Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Fixed Charge Transportation Problem: An Exact Algorithm Based on a New Integer Programming Formulation

Roberto Roberti, Enrico Bartolini, Aristide Mingozzi

MANAGEMENT SCIENCE
Vol. 61, No. 6, June 2015, pp. 1275–1291
ISSN 0025-1909 (print) | ISSN 1526-5501 (online)

informs®
http://dx.doi.org/10.1287/mnsc.2014.1947

The Fixed Charge Transportation Problem: An Exact Algorithm Based on a New Integer Programming Formulation

Roberto Roberti
Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi" (DEI), University of Bologna, 40126 Bologna, Italy, roberto.roberti6@unibo.it

Enrico Bartolini
Department of Mathematics and Systems Analysis, Aalto University School of Science, 00076 Aalto, Finland, enrico.bartolini@aalto.fi

Aristide Mingozzi
Department of Mathematics, University of Bologna, 40126 Bologna, Italy, aristide.mingozzi@unibo.it

The fixed charge transportation problem generalizes the well-known transportation problem where the cost of sending goods from a source to a sink is composed of a fixed cost and a continuous cost proportional to the amount of goods sent. In this paper, we describe a new integer programming formulation with exponentially many variables corresponding to all possible flow patterns to sinks. We show that the linear relaxation of the new formulation is tighter than that of the standard mixed integer programming formulation. We describe different classes of valid inequalities for the new formulation and a column generation method to compute a valid lower bound embedded into an exact branch-and-price algorithm. Computational results on test problems from the literature show that the new algorithm outperforms the state-of-the-art exact methods from the literature and can solve instances with up to 70 sources and 70 sinks.

Data, as supplemental material, are available at http://dx.doi.org/10.1287/mnsc.2014.1947.

Keywords: transportation; fixed charge; column generation
History: Received April 12, 2013; accepted March 1, 2014, by Dimitris Bertsimas, optimization. Published online in *Articles in Advance* August 14, 2014.

1. Introduction

The *fixed charge transportation problem* (FCTP) generalizes the well-known *transportation problem* where the cost of sending goods from a source to a sink is composed of a fixed cost and a continuous cost proportional to the amount of goods sent. The FCTP is a special case of the *single-commodity, uncapacitated, fixed-charge network flow problem* (Ortega and Wolsey 2003, Rardin and Wolsey 1993), which itself is a special case of the more general *fixed charge* (*linear programming*) *problem* formulated by Hirsch and Dantzig (1954). In practical applications, the fixed costs may represent toll charges on highways, landing fees at airports, setup costs in production systems, or the cost for building roads (see Palekar et al. 1990). The FCTP arises not only in distribution, transportation, scheduling, and location systems (see Adlakha and Kowalski 2003) but also in allocation of launch vehicles to space missions (Stroup 1967), solid-waste management (Walker 1976), process selection (Hirsch and Dantzig 1968), and teacher assignment (Hultberg and Cardoso 1997). In practice, the FCTP is difficult to solve exactly, and although it has attracted considerable research attention in the literature, current state-of-the-art exact solution methods are only able to consistently solve instances with up to 15 sources and 15 sinks.

The FCTP can be described as follows. A set $S = \{1, 2, \ldots, m\}$ of m sources and a set $T = \{1, 2, \ldots, n\}$ of n sinks are given. Each source $i \in S$ has an available supply a_i of goods, and each sink $j \in T$ requires a quantity b_j of goods from the sources. Both a_i, $i \in S$, and b_j, $j \in T$, are assumed integer, and without loss of generality, we also assume that $\sum_{i\in S} a_i = \sum_{j\in T} b_j$. A complete bipartite graph $G = (S, T, A)$ is associated with the FCTP, where the arc set A is defined as $A = \{(i, j): i \in S, j \in T\}$. A unit cost c_{ij} for transporting a unit of goods from source $i \in S$ to sink $j \in T$ and a fixed cost f_{ij} for using arc (i, j) are associated with each arc $(i, j) \in A$. The FCTP asks to transport all the goods from the sources to the sinks while minimizing the overall fixed and variable costs. Whenever $c_{ij} = 0$, $(i, j) \in A$, the problem is called a *pure fixed charge transportation problem* (PFCTP); see Fisk and McKeown (1979) and Göthe-Lundgren and Larsson (1994).

Most of the exact algorithms presented in the literature for the FCTP are based on the following mixed integer formulation. Define nonnegative variables x_{ij} representing the quantity transported along arc $(i, j) \in A$ and binary variables y_{ij} taking a value 1 if and only if x_{ij} is positive (i.e., arc (i, j) is used). Let $m_{ij} = \min\{a_i, b_j\}$, $(i, j) \in A$. The FCTP can be formulated as follows:

$$\text{(F0)} \quad z(\text{F0}) = \min \sum_{i \in S} \sum_{j \in T} (c_{ij} x_{ij} + f_{ij} y_{ij}) \tag{1}$$

$$\text{s.t.} \sum_{j \in T} x_{ij} = a_i \quad i \in S, \tag{2}$$

$$\sum_{i \in S} x_{ij} = b_j \quad j \in T, \tag{3}$$

$$x_{ij} \le m_{ij} y_{ij} \quad (i, j) \in A, \tag{4}$$

$$x_{ij} \ge 0 \quad (i, j) \in A, \tag{5}$$

$$y_{ij} \in \{0, 1\} \quad (i, j) \in A. \tag{6}$$

In the following, we denote by LF0 the linear relaxation of problem F0 and by $z(\text{LF0})$ its optimal solution cost. Notice that, in any LF0 optimal solution, variables $x_{ij} > 0$ correspond to a basic feasible solution of constraints (2) and (3), and $y_{ij} = x_{ij}/m_{ij}$, $(i, j) \in A$.

1.1. Literature Review

The FCTP was first formulated by Hirsch and Dantzig (1954), who observed that an FCTP optimal solution is attained at an extreme point of the feasible region of the corresponding transportation problem defined by Equations (2) and (3) and inequalities (5). Early exact solution methods have been presented by Murty (1968) and Gray (1971), among others. These methods were mainly based on the idea, introduced by Murty, of searching among the extreme points of the corresponding transportation problem.

Early branch-and-bound algorithms were proposed by Kennington and Unger (1976) and Barr et al. (1981). These methods are based on relaxation LF0 to obtain lower bounds. They use conditional penalties (see Driebeck 1966, Tomlin 1971) representing lower bounds on the change in the objective function caused by forcing the integrality of a fractional 0–1 variable to strengthen the lower bounds and guide the tree search. The algorithm of Barr et al. was specially designed for sparse transportation networks and could solve instances with up to 1,200 fixed charge variables.

Branch-and-bound algorithms using conditional penalties were also developed by Cabot and Erenguc (1984), Cabot and Erenguc (1986), Palekar et al. (1990), and Lamar and Wallace (1997). Cabot and Erenguc (1986) described a branch-and-bound algorithm based on a Lagrangian relaxation of the problem that is used to derive stronger penalties than those used by Kennington and Unger (1976) and Barr et al. (1981). Palekar et al. derived new bounds based on further strengthening the penalties proposed by Cabot and Erenguc (1984). Lamar and Wallace showed that the modified penalties of Palekar et al. are not valid, but a simple change in calculating them restores their validity.

Göthe-Lundgren and Larsson (1994) described an exact algorithm for the PFCTP based on a set covering formulation. This formulation is derived from F0 by dropping the continuous flow variables and adding an exponential number of set covering inequalities that are sufficient to define the feasible set of solutions. Recently, Agarwal and Aneja (2012) studied the structure of the projection polyhedron of formulation F0 in the space of variables y_{ij}. They developed several classes of valid inequalities, which generalize the set covering inequalities, and derived conditions under which such inequalities are facet defining. Their exact method could solve randomly generated instances of both FCTP and PFCTP with up to 15 sources and 15 sinks. To the best of our knowledge, the method of Agarwal and Aneja is currently the state-of-the-art exact method for solving both the FCTP and the PFCTP.

1.2. Contribution of This Paper

In this paper, we propose an exact method for the FCTP based on a new integer programming formulation involving an exponential number of binary variables, each one corresponding to a feasible supply pattern from sources to sinks. This formulation requires choosing a supply pattern for each source to satisfy all sink demands.

We show that the lower bound provided by the linear relaxation of the new formulation is stronger than $z(\text{LF0})$, and we propose several classes of valid inequalities that are shown to significantly improve this lower bound. Such inequalities are either adaptations to the new formulation of known inequalities for LF0 or new ones introduced in this paper.

The new formulation is used to develop a column-and-cut generation method to compute a valid lower bound on the FCTP and an exact branch-and-price algorithm for solving it. Extensive computational results over benchmark instances from the literature indicate that the new exact method is superior to that of Agarwal and Aneja (2012). Moreover, the proposed exact branch-and-price algorithm outperforms the latest version of CPLEX on new randomly generated FCTP and PFCTP instances involving up to 70 sources and 70 sinks.

The rest of the paper is organized as follows. The new integer programming formulation is introduced in §2. Valid inequalities for strengthening the linear relaxation of the new formulation are presented in §3. The pricing problem and the solution method for solving it are described in §4. Section 5 presents the new exact branch-and-price algorithm. Computational results are illustrated in §6. Some conclusions are drawn in §7.

2. A New Formulation for the FCTP

In this section, we propose a new integer programming formulation of the FCTP, called F1, that involves an exponential number of binary variables. We show that the linear relaxation of formulation F1 is stronger than LF0 and can be further strengthened with any valid inequality derived for LF0. In addition, it can be enhanced with new inequalities that cannot easily be added to LF0.

For each source $i \in S$, let $W_i = \{\mathbf{w} \in \mathbb{Z}_+^n\colon \sum_{j\in T} w_j = a_i,\ w_j \le m_{ij},\ j \in T\}$. Let the vectors of the sets W_i, $i \in S$ be indexed in such a way that the vectors $\{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^{|W_1|}\}$ correspond to W_1, $\{\mathbf{w}^{|W_1|+1}, \ldots, \mathbf{w}^{|W_1|+|W_2|}\}$ corresponds to W_2, and so on. Let $\mathcal{W}_i$ be the index set of all vectors of the set W_i (i.e., $\mathcal{W}_i = \{l \in \mathbb{Z}\colon \sum_{s=1}^{i-1}|W_s|+1 \le l \le \sum_{s=1}^{i}|W_s|\}$), and let $\mathcal{W} = \bigcup_{i\in S}\mathcal{W}_i$. In the following, we refer to any vector $\mathbf{w}^l$, $l \in \mathcal{W}_i$ as a *pattern* of source $i \in S$.

