).

The cutting-plane algorithm iterates solving the LP relaxation, generating violated valid inequalities and adding them to the LP formulation, solving again the LP, and so on. The families of cuts are generated in the SI, CI, and FPI sequence. The procedure terminates when either the optimal solution is found, which is unlikely to happen very often, or when the improvement is smaller than ϵ.

Three series of statistical information on attributes of the LP solutions are gathered during the cutting planes computations. The corresponding memories (all initialized at zero) are:

F: Frequency of opening a design arc; Updated at each LP solution $(\bar{x}, \bar{y})$ by setting

$$F_{ij} = F_{ij} + 1, \text{if } \bar{y}_{ij} > \beta,\ \forall (i,j) \in \mathcal{A},$$

where β is a threshold on when an arc (i,j) in a LP solution would actually belong to the final design.

L: Frequency of design arcs to be included in violated cover inequalities; Updated for each identified violated CI

$$L_{ij} = L_{ij} + 1,\ \forall (i,j) \in \mathcal{C},$$

where $\mathcal{C} \subseteq \mathcal{A}$ is the minimal cover obtained in the CI.

R Accumulated reduced cost for design arcs; Updated at each LP solution:

$$R_{ij} = R_{ij} + \bar{R}_{ij},\ \forall (i,j) \in \mathcal{A},$$

where $\bar{R}$ is the reduced cost vector associated to the current LP optimal solution $(\bar{x}, \bar{y})$.

3.2 First feasible solution

The main goal is to find rapidly a good starting feasible solution. To obtain it, we first reduce the size of the problem, and then solve the reduced CMND by a MIP algorithm. The selection of arcs to open/close must thus yield a network sufficiently small to be computationally tractable and with sufficient capacity and connectivity to satisfy the demand.

The heuristic we propose, denoted α_0-*fixing*, selects the reduced set $\widetilde{\mathcal{A}}$ based on the information gathered by the cutting-plane algorithm in the memories F, L, and R. The heuristic selects arcs in three steps denoted, respectively, consistency, connectivity, and feasibility. The first step selects arcs that consistently appeared in the optimal solutions of the LP relaxations, i.e., arc (i, j) is added to $\widetilde{\mathcal{A}}$ if $F_{ij} \geq \alpha_0 N_{LP}$, where α_0 is a predefined threshold and N_{LP} is the number of LPs solved during the cutting-plane procedure.

The connectivity step aims to ensure a connected reduced network, by making sure that all transfer nodes in $\widetilde{\mathcal{A}}$ with at least one outgoing/ingoing arc opened, have also at least one ingoing/outgoing arc opened. The arcs with high values in F were selected in the previous step, so we now turn to the L memory for guidance. Actually, a frequent appearance of an arc in a minimal cover of a violated CI means that the arc has a good chance to be open in feasible solutions and thus represents a good candidate for this heuristic. Define $\widetilde{\mathcal{N}}_i^- = \{j \in \mathcal{N}_i^- : (j, i) \in \widetilde{\mathcal{A}}\}$ and $\widetilde{\mathcal{N}}_i^+ = \{j \in \mathcal{N}_i^+ : (i, j) \in \widetilde{\mathcal{A}}\}$. Then, for each node $i \in \mathcal{N}$ with at least one ingoing/outgoig arc and no outgoing/ingoing arc, add to $\widetilde{\mathcal{A}}$ the arc that realizes $\text{argmax}_{j \in \mathcal{N}_i^+ \setminus \widetilde{\mathcal{N}}_i^+}(F_{ij} + L_{ij})$ or $\text{argmax}_{j \in \mathcal{N}_i^- \setminus \widetilde{\mathcal{N}}_i^-}(F_{ji} + L_{ji})$, respectively.

