CIRRELT

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

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

MIP-Based Matheuristic for Service Network Design with Design-Balanced Requirements

Mervat Chouman
Teodor Gabriel Crainic

December 2012

CIRRELT-2012-79

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

MIP-Based Matheuristic for Service Network Design with Design-Balanced Requirements

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

[1] Department of Operation and Information Management, Effat University, Jeddah, Kingdom of Saudi Arabia

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

Abstract. The paper introduces a MIP-based matheuristic for the design-balanced capacitated multicommodity network design problem, one of the premier formulations for the service network design problem with asset management concerns increasingly faced by carriers within their tactical planning processes. The matheuristic combines a cutting-plane procedure efficiently computing tight lower bounds and a variable-fixing procedure feeding a MIP solver. Learning mechanisms embedded into the cutting-plane procedure provide the means to identify promising variables and thus, both reduce the dimension of the problem instance making it addressable by a MIP solver, and guide the latter toward promising solution spaces. Extensive computational experiments show the efficiency of the proposed procedures in obtaining high-quality solutions, outperforming the current best methods from the literature.

Keywords. Service network design, design-balanced constraints, tabu search, cutting-plane method, matheuristic.

Acknowledgements. While working on this paper, 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. M. Chouman was postdoctoral scientist with the CIRRELT and the Chair. The authors thank Ms. Geneviève Hernu, analyst with the Chair, for her much welcome assistance in conducting the experimental phase of this project, and the CIRRELT for access to the material and human resources of its computing network. 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 ministère des Transports du Québec. The computing equipment principally used was acquired with the support of the Canadian Foundation for Innovation and its partners, mainly the Ministère de l'Éducation, du Loisir et du Sport and UQAM.

---

* Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2012

1 Introduction

Network design formulations are used to model a wide variety of problems in several fields such as transportation, logistics, distribution, production, etc. Surveys on network design may be found in Magnanti and Wong (1984), Minoux (1989), and Crainic (2000). We are particularly interested in fixed-cost, multicommodity, capacitated formulations characterized by a network with link capacities and a set of known demands between origin-destination nodes. The network design problem then aims to construct a network, by choosing the arcs to be used, and to satisfy the demand, by determining the flow distribution on each arc, at minimum cost. A fixed cost is payed as soon as a link is used, in addition to the usual per-unit routing cost.

Service network design belongs to this broad problem class, where links represent "services" to operate within a given system. Service network design is particularly used to address tactical planning issues for consolidation-based transportation carriers (Crainic, 2003; Crainic and Kim, 2007). More precisely, it relates to the decision problem of selecting transportation services to operate over a mid-term planning horizon, together with their frequencies or schedules as well as the main strategies for moving loads through the resulting service network, to optimize the economic and service criteria of the carrier and achieve an efficient allocation and utilization of its resources, given forecast origin-to-destination demand. The result of the tactical planning process usually is a transportation plan and schedule for a given time length, e.g., a day or a week, to be repeatedly operated over the planning horizon of the "next season" (i.e., from a few months to a year). One calls such a schedule periodic and circular.

The management of assets, e.g., power units, vehicles, crews, etc., was generally not detailed in most of the contributions in the literature (Crainic, 2003; Crainic and Kim, 2007), with a few exceptions where the cost of owning and operating particular assets, planes or ships, for example, was dominating the other cost considerations (e.g., Armacost et al., 2002; Smilowitz et al., 2003; Lai and Lo, 2004). Constraints requiring that the same number of assets enter and exit each terminal, called *design-balanced constraints* by Pedersen et al. (2009), and ad-hoc solution methods were proposed to better include resource-management considerations within tactical planning processes and models. Such approaches are becoming wide spread, as so-called full-asset-utilization policies aiming to use assets continuously following circular routes (Crainic and Kim, 2007; Bektas and Crainic, 2008) are being adopted by carriers of all modes. Andersen et al. (2009b,a) give an up-to-date review of previous contributions to the field and study formulations for various asset-management considerations within service network design settings.

In this paper, we focus on the *design-balanced capacitated multicommodity network design* problem (*DBCMND*), a generic network design problem with design-balanced requirements, formally introduced in Pedersen et al. (2009) together with a Tabu Search meta-heuristic. The DBCMND is NP-Hard, as it is a special case of the well-known

NP-Hard multicommodity fixed charge network design problem (Magnanti and Wong, 1984) and, thus, exact methods reach their limits rather rapidly (Andersen et al., 2011). Moreover, as shown , even feasible solutions are difficult to obtain for the DBCMND, as shown by Pedersen et al. (2009) and the recent Tabu Search - Path Relinking metaheuristic of Vu et al. (2012).

Our goal is to address these challenges and propose a methodology to efficiently identify good-quality feasible solutions for realistically-dimensioned instances. We propose a matheuristic combining an exact lower-bound computing method and a variable-fixing procedure feeding a MIP solver. We use a well-known commercial software for the latter. The former method is based on the cutting-plane procedure proposed by Chouman et al. (2009, 2011) for the *capacitated multicommodity fixed charge network design* problem (*CMND*) and computes tight lower bounds in short computational times. We introduce learning mechanisms embedded into the cutting-plane procedure, which yield the information used to fix variables and guide the MIP solver toward good-quality solutions. The performance of the proposed matheuristic is evaluated through an extensive computational study performed on a large set of test instances used in the literature. Comparisons with leading methods in the literature underlines the quality and efficiency of the proposed matheuristic.

The contributions of this paper are twofold. First, we propose a new matheuristic for the DBCMND that efficiently obtains high-quality solutions, outperforming existing solution methods in solution quality and computational effort, particularly as instance dimensions increase. Second, the paper introduces a MIP-based learning process to identify characteristics of good-quality solutions. More precisely, it shows how to take advantage of the time spent computing lower bounds to compile statistics characterizing attributes of already-encountered solutions. These statistics can then be used to identify key variables that, once fixed, significantly reduce the size of the problem instance providing the means to rapidly find good feasible solutions.

This paper is organized as follows. We recall the problem formulation in the next section. Section 3 describes the proposed matheuristic. Computational results are reported in Section 4. Conclusions and perspectives are given 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 noted $c_{ij}^k \geq 0$, while $f_{ij} \geq 0$ is the fixed design cost for arc (i, j). A limited capacity, 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_{ij}^k, which stand for 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 Design-Balanced Multicommodity Capacitated Fixed Charge Network Design problem becomes

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

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

$$\sum_{j\in\mathcal{N}_i^+} y_{ij} - \sum_{j\in\mathcal{N}_i^-} y_{ji} = 0, \qquad \forall i \in \mathcal{N}, \tag{2.3}$$

$$\sum_{k\in\mathcal{K}} x_{ij}^k \leq u_{ij}y_{ij}, \quad \forall\, (i,j) \in \mathcal{A}, \tag{2.4}$$

$$x_{ij}^k \geq 0, \qquad \forall\, (i,j) \in \mathcal{A},\ \forall\, k \in \mathcal{K}, \tag{2.5}$$

$$y_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in \mathcal{A}, \tag{2.6}$$

where $\mathcal{N}_i^- = \{j \in \mathcal{N} : (j,i) \in \mathcal{A}\}$, $\mathcal{N}_i^+ = \{j \in \mathcal{N} : (i,j) \in \mathcal{A}\}$, and

$$d^k = \begin{cases} w^k, & \text{if } i = O(k), \\ -w^k, & \text{if } i = D(k), \\ 0, & \text{otherwise.} \end{cases}$$

The objective function (2.1) minimizes the total cost computed as the sum of the total fixed cost for the arcs included in the optimal design (denoted as *open*) plus the total commodity transportation cost. Constraints (2.2) correspond to the flow conservation equations for each node and each commodity, while Constraints (2.3) are the design-balanced constraints ensuring that the total number of open arcs entering a node is equal to the total number of open arcs leaving that node. Relations (2.4) represent capacity constraints for each arc that also link flow and design variables by forbidding any flow to pass through an arc not already chosen as part of the design.

Note that, the linear relaxation (LP) of this formulation is obtained by replacing the integrality constraints (2.6) by $0 \leq y_{ij} \leq 1,\ \ \forall(i,j) \in \mathcal{A}$. Note also that, removing constraint set (2.3) yields the well known CMND, which is NP-Hard (Magnanti and Wong, 1984; Balakrishnan et al., 1997) and thus makes the DBCMND NP-hard as well. Practically, considerable algorithmic challenges are associated with addressing realistically-sized problem instances. These challenges are due to the trade-offs to be found between variable and fixed costs, and to the competition among commodities for the limited capacity on the arcs. In addition, by linking the design choices, the design-balanced constraints

(2.3) make the exploration of the solution/search space defined by the design variables much more challenging (e.g., CMND classical implicit enumeration and status changing, from open to close an vice-versa, are no longer possible), adding to the algorithmic difficulties.