Let d_l be the cost of pattern $l \in \mathcal{W}_i$, $i \in S$, defined as $d_l = \sum_{j\in T} c_{ij} w_j^l + \sum_{j\in T\colon w_j^l>0} f_{ij}$, and let ξ_l be a binary variable equal to 1 if and only if pattern $l \in \mathcal{W}$ is in the solution (otherwise, it is equal to 0). The new FCTP formulation, called F1, is the following:

$$\text{(F1)} \quad z(\text{F1}) = \min \sum_{l\in\mathcal{W}} d_l \xi_l \tag{7}$$

$$\text{s.t.} \quad \sum_{l\in\mathcal{W}} w_j^l \xi_l = b_j \quad j \in T, \tag{8}$$

$$\sum_{l\in\mathcal{W}_i} \xi_l = 1 \quad i \in S, \tag{9}$$

$$\xi_l \in \{0,1\} \quad l \in \mathcal{W}. \tag{10}$$

The objective function (7) aims to minimize the costs of the selected patterns. Constraints (8) state that all sink requests must be satisfied. Constraints (9) require that exactly one pattern for each source be chosen. Constraints (10) impose integrality on the variables.

Hereafter, we denote by LF1 the linear relaxation of formulation F1 and by $z(\text{LF1})$, its optimal solution cost.

Notice that any F1 solution $\bar{\boldsymbol{\xi}}$ of cost $\bar{z}(\text{LF1})$ can be transformed into a feasible LF0 solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ of cost $\bar{z}(\text{LF0}) = \bar{z}(\text{LF1})$ by setting

$$\bar{x}_{ij} = \sum_{l\in\mathcal{W}_i} w_j^l \bar{\xi}_l \quad \text{and} \quad \bar{y}_{ij} = \sum_{l\in\mathcal{W}_i\colon w_j^l>0} \bar{\xi}_l \quad (i,j) \in A. \tag{11}$$

The following proposition describes the relationship between $z(\text{LF1})$ and $z(\text{LF0})$.

Proposition 1. *Let $z(\text{LF0})$ and $z(\text{LF1})$ be the cost of an optimal LF0 and LF1 solution, respectively. Then, inequality $z(\text{LF0}) \le z(\text{LF1})$ holds, and this inequality can be strict.*

Proof. Let $\bar{\boldsymbol{\xi}}$ be an optimal LF1 solution of cost $z(\text{LF1})$, and let $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ be the LF0 solution of cost $\bar{z}(\text{LF0}) = z(\text{LF1})$ derived from $\bar{\boldsymbol{\xi}}$ according to expressions (11). Notice that if $0 < w_j^l < m_{ij}$ for some arc $(i,j) \in A$ and some $l \in \mathcal{W}_i$ such that $\bar{\xi}_l > 0$, then from expressions (11), it follows that

$$\bar{x}_{ij} = \sum_{l\in\mathcal{W}_i} w_j^l \bar{\xi}_l = \sum_{l\in\mathcal{W}_i\colon w_j^l>0} w_j^l \bar{\xi}_l < m_{ij} \sum_{l\in\mathcal{W}_i\colon w_j^l>0} \bar{\xi}_i^l = m_{ij}\bar{y}_{ij}.$$

Thus, $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ cannot be an optimal LF0 solution since, from $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, a feasible LF0 solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of cost $\hat{z}(\text{LF0}) < \bar{z}(\text{LF0})$ can be derived by setting $\hat{\mathbf{x}} = \bar{\mathbf{x}}$, and $\hat{y}_{ij} = \hat{x}_{ij}/m_{ij}$, $(i,j) \in A$. Solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is not necessarily an optimal LF0 solution (i.e., $\hat{z}(\text{LF0}) \ge z(\text{LF0})$) but saturates constraints (4) as any LF0 solution does. Therefore, $z(\text{LF1}) > \hat{z}(\text{LF0}) \ge z(\text{LF0})$. □

Notice that because the FCTP is symmetric in the sources and sinks, variables ξ_l could have been defined for sink patterns instead of source patterns, leading to another formulation of the problem. This latter formulation corresponds to formulation F1, where sources and sinks are interchanged. Therefore, two different valid lower bounds on the FCTP can be achieved by solving LF1 with and without interchanging sources and sinks; one of the two lower bounds may be much tighter than the other one. From our computational experience, we observe that whenever $m < n$, it is not convenient to reverse sources and sinks, and it is always more effective to solve the problem with fewer sources than sinks. Whenever $m = n$, one of the two lower bounds may be much tighter than the other, so it is computationally effective to compute both and use the better of the two to solve the FCTP.

3. Valid Inequalities

Relaxation LF1 must be solved by column generation methods since the number of variables can be too large for nontrivial FCTP instances. In this section, we describe several classes of valid inequalities for LF1—either newly introduced in this paper or taken from the literature and adapted to the problem LF1—that do not modify the structure of the pricing problem and whose dual variables can thus be taken into account when solving the pricing problem without affecting its *robustness*.

3.1. Set Covering Inequalities

Set covering (SC) inequalities have been proposed by Göthe-Lundgren and Larsson (1994) and generalized by Agarwal and Aneja (2012) for relaxation LF0 but can also be added to relaxation LF1 as described in the following. These inequalities state that if the total demand of a set of sinks, $L \subseteq T$, is larger than the total supply of a set of sources, $K \subseteq S$, then any feasible F0 solution must contain at least one variable $y_{ij} = 1$ for some arc $(i,j) \in A(S\backslash K, L)$, where $A(S\backslash K, L) = \{(i,j) \in A\colon i \in S\backslash K,\ j \in L\}$.

Let $\mathcal{S} = \{(K, L)\colon K \subseteq S,\ L \subseteq T \text{ s.t. } \sum_{i\in K} a_i < \sum_{j\in L} b_j\}$. Lower bound $z(\text{LF0})$ can be improved by adding, to relaxation LF0, the following inequalities:

$$\sum_{i\in S\setminus K}\sum_{j\in L} y_{ij} \geq 1 \quad (K, L) \in \mathcal{S}. \tag{12}$$

By using inequalities (12), Göthe-Lundgren and Larsson (1994) derived a set covering formulation for the PFCTP. Agarwal and Aneja (2012) showed that, under certain conditions, inequalities (12) are facet defining for the projection of LF0 in the space of the binary variables y_{ij}.

Define for each pattern $l \in \mathcal{W}_i$, $i \in S\setminus K$ the coefficient $\tau_l(K, L)$ indicating the number of sinks of the set L served by pattern l; that is,

$$\tau_l(K, L) = |\{j \in L\colon w_j^l > 0\}| \quad l \in \mathcal{W}_i,\ i \in S\setminus K.$$

Because any LF1 solution can be transformed into a feasible LF0 solution by means of expressions (11), then inequalities (12) can be cast into relaxation LF1 as follows:

$$\text{(SC)} \quad \sum_{i\in S\setminus K}\sum_{l\in \mathcal{W}_i} \tau_l(K, L)\xi_l \geq 1 \quad (K, L) \in \mathcal{S}. \tag{13}$$

3.1.1. Separation of SC Inequalities (13)**.** Given an optimal LF1 solution, $\boldsymbol{\xi}$, a subset $(K^*, L^*) \in \mathcal{S}$ corresponding to a maximally violated inequality (13) can be identified by solving the following binary problem (see Agarwal and Aneja 2012). Define m binary variables ϑ_i, $i \in S$, where $\vartheta_i = 0$ if and only if $i \in K^*$ ($\vartheta_i = 1$ otherwise), and n binary variables φ_j, $j \in T$, where $\varphi_j = 1$ if and only if $j \in L^*$ ($\varphi_j = 0$ otherwise). Furthermore, define binary variables ζ_{ij}, for each arc $(i, j) \in A$, such that $\zeta_{ij} = 1$ if and only if $i \notin K^*$ and $j \in L^*$ ($\zeta_{ij} = 0$ otherwise). Let $\bar{y}_{ij} = \sum_{l\in\mathcal{W}_i\colon w_j^l>0} \xi_l$. The solution to the following problem provides a pair $(K^*, L^*) \in \mathcal{S}$, if such a couple exists, that violates the corresponding inequality (13):

$$z^* = \min \sum_{(i,j)\in A} \bar{y}_{ij}\zeta_{ij} \tag{14}$$

$$\text{s.t.} \quad \zeta_{ij} \geq \vartheta_i + \varphi_j - 1 \quad (i, j) \in A, \tag{15}$$

$$\sum_{i\in S} a_i\vartheta_i + \sum_{j\in T} b_j\varphi_j \geq \sum_{i\in S} a_i + 1, \tag{16}$$

$$\zeta_{ij} \in \{0, 1\} \quad (i, j) \in A, \tag{17}$$

$$\vartheta_i \in \{0, 1\} \quad i \in S, \tag{18}$$

$$\varphi_j \in \{0, 1\} \quad j \in T. \tag{19}$$

If $z^* < 1$, then the inequality (13) defined by the pair $(K^*, L^*) \in \mathcal{S}$, where $K^* = \{i \in S\colon \vartheta_i = 0\}$ and $L^* = \{j \in T\colon \varphi_j = 1\}$, is violated by solution $\boldsymbol{\xi}$; otherwise (i.e., if $z^* \geq 1$), no such an inequality exists. Ortega and Wolsey (2003) showed that problem (14)–(19) is $\mathcal{NP}$-hard. Nonetheless, in our computational experience, it turned out to be not too time consuming to solve problem (14)–(19) by using a generic integer programming (IP) solver.

3.2. Extended Generalized Upper Bound Cover Inequalities

Generalized upper bound cover (GUBC) inequalities are well-known inequalities for general mixed integer problems having binary variables subject to both knapsack and generalized upper bound constraints. The GUBC inequalities with different lifting procedures have been thoroughly studied in the literature (for an exhaustive introduction on the subject, the reader is referred to Wolsey 1998; Gu et al. 1998, 1999). In this section, we describe new extended GUBC inequalities for the FCTP that arise with reference to constraint (8) for a given sink $j \in T$ and to constraints (9). The coefficients of these new extended GUBC inequalities can be computed even if the number of variables is exponential (as in problem LF1), and their dual variables can easily be handled when pricing out columns for solving relaxation LF1.

Observe that formulation (7)–(10) is still valid if constraints (9) are relaxed to $\sum_{l\in\mathcal{W}_i} \xi_l \leq 1$, $i \in S$. This is because we assumed that $\sum_{i\in S} a_i = \sum_{j\in T} b_j$.

For a sink $j \in T$, let $\mathcal{P}_j$ be the set of all feasible integer solutions of the corresponding equality knapsack constraint (8) along with the m constraints (9), where the equals sign is replaced with a less than or equal to sign (i.e., $\mathcal{P}_j = \{\xi_l \in \{0, 1\},\ l \in \mathcal{W}\colon \sum_{l\in\mathcal{W}} w_j^l\xi_l = b_j \text{ and } \sum_{l\in\mathcal{W}_i} \xi_l \leq 1,\ i \in S\}$).