The feasibility step aims to provide sufficient capacity at each origin or destinati0on node for flow to get out or into it. The choice of arcs in this step is made using the information provided in the average reduced cost memory R, arcs with low accumulated reduced costs being good candidates. Thus, for any origin i with insufficient outgoing capacity (i.e., $\sum_{j \in \widetilde{\mathcal{N}}_i^+} u_{ij} < w^k$), Add to $\widetilde{\mathcal{A}}$ the arc that realizes: $\text{argmin}_{j \in \mathcal{N}_i^+ \setminus \widetilde{\mathcal{N}}_i^+}(R_{ij})$. Similarly, for any destination i with insufficient ingoing capacity, add the arc that realizes: $\text{argmin}_{j \in \mathcal{N}_i^- \setminus \widetilde{\mathcal{N}}_i^-}(R_{ji})$.

Once the set $\widetilde{\mathcal{A}}$ is determined, we consider the restriction CMND$_{\widetilde{\mathcal{A}}}$ where all arcs in $\widetilde{\mathcal{A}}$ are free and all arcs in $\mathcal{A} \setminus \widetilde{\mathcal{A}}$ are closed. The restricted CMND$_{\widetilde{\mathcal{A}}}$ problem is solved using any available commercial MIP software. Time and node limits are imposed. If the problem is unfeasible, we decrease the value of α_0 and we repeat the α_0-fixing heuristic to find a new and larger set $\widetilde{\mathcal{A}}$. We iterate as needed.

According to our computational results, this heuristic has proved effective to identify rapidly high quality feasible solutions which are competitive with those found by the best heuristic approaches available in the literature.

3.3 Tabu Search

In the current version of the methodology, the local search component of TS is used to explore the design-variable search space, starting from the feasible solution coming from the MIP method. The procedure is based on a large neighborhood defined by moves obtained by performing an *Arc-Balanced Cycle* procedure. Neighbors are evaluated by solving the associated minimum cost network flow problem using a LP software and the best neighbor is chosen to be the next incumbent. The method allows the deterioration of the solution quality when the incumbent is a local optimum. Recent moves are included in the tabu list, with a random number of iterations forbidding a closed arc to be included in any move. Besides the tabu list, a memory, F^{TS}, is updated at each iteration to record the frequency of an arc being open in TS solutions. This memory is used to identify new feasible solutions in the intensification phase (Section 3.4). The heuristic stops when no improvement in the best solution is observed for a prespecified number of iterations or the overall computational time exceeds a time-limit.

The main feature of the proposed TS method is the definition of the *arc-balanced cycle* neighborhood/move. Inspired by the cycle-based move introduced in Ghamlouche et al. (2003), the new procedure preserves the number of open in and out arcs at each node (hence the name of the move), which makes it suitable for a broad range of problem settings, e.g., the network design problems with arc-balance requirements Pedersen et al. (2009).

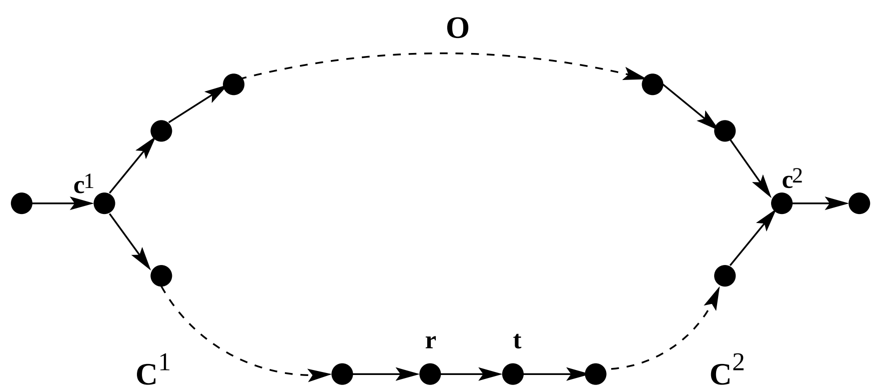

Figure 2: Structure of the arc-balanced cycle

The move aims to close a selected arc by deviating the flow moving on it, more precisely, by replacing a sub-path containing the arc by a different sub-path. Note that this deviations may appear promising when the move is generated but may prove unfeasible from the point of view of the flow distribution. Then, in this version of the procedure, the neighbor is discarded and the next one is evaluated. Thus, to close an arc (r,t),we aim to find cycles with the special property, denoted *three-phases* property, given by $(C^1, \{(r,t)\}, C^2, O)$ illustrated in Figure 2, where