3 Methodological Approach

We now present the matheuristic we propose for the DBCMND. It is motivated essentially by three observations. First, the cutting-plane procedure proposed by Chouman et al. (2009, 2011) has proved to be effective in computing tight lower bounds for the CMND, in relatively short computational times when compared to state-of-the-art software. Given the similitude between the two design problems, we expect the procedure to be efficient for the DBCMND as well. Second, various memories characterizing attributes of LP solutions can be built during the cutting-plane procedure and may then be used to identify promising solution elements - design arcs - providing means to guide the search toward good-quality feasible solutions. The same memories can then be used to fix part of the solution, closing unpromising design arcs, and, thus, reduce the dimensions of the problem instance, bringing it within the efficiency range of exact MIP solvers. It was actually interesting to investigate the effectiveness of a commercial MIP solver in identifying good solutions when the search is guided by a MIP-based learning process.

Algorithm 1 Cut&Fix Matheuristic

Phase I: Lower-bound computation (LB).
Run the cutting-plane algorithm (Algorithm 2) and compile LB memories;

Phase II: Feasible solution (FS).
Perform the α-fixing heuristic (Algorithm 3) based on the LB memories
Solve the resulting reduced DBCMND with a MIP solver to obtain a feasible solution

Algorithm 1 illustrates the proposed matheuristic combining two algorithmic components. The methods starts with a Phase I, which computes lower bounds on the optimal value of DBCMND using the cutting-plane procedure of Chouman et al. (2009, 2011) and compiles a number of statistics on solution characteristics. These memories are used in Phase II to fix a number of design variables and identify a feasible solution by solving the restricted problem using a MIP solver (e.g., the branch-and-cut of a commercial software). These components are detailed in the following subsections.

3.1 Lower bounds and memories

Any valid inequality (VI) that is valid for a relaxation of a problem, is valid for the problem itself. Therefore, as by dropping constraints (2.3) one obtains a CMND, any valid inequality for the CMND is valid for the DBCMND. More precisely, the families of inequalities studied in Chouman et al. (2009) and Chouman et al. (2011) are valid for the DBCMND and can be used to improve the formulation of the problem and strengthen the quality of its LP bounds.

The same studies have shown, however, that different VI families display quite different behaviors relative to their capability to improve the quality of bounds in reasonable computation times. Based on those studies and aiming for a combination of VI yielding a good trade-off between solution quality and computing time, and, thus, an efficient cutting-plane method, we consider three families of VIs only: the strong, cover, and flow-pack inequalities. We briefly describe these families of inequalities. More detailed discussions relative to the associated separation problems and implementation issues are to be found in the two references above.

Strong Inequalities (SI) are defined as

$$x_{ij}^k \leq d^k y_{ij}, \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}. \tag{3.1}$$

Adding SI to the model significantly improves the quality of the LP lower bounds (Crainic et al., 1999; Gendron and Crainic, 1994).

Cover Inequalities (CI) are defined in terms of cutsets of the network. Let $\mathcal{S} \subset \mathcal{N}$ be any non-empty subset of $\mathcal{N}$ and $\bar{\mathcal{S}} = \mathcal{N} \backslash \mathcal{S}$ its complement. We identify the corresponding cutset by $(\mathcal{S}, \bar{\mathcal{S}})$, i.e., the set of arcs that connect a node in $\mathcal{S}$ to a node in $\bar{\mathcal{S}}$. Let $d_{(\mathcal{S},\bar{\mathcal{S}})} = \sum_{k \in \mathcal{K}(\mathcal{S},\bar{\mathcal{S}})} d^k$ where $\mathcal{K}(\mathcal{S}, \bar{\mathcal{S}}) \subseteq \mathcal{K}$, be the set of commodities with their origin in $\mathcal{S}$ and their destination in $\bar{\mathcal{S}}$. $d_{(\mathcal{S},\bar{\mathcal{S}})}$ is then a lower bound on the amount of flow that must circulate across the cutset in any feasible solution. A set $\mathcal{C} \subseteq (\mathcal{S}, \bar{\mathcal{S}})$ is a cover if the total capacity of the arcs in $(\mathcal{S}, \bar{\mathcal{S}}) \backslash \mathcal{C}$ does not cover the demand, i.e., $\sum_{(i,j) \in (\mathcal{S},\bar{\mathcal{S}}) \backslash \mathcal{C}} u_{ij} < d_{(\mathcal{S},\bar{\mathcal{S}})}$. Moreover, the cover $\mathcal{C} \subseteq (\mathcal{S}, \bar{\mathcal{S}})$ is minimal if it is sufficient to open any arc in $\mathcal{C}$ to cover the demand. For every cover $\mathcal{C} \subseteq (\mathcal{S}, \bar{\mathcal{S}})$, the cover inequality

$$\sum_{(i,j) \in \mathcal{C}} y_{ij} \geq 1 \tag{3.2}$$

is valid for the DBCMND. The basic idea of this inequality is that one has to open at least one arc from the set $\mathcal{C}$ in order to meet the demand. In addition, it has been proven (Balas, 1975; Wolsey, 1975) that if $\mathcal{C}$ is a minimal cover, applying a lifting procedure yields a stronger inequality.

Flow Pack Inequalities (FPI). For any $\mathcal{L} \subseteq \mathcal{K}$ and cutset $(\mathcal{S}, \bar{\mathcal{S}})$, let

$$x_{ij}^L = \sum_{k \in \mathcal{L}} x_{ij}^k, \quad b_{ij}^L = \min\{u_{ij}, \sum_{k \in \mathcal{L}} d^k\}, \quad \text{and} \quad d_{(\mathcal{S},\bar{\mathcal{S}})}^L = \sum_{k \in \mathcal{K}(\mathcal{S},\bar{\mathcal{S}}) \cap L} d^k.$$

A flow pack $(\mathcal{C}_1, \mathcal{C}_2)$ is defined by two sets $\mathcal{C}_1 \subseteq (\mathcal{S}, \bar{\mathcal{S}})$ and $\mathcal{C}_2 \subseteq (\bar{\mathcal{S}}, \mathcal{S})$ such that $\mu = \sum_{(i,j) \in \mathcal{C}_1} b_{ij}^L - \sum_{(j,i) \in \mathcal{C}_2} b_{ji}^L - d_{(\mathcal{S},\bar{\mathcal{S}})}^L < 0$. Let $\mathcal{D}_1 \subset (\mathcal{S}, \bar{\mathcal{S}}) \backslash \mathcal{C}_1$. The flow pack inequality is then defined as (Atamturk, 2001; Stallaert, 97)

$$\sum_{(i,j) \in \mathcal{C}_1} x_{ij}^L + \sum_{(i,j) \in \mathcal{D}_1} (x_{ij}^L - \min\{b_{ij}^L, -\mu\} y_{ij})) \leq - \sum_{(j,i) \in \mathcal{C}_2} (b_{ji}^L + \mu)^+ (1 - y_{ji}) + \sum_{(j,i) \in (\bar{\mathcal{S}},\mathcal{S}) \backslash \mathcal{C}_2} x_{ji}^L + \sum_{(i,j) \in \mathcal{C}_1} b_{ij}^L. \tag{3.3}$$

The cutting-plane procedure then iterates on solving the linear relaxation, LP, of the DBCMND and generating violated valid inequalities that are added to the LP formulation. Memories are being updated at each such iteration as described bellow. The Phase I terminates when either the optimal solution is found, which is unlikely to happen very often, or when the improvement is smaller than ϵ. With $\bar{Z}$ and $(\bar{x}, \bar{y})$ standing for the optimal value and solution vector, respectively, of the LP with the currently generated VIs, Algorithm 2 displays the main steps of the procedure.

Three different statistics characterizing attributes of LP solutions found while running the cutting plane algorithm are collected in three particular memories for subsequent utilization:

Design-variable frequencies, $F \in \mathbb{N}^{|\mathcal{A}|}$, representing how often an arc has been used in the LP solutions. Set initially to the null vector, F is updated at each LP solution $(\bar{x}, \bar{y})$ by setting $F_{ij} = F_{ij} + 1$if $\bar{y}_{ij} > \beta, \ \forall (i,j) \in \mathcal{A}$, for a given threshold β indicating the importance of an arc (i,j) in the current LP solution.

Violated cover inequality frequencies, $L \in \mathbb{N}^{|\mathcal{A}|}$, counting how often design arcs were included in violated CIs generated during the cutting-plane procedure. Similarly to F, initialized to the null vector, L is updated at each violated CI found $L_{ij} = L_{ij} + 1, \ \forall (i,j) \in \mathcal{C}$, where $\mathcal{C} \subseteq \mathcal{A}$ is the minimal cover obtained in the CI.

Accumulated reduced costs, $R \in \mathbb{R}^{|\mathcal{A}|}$. Similarly to the two previous memories, R is initialized to the null vector and is 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 $(\bar{x}, \bar{y})$.