A GUBC for the set $\mathcal{P}_j$ (see Gu et al. 1999) is a subset $C \subseteq \mathcal{W}$ of variables of total weight greater than b_j (i.e., $\sum_{l\in C} w_j^l > b_j$) and such that no two variables of C belong to the same set $\mathcal{W}_i$ (i.e., $|C \cap \mathcal{W}_i| \leq 1$, $i \in S$). A GUBC is *minimal* if $\sum_{l\in C\setminus\{r\}} w_j^l \leq b_j$ for each variable $r \in C$. Let $\mathcal{C}_j$ be the set of all minimal GUBCs of the set $\mathcal{P}_j$. Any F1 solution satisfies the following constraints, known as GUBC inequalities:

$$\text{(GUBC)} \quad \sum_{l\in C} \xi_l \leq |C| - 1 \quad C \in \mathcal{C}_j,\ j \in T. \tag{20}$$

Different lifting strategies of inequalities (20) can be performed (see Gu et al. 1999). In the following, we propose a lifting strategy that can easily be handled within a column generation where an exponential number of variables must be considered.

For each GUBC $C \in \mathcal{C}_j$ of a given sink $j \in T$, let $\rho(C)$ be the maximum coefficient w_j^l associated with the variables of C (i.e., $\rho(C) = \max_{l\in C}\{w_j^l\}$), and let vector $\boldsymbol{\gamma}(C) = \{\gamma_1(C), \gamma_2(C), \ldots, \gamma_m(C)\}$ be defined as

$$\gamma_i(C) = \begin{cases} w_j^l & \text{if } \exists\, l \in C \cap \mathcal{W}_i, \\ \rho(C) & \text{otherwise}, \end{cases} \quad i \in S,\ C \in \mathcal{C}_j,\ j \in T.$$

Then, the following inequalities, called *extended generalized upper bound cover* (EGUBC) inequalities, lift up inequalities (20) and are valid for problem F1:

$$\text{(EGUBC)} \quad \sum_{i\in S}\ \sum_{l\in\mathcal{W}_i\colon w_j^l\geq\gamma_i(C)} \xi_l \leq |C| - 1 \quad C \in \mathcal{C}_j,\ j \in T. \tag{21}$$

3.2.1. Separation of EGUBC Inequalities (21). Given a fractional LF1 solution, $\boldsymbol{\xi}$, the separation of a violated EGUBC inequality $C^* \in \mathscr{C}_j$ for a given sink $j \in T$ is performed with a heuristic algorithm that identifies violated EGUBC inequalities in their downlifted version where $\rho(C)$ is set equal to ∞. The problem of separating such inequalities can be formulated as follows. Let f_i^q be the fractional number of patterns sending at least q units from source $i \in S$ to sink $j \in T$, defined as $f_i^q = \sum_{l \in \mathscr{W}_i: w_j^l \geq q} \xi_i^l$. Let φ_i^q, $i \in S$, $q = 1, \ldots, m_{ij}$, be a binary variable equal to 1 if and only if a pattern $l \in \mathscr{W}_i$ such that $w_j^l = q$ belongs to C^*, and then, $\gamma_i(C^*) = q$. The separation problem can be formulated as follows:

$$z^* = \max \sum_{i \in S} \sum_{q=1}^{m_{ij}} (f_i^q - 1)\varphi_i^q \tag{22}$$

$$\text{s.t.} \sum_{i \in S} \sum_{q=1}^{m_{ij}} q\varphi_i^q = b_j + 1, \tag{23}$$

$$\sum_{q=1}^{m_{ij}} \varphi_i^q \leq 1 \quad i \in S, \tag{24}$$

$$\varphi_i^q \in \{0, 1\} \quad i \in S,\ q = 1, \ldots, m_{ij}. \tag{25}$$

Whenever the optimal solution cost z^* of problem (22)–(25) is strictly greater than -1, there exists a violated inequality (21) given by $C^* = \bigcup_{i \in S} \{l_i^*\}$, where $l_i^* = \min\{l \in \mathscr{W}_i: \varphi_i^q = 1,\ w_j^l = q\}$, $i \in S$. Problem (22)–(25) can be solved by using a dynamic programming recursion.

3.3. Couple Inequalities

Consider the set $\mathscr{P}_j$ of solutions as defined in §3.2 for a given sink $j \in T$. It is easy to observe that any solution $\boldsymbol{\xi} \in \mathscr{P}_j$ contains at most two variables $h, k \in \mathscr{W}$ equal to 1 such that $b_j/2 < w_j^h < b_j$ and $(b_j - w_j^h)/2 < w_j^k < b_j - w_j^h$. In this case, there must be at least one variable $s \in \mathscr{W}$ such that $0 < w_j^s \leq b_j - w_j^h - w_j^k$, and $\xi_s = 1$. Consider a source $i \in S$ and $q_1, q_2 \in \mathbb{Z}_+$ such that $b_j/2 < q_1 < b_j$, $(b_j - q_1)/2 < q_2 < b_j - q_1$. For each quadruplet (i, j, q_1, q_2) as defined above, the inequality

$$\sum_{l \in \mathscr{W}: w_j^l \leq b_j - q_1 - q_2} \xi_l - \sum_{l \in \mathscr{W}: w_j^l = q_1} \xi_l - \sum_{l \in \mathscr{W}_i: w_j^l = q_2} \xi_l \geq -1 \tag{26}$$

is satisfied by any solution $\boldsymbol{\xi} \in \mathscr{P}_j$. Notice that, in the third summation of (26), the restriction that l must belong to $\mathscr{W}_i$ instead of $\mathscr{W}$ is necessary for the correctness of the cut. Consider the following example where $b_j = 12$, $q_1 = 7$, and $q_2 = 4$; sink j could be served by three patterns from three different sources, each one taking q_2 units to sink j. If $\mathscr{W}_i$ is replaced by $\mathscr{W}$ under the third summation of (26), the aforementioned solution would be cut off.

Let $\mathscr{Q}$ be defined as $\mathscr{Q} = \{(i, j, q_1, q_2): i \in S,\ j \in T,\ b_j/2 < q_1 < b_j,\ (b_j - q_1)/2 < q_2 < b_j - q_1\}$. For each $Q = (i, j, q_1, q_2) \in \mathscr{Q}$, define the following coefficient for each pattern $l \in \mathscr{W}$:

$$\pi_l(Q) = \begin{cases} 1 & \text{if } w_j^l \leq b_j - q_1 - q_2, \\ -1 & \text{if } w_j^l = q_1, \\ -1 & \text{if } w_j^l = q_2 \text{ and } l \in \mathscr{W}_i, \\ 0 & \text{otherwise.} \end{cases}$$

The following family of inequalities, which we call *couple* (CPL) inequalities, can be added to problem LF1:

$$\text{(CPL)} \quad \sum_{l \in \mathscr{W}} \pi_l(Q)\xi_l \geq -1 \quad Q \in \mathscr{Q}. \tag{27}$$

3.3.1. Separation of CPL Inequalities (27). Given an optimal solution $\boldsymbol{\xi}$ of problem LF1, inequalities (27) can easily be separated by complete enumeration of the quadruple $Q \in \mathscr{Q}$.

3.4. Feasibility Inequalities

In this section, we introduce a family of valid inequalities derived by considering constraint (8) for a given sink $j \in T$. These inequalities are valid for any feasible solution of a binary integer problem where variables have positive integer coefficients and are subject to an equality constraint, but they do not hold whenever the equality constraint is relaxed to "less than or equal to" or "greater than or equal to."

Consider a feasible FCTP solution $\boldsymbol{\xi}$ and a given sink $j \in T$. If there exists a pattern $k \in \mathscr{W}$ such that $\xi_k = 1$ and $w_j^k > b_j/2$, then for any pattern $l \in \mathscr{W}$ such that $w_j^l > b_j - w_j^k$, variable ξ_l must be equal to zero. Therefore, solution $\boldsymbol{\xi}$ satisfies the following equality:

$$\sum_{l \in \mathscr{W}: w_j^l \leq b_j - w_j^k} w_j^l \xi_l = b_j - w_j^k.$$

Given an integer $q \in \mathbb{Z}_+$ such that $b_j/2 < q < b_j$, solution $\boldsymbol{\xi}$ can contain at most one variable $\xi_k = 1$ such that $w_j^k \geq q$. In this case, $\boldsymbol{\xi}$ must contain one or more variables ξ_l with $w_j^l \leq b_j - w_j^k$ such that $\sum_{l \in \mathscr{W}: w_j^l \leq b_j - w_j^k} w_j^l \xi_l = b_j - w_j^k$. This observation leads to the following valid inequality, complied by any feasible FCTP solution, for the given sink $j \in T$ and integer q as defined above:

$$\sum_{l \in \mathscr{W}: w_j^l \leq b_j - q} w_j^l \xi_l \geq \sum_{l \in \mathscr{W}: q \leq w_j^l < b_j} (b_j - w_j^l)\xi_l. \tag{28}$$

Let $\mathscr{F}$ be the set of all couples (j, q) such that $j \in T$ and $b_j/2 < q < b_j$ (i.e., $\mathscr{F} = \{(j, q): j \in T,\ b_j/2 < q < b_j\}$). For each couple (j, q) and each pattern $l \in W$, define the following coefficients:

$$\lambda_l(j, q) = \begin{cases} w_j^l & \text{if } w_j^l \leq b_j - q, \\ w_j^l - b_j & \text{if } q \leq w_j^l < b_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then, inequalities (28) can be cast into problem LF1 as

$$\text{(FSB)}\quad \sum_{l\in\mathscr{W}} \lambda_l(j,q)\xi_l \geq 0 \quad (j,q)\in\mathscr{F}. \tag{29}$$

Even though inequalities (29) are quite general, to the best of our knowledge, they have not been proposed in the literature. In the following, we will refer to them as *feasibility* (FSB) inequalities.

3.4.1. Separation of FSB Inequalities (29)**.** Given an optimal solution $\boldsymbol{\xi}$ of problem LF1, inequalities (29) can easily be separated by complete enumeration of the couples $(j,q)\in\mathscr{F}$.