$\mathcal{C}^1 \subseteq \widetilde{\mathcal{A}}$ is a subset of successive open arcs making up a path between a given node c^1 and r;

$\mathcal{C}^2 \subseteq \widetilde{\mathcal{A}}$ is a subset of successive open arcs representing a path between t and a given node c^2;

$\mathcal{O} \subseteq \mathcal{A} \backslash \widetilde{\mathcal{A}}$ is a subset of successive closed arcs representing a path between c^1 and c^2.

The move then consists in moving the flow, or a sub-flow, passing on the sub-path $\mathcal{C}^1 \cup \{(r,t)\} \cup \mathcal{C}^2$ to the sub-path $\mathcal{O}$. Closing the full sub-path $\mathcal{C}^1 \cup \{(r,t)\} \cup \mathcal{C}^2$ and replacing it by the sub-path $\mathcal{O}$, could be too restrictive to obtain feasible neighbors. Instead, we propose to close only a few arcs. Let δ given by $\delta = \min \{ \{\sum_{k \in \mathcal{K}} x_{ij}^k, \forall (i,j) \in \mathcal{C}^1 \cup \{(r,t)\} \cup \mathcal{C}^2\} \cup \{u_{ij}, \forall (i,j) \in \mathcal{O}\} \}$ be the maximum flow that can be moved using the identified cycle. The move is then performed by closing all arcs in $\mathcal{C}^1 \cup \{(r,t)\} \cup \mathcal{C}^2$ carrying a flow equal to δ and opening all arcs in O. Moreover, we concentrate the search for $\mathcal{C}^1$ and $\mathcal{C}^2$ on arcs carrying the same subset of commodities as those moving on the arc to close (r,t). By doing so, we aim to increase the number of arcs to close (those shared by the same commodities) while increasing the chance to get flow-feasible arc-balanced cycles. In addition, such selection of arcs enables the generation of neighbors that are not too close to the current solution.

The procedure to identify an arc-balanced cycle move is performed in several steps. First, one determines the subset of opened arcs that could be used in a cycle involving the arc to close. This is done by simple labeling procedures starting from r and t and going backward and forward, respectively. The next step consists in identifying the restricted network used to find the cycle including opened and closed arcs by reversing all selected opened arcs and by associating to each of them their negative fixed cost,$-f_{.}$. Then, one adds all the closed arcs with their associated fixed cost. The next step finds a shortest path in the restricted network from r to t by applying a particular labeling correcting procedure that yields non-negative cycles satisfying the three-phases property.

For each current solution, there are many possible moves since any opened arc with positive flow is a possible candidate. Evaluating all such moves would be computationally prohibitive, particularly since we use a commercial LP solver for the minimum cost network flow problem. We therefore build candidate lists based on the following three different criteria:

Fixed cost, to reach a good balance between the fixed cost of the arc and the amount of flow moving through. Thus, we define the $|\mathcal{A}|$-dimensional vector ξ^1:

$$\xi_{ij}^1 = \begin{cases} \frac{f_{ij}}{\sum_k x_{ij}^k}, & \text{if } \sum_k x_{ij}^k > 0 \\ 0, & \text{otherwise} \end{cases}$$

Capacity, to reach a good level of used capacity with respect to the flow moving on the arc. Thus, we define the $|\mathcal{A}|$-dimensional vector ξ^2:

$$\xi_{ij}^2 = \begin{cases} \frac{\sum_k x_{ij}^k}{u_{ij}}, & \text{if } \sum_k x_{ij}^k > 0 \\ 0, & \text{otherwise} \end{cases}$$

Unit price, to achieve a good level of unit price. Thus, we define the $|\mathcal{A}|$-dimensional vector ξ^3:

$$\xi^3_{ij} = \begin{cases} \frac{f_{ij}+\sum_k c^k_{ij}x^k_{ij}}{u_{ij}}, & \text{if } \sum_k x^k_{ij} > 0 \\ 0, & \text{otherwise} \end{cases}$$

Candidate lists are then defined by choosing the first *MaxCandidList* of non-tabu candidates from each considered vector, where ξ^1 and ξ^3 are sorted in descending order and ξ^2 is sorted in ascending order. The respective orders are chosen with respect to the idea of closing expensive arcs used to move only small amount of flow in the current solution.