Algorithm 2 Lower-bound Computation & Memory Building

$Z_{Last} \leftarrow 0$; $F \leftarrow 0^{|\mathcal{A}|}$; $L \leftarrow 0^{|\mathcal{A}|}$; $R \leftarrow 0^{|\mathcal{A}|}$;
Solve the LP relaxation;
while $\bar{Z} - Z_{Last} > \epsilon$ **do**
  $Z_{Last} \leftarrow \bar{Z}$;
  if $\bar{y}$ is integer **then**
    return $\bar{Z}$ and $(\bar{x}, \bar{y})$
  end if
  Generate and add violated SIs;
  Generate and add violated CIs;
  Update the L memory;
  Generate and add violated FPIs;
  if New inequalities are added **then**
    Solve the LP relaxation;
    Update Nb_{LP};
    Update the F and R memories;
  end if
end while
return the lower bound $\bar{Z}$, the corresponding solution $(\bar{x}, \bar{y})$, and the memories F, L, and R.

3.2 α-*fixing* heuristic

Almost all the metaheuristics and approximate methods in the literature are based on the availability of a first feasible solution. Contrary to the CMND, it is not obvious to find a feasible initial solution for the DBCMND. The classical approach of first solving a linear relaxation and then rounding-up all design variables corresponding to used arcs in the LP solution (often used for the CMND) is not appropriate for the DBCMND. Indeed, except for some special cases, the integral solution obtained does not satisfy the design-balanced constrains. Figure 1 illustrates the infeasibility of such a rounding-up method for a small graph consisting of four nodes and five arcs. Notice that the LP solution satisfies the design-balanced requirements while the round-up solution does not (for nodes 3 and 4). We therefore introduce the first generic procedure providing the means to obtain efficiently good initial solutions for the DBCMND.

To obtain a good starting feasible solution that might be embedded into any exploration heuristic, we propose to first reduce the size of the DBCMND instance, by closing a suitable subset of arcs and, then, solve the resulting reduced DBCMND problem using any available exact MIP code. The challenge is in selecting the suitable subset of arcs to close. On the one hand, one desires to close a sufficiently high number to yield a reduced design problem "easy" to address with a good MIP solver. On the other hand, closing too many arcs may yield an easily addressed problem but a network too small to carry

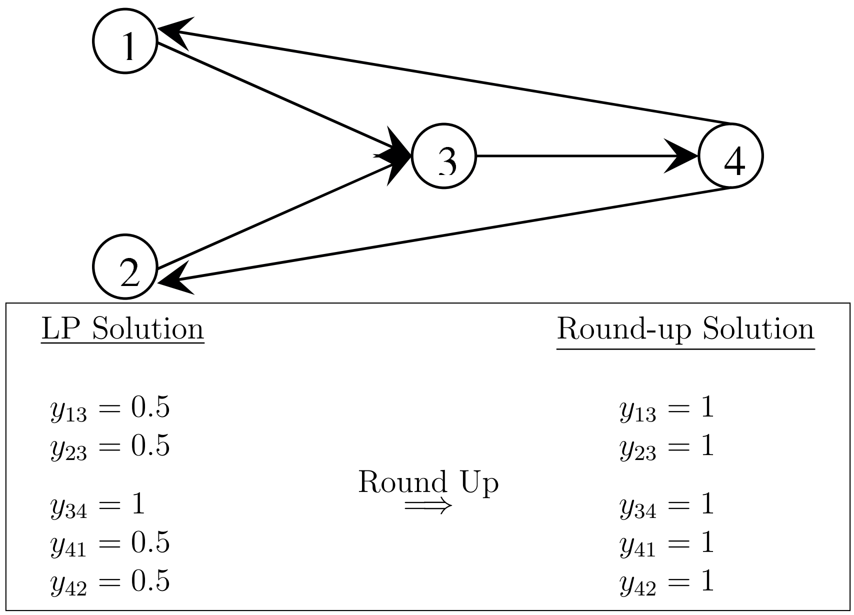

Figure 1: Infeasibility of rounding-up LP solutions

all the demand and, thus, inappropriate for the task at hand.

To efficiently address this challenge, we propose the *α-fixing* heuristic, which selects based on the compiled memories F, L, and R (Section 3.1) two suitable complementary sets $\widetilde{\mathcal{A}}$ and $\mathcal{A}\backslash\widetilde{\mathcal{A}}$ of design arcs to keep and close, respectively. As illustrated in Algorithm 3, the heuristic starts with an empty $\widetilde{\mathcal{A}}$ set, and gradually adds arcs that are attractive given the information gathered while executing the cutting-plane algorithm, and that provide sufficient connectivity and capacity to the resulting network.

Algorithm 3 *α-Fixing Heuristic*

Require: F, L, R as given by the cutting-plane Algorithm 2

$\widetilde{A} = \emptyset$;

Selection step based on F;

Add to $\widetilde{\mathcal{A}}$ all arcs with frequency $\geq \alpha N b_{LP}$;

Connectivity step based on $F + L$;

Add arcs to $\widetilde{\mathcal{A}}$ to ensure each transshipment node has at least one incoming and one outgoing arc;

Feasibility step based on R;

Add arcs to $\widetilde{\mathcal{A}}$ to provide sufficient capacity for flows out of and into demand and supply nodes, respectively.

Arcs are added in three consecutive steps. The *selection step* aims to select attractive arcs as defined by F, the utilization frequency in cutting-plane LP solutions. The idea is that arcs that are repeatedly used in optimal LP solutions are most likely to also be part of good, hopefully optimal, feasible solutions. Thus, given a threshold α, arc (i,j) is added to $\widetilde{\mathcal{A}}$ if $F_{ij} \geq \alpha Nb_{LP}$, where Nb_{LP} is the number of LPs solved by the cutting-plane.

The *connectivity step* aims to provide the means for commodities to pass through each selected transshipment node in the network. This means that each transshipment node already in $\widetilde{\mathcal{A}}$, i.e., with at least an incoming/outgoing arc open, must have at least an outgoing/incoming arc open. Because the choice has to be made among arcs in $\mathcal{A}\backslash\widetilde{\mathcal{A}}$, which did not appear often in cutting-plane LP solutions, we combine the measures of frequency F and the CI-frequency L. In fact, a frequent appearance of an arc in minimal covers of violated CIs means the arc has a good chance to be open and used in feasible solutions, and is therefore a good candidate for the fixing heuristic. Consequently, for each node $i \in \mathcal{N}$ with at least one

- Incoming arc, i.e., $\sum_{j\in\widetilde{\mathcal{N}}_i^-} y_{ji} \geq 1$, and no outgoing arc, i.e., $\sum_{j\in\widetilde{\mathcal{N}}_i^+} y_{ij} = 0$, add to $\widetilde{\mathcal{A}}$ the arc $(i,j) = \text{argmax}_{j\in\mathcal{N}_i^+\backslash\widetilde{\mathcal{N}}_i^+}(F_{ij} + L_{ij})$;
- Outgoing arc, i.e., $\sum_{j\in\widetilde{\mathcal{N}}_i^+} y_{ij} \geq 1$, and no incoming arc, i.e., $\sum_{j\in\widetilde{\mathcal{N}}_i^-} y_{ji} = 0$, add to $\widetilde{\mathcal{A}}$ the arc $(j,i) = \text{argmax}_{j\in\mathcal{N}_i^-\backslash\widetilde{\mathcal{N}}_i^-}(F_{ji} + L_{ji})$.

Finally, the feasibility step opens enough arcs at origins and destinations to provide sufficient capacity to satisfy the demand requirements. Because the previous step used the L memory, we aim for a certain degree of diversification in our selection and, thus, we use information based on the reduced costs R in this step. Therefore, for any supply or demand node such that

$$\sum_{j\in\widetilde{\mathcal{N}}_i^+} u_{ij} < w^k,\ i = O(k),\ \forall k \in \mathcal{K} \quad \Rightarrow \text{Add the arc } (i,j) = \text{argmin}_{j\in\mathcal{N}_i^+\backslash\widetilde{\mathcal{N}}_i^+}(R_{ij}),$$
$$\sum_{j\in\widetilde{\mathcal{N}}_i^-} u_{ji} < w^k,\ i = D(k),\ \forall k \in \mathcal{K} \quad \Rightarrow \text{Add the arc } (j,i) = \text{argmin}_{j\in\mathcal{N}_i^-\backslash\widetilde{\mathcal{N}}_i^-}(R_{ji}).$$

Once the set $\widetilde{\mathcal{A}}$ is determined using the α-fixing heuristic, we consider the restriction $\text{DBCMND}_{\widetilde{\mathcal{A}}}$ where all arcs in $\widetilde{\mathcal{A}}$ are free and all arcs in $\mathcal{A}\backslash\widetilde{\mathcal{A}}$ are closed. $\text{DBCMND}_{\widetilde{\mathcal{A}}}$ is then solved using a MIP solver until optimality or a time or node limit is reached, the latter limits aiming to achieve the goal of obtaining a solution as good as possible in a short computational time. If the reduced problem is feasible, we stop. Otherwise, we decrease the value of α and repeat the α-fixing heuristic to find a larger set $\widetilde{\mathcal{A}}$. We iterate until a feasible solution is found.