3.5. Chvátal–Gomory Inequalities

Any F1 solution must satisfy the following well-known *Chvátal–Gomory* inequalities (see Nemhauser and Wolsey 1988), defined by considering a single constraint (8) and an integer multiplier $q\in\mathbb{Z}$:

$$\text{(CGD)}\quad \sum_{l\in\mathscr{W}} \left\lfloor \frac{w_j^l}{q} \right\rfloor \xi_l \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2\leq q<b_j,\ j\in T, \tag{30}$$

$$\text{(CGU)}\quad \sum_{l\in\mathscr{W}} \left\lceil \frac{w_j^l}{q} \right\rceil \xi_l \geq \left\lceil \frac{b_j}{q} \right\rceil \quad 2\leq q<b_j,\ j\in T. \tag{31}$$

In the following, we refer to inequalities (30) and (31) as CGD and CGU inequalities, respectively.

3.5.1. Separation of CGD and CGU Inequalities. Given an optimal solution $\boldsymbol{\xi}$ of problem LF1, inequalities (30) and (31) can easily be separated by complete enumeration.

3.6. Lifted CGD Inequalities

In this section, we introduce a class of valid inequalities for problem (7)–(10) that lift up inequalities (30) and are computationally effective in tightening lower bound $z(\text{LF1})$ without affecting the robustness of the pricing problem.

Consider inequality (30) for a given sink $j\in T$ and a given $2\leq q<b_j$. Such an inequality states that at most $\lfloor b_j/q\rfloor$ patterns ξ_l sending a flow q to sink j can belong to any feasible FCTP solution and that if a pattern l sends a flow strictly greater than q to sink j, it can be considered with a coefficient $\lfloor w_j^l/q\rfloor$. Now consider a source $r\in S$ and a pattern $l^*\in\mathscr{W}_r$ such that $w_j^{l^*} = b_j-\lfloor b_j/q\rfloor q+1$. It is easy to see that inequality (30) can be lifted up by adding variable ξ_{l^*} as follows:

$$\sum_{l\in\mathscr{W}:\, w_j^l\geq q} \left\lfloor \frac{w_j^l}{q} \right\rfloor \xi_l + \xi_{l^*} \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2\leq q<b_j,\ j\in T. \tag{32}$$

Inequality (32) can be further strengthened by adding any other variable ξ_l, $l\in\mathscr{W}_r$ such that $w_j^l\geq w_j^{l^*}$. Therefore, because of constraint (9) for a given source $r\in S$, we can further lift up constraints (32) as follows:

$$\sum_{i\in S}\sum_{l\in\mathscr{W}_i:\, w_j^l\geq q} \left\lfloor \frac{w_j^l}{q} \right\rfloor \xi_l + \sum_{l\in\mathscr{W}_r:\, \rho_{qj}\leq w_j^l<q} \xi_l \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2\leq q<b_j,\ r\in S,\ j\in T, \tag{33}$$

where $\rho_{qj} = b_j-\lfloor b_j/q\rfloor q+1$.

For each source $r\in S$, sink $j\in T$, and flow q, $2\leq q<b_j$, by defining

$$s_l(q,r,j) = \begin{cases} 1 & \text{if } l\in\mathscr{W}_r \text{ and } \rho_{qj}\leq w_j^l<q,\\ \lfloor w_j^l/q\rfloor & \text{if } w_j^l\geq q,\\ 0 & \text{otherwise,}\end{cases}$$

we can rewrite constraints (33) as

$$\text{(LCGD)}\quad \sum_{l\in\mathscr{W}} s_l(q,r,j)\xi_j^l \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2\leq q<b_j,\ r\in S,\ j\in T. \tag{34}$$

3.6.1. Separation of LCGD Inequalities. Given an optimal solution $\boldsymbol{\xi}$ of problem LF1, inequalities (34) can easily be separated by complete enumeration.

4. Solving the Pricing Problem

Let $\overline{\text{LF1}}$ be the problem obtained by adding inequalities (13), (21), (27), (29)–(31), and (34) to problem LF1. In this section, we describe how to price out columns when solving problem $\overline{\text{LF1}}$ in order to compute a valid lower bound to the FCTP.

Let $v_j\in\mathbb{R}$ $(j\in T)$, $u_i\in\mathbb{R}$ $(i\in S)$, $g_{KL}\in\mathbb{R}_+$ $((K,L)\in\mathscr{S})$, $h_{Cj}\in\mathbb{R}_-$ $(C\in\mathscr{C}_j,\ j\in T)$, $\eta_Q\in\mathbb{R}_+$ $(Q\in\mathscr{Q})$, $\alpha_{jq}\in\mathbb{R}_+$ $((j,q)\in\mathscr{F})$, $\mu_{qj}\in\mathbb{R}_-$ $(2\leq q\leq b_j,\ j\in T)$, $\chi_{qj}\in\mathbb{R}_+$ $(2\leq q\leq b_j,\ j\in T)$, and $\psi_{qrj}\in\mathbb{R}_-$ $(2\leq q\leq b_j,\ r\in S,\ j\in T)$ be the dual variables associated with constraints (8), (9), (13), (21), (27), (29), (30), (31), and (34), respectively.

Given an $\overline{\text{LF1}}$ dual solution $(\mathbf{v},\mathbf{u},\mathbf{g},\mathbf{h},\boldsymbol{\eta},\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\chi},\boldsymbol{\psi})$, the pricing problem consists of finding patterns of the set $\mathscr{W}$ having the most negative reduced cost. For a given source $i\in S$, the reduced cost, d_l', of pattern $l\in\mathscr{W}_i$ is computed as

$$\begin{aligned} d_l' = d_l &- \sum_{j\in T} w_j^l v_j - u_i - \sum_{(K,L)\in\mathscr{S}} \tau_l(K,L) g_{KL}\\ &- \sum_{j\in T}\sum_{C\in\mathscr{C}_j:\, w_j^l\geq\gamma_i(C)} h_{Cj} - \sum_{Q\in\mathscr{Q}} \pi_l(Q)\eta_Q\\ &- \sum_{(j,q)\in\mathscr{F}} \lambda_l(j,q)\alpha_{jq} - \sum_{j\in T}\sum_{q=2}^{b_j-1} \left\lfloor \frac{w_j^l}{q}\right\rfloor \mu_{qj}\\ &- \sum_{j\in T}\sum_{q=2}^{b_j-1} \left\lceil \frac{w_j^l}{q}\right\rceil \chi_{qj} - \sum_{r\in S}\sum_{j\in T}\sum_{q=2}^{b_j-1} s_l(q,r,j)\psi_{qrj}. \end{aligned}$$

The reduced cost d'_l of pattern $l \in \mathscr{W}_i$ can be decomposed by its arcs as follows. Define the modified cost δ^{σ}_{ij} with respect to $(\mathbf{v}, \mathbf{u}, \mathbf{g}, \mathbf{h}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\chi}, \boldsymbol{\psi})$ for sending σ units of goods $(1 \le \sigma \le m_{ij})$ from source $i \in S$ to sink $j \in T$ as

$$\begin{aligned}\delta^{\sigma}_{ij} &= f_{ij} + \sigma c_{ij} - \sigma v_j - \sum_{\substack{(K,L)\in\mathscr{S}:\\ i\in S\setminus K,\, j\in K}} g_{KL} - \sum_{\substack{C\in\mathscr{C}_j:\\ \gamma_i(C)\le\sigma}} h_{Cj}\\
&\quad - \sum_{\substack{(j,r)\in\mathscr{F}:\\ b_j - r\ge\sigma}} \sigma\alpha_{jr} - \sum_{\substack{(j,r)\in\mathscr{F}:\\ r\le\sigma}} (\sigma - b_j)\alpha_{jr} - \sum_{\substack{Q=(r,j,q_1,q_2)\in\mathscr{Q}:\\ b_j - q_1 - q_2\ge\sigma}} \eta_Q\\
&\quad + \sum_{Q=(r,j,\sigma,q_2)\in\mathscr{Q}} \eta_Q + \sum_{Q=(i,j,q_1,\sigma)\in\mathscr{Q}} \eta_Q - \sum_{q=2}^{\sigma} \left\lfloor \frac{\sigma}{q} \right\rfloor \mu_{qj}\\
&\quad - \sum_{q=2}^{b_j-1} \left\lceil \frac{\sigma}{q} \right\rceil \chi_{qj} - \sum_{r\in S}\sum_{q=2}^{\sigma} \left\lfloor \frac{\sigma}{q} \right\rfloor \psi_{qrj} - \sum_{q=\rho_{qj}}^{\sigma-1} \psi_{qij}.\end{aligned}$$

By using the modified cost δ^{σ}_{ij}, the reduced cost of pattern $l \in \mathscr{W}_i$, $i \in S$, can equivalently be expressed as

$$d'_l = \sum_{j\in T:\, w^l_j>0} \delta^{w^l_j}_{ij} - u_i.$$

The problem of generating a pattern of minimum reduced cost for a given source $i \in S$ can be formulated as follows. Let $\varphi_{j\sigma}$ be a binary variable equal to 1 if σ units of goods $(1 \le \sigma \le m_{ij})$ are sent from source i to sink $j \in T$. The pricing problem for source $i \in S$ can be formulated as a *multiple choice knapsack problem* as

$$z_i = \min \sum_{j\in T}\sum_{\sigma=1}^{m_{ij}} \delta^{\sigma}_{ij}\varphi_{j\sigma} \tag{35}$$

$$\text{s.t. } \sum_{j\in T}\sum_{\sigma=1}^{m_{ij}} \sigma\varphi_{j\sigma} = a_i, \tag{36}$$

$$\sum_{\sigma=1}^{m_{ij}} \varphi_{j\sigma} \le 1 \quad j \in T, \tag{37}$$

$$\varphi_{j\sigma} \in \{0, 1\} \quad j \in T,\ 1 \le \sigma \le m_{ij}. \tag{38}$$

Let $\boldsymbol{\varphi}^*$ be the optimal solution of problem (35)–(38) of cost z^*_i. If $z^*_i - u_i < 0$, there exists a negative reduced-cost pattern $l \in W_i$, defined by the vector $(w^l_1, w^l_2, \ldots, w^l_n)$, where $w^l_j = \sum_{\sigma=1}^{m_{ij}} \sigma\varphi^*_{j\sigma}$. Otherwise (i.e., $z^*_i - u_i \ge 0$), no negative reduced-cost pattern for source $i \in S$ exists.

Problem (35)–(38) can be solved by a dynamic programming recursion in time $O(na_i^2)$. Let $f(\sigma, j)$ be the optimal solution of the subproblem derived from problem (35)–(38) by replacing T with $\{1, 2, \ldots, j\}$ and the right-hand side a_i of constraint (36) with σ $(1 \le \sigma \le a_i)$. For a given source $i \in S$, the recursion for computing functions $f(\sigma, j)$ is

$$f(\sigma, j) = \min\left\{ f(\sigma, j-1),\ \min_{1\le\sigma'\le\max\{m_{ij},\sigma\}} \{f(\sigma - \sigma', j-1) + \delta^{\sigma'}_{ij}\}\right\}$$

for each $j = 1, \ldots, n$ and $\sigma = 1, \ldots, \max\{a_i, \sum_{r=1}^{j} b_r\}$. The initialization $f(0, j) = 0$, $j \in T$, and $f(\sigma, 0) = \infty$, $\sigma = 1, \ldots, a_i$ is required. The cost of a pattern of minimum reduced cost corresponds to the value of function $f(a_i, n)$.