3.4 Intensification - New Feasible Solution

Motivated by the idea of using an exact method to revitalize the tabu search algorithm, we include an intensification mechanism around a set of *good* attributes. Thus, when the local search phase of the TS method appears trapped at a local optima, a new feasible solution is identified by reducing first the size of the CMND problem and next by solving the resulting restricted problem using an available commercial MIP software.

Contrary to the α_0-fixing heuristic, here we reduce the size of the problem by opening some arcs in the network and leaving all the remaining arcs free for the MIP method. The selection is performed randomly among the arcs with a frequency in F^{TS} higher than α_1.

4 Computational Results

We tested this basic variant of the methodology - cutting planes, variable fixing + MIP, tabu search, variable fixing + MIP, tabu search - on the benchmark problem sets C and R used also in Ghamlouche et al. (2003) and Hewitt et al. (2009)). The parameter values are given in Table 1.

Table 2 displays results for the problems is set C. The first column gives the identification of the instance in terms of numbers of nodes, arcs (all are design arcs), and commodities, as well as whether the dominating costs are the Variable or the Fixed ones and whether the total possible capacity it relatively Loose or Tight with respect to the total demand. Columns *LB*, *BS*, and *FS* display the solution values of, respectively, the lower bound produced by the cutting-plane procedure, the best solution of the proposed

β	0.3
α_0	0.45
α_1	0.90
$NbIterTS$	10
$NbIterProcess$	3
$MaxCandidList$	7

Table 1: Parameter values

Instance	LB	BS	FS	cpu	cpu-BS	Gap	Gap-FS	BS-2h	Gap-FS-2h
100,400,10,V,L	27915	28423	29693	93	49	1.79	5.99	28423	-4.55
100,400,10,F,L	21615	24161	27882	308	109	10.54	22.48	24161	-17.21
100,400,10,F,T	57230	67233	68374	1283	918	14.88	16.30	67233	-1.99
100,400,30,V,T	382584	384940	385217	279	85	0.61	0.68	384940	-0.07
100,400,30,F,L	46570	49682	58697	4758	2068	6.26	20.66	49682	-19.36
100,400,30,F,T	126629	144349	147957	1789	145	12.28	14.41	144349	-2.85
20,230,40,V,L	422998	423848	424385	203	116	0.20	0.33	423848	-0.13
20,230,40,V,T	636882	643538	658887	148	37	1.03	3.34	643538	-2.41
20,230,40,F,T	370016	371475	371800	157	63	0.39	0.48	371475	-0.09
20,230,200,V,L	91359	94218	94295	13229	8328	3.03	3.11	94254	-0.08
20,230,200,F,L	132500	138491	144990	22047	15006	4.33	8.61	140192	-4.90
20,230,200,V,T	95672	98612	98742	7849	3965	2.98	3.11	98612	-0.14
20,230,200,F,T	131611	136309	136309	10243	835	3.45	3.45	136309	0.00
20,300,40,V,L	427947	429398	430140	178	33	0.34	0.51	429398	-0.17
20,300,40,F,L	575530	588464	590164	239	46	2.20	2.48	588464	-0.30
20,300,40,V,T	461821	464509	464628	290	161	0.58	0.60	464509	-0.03
20,300,40,F,T	600867	604198	605505	168	35	0.55	0.77	604198	-0.22
20,300,200,V,L	73127	75045	75555	10804	4476	2.56	3.21	75045.3	-0.70
20,300,200,F,L	110977	116259	117438	15207	9109	4.54	5.50	116259	-1.06
20,300,200,V,T	74048	74995	75834	5924	1913	1.26	2.36	74995	-1.13
20,300,200,F,T	103776	109164	109164	7648	542	4.94	4.94	109164	0.00
30,520,100,V,L	53141	54008	54138	3587	2415	1.60	1.84	54008	-0.24
30,520,100,F,L	90304	93967	94588	8711	2925	3.90	4.53	93967	-0.69
30,520,100,V,T	51486	52156	52329	4175	2521	1.29	1.61	52156	-0.34
30,520,100,F,T	94272	97490	98754	12043	4161	3.30	4.54	97490	-1.34
30,520,400,V,L	111770	112927	112948	36025	22797	1.02	1.04	112947	-0.02
30,520,400,F,L	146726	149920	149920	36159	5769	2.13	2.13	149920	0.00
30,520,400,V,T	114061	114664	114869	38793	38793	0.53	0.70	114847	-0.18
30,520,400,F,T	149798	152929	152929	36041	8556	2.05	2.05	152929	0.00
30,700,100,V,L	47309	47603	47678	5331	3938	0.62	0.77	47603	-0.16
30,700,100,F,L	58209	60184	60509	10464	5650	3.28	3.80	60184	-0.56
30,700,100,V,T	45195	45880	46132	6156	4263	1.49	2.03	45880	-0.56
30,700,100,F,T	53783	54926	55823	6772	3018	2.08	3.65	54925.5	-1.67
30,700,400,V,L	96605	97982	98015	37540	35241	1.41	1.44	98015	-0.03
30,700,400,F,L	130779	135109	135166	36679	21429	3.20	3.25	135166	-0.04
30,700,400,V,T	94013	95781	95886	36327	15372	1.85	1.95	95886	-0.11
30,700,400,F,T	127572	130856	131087	38184	32531	2.51	2.68	131087	-0.18
Average				12320	6957	3.00	4.36		-1.72