4 Computational Results

The objectives of the computational experiments are twofold: 1) to test the effectiveness of the cutting-plane procedure, developed originally for the CMND, in the context of DBCMND; and 2) to evaluate the quality of the solutions obtained by the Cut&Fix matheuristic we propose. We compare our results with those obtained by the meta-heuristic of Pedersen et al. (2009), the matheuristic of Vu et al. (2012), as well as to the best solutions obtained by the Branch-and-Cut method (B&C) of CPLEX (version 12) after one hour and 10 hours. A computational time limit of 10 hours was imposed to all the methods.

The procedures were coded in C++. The LP relaxations within the cutting-plane procedure were solved to optimality using the option *Dualopt* of CPLEX (version 12). All the restrictions DBCMND$_{\widetilde{\mathcal{A}}}$ were solved using the B&C of CPLEX with a time limit of one hour and a node limit of 200. Experiments were performed on a network of Dual-Core AMD Opteron (using a single thread) workstations with 8 Gigabytes of RAM operating under SunOS 5.1.

The performance of the proposed matheuristic is evaluated on a set of network design instances with various characteristics used in several papers (Ghamlouche et al., 2003; Pedersen et al., 2009; Vu et al., 2012; Chouman et al., 2011) and described in Crainic et al. (2001). These problem instances, identified as Sets **C** and **R**, consist of general transshipment networks with one commodity per origin-destination pair and no parallel arcs. Positive transportation cost, fixed cost, and capacity are associated with each arc. Note that the transportation costs on any given arc are the same for all commodities.

Set **C** consists of 43 instances characterized by their number of nodes, arcs, and commodities, noted $|N|, |A|$, and $|K|$, respectively. Two additional letters are used to characterize the fixed cost level, “F” for high and “V” for low, relatively to the transportation cost, and the capacity level, “T” for tight and “L” for loose, compared to the total demand. The set of instances **R** consists of 81 problems, nine sets of nine instances each. Each set is characterized by the same number of nodes, arcs, and commodities, instances displaying various levels of fixed cost and capacity ratios. Thus, “F01” for low, “F05” for medium, and “F10” for high, are used to qualify the importance of the fixed cost with respect to the transportation cost, while “C1” for loose, “C2” for medium, and “C8” for tight, to qualify the tightness of the capacity compared to the total demand. To facilitate comparisons, we present the results for the 24 **C** instances and the 54 **R** instances used in Pedersen et al. (2009) and Vu et al. (2012). These are medium to large-size instances with various levels of cost and capacity ratios.

Our primary measure of performance is the gap between the reference solution z^* and a given solution z computed as:

$$\Delta z^*/z = \frac{100(z^* - z)}{z^*} \tag{4.1}$$

In order not to overload the paper, we report average results over the **C** and **R** instances. Detailed results for each instance are included in the Appendix. The next two subsections addresses the objectives stated above.

4.1 Evaluation of the lower bound procedure

The scope of this section is to analyze the performance of the cutting-plane method in the context of the DBCMND. We aim to examine, in particular, the effectiveness of this algorithm in improving the lower bound of the DBCMND while compiling characteristics and attributes of good solutions.

Table 1 displays the results obtained by the cutting-plane algorithm for the **C** and **R** sets averaged according to the problem dimensions in terms of numbers of nodes, arcs, and commodities. For each such group of instances (Column DESCRIPTION), the table indicates the number of instances in the group (Column NB), the gap between the lower bound obtained by the cutting-plane and the first LP bound (Column GAPLB), the total number of cuts generated in the cutting-plane (Column CUTS), the total number of LPs solved (Column NBLP), and the CPU computational time required by the lower bound procedure.

The results show clearly the effectiveness of the cutting-plane algorithm in improving the quality of the LP bounds. The overall averaged gap improvement for the **C** instances reaches 21.23%, while it is 36.78% for the **R** instance set. Although the average numbers of cuts generated and LP solved may seem relatively high, the associated computational effort is low reaching an average of 976 and 347 CPU seconds (equivalent to 16 and 5 minutes) for the **C** and **R** sets, respectively. We actually observe very short solving times for the first LP, with averages of 5.75 and 2.45 CPU seconds on average for the **C** and **R** sets, respectively. We also observe that the time required for the cut generations is almost negligible, while solving the LP relaxation after each round of cut generation is more efficient than for the first LP, because it consists in re-optimizing from a previous optimal basis (the simplex method of CPLEX is applied with the *Dualopt* option).

These results support our claim that the cutting-plane algorithm is effective in improving the bounds within a short computing effort, even when repetitively solving different LP models: 13 and 14 on average for the **C** and **R** instance sets, respectively. As discussed in Section 3.2, these multiple solutions of LP formulations provide the means to compile the memories that are then used in the *α-fixing* heuristic to guide the search towards good feasible solutions.

Set **C**					
Description	Nb	GapLB	Cuts	NbLP	CpuLB
20,230,200	(4)	30.85%	3168	14	301
20,300,200	(4)	21.59%	2371	16	176
30,520,100	(4)	21.10%	2022	13	137
30,520,400	(4)	16.18%	3616	12	2221
30,700,100	(4)	18.99%	1784	13	74
30,700,400	(4)	18.70%	3974	13	2950
Average	(24)	21.23%	2822	13	976
Set **R**					
Description	Nb	GapLB	Cuts	NbLP	CpuLB
20,220,40	(9)	38.87%	1101	12	12
20,220,100	(9)	33.60%	1703	15	82
20,220,200	(9)	28.86%	2080	11	318
20,320,40	(9)	45.05%	1872	17	80
20,320,100	(9)	39.62%	2607	14	424
20,320,200	(9)	34.67%	3312	15	1166
Average	(54)	36.78%	2113	14	347

Table 1: Evaluation of the Cutting-Plane Algorithm

4.2 Evaluation of the Cut&Fix matheuristic

This section is dedicated to the evaluation of the performance of the proposed Cut&Fix matheuristic, which is strongly linked to the capability of the α-*fixing* heuristic to efficiently identify high-quality feasible solutions based on the information compiled during the computation of the cutting-plane algorithm. We present comparative results for the Cut&Fix matheuristic, the Tabu Search of Pedersen et al. (2009), the matheuristic of Vu et al. (2012), and the B&C of CPLEX, version 12, collected after 1 hour and 10 hours of computing time.

The value of the parameters in the implementation of the α-*fixing* procedure are $\alpha = 0.45$ and $\beta = 0.3$. These values were selected based on computational experiments where, with the objective of including a suitable number of arcs in the network, the median values of the F and L memories were first computed, then several values around these medians were tested. The values of 100 and 200 were also tested to limit the number of nodes allowed to the MIP solver to work on the DBCMND$_{\widetilde{\mathcal{A}}}$. The value 200 was selected since experimental results indicated small improvements, of 0.36% on average obtained on 19 instances, for an extra 5 minutes of computational time. Moreover, we observed that the heuristic was not finding a feasible solution for one of the **R** instances with a 100-node limit.

Tables 2 and 3 display the results for the **C** and **R** instance sets, respectively, averaged according to problem dimensions. In addition to the DESCRIPTION and NB columns, Columns C&F/TS, C&F/TS-PR, C&F/CPLEX1h and C&F/CPLEX10h display the gaps between the solution found by the Cut&Fix, C&F, matheuristic and the solutions obtained by the Tabu Search of Pedersen et al. (2009), the matheuristic of Vu et al. (2012), CPLEX after 1 hour, and CPLEX after 10 hours, respectively. Note that negative values in any of these columns indicate that the Cut&Fix matheuristic outperforms the method in the corresponding column for the designated group of instances. Columns C&F/LB_CPLEX10h and C&F/LB indicate the gap of the C&F solution with respect to the best lower bound found by the B&C of CPLEX after 10 hours of CPU time and the lower bound found by the cutting-plane, respectively. The last column, CpuCF, indicates the computational time required by the proposed matheuristic.