5. An Exact Algorithm for the FCTP

In this section, we describe an exact branch-and-price algorithm for solving problem F1. First, we describe how we compute the lower bound at the root node by using a column-and-cut generation algorithm for solving problem $\overline{\text{LF1}}$. Then we illustrate the branching strategy for finding an optimal FCTP solution.

5.1. A Column-and-Cut Generation Bounding Procedure

We denote by MP a problem $\overline{\text{LF1}}$ with a limited subset of columns of the sets W_i, $i \in S$ and a limited subset of valid inequalities (13), (21), (27), (29)–(31), and (34). First, the bounding procedure solves *LF*1 by column generation to compute the corresponding lower bound $z(\text{LF1})$. Second, it solves $\overline{\text{LF1}}$ by iteratively adding, in a cut-and-column generation fashion, the six classes of valid inequalities described in §3 and columns of negative reduced cost. The bounding procedure terminates when no negative reduced-cost patterns exist and no valid inequalities are violated, and it provides a lower bound, $z(\overline{\text{LF1}})$, corresponding to the optimal solution cost of problem $\overline{\text{LF1}}$. As observed at the end of §2, whenever $m = n$, the bounding procedure described below is executed twice (with and without interchanging sources and sinks), and then, branching to close the gap starts from the formulation that has led to the best lower bound.

A step-by-step description of the bounding procedure is as follows. A parameter Δ is used to indicate how many classes of valid inequalities are considered in the cutting phase. In the following, we indicate by $\boldsymbol{\xi}$ and $(\mathbf{v}, \mathbf{u}, \mathbf{g}, \mathbf{h}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\chi}, \boldsymbol{\psi})$ the optimal primal and dual solutions of problem MP, respectively, and we indicate by $z(\text{MP})$ its optimal solution cost.

Description of the Bounding Procedure

0. *Initialize MP.* Initialize MP with the set of patterns corresponding to a feasible FCTP solution computed with a greedy algorithm.

1. *Solve problem* LF1 *by column generation*.

   a. *Solve MP.* Solve problem MP with a generic linear programming (LP) solver.

   b. *Generate columns*.

      i. Compute, for each source $i \in S$ the pattern of minimum reduced cost with respect to $(\mathbf{v}, \mathbf{u}, \mathbf{g}, \mathbf{h}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\chi}, \boldsymbol{\psi})$, as described in §4.

      ii. If columns of negative reduced cost exist, add them to MP and go to Step 1a.

      iii. Set $z(\overline{\text{LF1}})^0 = z(\text{MP})$ and $\Delta = 1$.

Table 1 **Results on** Data Set 1

		CPLEX			AA12	B&P							
Inst	Opt	LF0	LBr	T_{Tot}	T_{Tot}	LF1	$z(\overline{LF1})$	$\overline{LF1}$	$T_{\overline{LF1}}$	Cols	Cuts	Nds	T_{Tot}
4×64	9,711	0.8	0.2	0.2	1.1	0.2	9,711	0.0	1.9	1,875	56	0	1.9
8×32	5,247	5.9	1.4	2.6	11.0	3.0	5,247	0.0	0.3	735	87	0	0.3
10×10a	1,499	16.4	0.5	0.3	0.4	10.0	1,499	0.0	0.1	130	44	0	0.1
10×10b	3,073	14.9	2.1	0.3	0.7	8.4	3,073	0.0	0.1	93	25	0	0.1
10×10c	13,007	13.9	4.0	0.8	1.9	8.9	12,902	0.8	0.4	136	62	17	0.5
10×12	2,714	10.6	1.3	0.2	0.6	5.6	2,714	0.0	0.1	127	23	0	0.1
10×26	4,270	9.6	4.0	12.3	22.4	4.7	4,243	0.6	1.0	693	244	7	2.2
12×12	2,291	20.3	6.8	5.4	5.6	12.5	2,260	1.4	0.9	283	140	23	1.0
12×21	3,664	13.8	5.7	36.2	19.3	8.5	3,624	1.1	0.5	483	128	35	1.0
13×13	3,252	17.2	6.4	9.0	10.4	12.3	3,207	1.4	0.8	308	131	55	1.2
14×18	3,712	18.7	9.0	818.8	1,307.5	11.9	3,660	1.4	0.5	477	187	81	2.0
16×16	3,823	18.5	6.9	91.9	83.7	11.8	3,753	1.8	1.9	457	200	145	4.0
17×17	1,373	11.4	1.7	2.6	5.5	6.9	1,373	0.0	1.1	515	228	0	1.1
Avg		13.2	3.8	75.4	113.1	8.0		0.7	0.7	486	120	28	1.2

Table 2 **Results on** Data Set 2 **with** $m = 15$, $n = 15$, $B = 20$, **and** $\theta = 0.0$

		CPLEX			AA12		B&P							
Inst	Opt	LF0	LBr	T_{Tot}	LBr	T_{Tot}	LF1	$z(\overline{LF1})$	$\overline{LF1}$	$T_{\overline{LF1}}$	Cols	Cuts	Nds	T_{Tot}
1	6,683	25.5	8.5	14.3	12.2	118.8	15.0	6,603	1.2	0.7	259	99	23	0.9
2	6,903	26.8	8.5	15.8	9.8	95.0	18.1	6,884	0.3	0.8	309	129	5	0.9
3	6,210	22.3	4.8	1.4	14.2	44.6	13.4	6,205	0.1	0.9	264	109	3	0.9
4	7,753	28.9	9.0	47.8	9.8	291.4	20.9	7,675	1.0	0.8	381	143	21	1.1
5	7,360	30.9	8.6	21.1	9.6	165.9	18.1	7,289	1.0	0.5	311	140	9	0.5
6	6,911	19.8	4.7	4.1	8.7	46.5	10.6	6,911	0.0	0.5	255	96	0	0.5
7	6,434	22.3	6.0	7.8	6.8	28.2	11.0	6,434	0.0	0.1	259	120	0	0.1
8	7,254	26.4	8.6	50.1	11.1	400.7	21.6	7,187	0.9	0.6	400	171	17	0.9
9	7,119	20.0	4.3	0.9	11.6	45.6	12.9	7,102	0.2	1.2	275	125	3	1.3
10	6,843	27.9	7.9	39.6	10.1	152.8	18.5	6,817	0.4	0.6	338	131	9	0.7
Avg		25.1	7.1	20.3	10.4	139.0	16.0		0.5	0.7	305	126	9	0.8

2. *Solve problem $\overline{LF1}$ by cut-and-column generation.*
 a. *Separate cuts.*
  i. For each sink $j \in T$, add the CGD inequality (30) and the CGU inequality (31) most violated by $\boldsymbol{\xi}$.
  ii. If $\Delta \geq 2$, for each sink $j \in T$, add the LCGD inequality (34) most violated by $\boldsymbol{\xi}$.
  iii. If $\Delta \geq 3$, for each sink $j \in T$, add the FSB inequality (29) most violated by $\boldsymbol{\xi}$.
  iv. If $\Delta \geq 4$, for each sink $j \in T$, add the GUBC inequality (21) most violated by $\boldsymbol{\xi}$.
  v. If $\Delta \geq 5$, for each sink $j \in T$, add the CPL inequality (27) most violated by $\boldsymbol{\xi}$.

Table 3 **Results on** Data Set 2 **with** $m = 15$, $n = 15$, $B = 20$, **and** $\theta = 0.2$

		CPLEX			AA12		B&P							
Inst	Opt	LF0	LBr	T_{Tot}	LBr	T_{Tot}	LF1	$z(\overline{LF1})$	$\overline{LF1}$	$T_{\overline{LF1}}$	Cols	Cuts	Nds	T_{Tot}
1	10,017	18.2	5.3	14.8	7.5	93.3	11.8	9,917	1.0	0.6	317	108	13	0.7
2	10,075	17.8	5.0	10.5	7.2	89.5	9.2	10,075	0.0	0.1	296	126	0	0.1
3	9,327	16.5	3.7	1.3	10.3	60.2	11.3	9,327	0.0	0.1	225	82	0	0.1
4	11,093	21.3	7.0	214.2	8.3	tl	15.0	11,002	0.8	0.6	361	148	43	1.1
5	10,312	19.7	5.4	9.5	6.2	45.2	13.0	10,286	0.3	0.4	415	132	3	0.4
6	10,086	13.4	2.4	1.4	5.8	21.1	6.6	10,086	0.0	0.4	289	80	0	0.4
7	9,913	17.9	4.4	11.2	6.2	41.5	13.5	9,881	0.3	0.7	423	147	9	0.8
8	10,495	19.7	7.6	95.4	8.5	tl	14.3	10,412	0.8	0.7	353	146	17	0.8
9	10,137	16.5	1.7	0.5	8.9	17.9	10.1	10,137	0.0	0.1	294	101	0	0.1
10	9,939	19.8	4.2	10.2	6.4	41.6	12.7	9,856	0.8	0.6	386	151	21	0.9
Avg		18.1	4.7	36.9	7.5	161.0	11.8		0.4	0.4	336	122	11	0.5

Table 4 **Results on Data Set 2 with $m=15$, $n=15$, $B=20$, and $\theta=0.5$**

		CPLEX			AA12		B&P							
Inst	Opt	LF0	LBr	T_{Tot}	LBr	T_{Tot}	LF1	$z(\overline{LF1})$	$\overline{LF1}$	$T_{\overline{LF1}}$	Cols	Cuts	Nds	T_{Tot}
1	14,161	14.8	4.8	10.1	7.5	53.4	9.5	14,128	0.2	0.6	321	113	7	0.7
2	13,793	12.6	1.5	0.8	4.8	18.3	3.8	13,793	0.0	0.1	223	37	0	0.1
3	13,699	14.3	4.3	8.7	9.0	85.3	10.3	13,656	0.3	0.6	319	128	5	0.7
4	15,246	16.5	4.7	75.6	6.9	tl	9.1	15,196	0.3	0.8	363	158	9	0.9
5	14,593	15.2	4.3	41.4	6.1	96.9	8.8	14,497	0.7	1.0	317	126	15	1.2
6	14,680	11.4	3.4	4.4	5.3	43.7	6.1	14,599	0.6	0.5	263	77	35	0.6
7	14,255	12.5	2.3	7.9	4.6	29.3	4.7	14,255	0.0	0.1	319	122	0	0.1
8	14,235	15.3	4.4	20.0	6.7	141.1	8.4	14,235	0.0	0.1	278	110	0	0.1
9	14,281	16.2	2.9	7.0	8.4	57.7	6.9	14,231	0.4	0.9	278	108	29	1.1
10	13,953	14.6	3.1	5.6	5.0	28.5	9.1	13,857	0.7	0.4	379	148	19	0.6
Avg		14.3	3.6	18.2	6.4	115.4	7.7		0.3	0.5	306	113	12	0.6

vi. If $\Delta \geq 6$, add the SC inequality (13) most violated by $\boldsymbol{\xi}$.

vii. If at least an inequality was added in the current iteration, solve problem MP with a generic LP solver.

b. *Generate columns.*

i. For each source $i \in S$, compute the pattern of minimum reduced cost with respect to $(\mathbf{v}, \mathbf{u}, \mathbf{g}, \mathbf{h}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\chi}, \boldsymbol{\psi})$, as described in §4.