Table 2: MIP-TS results for C problems: 37 instances

MIP-TS matheuristic, and the value of the first feasible solution (obtained solving the restricted CMND obtained out of the cutting-plane procedure). The following two columns display the total CPU time required by the proposed method (limited at 10 hours) and to identify the best solution, respectively. The *Gap* and *Gap-FS* columns give the gaps relative to the lower bound of the best and first feasible solutions, respectively. Finally, to explore the efficiency of the method, the last two columns display the best-solution value after 2 hours of CPU time and the improvement gap of that solution relative to the first feasible one. These results show that the tabu search and the associated variable fixing procedure improve, in some cases quite significantly, the solution obtained from the lower bound.

Comparisons to leading methods in the literature, CPLEX, the Path Relinking of

Ghamlouche et al. (2004), and the matheuristic of Hewitt et al. (2009), show that the approach we propose performs very well. The performance measures are the number of instances of each set the MIP-TS method improves compared to the other method, and the average gap over all instances in the set between the two methods.

Tables 3 and 4 sum up the results. Both tables display in the first column the comparison with CPLEX after 2 hours CPU time. The second and last columns of both tables are dedicated to the comparison with the Path Relinking (identified as *TS-PR*) and the first feasible solution, respectively (the proposed procedure stopped normally). Table 3 also displays the comparison to the matheuristic of Hewitt et al. (2009) (identified as *IP*).

	BS-2H $\leq$ CPLEX-2H	BS $\leq$ TS-PR	BS $\leq$ IP	BS $<$ FS
Number of Instances	16	33	31	33
Average Gap	0.33%	-3.59%	-0.81%	-1.72%

Table 3: Improved feasible solution over 37 C-instances

	BS-2H $\leq$ CPLEX-2H	BS $\leq$ TS-PR	BS $<$ FS
Number of Instances	25	33	39
Average Gap	0.54%	-4.48%	-1.30%

Table 4: Improved feasible solution over 54 R-instances

The results indicate that the proposed matheuristic performs very well and even in its simplest form appears to be competitive with the best current methods. It actually compares very well with the exact algorithms in CPLEX for moderate-size instances and outperforms the leading meta-heuristics in the literature.