DESCRIPTION	NB	C&F /TS	C&F /TS-PR	C&F /CPLEX1h	C&F /CPLEX10h	C&F /LB_CPLEX10h	C&F /LB	CpuCF
20,230,200	(4)	-4.01%	2.00%	0.73%	1.81%	4.15%	5.19%	416
20,300,200	(4)	-3.72%	1.34%	1.26%	1.45%	3.37%	4.25%	503
30,520,100	(4)	-0.52%	2.65%	2.37%	2.97%	5.13%	5.72%	564
30,520,400	(4)	-4.07%	-0.04%	-0.18%	0.45%	1.83%	1.93%	6030
30,700,100	(4)	-1.63%	1.27%	1.31%	1.40%	2.49%	3.28%	211
30,700,400	(4)	-7.32%	-0.73%	-1.26%	0.20%	2.53%	2.59%	7042
Average	(24)	-3.40%	1.73%	0.19%	2.07%	4.11%	5.40%	2461

Table 2: Performance comparisons - **C** instance set

DESCRIPTION	NB	C&F /TS	C&F /TS-PR	C&F /CPLEX1h	C&F /CPLEX10h	C&F /LB_CPLEX10h	C&F /LB	CpuCF
20,220,40	(9)	-1.52%	2.76%	3.00%	3.08%	3.33%	7.09%	13
20,220,100	(9)	-3.62%	1.11%	0.88%	1.35%	3.01%	5.21%	505
20,220,200	(9)	-5.42%	0.43%	-0.30%	0.64%	2.18%	2.98%	684
20,320,40	(9)	-0.76%	4.54%	4.56%	5.08%	5.77%	8.99%	154
20,320,100	(9)	-3.78%	2.11%	0.87%	2.19%	5.23%	6.70%	884
20,320,200	(9)	-4.91%	0.31%	-6.44%	0.78%	3.39%	4.29%	1740
Average	(54)	-3.33%	1.87%	0.43%	2.19%	3.82%	5.88%	663

Table 3: Performance comparisons - **R** instance set

The results indicate clearly the superiority of the proposed matheuristic, relative to the other methods, in identifying high-quality feasible solutions in short computational times. Indeed, it significantly outperforms the meta-heuristic of Pedersen et al. (2009), improving the results for 23 out of 24 **C** instances and for 46 out of 54 **R** instances. Moreover, the improvement gap increases for difficult problems characterized by large number of commodities, reaching a maximum gap improvement of 7.32% on average for the instances with 700 arcs and 400 commodities. The results indicate also that Cut&Fix matheuristic is very competitive with the current best heuristic method known in the literature, namely the matheuristic of Vu et al. (2012). Thus, for almost one fifth of the computational time of the TS-PR method, C&F succeed in achieving high-quality feasible solutions with an average gap of 1.73% and 1.87% for the **C** and **R** instances, respectively. Moreover, the C&F solutions outperform the TS-PR solutions for the large size instances with 400 commodities.

The proposed method also compares very well with the B&C of CPLEX (version 12). It identifies better solutions for 8 of the 24 **C** instances and for 13 of the 54 **R** instances. It is very competitive for instances characterized by small to medium number of commodities, while outperforming CPLEX when the number of commodities increases. Indeed, CPLEX cannot find feasible solutions for 4 out of the 8 instances with 400 commodities, while the method we propose provides solutions with optimal gaps ranging from 1.83% to 2.53% for those hard instances. Indeed, even comparing with the CPLEX results after 10 hours of computational effort, the proposed matheuristic is competitive in solution quality, with overall gap differences as low as 2.07% and 2.19% for the **C** and **R** sets, respectively, in a fraction of CPU time. The improvement is actually more important for difficult problem instances characterized by large numbers of commodities. Notice finally that, only instances for which CPLEX found a feasible solution (within one or ten hours of CPU time) are included in the figures of Tables 2 and 3, which underestimates the performance of the proposed matheuristic.

Cost	Cap	Nb	C&F /TS	C&F /TS-PR	C&F /CPLEX1h	C&F /CPLEX10h	C&F /LB_CPLEX10h	C&F /LB	CpuCF
V	L	(6)	-2.78%	0.94%	0.56%	1.27%	2.57%	3.27%	1897
V	T	(6)	-2.73%	0.46%	0.45%	0.71%	1.39%	2.03%	1999
F	L	(6)	-5.63%	1.17%	1.79%	1.60%	4.88%	5.37%	3061
F	T	(6)	-3.04%	1.75%	1.64%	1.93%	4.16%	4.63%	2888
Average		(24)	-3.40%	1.73%	0.19%	2.07%	4.11%	5.40%	2461

Table 4: Aggregated comparative results by fixed cost and capacity ratios - **C** instancces

Cost	Cap	Nb	C&F /TS	C&F /TS-PR	C&F /CPLEX1h	C&F /CPLEX10h	C&F /LB_CPLEX10h	C&F /LB	CpuCF
	C1	(6)	-1.10%	0.73%	0.73%	0.75%	0.93%	1.62%	51
F01	C2	(6)	-0.12%	2.10%	2.33%	2.39%	2.61%	3.38%	194
	C8	(6)	-2.05%	1.06%	0.55%	1.26%	1.68%	3.16%	220
	C1	(6)	0.48%	4.37%	2.98%	4.53%	6.52%	9.50%	452
F05	C2	(6)	-3.41%	1.95%	-1.17%	2.24%	4.10%	6.47%	459
	C8	(6)	-6.10%	0.38%	-0.46%	0.82%	2.06%	4.04%	272
	C1	(6)	-3.42%	2.73%	0.12%	3.45%	7.12%	10.20%	3451
F10	C2	(6)	-4.72%	2.56%	-0.54%	2.88%	6.49%	9.02%	497
	C8	(6)	-9.55%	0.99%	-0.70%	1.37%	2.85%	5.52%	374
Average		(54)	-3.33%	1.87%	0.43%	2.19%	3.82%	5.88%	663

Table 5: Aggregated comparative results by fixed cost and capacity ratios - **R** instancces

Tables 4 and 5 display the same results averaged according to the different levels of fixed cost (first column) and capacity (second column) ratios. These figures indicate that the performance of the proposed method increases with the problem difficulty. Indeed, the gaps of the best solution obtained by the Cut&Fix matheuristic with respect to the lower bounds provided by CPLEX and the cutting-plane procedure (last two columns) are smaller when, for the same level of cost ratio, the level of the capacity ratio increases from loose (C1) to tight (C8). We take these results to support the claim that the proposed heuristic is suitable for hard real-world problems characterized by large size and limited resources.

Cost	Cap	Nb	TS-PR	CPLEX1h	CPLEX10h	C&F
V	L	(6)	7690	3600	30982	2686
V	T	(6)	8380	3600	31406	2475
F	L	(6)	6990	3600	39169	5094
F	T	(6)	8080	3600	36000	4070
Average		(24)	7785	3600	34389	3581

Table 6: CPU times, fixed cost and capacity ratio aggregtion, **C** instances

Cost	Cap	Nb	TS-PR	CPLEX1h	CPLEX10h	C&F
	C1	(6)	4670	1586	7083	92
F01	C2	(6)	5770	1593	12393	224
	C8	(6)	5940	3184	20959	249
	C1	(6)	5340	3600	27313	1300
F05	C2	(6)	5240	3259	25272	802
	C8	(6)	7070	3342	30342	337
	C1	(6)	5270	3591	27867	4634
F10	C2	(6)	6150	3600	36000	968
	C8	(6)	5950	3061	28161	432
Average		(54)	5711	2980	23932	1004

Table 7: CPU times, fixed cost and capacity ratio aggregation, **R** instances

To sum up the comparative results, the proposed Cut&Fix matheuristic yields high-quality solutions, outperforming or being very competitive with the best methods in the literature and the state-of-the-art MIP solver CPLEX. This performance appears even more remarkable when comparing the computational efforts required by each of the methods. Tables 6 and 7 display the average CPU times required by TS-PR (Vu et al., 2012), CPLEX run for 1 and 10 hours, and the proposed C&F matheuristic. A first general conclusion is that the proposed matheuristic is efficient, the proportion of CPU time required by each of its phase being on average one third and two thirds of the total effort, respectively (tables in the Appendix detail these results). Comparing performances shows that for the **C** instances, C&F requires an average of 1 hour CPU time, which is half the time required by TS-PR. Compared with CPLEX, one may recall that for about the same computational effort of one hour, CPLEX could not identify a feasible solution for 4 out of the 8 instances with 400 commodities. C&F outperforms CPLEX for large instances for similar computational efforts. The results for the **R** instances further support our claims about the performance of the proposed matheuristic. Indeed, on average, C&F requires one third of the CPU time of CPLEX1h, one fifth of the CPU time of TS-PR, and 1/24 the CPU time of CPLEX10h for similar or better-quality solutions.

5 Conclusions and Perspectives

We introduced Cut&Fix, a new matheuristic for the design-balanced capacitated multicommodity network design problem, one of the premier formulations for the service network design problem with asset management concerns increasingly faced by carriers within their tactical planning processes.

The matheuristic combines a cutting-plane procedure efficiently computing tight lower bounds and a variable-fixing heuristic reducing the problem size and feeding an exact MIP solver. Learning mechanisms embedded into the cutting-plane procedure are used to identify characteristics of good solutions and guide the variable-fixing heuristic.

An extensive computational study first showed that the cutting-plane procedure, initially proposed for the fixed-charge, multicommodity capacitated network design problem, is also very efficient for the special structure of DBCMND, cutset-based inequalities in particular.

The computational study also shown the merit of the Cut&Fix idea of combining this cutting-plane procedure, together with appropriate learning mechanisms, and variable-fixing techniques into an efficient algorithm able to rapidly identify high-quality feasible solutions. This capability of the proposed Cut&Fix method is by itself remarkable as identifying feasible solutions to the DBCMND was previously shown to be difficult. Yet, the numerical experiments shown that the proposed matheuristic is outperforming existing solution methods in solution quality and computational effort. It thus currently stands as the best, or one of the best, heuristic for the DBCMND.