Table 5 **Results on Data Set 3 with $B=20$ and $\theta=0.0$**

		CPLEX				B&P											
Inst	UB	LF0	LBr	bLB	T_{Tot}	LF1	$z(\overline{LF1})$	$\overline{LF1}$	$T_{\overline{LF1}}$	Cols	Cuts	bLB	Nds	T_{LP}	T_{Col}	T_{Cut}	T_{Tot}
30 × 30																	
1	**10,690**	17.8	5.6	0.0	6,410	11.0	10,628	0.6	4	655	207	0.0	29	1	0	3	4
2	**10,443**	19.5	7.3	2.0	tl	12.8	10,328	1.1	4	664	214	0.0	7	1	0	4	5
3	**10,918**	20.0	6.3	0.7	tl	13.9	10,809	1.0	4	680	214	0.0	91	2	0	3	6
4	**11,365**	20.8	6.1	1.2	tl	13.9	11,317	0.4	7	860	191	0.0	29	1	0	6	7
5	**10,543**	17.8	5.6	0.6	tl	11.0	10,437	1.0	16	760	224	0.0	173	4	1	15	21
6	**10,799**	18.3	5.7	2.0	tl	11.0	10,707	0.9	2	644	187	0.0	447	6	2	2	10
7	**10,939**	15.1	4.2	0.0	2,938	9.3	10,873	0.6	21	612	182	0.0	29	1	0	20	22
8	**10,588**	16.5	4.5	0.0	2,139	9.6	10,575	0.1	7	643	212	0.0	5	0	0	6	7
9	**10,558**	20.3	7.4	2.7	tl	12.6	10,376	1.7	14	705	199	0.0	1,437	20	5	13	39
10	**10,747**	16.3	5.2	0.0	1,396	10.4	10,625	1.1	6	597	161	0.0	95	1	0	5	7
Avg		18.2	5.8	0.9	3,221	11.6		0.9	9	682	199	0.0	234	4	1	8	13
50 × 50																	
1	**15,972**	17.4	4.8	3.0	tl	12.3	15,833	0.9	57	1,316	294	0.0	1,755	46	14	54	121
2	**16,154**	17.8	6.2	4.1	tl	11.8	16,098	0.3	38	1,334	294	0.0	81	4	1	36	42
3	**15,996**	17.5	5.0	2.9	tl	8.1	15,939	0.4	29	1,339	248	0.0	143	5	2	27	35
4	**16,317**	17.8	4.4	2.8	tl	7.1	16,159	1.0	43	1,208	273	0.0	1,853	49	18	41	116
5	**16,147**	17.3	5.1	3.3	tl	9.9	16,002	0.9	50	1,418	361	0.0	2,117	73	22	48	151
6	**16,576**	16.2	5.7	4.2	tl	11.2	16,425	0.9	46	1,297	317	0.0	8,427	206	64	44	341
7	**15,854**	16.7	5.8	3.6	tl	11.8	15,754	0.6	54	1,408	291	0.0	881	24	7	51	87
8	**16,043**	17.3	5.9	3.8	tl	11.4	15,914	0.8	11	1,280	254	0.0	4,785	94	36	11	153
9	**16,326**	17.2	4.8	3.0	tl	9.4	16,229	0.6	44	1,194	281	0.0	787	24	7	42	76
10	**15,898**	16.8	5.4	3.4	tl	10.1	15,810	0.6	22	1,377	301	0.0	267	10	3	20	34
Avg		17.2	5.3	3.4		10.3		0.7	39	1,317	291	0.0	2,110	53	18	37	116
70 × 70																	
1	**21,155**	16.3	5.3	4.4	tl	9.7	21,029	0.6	270	2,351	489	0.0	6,881	399	173	257	896
2	**21,614**	14.3	4.4	3.6	tl	10.6	21,524	0.4	152	1,972	350	0.0	883	44	14	145	211
3	**21,346**	14.0	5.6	4.2	tl	10.0	21,205	0.7	92	1,954	392	0.0	15,943	679	286	87	1,154
4	**20,771**	15.9	4.3	3.1	tl	8.8	20,640	0.6	364	2,285	499	0.0	1,407	109	46	346	528
5	**21,107**	14.8	4.1	3.0	tl	9.1	21,027	0.4	139	2,033	290	0.0	671	28	12	133	179
6	**20,343**	15.6	4.3	3.2	tl	9.1	20,226	0.6	32	2,028	318	0.0	2,701	104	48	31	194
7	**21,033**	15.9	5.6	4.2	tl	10.4	20,936	0.5	216	1,942	485	0.0	889	48	21	206	288
8	**20,816**	14.7	4.1	3.2	tl	8.0	20,740	0.4	259	2,087	378	0.0	229	12	5	244	275
9	**21,123**	13.9	4.1	3.0	tl	8.1	21,010	0.5	161	1,913	369	0.0	2,409	105	43	154	320
10	**21,010**	14.4	5.2	4.1	tl	9.7	20,936	0.4	125	1,943	405	0.0	495	26	9	119	162
Avg		15.0	4.7	3.6		9.3		0.5	181	2,051	398	0.0	3,251	155	66	172	421

ii. If columns of negative reduced cost exist, add them to MP and solve MP with a generic LP solver.

c. *Stopping criteria*.

i. If at the current iteration no cuts were added at Step 2a and no columns were added at Step 2b, set $z(\overline{\mathrm{LF1}})^{\Delta} = z(\mathrm{MP})$ and $\Delta = \Delta + 1$.

ii. If $\Delta \le 6$, go to Step 2a. Otherwise, stop.

In solving LF1, notice that at Step 1, no cuts are separated so that lower bound $z(\overline{\mathrm{LF1}})^0$ corresponds to $z(\mathrm{LF1})$. At Step 2, lower bound $z(\overline{\mathrm{LF1}})^{\Delta}$, $\Delta = 1, \ldots, 6$, represents the lower bound achieved by adding the first Δ types of inequalities only. For example, $z(\overline{\mathrm{LF1}})^3$ is the lower bound achieved by adding CGD, CGU, LCGD, and FSB inequalities only. Notice also that, at the very last iteration, parameter Δ is equal to 7, so all six classes of valid inequalities are separated. We tried different orderings for separating cuts and chose the one that provided the minimum average computing time without affecting the quality of the final lower bound.

5.2. Branching Strategy

The branch-and-price algorithm, hereafter called B&P, is executed after running the bounding procedure described in §5.1 if lower bound $z(\overline{\mathrm{LF1}})$ does not correspond to an integer feasible FCTP solution. Algorithm B&P solves the integer problem obtained by adding to F1 all the valid inequalities separated by the bounding procedure.

Branching is on aggregated variables $y_{ij} = \sum_{l \in \mathcal{W}_i:\, w_j^l > 0} \xi^l$. Given a fractional solution $\boldsymbol{\xi}$, the algorithm chooses the arc $(i, j) \in A$ with the associated variable y_{ij} having a value closest to 0.6 and imposes

Table 6 **Results on** Data Set 3 **with** $B = 20$ **and** $\theta = 0.2$

		CPLEX				B&P											
Inst	UB	LF0	LBr	bLB	T_{Tot}	LF1	$z(\overline{\mathrm{LF1}})$	$\overline{\mathrm{LF1}}$	$T_{\overline{\mathrm{LF1}}}$	Cols	Cuts	bLB	Nds	T_{LP}	T_{Col}	T_{Cut}	T_{Tot}
30 × 30																	
1	**12,769**	12.3	3.2	0.0	322	6.3	12,744	0.2	3	538	152	0.0	13	0	0	2	3
2	**12,979**	17.4	6.1	0.0	9,929	10.3	12,956	0.2	3	666	213	0.0	5	1	0	2	3
3	**14,109**	14.4	5.4	1.1	tl	10.3	13,967	1.0	4	681	207	0.0	179	3	1	4	7
4	**13,271**	14.2	4.5	0.0	6,226	9.6	13,175	0.7	7	727	212	0.0	85	2	0	6	9
5	**13,756**	17.0	4.4	0.0	9,129	7.5	13,713	0.3	10	646	173	0.0	13	1	0	10	11
6	**13,540**	14.1	4.8	0.0	1,557	9.5	13,474	0.5	17	566	207	0.0	29	1	0	16	18
7	**13,547**	17.0	6.2	2.3	tl	10.9	13,429	0.9	4	729	201	0.0	403	5	1	4	11
8	**13,116**	16.6	5.7	1.6	tl	12.9	12,929	1.4	12	769	219	0.0	907	20	4	11	36
9	**13,836**	15.7	5.4	0.7	tl	9.4	13,760	0.5	6	653	188	0.0	111	2	1	6	9
10	**13,371**	13.8	3.9	0.0	1,540	9.1	13,343	0.2	5	683	207	0.0	11	1	0	5	6
Avg		15.3	5.0	0.6	4,784	9.6		0.6	7	666	198	0.0	176	3	1	7	11
50 × 50																	
1	**20,451**	12.4	4.8	2.8	tl	8.8	20,336	0.6	41	1,282	313	0.0	573	21	7	39	69
2	**20,704**	13.5	3.7	1.7	tl	7.7	20,612	0.4	69	1,329	330	0.0	255	11	3	65	82
3	**20,672**	17.5	5.8	4.1	tl	11.3	20,547	0.6	24	1,354	329	0.0	2,871	97	27	23	154
4	**20,757**	14.2	4.1	2.8	tl	8.1	20,676	0.4	91	1,396	337	0.0	123	6	1	86	97
5	**21,097**	14.2	5.3	3.3	tl	9.3	20,899	0.9	35	1,223	320	0.0	3,087	88	28	33	159
6	**20,751**	15.9	6.0	4.1	tl	11.0	20,647	0.5	18	1,306	252	0.0	403	12	3	17	32
7	**20,475**	14.3	4.7	2.9	tl	9.8	20,370	0.5	35	1,286	306	0.0	367	11	3	33	50
8	**20,927**	14.8	5.8	4.1	tl	10.3	20,768	0.8	44	1,347	382	0.0	2,273	94	25	42	169
9	**20,903**	15.1	5.8	4.0	tl	10.5	20,780	0.6	44	1,319	326	0.0	1,045	37	9	42	92
10	**20,320**	13.1	4.0	2.1	tl	7.0	20,238	0.4	45	1,161	254	0.0	97	3	1	42	48
Avg		14.5	5.0	3.2		9.4		0.6	45	1,300	315	0.0	1,109	38	11	42	95
70 × 70																	
1	27,868	13.8	4.9	4.2	tl	10.3	27,695	0.6	104	2,222	367	0.1[a]	*75,817*	*2,980*	*942*	*100*	*5,854*
2	**27,087**	12.3	4.5	3.5	tl	8.2	26,952	0.5	197	2,323	366	0.0	3,589	159	68	192	443
3	**27,547**	14.2	4.3	3.6	tl	8.3	27,414	0.5	91	1,988	395	0.0	4,361	199	79	87	390
4	**26,832**	11.4	3.6	2.7	tl	8.6	26,743	0.3	66	1,896	429	0.0	285	19	6	92	90
5	**27,685**	14.1	4.4	3.4	tl	8.4	27,562	0.4	90	2,306	398	0.0	1,847	99	32	86	228
6	**26,972**	12.7	4.0	2.9	tl	7.9	26,860	0.4	133	2,057	315	0.0	1,065	49	20	127	206
7	**27,485**	13.5	4.4	3.6	tl	6.8	27,302	0.7	137	2,021	399	0.0	14,289	676	268	133	1,171
8	**26,944**	13.8	5.1	4.1	tl	9.1	26,760	0.7	114	2,004	418	0.0	87,329	4,351	1,542	109	6,528
9	**27,769**	13.4	4.1	3.3	tl	9.7	27,646	0.4	175	2,122	326	0.0	1,351	67	22	169	267
10	**27,256**	12.7	4.3	3.5	tl	8.2	27,138	0.4	83	2,121	315	0.0	1,353	61	21	78	169
Avg		13.2	4.4	3.5		8.6		0.5	119	2,106	373	0.0	12,830	631	229	119	1,055