5 Perspectives

We introduced a general framework methodology based on the integration of mathematical programming methods and neighborhood-based meta-heuristics for the capacitated multicommodity fixed-charge network design problem. The exact part of the method includes lower bound computation by a cutting-plane method and an exact algorithm performed on restricted formulations of the problem. T The upper bound method is a Tabu search meta-heuristic, which explores the space of the arc-design variables through a new arc-balanced cycle-based neighborhood. The restricted formulations are obtained through partial variable fixing based on long-term memories build by the cutting-plane and the tabu search methods.

First computational results on standard benchmark instances indicate that the proposed method performs very well. It actually compares very well with the exact algorithms in CPLEX for moderate-size instances and outperforms the leading meta-heuristics in the literature. The method may also be easily generalized to other classes of network design problems, e.g., the design-balanced formulations of Pedersen et al. (2009), for which the preliminary results are very encouraging. We are also developing a more advanced version of the method where the information in the tabu search long-term memories are used to generate new cuts, the method returning to the cutting-plane algorithm for a major diversification phase.

Acknowledgments

While working on this project, T.G. Crainic was the NSERC Industrial Research Chair on Logistics Management, ESG UQAM, and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway, while M. Chouman was postdoctoral scientist with the Chair.

Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs, and by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec.

References

M. Chouman, T. Crainic, and B. Gendron. A cutting-plane algorithm for multicommodity capacitated fixed charge network design. Technical Report CIRRELT-2009-20, Interuniversity Research Centre onEnterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, 2009.

A. Costa, J. Cordeau, and B. Gendron. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications*, 42:371–392, 2009.

T. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.

T. Crainic and M. Gendreau. Cooperative parallel tabu search for capacitated network design. *Journal of Heuristics*, 8:601–627, 2002.

T. Crainic, A. Frangioni, and B. Gendron. Multicommodity capacitated network design. In P. Soriano and B. Sanso, editors, *Telecommunications Network Planning*, pages 1–19. Kluwer Academics Publisher, 1999.

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

T. Crainic, B. Gendron, and G. Hernu. A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10(5):525–545, 2004.

T. Crainic, Y. Li, and M. Toulouse. A First Multilevel Cooperative Algorithm for the Capacitated Multicommodity Network Design. *Computers & Operations Research*, 33 (9):2602–2622, 2006.

T. G. Crainic, M. Gendreau, and J. Farvolden. A simplex-based tabu search method for capacitated network design. *INFORMS Journal on Computing*, 12(3):223–236, 2000.

B. Gendron and T. Crainic. Bounding procedures for multicommodity capacitated fixed charge network design problem. Technical Report CRT-96-06, Center for research on transportation, Université de Montréal, 1996.

I. Ghamlouche, T. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed charge capacitated multicommodity network design. *Operations Research*, 51(4):655–667, 2003.

I. Ghamlouche, T. Crainic, and M. Gendreau. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals of Operations Research*, 131: 109–133, 2004.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 1(13):533–549, 1986.

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, 1997.

M. Hewitt, G. Nemhauser, and M. Savelsbergh. Combining Exact and Heuristic Approaches for the Capacitated Fixed Charge Network Flow Problem. *INFORMS Journal on Computing*, 2009. to appear.

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.

G. Kliewer and L. Timajev. *Relax-and-cut for capacitated network design*, pages 47–58. Springer Berlin-Heidelberg, 2005.

T. Magnanti and R. Wong. Network design and transportation planning: models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

M. Minoux. Network synthesis and optimum network design problems: models, solution methods and applications. *Networks*, 19:313–360, 1989.

M. Pedersen, T. Crainic, and O. Madsen. Models and tabu search meta-heuristics for service network design with asset-balance requirements. *Transportation Science*, 43(2): 158–177, 2009.

M. Sellmann, G. Kliewer, and A. Koberstein. Capacitated network design, cardinality cuts and coupled variable fixing algorithms based on lagrangian relaxations. Technical Report tr-ri-02-234, Department of Computer Science, University of Paderborn, 2002.