The fundamental ideas on which the new matheuristic is built are general in nature and open interesting research perspectives in hybridizing mathematical programming and meta-heuristics for network design problems. We are currently following some of these avenues, including adapting these ideas for other hard transportation planning problems, such as the management of power-unit fleets (e.g., locomotives in rail transportation). We plan to report on these developments in the near future.

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.

The authors thank Ms. Geneviève Hernu, analyst with the Chair, for her much welcome assistance in conducting the experimental phase of this project, and the CIRRELT for access to the material and human resources of its computing network. 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 Quebec. The computing equipment principally used was acquired with the support of the Canadian Foundation for Innovation and its partners, mainly the Quebec Ministry of Education and UQAM.

References

J. Andersen, T.G. Crainic, and M. Christiansen. Service Network Design with Management and Coordination of Multiple Fleets. *European Journal of Operational Research*, 193(2):377–389, 2009a.

J. Andersen, T.G. Crainic, and M. Christiansen. Service Network Design with Asset Management: Formulations and Comparative Analyzes. *Transportation Research Part C: New Technologies*, 17(2):397–207, 2009b.

J. Andersen, M. Christiansen, T.G. Crainic, and R. Grønhaug. Branch-and-Price for Service Network Design with Asset Management Constraints. *Transportation Science*, 46(1):33–49, 2011.

A.P. Armacost, C. Barnhart, and K.A. Ware. Composite Variable Formulations for Express Shipment Service Network Design. *Transportation Science*, 36(1):1–20, 2002.

A. Atamturk. Flow pack facets of the single node fixed-charge flow polytope. *Operations Research Letters*, 29:107–114, 2001.

A. Balakrishnan, T. L. Magnanti, and P. B. Mirchandani. Network design. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 311–334. John Wiley & Sons, NY, 1997.

E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.

T. Bektas and T.G. Crainic. A brief overview of intermodal transportation. In G. Taylor, editor, *Logistics Engineering Handbook*, chapter 28, pages 1 – 16. Taylor and Francis Group, 2008.

M. Chouman, T.G. Crainic, and B. Gendron. A cutting-plane algorithm for multicommodity capacitated fixed charge network design. Technical report, Center for research on transportation, Université de Montréal, 2009.

M. Chouman, T.G. Crainic, and B. Gendron. Commodity Representations and Cutset-Based Inequalities for Multicommodity Capacitated Fixed Charge Network Design. Technical Report CIRRELT-2011-56, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et les transports, Université de Montréal, Montréal, QC, Canada, 2011.

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

T.G. Crainic. Long-Haul Freight Transportation. In Hall, R.W., editor, *Handbook of Transportation Science*, pages 451–516. Kluwer Academic Publishers, Norwell, MA, second edition, 2003.

T.G. Crainic and K. Kim. Intermodal Transportation. In Barnhart, C. and Laporte, G., editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, chapter 8, pages 467–537. North-Holland, Amsterdam, 2007.

T.G. 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.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.

B. Gendron and T.G. Crainic. Relaxations for multicommodity capacitated network design problems. Technical report, Publication CRT-945, Centre de recherche sur les transports, Université de Montréal, 1994.

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

M.F. Lai and H.K. Lo. Ferry Service Network Design: Optimal Fleet Size, Routing and Scheduling. *Transportation Research Part A: Policy and Practice*, 38:305–328, 2004.

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

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

M.B. Pedersen, T.G. Crainic, and O.B.G. Madsen. Models and Tabu Search Metaheuristics for Service Network Design with Asset-Balance Requirements. *Transportation Science*, 43(2):158–177, 2009.

K.R. Smilowitz, A. Atamtürk, and C.F. Daganzo. Deferred Item and Vehicle Routing within Integrated Networks. *Transportation Research Part E: Logistics and Transportation*, 39:305–323, 2003.

J. Stallaert. The complementary class of generalized flow cover inequalities. *Discrete Applied Mathematics*, 77:73–80, 97.

D.M.. Vu, T.G. Crainic, and M. Toulouse. A Three-Stage Matheuristic for the Capacitated Multi-commodity Fixed-Cost Network Design with Design-Balance Constraints. Technical Report CIRRELT-2012-21, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et les transports, Université de Montréal, Montréal, QC, Canada, 2012.

L. Wolsey. Faces of linear inequalities in 0-1 variables. *Mathematical Programming*, 8: 165–178, 1975.

Appendix

Tables 8 and 9 display the detailed results for the instances of sets **C** and **R**, respectively. For each instance, the respective table gives its description, the best feasible solution obtained by the Tabu Search of Pedersen et al. (2009) (Column TS), the best feasible solution obtained by the Tabu-Search Path Relinking matheuristic of Vu et al. (2012), the best feasible solutions of the B&C of CPLEX obtained after 1 hour and 10 hours, and the lower bound obtained by the B&C of CPLEX (Columns CPLEX1h, CPLEX10h, and LB_CPLEX10h, respectively), the feasible solution obtained by the Cut&Fix matheuristic (Column C&F), and the lower bound and the number of cuts generated by the cutting-plane algorithm (Columns LB and Cuts, respectively).

Instance	TS	TS-PR	CPLEX			C&F		
			CPLEX1h	CPLEX10h	LB_CPLEX10h	C&F	LB	Cuts
C20,230,200,V,L	102919	97274	98512	97274	94993	98699	93774	3291
C20,230,200,F,L	150764	139395	140843	140843	135107	141744	134043	3495
C20,230,200,V,T	103371	100720	101089	100221	99525	103103	98383	2779
C20,230,200,F,T	149942	138962	142452	139252	135941	142638	134415	3108
C20,300,200,V,L	82533	77584	77570	77570	75986	79953	75267	2250
C20,300,200,F,L	128757	119987	119945	119890	115992	120979	114762	2812
C20,300,200,V,T	78571	76450	76350	76208	76201	76545	75507	2099
C20,300,200,F,T	116338	111776	112358	111743	108920	113412	108154	2323
C30,520,100,V,L	55981	54783	54810	54683	54390	55733	53906	1569
C30,520,100,F,L	104533	100098	99717	98871	94079	104235	93625	3251
C30,520,100,V,T	54493	53035	53034	53032	52871	53224	52493	1042
C30,520,100,F,T	105167	101412	102919	101495	98072	106251	97684	2225
C30,520,400,V,L	119735	115528	115487	114730	113469	115220	113294	3303
C30,520,400,F,L	162360	153409	na	152891	149910	153737	149757	4260
C30,520,400,V,T	120421	117226	117214	116763	115859	117056	115738	2886
C30,520,400,F,T	161978	155906	na	155025	152375	155942	152317	4013
C30,700,100,V,L	49429	48807	48693	48693	48688	49268	48187	1442
C30,700,100,F,L	63889	61408	61430	61408	59834	62267	59442	2527
C30,700,100,V,T	48202	46812	46750	46750	46531	46928	46091	1478
C30,700,100,F,T	58204	56237	56337	56169	55372	57701	55042	1690
C30,700,400,V,L	103932	100589	101866	99716	97888	99458	97814	4180
C30,700,400,F,L	157043	141037	na	138229	133536	139607	133503	4888
C30,700,400,V,T	103085	97875	97838	97694	95885	97737	95807	3140
C30,700,400,F,T	141917	133686	na	132827	129807	132855	129717	3686