[a]Corresponding final best lower bound is 27,850.

the disjunction $y_{ij} = 0 \vee y_{ij} = 1$. Ties are broken by selecting the arc having the largest fixed cost f_{ij}.

The lower bound at each node of the branch-and-bound tree is computed by using column generation without adding any further valid inequalities. Nodes are explored with the best-bound-first strategy.

6. Computational Results

This section reports a detailed computational analysis of our branch-and-price algorithm, B&P, described in §5. The algorithm was implemented in C and compiled with Visual Studio 2010 64-bit. CPLEX 12.5 was used as the LP solver at Steps 1 and 2 of the bounding procedure of §5.1 and IP solver for separating SC inequalities described in §3.1.

We considered three sets of instances. The first set (Data Set 1) is publicly available at http://plato.asu.edu/ftp/lptestset/fctp (last accessed July 29, 2014), maintained by Arizona State University. Data Set 1 was used in Agarwal and Aneja (2012) and is made up of 13 instances with up to 17 origins and 64 destinations. The second data set (Data Set 2) was introduced by Agarwal and Aneja (2012) and consists of 30 instances with 15 origins and 15 destinations. In Data Set 2, quantities a_i and b_j were randomly generated in the interval [1, 20] with uniform distribution. Fixed and unit costs were generated in the interval [200, 800], but unit costs were scaled so as to maintain a predefined ratio θ between the total variable and fixed costs. The instances are grouped into three classes characterized by different values of the parameter θ; i.e., $\theta = 0.0$, $\theta = 0.2$, and $\theta = 0.5$. Instances with $\theta = 0.0$ correspond to PFCTP instances. On instances with $\theta = 0.2$ and $\theta = 0.5$, the unit costs account for roughly 20% and 50% of the total cost, respectively. The Data Set 2 instances were kindly provided to us by the author (Agarwal 2013).

Table 7 Results on Data Set 3 **with** $B = 20$ **and** $\theta = 0.5$

		CPLEX				B&P											
Inst	UB	LF0	LBr	bLB	T_{Tot}	LF1	$z(\overline{LF1})$	$\overline{LF1}$	$T_{\overline{LF1}}$	Cols	Cuts	bLB	Nds	T_{LP}	T_{Col}	T_{Cut}	T_{Tot}
30 × 30																	
1	**18,291**	12.8	4.2	0.0	4,401	7.7	18,188	0.6	6	755	218	0.0	55	1	0	6	7
2	**19,106**	15.1	4.3	0.4	tl	8.7	19,054	0.3	4	740	212	0.0	25	1	0	4	5
3	**18,034**	10.1	2.7	0.0	159	6.3	18,003	0.2	4	523	199	0.0	7	0	0	4	5
4	**17,637**	11.6	3.8	0.0	1,098	8.3	17,535	0.6	2	592	197	0.0	93	1	0	2	4
5	**18,548**	12.2	2.6	0.0	196	6.0	18,485	0.3	10	728	216	0.0	21	1	0	10	11
6	**17,781**	13.2	3.8	0.0	1,459	9.6	17,640	0.8	14	698	247	0.0	185	5	1	14	19
7	**17,969**	15.4	5.5	1.6	tl	10.1	17,853	0.6	3	758	276	0.0	239	5	1	2	9
8	**18,198**	12.1	3.9	0.0	3,484	8.5	18,064	0.7	7	691	209	0.0	129	2	1	7	10
9	**17,744**	12.0	3.0	0.0	740	7.9	17,699	0.3	6	710	251	0.0	13	1	0	6	7
10	**18,760**	14.3	4.3	0.0	3,265	7.5	18,627	0.7	4	696	173	0.0	137	4	1	4	8
Avg		12.9	3.8	0.2	1,850	8.1		0.5	6	689	220	0.0	90	2	0	6	9
50 × 50																	
1	**27,147**	10.7	3.9	2.2	tl	7.1	27,033	0.4	42	1,301	289	0.0	1,703	47	15	40	107
2	**27,574**	12.1	4.1	2.6	tl	7.7	27,394	0.7	47	1,418	332	0.0	3,377	118	33	46	209
3	**27,668**	13.0	4.2	2.3	tl	6.8	27,495	0.6	42	1,422	413	0.0	2,579	119	31	40	200
4	**27,603**	12.7	4.1	2.6	tl	6.2	27,405	0.7	40	1,194	360	0.0	27,701	788	274	39	1,207
5	**27,085**	9.7	3.1	1.3	tl	5.8	26,890	0.7	35	1,154	250	0.0	36,053	784	242	34	1,170
6	**28,256**	10.9	2.9	1.7	tl	5.8	28,097	0.6	27	1,379	310	0.0	2,321	66	21	26	121
7	**27,621**	10.1	2.9	1.6	tl	6.3	27,547	0.3	56	1,241	359	0.0	291	12	4	55	72
8	**27,493**	11.1	3.2	1.6	tl	6.1	27,435	0.2	29	1,154	264	0.0	25	2	0	28	30
9	**27,445**	11.6	3.7	2.4	tl	8.0	27,359	0.3	38	1,370	349	0.0	119	7	2	37	45
10	**27,888**	11.3	4.4	2.8	tl	6.8	27,732	0.6	44	1,163	361	0.0	719	25	7	43	76
Avg		11.3	3.7	2.1		6.7		0.5	40	1,280	329	0.0	7,489	197	63	39	324
70 × 70																	
1	**36,584**	9.1	2.9	2.3	tl	5.4	36,431	0.4	100	1,505	399	0.0	5,679	248	94	98	477
2	**37,551**	11.2	3.9	3.1	tl	8.1	37,378	0.5	176	2,206	474	0.0	25,323	1,494	503	174	2,349
3	**36,982**	10.0	3.3	2.5	tl	5.6	36,819	0.4	257	2,040	445	0.0	3,875	214	87	254	585
4	**36,669**	11.7	4.7	3.7	tl	7.9	36,468	0.5	71	2,174	521	0.0	42,797	3,285	994	67	4,657
5	**37,090**	12.0	4.6	3.9	tl	6.4	36,900	0.5	56	2,189	486	0.0	18,725	1,382	394	50	1,946
6	**37,152**	9.2	3.0	2.2	tl	6.4	37,004	0.4	114	1,777	343	0.0	3,635	154	56	112	342
7	**36,715**	10.1	3.4	2.6	tl	7.1	36,541	0.5	187	2,001	437	0.0	19,177	993	354	184	1,660
8	**37,007**	10.3	3.6	2.9	tl	6.6	36,902	0.3	180	2,190	440	0.0	249	18	5	176	201
9	**37,679**	10.3	3.6	2.8	tl	6.9	37,490	0.5	86	1,856	384	0.0	23,501	956	359	83	1,546
10	**36,588**	9.7	2.7	1.9	tl	6.9	36,512	0.2	172	2,137	415	0.0	95	10	3	169	182
Avg		10.4	3.6	2.8		6.7		0.4	140	2,008	434	0.0	14,306	875	285	137	1,395

The third data set (Data Set 3) is introduced in this paper and contains 180 instances with up to 70 origins and 70 destinations. These instances are similar to the Data Set 2 instances with a_i and b_j values ranging in the interval $[1, B]$, $B \in \{20, 50\}$; $\theta \in \{0.0, 0.2, 0.5\}$; and fixed costs randomly generated in the range $[200, 800]$. Variable costs were generated proportional to fixed costs as $c_{ij} = \lfloor(\theta f_{ij}(m+n-1))/(\sum_{i\in S} a_i)\rfloor$. Instances obtained this way turned out to be computationally more challenging for B&P than instances generated by Agarwal and Aneja (2012), where the variable costs c_{ij} are generated independently of fixed costs f_{ij} and are not proportional to them. The Data Set 3 instances are grouped into six sets of 30 instances characterized by different values of parameters B and θ. Data Set 3 contains instances with 20, 30, and 40 origins and destinations for values of $B = 50$ and 30, 50, and 70 origins and destinations for values of $B = 20$.