Table 8: Results for the 24 instances of Set **C**

Instance	TS	TS-PR	CPLEX			C&F		
			CPLEX1h	CPLEX10h	LB_CPLEX10h	C&F	LB	Cuts
r13,F01,C1	147837	147349	147349	147349	147349	148494	147089	642
r13,F05,C1	281668	279389	277944	277891	277864	298494	260671	1838
r13,F10,C1	404434	385396	385396	385396	385358	417877	358674	2317
r13,F01,C2	159852	156616	155887	155887	155881	155887	154156	643
r13,F05,C2	311209	295180	295180	295180	295152	298582	280919	1194
r13,F10,C2	470034	434383	434383	431140	425171	454625	397156	1572
r13,F01,C8	225339	218787	218787	218787	218765	224632	213542	293
r13,F05,C8	512027	491959	491804	491560	487012	497877	471640	615
r13,F10,C8	875984	791213	782049	782049	781971	798947	749575	792
r14,F01,C1	431562	422709	422709	422709	422667	423538	418358	1313
r14,F05,C1	811102	784626	790716	784626	780474	812423	744067	2752
r14,F10,C1	1193950	1137820	1145783	1123130	1070240	1156950	1032480	3524
r14,F01,C2	465762	453434	452591	452591	452546	457421	449359	1061
r14,F05,C2	942678	891138	884673	884673	862760	890673	845183	1757
r14,F10,C2	1401880	1307770	1317261	1316857	1246421	1336490	1234430	2313
r14,F01,C8	720882	702614	702781	702614	702544	708444	692487	582
r14,F05,C8	1795650	1693240	1695949	1690451	1668787	1706840	1629040	1042
r14,F10,C8	2997290	2769360	2787042	2755700	2732725	2772750	2629770	983
r15,F01,C1	1039440	1017740	1017740	1017740	1017642	1019180	1008050	1791
r15,F05,C1	2170310	2055803	2024138	2024138	1963704	2028140	1943490	3707
r15,F10,C1	3194270	2971500	3028908	2986773	2845332	3003990	2819330	4407
r15,F01,C2	1205790	1174520	1176047	1174518	1172816	1182020	1163080	1475
r15,F05,C2	2698680	2561060	2681189	2571081	2492512	2574700	2476560	2477
r15,F10,C2	4447950	4045030	4125923	4016242	3892952	4176330	3839540	2697
r15,F01,C8	2472860	2408210	2401176	2401115	2400875	2403330	2378240	551
r15,F05,C8	6067350	5796510	5795320	5795320	5794821	5797170	5766980	789
r15,F10,C8	10263600	9129360	9105014	9105014	9104291	9115830	9082920	830
r16,F01,C1	142692	140082	140082	140082	140077	142797	139829	1045
r16,F05,C1	261775	248703	251554	248703	248679	277712	237068	3122
r16,F10,C1	374819	350958	348805	340641	340610	359648	320039	3712
r16,F01,C2	145266	142607	142381	142381	142367	159168	141588	1004
r16,F05,C2	277307	260822	259639	259313	259287	285509	246645	2244
r16,F10,C2	391386	368572	368753	365001	357878	376114	338684	2610
r16,F01,C8	187176	180228	180132	179639	179621	183475	174549	489
r16,F05,C8	423320	388180	387580	387580	381166	393541	369945	1229
r16,F10,C8	649121	598835	599513	599513	582877	610267	562970	1392
r17,F01,C1	374016	365788	364784	364784	364750	368841	361737	2054
r17,F05,C1	718135	676528	693562	686149	648769	717089	635288	4183
r17,F10,C1	1041450	966116	1006780	964373	890301	991205	872746	5310
r17,F01,C2	393608	384579	382593	382593	382555	388625	379451	1559
r17,F05,C2	786198	741744	739859	739041	717796	744146	702961	3135
r17,F10,C2	1162290	1086640	1138826	1082684	1015353	1126380	999737	3496
r17,F01,C8	539817	529876	530029	529350	523241	535474	518813	819
r17,F05,C8	1348750	1230910	1229810	1226834	1198304	1241990	1185310	1379
r17,F10,C8	2227780	1999950	2024019	2000342	1951428	2064630	1894980	1528
r18,F01,C1	864425	844260	846152	845529	836211	848636	828063	2930
r18,F05,C1	1640200	1588890	1689474	1585372	1533535	1615730	1523250	5958
r18,F10,C1	2399230	2264470	2484100	2253909	2133408	2286290	2122450	6623
r18,F01,C2	962402	944708	942674	940628	929343	945562	923086	2420
r18,F05,C2	1958160	1883870	2178442	1873785	1821051	1904830	1812830	3993
r18,F10,C2	2986000	2806020	3123686	2809646	2689857	2809030	2680660	3977
r18,F01,C8	1617320	1542500	1593899	1534865	1513808	1547430	1504740	1150
r18,F05,C8	4268580	4039410	4243216	3965429	3910784	3961280	3829710	1436
r18,F10,C8	7440780	6603500	7238683	6570234	6375563	6602450	6243600	1321

Table 9: Results for the 54 instances of Set **R**

Tables 10 and 11 display the detailed comparative results for instances in sets **C** and **R**, respectively. Columns C&F/TS, C&F/TS-PR, C&F/CPLEX1h, and C&F/CPLEX10h correspond to the improvement gaps of the fixing matheuristic solution with respect to the Tabu Search of Pedersen et al. (2009), the Tabu Search Path Relinking of Vu et al. (2012), and CPLEX after 1 hour and 10 hours, respectively. The last two columns display the gaps between C&F and the lower bounds computed by CPLEX and the cutting-plane procedure, respectively.

Instance	C&F /TS	C&F /TS-PR	C&F /CPLEX1h	C&F /CPLEX10h	C&F /LB_CPLEX10h	C&F /LB
C20,230,200,V,L	-4.28%	1.44%	0.19%	1.44%	3.75%	4.99%
C20,230,200,F,L	-6.36%	1.66%	0.64%	0.64%	4.68%	5.43%
C20,230,200,V,T	-0.26%	2.31%	1.95%	2.80%	3.47%	4.58%
C20,230,200,F,T	-5.12%	2.58%	0.13%	2.37%	4.69%	5.76%
C20,300,200,V,L	-3.23%	2.96%	2.98%	2.98%	4.96%	5.86%
C20,300,200,F,L	-6.43%	0.82%	0.85%	0.90%	4.12%	5.14%
C20,300,200,V,T	-2.65%	0.12%	0.26%	0.44%	0.45%	1.36%
C20,300,200,F,T	-2.58%	1.44%	0.93%	1.47%	3.96%	4.64%
C30,520,100,V,L	-0.44%	1.70%	1.66%	1.88%	2.41%	3.28%
C30,520,100,F,L	-0.29%	3.97%	4.33%	5.15%	9.74%	10.18%
C30,520,100,V,T	-2.38%	0.36%	0.36%	0.36%	0.66%	1.37%
C30,520,100,F,T	1.02%	4.55%	3.14%	4.48%	7.70%	8.06%
C30,520,400,V,L	-3.92%	-0.27%	-0.23%	0.43%	1.52%	1.67%
C30,520,400,F,L	-5.61%	0.21%	na	0.55%	2.49%	2.59%
C30,520,400,V,T	-2.87%	-0.15%	-0.14%	0.25%	1.02%	1.13%
C30,520,400,F,T	-3.87%	0.02%	na	0.59%	2.29%	2.32%
C30,700,100,V,L	-0.33%	0.94%	1.17%	1.17%	1.18%	2.19%
C30,700,100,F,L	-2.61%	1.38%	1.34%	1.38%	3.91%	4.54%
C30,700,100,V,T	-2.71%	0.25%	0.38%	0.38%	0.85%	1.78%
C30,700,100,F,T	-0.87%	2.54%	2.36%	2.66%	4.04%	4.61%
C30,700,400,V,L	-4.50%	-1.14%	-2.42%	-0.26%	1.58%	1.65%
C30,700,400,F,L	-12.49%	-1.02%	na	0.99%	4.35%	4.37%
C30,700,400,V,T	-5.47%	-0.14%	-0.10%	0.04%	1.89%	1.97%
C30,700,400,F,T	-6.82%	-0.63%	na	0.02%	2.29%	2.36%
Average	-3.40%	1.73%	0.19%	2.07%	4.11%	5.40%

Table 10: Improvement with respect to other methods, **C** instances

Tables 12 and 13 display for each instance in sets **C** and **R**, respectively, the CPU computational time for each of the methods TS-PR, CPLEX1h, CPLEX10h, the cutting-plane (Column LB), the α-*fixing* heuristicc (Column αF), and the total time of the Cut&Fix matheuristic (Column C&F).

Instance	C&F /TS	C&F /TS-PR	C&F /CPLEX1h	C&F /CPLEX10h	C&F /LB_CPLEX10h	C&F /LB
r13,F01,C1	0.44%	0.77%	0.77%	0.77%	0.77%	0.95%
r13,F05,C1	5.64%	6.40%	6.88%	6.90%	6.91%	12.67%
r13,F10,C1	3.22%	7.77%	7.77%	7.77%	7.78%	14.17%
r13,F01,C2	-2.54%	-0.47%	0.00%	0.00%	0.00%	1.11%
r13,F05,C2	-4.23%	1.14%	1.14%	1.14%	1.15%	5.92%
r13,F10,C2	-3.39%	4.45%	4.45%	5.17%	6.48%	12.64%
r13,F01,C8	-0.31%	2.60%	2.60%	2.60%	2.61%	4.94%
r13,F05,C8	-2.84%	1.19%	1.22%	1.27%	2.18%	5.27%
r13,F10,C8	-9.64%	0.97%	2.12%	2.12%	2.12%	6.18%
r14,F01,C1	-1.89%	0.20%	0.20%	0.20%	0.21%	1.22%
r14,F05,C1	0.16%	3.42%	2.67%	3.42%	3.93%	8.41%
r14,F10,C1	-3.20%	1.65%	0.97%	2.92%	7.49%	10.76%
r14,F01,C2	-1.82%	0.87%	1.06%	1.06%	1.07%	1.76%
r14,F05,C2	-5.84%	-0.05%	0.67%	0.67%	3.13%	5.11%
r14,F10,C2	-4.89%	2.15%	1.44%	1.47%	6.74%	7.64%
r14,F01,C8	-1.76%	0.82%	0.80%	0.82%	0.83%	2.25%
r14,F05,C8	-5.20%	0.80%	0.64%	0.96%	2.23%	4.56%
r14,F10,C8	-8.10%	0.12%	-0.52%	0.61%	1.44%	5.16%
r15,F01,C1	-1.99%	0.14%	0.14%	0.14%	0.15%	1.09%
r15,F05,C1	-7.01%	-1.36%	0.20%	0.20%	3.18%	4.17%
r15,F10,C1	-6.33%	1.08%	-0.83%	0.57%	5.28%	6.15%
r15,F01,C2	-2.01%	0.63%	0.51%	0.63%	0.78%	1.60%
r15,F05,C2	-4.82%	0.53%	-4.14%	0.14%	3.19%	3.81%
r15,F10,C2	-6.50%	3.14%	1.21%	3.83%	6.79%	8.06%
r15,F01,C8	-2.89%	-0.20%	0.09%	0.09%	0.10%	1.04%
r15,F05,C8	-4.66%	0.01%	0.03%	0.03%	0.04%	0.52%
r15,F10,C8	-12.59%	-0.15%	0.12%	0.12%	0.13%	0.36%
r16,F01,C1	0.07%	1.90%	1.90%	1.90%	1.91%	2.08%
r16,F05,C1	5.74%	10.45%	9.42%	10.45%	10.45%	14.64%
r16,F10,C1	-4.22%	2.42%	3.01%	5.28%	5.29%	11.01%
r16,F01,C2	8.73%	10.40%	10.55%	10.55%	10.56%	11.04%
r16,F05,C2	2.87%	8.65%	9.06%	9.18%	9.18%	13.61%
r16,F10,C2	-4.06%	2.01%	1.96%	2.95%	4.85%	9.95%
r16,F01,C8	-2.02%	1.77%	1.82%	2.09%	2.10%	4.86%
r16,F05,C8	-7.57%	1.36%	1.51%	1.51%	3.14%	6.00%
r16,F10,C8	-6.37%	1.87%	1.76%	1.76%	4.49%	7.75%
r17,F01,C1	-1.40%	0.83%	1.10%	1.10%	1.11%	1.93%
r17,F05,C1	-0.15%	5.66%	3.28%	4.31%	9.53%	11.41%
r17,F10,C1	-5.07%	2.53%	-1.57%	2.71%	10.18%	11.95%
r17,F01,C2	-1.28%	1.04%	1.55%	1.55%	1.56%	2.36%
r17,F05,C2	-5.65%	0.32%	0.58%	0.69%	3.54%	5.53%
r17,F10,C2	-3.19%	3.53%	-1.10%	3.88%	9.86%	11.24%
r17,F01,C8	-0.81%	1.05%	1.02%	1.14%	2.28%	3.11%
r17,F05,C8	-8.60%	0.89%	0.98%	1.22%	3.52%	4.56%
r17,F10,C8	-7.90%	3.13%	1.97%	3.11%	5.48%	8.22%
r18,F01,C1	-1.86%	0.52%	0.29%	0.37%	1.46%	2.42%
r18,F05,C1	-1.51%	1.66%	-4.56%	1.88%	5.09%	5.72%
r18,F10,C1	-4.94%	0.95%	-8.65%	1.42%	6.69%	7.17%
r18,F01,C2	-1.78%	0.09%	0.31%	0.52%	1.72%	2.38%
r18,F05,C2	-2.80%	1.10%	-14.36%	1.63%	4.40%	4.83%
r18,F10,C2	-6.30%	0.11%	-11.20%	-0.02%	4.24%	4.57%
r18,F01,C8	-4.52%	0.32%	-3.00%	0.81%	2.17%	2.76%
r18,F05,C8	-7.76%	-1.97%	-7.12%	-0.10%	1.27%	3.32%
r18,F10,C8	-12.70%	-0.02%	-9.64%	0.49%	3.44%	5.44%
Average	-3.33%	1.87%	0.43%	2.19%	3.82%	5.88%

Table 11: Improvement with respect to other methods, **R** instances

Instance	TS-PR	Cplex		C&F		Total
		CPLEX1h	CPLEX10h	LB	αF	C&F
C20,230,200,V,L	5460	3600	36000	294	346	635
C20,230,200,F,L	5520	3600	36000	421	614	1026
C20,230,200,V,T	4680	3600	36000	189	359	544
C20,230,200,F,T	5040	3600	36000	301	346	634
C20,300,200,V,L	9720	3600	36000	125	313	432
C20,300,200,F,L	6600	3600	36000	247	787	1027
C20,300,200,V,T	5400	3600	8433	136	148	283
C20,300,200,F,T	8460	3600	36000	195	766	958
C30,520,100,V,L	5100	3600	36000	31	191	225
C30,520,100,F,L	7080	3600	36000	389	1243	1630
C30,520,100,V,T	6180	3600	36000	18	65	82
C30,520,100,F,T	9900	3600	36000	112	756	870
C30,520,400,V,L	13260	3600	36000	1316	5333	6830
C30,520,400,F,L	8760	3600	36000	3034	7182	10529
C30,520,400,V,T	12600	3600	36000	942	4456	5413
C30,520,400,F,T	8160	3601	36002	3592	7148	10696
C30,700,100,V,L	4440	3600	5894	29	72	101
C30,700,100,F,L	9180	3600	36000	175	437	631
C30,700,100,V,T	8100	3600	36000	37	89	125
C30,700,100,F,T	9120	3600	36000	55	245	309
C30,700,400,V,L	8160	3600	36000	2547	5127	7893
C30,700,400,F,L	4800	3600	55015	4513	8101	15723
C30,700,400,V,T	13320	3600	36001	1565	6873	8405
C30,700,400,F,T	7800	3600	36000	3174	8067	10953
Average	7785	3600	34389	976	2461	3581

Table 12: CPU time per procedure, **C** instances

Instance	TS-PR	Cplex		C&F		Total
		CPLEX1h	CPLEX10h	LB	αF	C&F
r13,F01,C1	3720	2	2	1	0	1
r13,F05,C1	3780	3600	8210	16	22	39
r13,F10,C1	3780	3548	3548	47	11	57
r13,F01,C2	3720	19	19	1	1	2
r13,F05,C2	3780	1553	1553	7	31	38
r13,F10,C2	3780	3600	36000	18	25	44
r13,F01,C8	3720	1103	1103	1	4	6
r13,F05,C8	4260	3600	36000	4	10	14
r13,F10,C8	3780	3600	24595	8	13	21
r14,F01,C1	4140	863	863	9	11	21
r14,F05,C1	3900	3600	36000	86	168	251
r14,F10,C1	4380	3600	36000	330	3276	3672
r14,F01,C2	3900	416	416	9	11	20
r14,F05,C2	5700	3600	36000	41	80	121
r14,F10,C2	5100	3600	36000	124	122	243
r14,F01,C8	6420	3600	14896	12	21	33
r14,F05,C8	6480	3600	36000	75	360	434
r14,F10,C8	10140	3600	36000	49	496	545
r15,F01,C1	4020	3600	4177	53	108	159
r15,F05,C1	6840	3600	36000	495	701	1199
r15,F10,C1	7500	3600	36000	1543	2981	4399
r15,F01,C2	5220	3600	36000	43	388	431
r15,F05,C2	8760	3600	36000	244	1132	1364
r15,F10,C2	10200	3600	36000	350	257	645
r15,F01,C8	7020	3600	31051	53	430	482
r15,F05,C8	8940	2050	2050	50	132	178
r15,F10,C8	5100	368	368	33	27	59
r16,F01,C1	3720	7	7	5	3	8
r16,F05,C1	3780	3600	11667	88	63	146
r16,F10,C1	3780	3600	19652	301	615	916
r16,F01,C2	3840	23	23	4	5	9
r16,F05,C2	3780	3600	6081	76	81	165
r16,F10,C2	4020	3600	36000	194	570	766
r16,F01,C8	3960	3600	6703	3	5	8
r16,F05,C8	3840	3600	36000	18	27	45
r16,F10,C8	4380	3600	36000	29	13	42
r17,F01,C1	4260	1446	1446	26	16	42
r17,F05,C1	7980	3600	36000	466	48	486
r17,F10,C1	7860	3600	36000	2157	6530	8579
r17,F01,C2	4800	1897	1897	21	28	48
r17,F05,C2	3960	3600	36000	331	377	695
r17,F10,C2	5400	3600	36000	682	617	1321
r17,F01,C8	6900	3600	36000	13	126	139
r17,F05,C8	11220	3600	36000	41	118	159
r17,F10,C8	8220	3600	36000	80	92	170
r18,F01,C1	8160	3600	36000	155	164	320
r18,F05,C1	5760	3600	36000	4002	1708	5677
r18,F10,C1	4320	3600	36000	3092	7290	10184
r18,F01,C2	13140	3600	36000	96	730	831
r18,F05,C2	5460	3600	36000	1497	1056	2428
r18,F10,C2	8400	3600	36000	1188	1392	2787
r18,F01,C8	7620	3600	36000	93	733	827
r18,F05,C8	7680	3600	36000	209	985	1191
r18,F10,C8	4080	3600	36000	158	1601	1756
Average	55711	2980	23932	347	663	1004

Table 13: CPU time per procedure, **R** instances