All instances of the three data sets fulfill the assumption made in §1 that $\sum_{i\in S} a_i = \sum_{j\in T} b_j$ (i.e., instances are already *balanced*) and dense (i.e., $|A| = mn$). If the problem is not balanced, an artificial source or an artificial sink must be added to balance the problem. We have noticed that the computational behavior of the proposed exact algorithm on balanced and unbalanced instances is similar. We also tested our exact algorithm on some sparse instances (with 20% and 40% of density, as done in Barr et al. 1981), and we noticed that sparse instances are, on average, easier than dense ones mainly because there are fewer variables to branch on and to consider in the pricing phase. However, the size of solvable instances does not increase substantially.

Table 8 **Results on** Data Set 3 **with** $B = 50$ **and** $\theta = 0.0$

		CPLEX				B&P											
Inst	UB	LF0	LBr	bLB	T_{Tot}	LF1	$z(\overline{\text{LF1}})$	$\overline{\text{LF1}}$	$T_{\overline{\text{LF1}}}$	Cols	Cuts	bLB	Nds	T_{LP}	T_{Col}	T_{Cut}	T_{Tot}
20×20																	
1	**7,710**	23.8	7.4	0.0	1,646	15.2	7,680	0.4	3	611	292	0.0	9	1	0	2	3
2	**8,039**	28.7	8.6	0.0	585	13.1	7,919	1.5	2	537	237	0.0	109	3	1	2	5
3	**8,483**	23.4	9.7	0.0	7,004	16.8	8,190	3.5	2	545	240	0.0	5,325	88	27	2	121
4	**8,223**	28.1	12.7	4.3	tl	21.4	7,979	3.0	2	619	272	0.0	3,423	73	16	1	94
5	**8,257**	28.1	10.5	1.2	tl	16.7	8,100	1.9	2	722	349	0.0	161	6	1	2	9
6	**8,684**	26.9	9.4	2.0	tl	14.3	8,439	2.8	2	741	294	0.0	1,721	49	11	1	63
7	**8,576**	25.1	10.3	0.0	9,040	18.4	8,485	1.1	2	679	295	0.0	37	2	0	1	3
8	**8,329**	27.0	11.3	2.7	tl	17.8	8,126	2.4	2	676	278	0.0	2,141	47	14	1	64
9	**7,863**	25.8	11.1	0.7	tl	19.2	7,770	1.2	2	707	311	0.0	59	4	1	1	5
10	**8,548**	26.9	10.9	2.5	tl	16.2	8,411	1.6	3	783	333	0.0	77	4	1	2	6
Avg		26.4	10.2	1.3	4,569	16.9		1.9	2	662	290	0.0	1,306	28	7	1	37
30×30																	
1	**11,513**	23.3	8.8	5.2	tl	13.7	11,369	1.3	8	1,018	362	0.0	517	24	7	6	36
2	**11,649**	20.3	7.8	4.0	tl	12.6	11,412	2.0	7	995	346	0.0	6,627	229	82	5	329
3	**11,144**	23.2	9.5	5.5	tl	14.9	10,968	1.6	9	1,141	396	0.0	3,907	172	54	8	242
4	**11,303**	24.8	10.7	6.5	tl	15.9	11,122	1.6	21	1,122	460	0.0	3,525	193	60	18	280
5	**11,543**	23.0	10.2	5.8	tl	16.3	11,325	1.9	9	1,255	433	0.0	3,073	221	55	6	290
6	**11,642**	27.1	10.6	6.6	tl	22.5	11,365	2.4	12	1,357	477	0.0	20,877	1,023	365	9	1,449
7	**11,400**	22.4	9.1	5.7	tl	15.9	11,235	1.4	16	1,161	425	0.0	475	31	8	14	54
8	**10,817**	24.8	11.1	6.1	tl	20.5	10,671	1.3	10	1,163	483	0.0	1,509	107	26	8	145
9	**11,911**	21.8	8.6	4.5	tl	12.8	11,734	1.5	9	992	344	0.0	2,311	81	29	7	121
10	**11,259**	23.1	9.8	5.2	tl	16.6	10,985	2.4	8	940	358	0.0	12,151	409	146	7	585
Avg		23.4	9.6	5.5		16.2		1.7	11	1,114	408	0.0	5,497	249	83	9	353
40×40																	
1	**14,285**	19.7	8.6	6.2	tl	13.0	14,042	1.7	27	1,284	443	0.0	49,145	2,655	967	23	3,798
2	**14,510**	23.7	9.4	7.1	tl	16.5	14,318	1.3	36	1,443	468	0.0	8,001	637	203	31	899
3	**14,409**	24.8	9.9	7.1	tl	18.3	14,145	1.8	17	1,622	603	0.0	50,723	5,028	1,392	12	6,638
4	14,651	25.7	11.6	8.9	tl	19.3	14,287	2.5	21	1,651	596	0.8[a]	*78,103*	*6,566*	*1,901*	*19*	*8,640*
5	**14,732**	22.1	10.3	7.9	tl	16.0	14,438	2.0	22	1,673	556	0.0	85,417	6,793	2,287	17	9,456
6	14,348	19.4	8.8	6.3	tl	12.8	14,052	2.1	46	1,277	468	0.1[b]	*88,617*	*5,923*	*1,215*	*43*	*7,289*
7	14,785	22.3	9.3	7.1	tl	15.7	14,509	1.9	26	1,541	544	0.0[c]	*100,000*	*6,539*	*2,522*	*25*	*9,341*
8	**13,615**	20.7	7.8	4.9	tl	14.9	13,354	1.9	36	1,339	465	0.0	62,785	3,086	1,160	32	4,510
9	14,408	24.5	12.0	8.9	tl	17.2	14,047	2.5	24	1,618	523	0.9[d]	*75,701*	*5,672*	*2,107*	*21*	*8,103*
10	14,169	21.1	8.4	6.3	tl	14.2	13,832	2.4	16	1,501	545	0.5[e]	*80,345*	*5,635*	*2,523*	*14*	*8,411*
Avg		22.4	9.6	7.1		15.8		2.0	27	1,495	521	0.2	51,214	3,640	1,202	23	5,060

[a]Corresponding final best lower bound is 14,534. [b]Corresponding final best lower bound is 14,328. [c]Corresponding final best lower bound is 14,783. [d]Corresponding final best lower bound is 14,275. [e]Corresponding final best lower bound is 14,098.

Therefore, we focus our computational experiments on balanced and dense instances only.

Algorithm B&P is compared with the branch-and-cut of Agarwal and Aneja (2012) (hereafter, AA12) on Data Set 1 and Data Set 2 and with CPLEX 12.5 solving formulation F0. All runs were conducted on a single core of an Intel Xeon X7350 @ 2.93 GHz server with 16 GB of RAM running under Windows Server 2008. Algorithm AA12 is based on CPLEX 11.2 and was tested on an Intel Pentium 1.8 GHz Dual Core CPU. According to the Standard Performance Evaluation Corporation (http://www.spec.org/benchmarks.html, last accessed July 29, 2014), our machine is two times faster than that used in Agarwal and Aneja. All computing times reported in this section are in seconds.

Tables 1–4 compare the performance of B&P, AA12, and CPLEX on Data Set 1 and Data Set 2. These tables report the instance name ("Inst") and the optimal solution cost ("Opt"). For CPLEX, the tables report the percentage gap ("LF0") of lower bound $z(\mathrm{LF0})$, the percentage gap ("LBr") of the lower bound achieved at the root node with all its cuts, and the total computing time ("T_{Tot}") for solving the instance. For algorithm AA12, the tables report the percentage gap ("LBr") of the lower bound achieved at the root node (when available) and the computing time ("T_{Tot}") for solving the instance to optimality. For B&P, the tables report the percentage gap ("LF1") of lower bound $z(\mathrm{LF1})$; the lower bound ("$z(\overline{\mathrm{LF1}})$"), its percentage gap ("$\overline{\mathrm{LF1}}$"), and the corresponding computing time ("$T_{\overline{\mathrm{LF1}}}$"); the number of columns ("Cols") and cuts ("Cuts") generated for solving $\overline{\mathrm{LF1}}$; the number of nodes ("Nds"); and the total computing time ("T_{Tot}"). Averages are reported in the last lines of the tables.

Table 9 **Results on** Data Set 3 **with** $B = 50$ **and** $\theta = 0.2$

		CPLEX				B&P											
Inst	UB	LF0	LBr	bLB	T_{Tot}	LF1	$z(\overline{\mathrm{LF1}})$	$\overline{\mathrm{LF1}}$	$T_{\overline{\mathrm{LF1}}}$	Cols	Cuts	bLB	Nds	T_{LP}	T_{Col}	T_{Cut}	T_{Tot}
20 × 20																	
1	**10,286**	22.0	7.7	0.0	5,679	12.3	10,075	2.1	3	618	298	0.0	1,361	33	8	2	45
2	**10,610**	21.2	7.2	0.0	791	12.5	10,485	1.2	3	677	302	0.0	59	3	0	2	5
3	**10,746**	20.8	7.5	1.9	tl	15.1	10,557	1.8	2	663	295	0.0	597	14	3	1	19
4	**10,769**	20.3	5.9	0.0	355	12.5	10,707	0.6	3	561	273	0.0	13	1	0	2	3
5	**10,521**	20.1	8.0	0.3	tl	13.1	10,317	1.9	2	570	209	0.0	753	14	4	1	19
6	**9,802**	20.0	7.7	0.0	936	13.3	9,655	1.5	7	714	344	0.0	71	5	1	6	11
7	**9,337**	20.5	6.0	0.0	154	15.8	9,161	1.9	3	610	238	0.0	479	9	2	2	14
8	**10,562**	20.6	7.4	0.0	7,425	12.7	10,374	1.8	3	692	251	0.0	713	19	4	2	25
9	**10,411**	21.0	7.7	0.9	tl	15.9	10,237	1.7	3	697	299	0.0	371	10	3	1	15
10	**9,947**	21.4	7.3	0.0	543	14.9	9,882	0.7	3	692	295	0.0	23	2	0	2	4
Avg		20.8	7.2	0.3	2,269	13.8		1.5	3	649	280	0.0	444	11	3	2	16
30 × 30																	
1	**13,969**	23.9	11.2	6.7	tl	17.9	13,682	2.1	5	1,258	475	0.0	39,251	2,203	580	2	2,886
2	**14,310**	20.9	8.8	5.1	tl	13.3	14,119	1.3	9	1,101	415	0.0	2,369	145	31	6	188
3	**13,707**	21.1	8.4	4.8	tl	11.8	13,477	1.7	14	1,299	475	0.0	14,177	923	279	10	1,250
4	**14,482**	24.5	10.7	6.9	tl	16.2	14,197	2.0	8	1,302	518	0.0	29,011	1,910	486	5	2,487
5	**13,888**	19.9	8.8	4.8	tl	12.1	13,644	1.8	8	1,103	367	0.0	9,511	380	113	6	519
6	**13,822**	21.9	8.9	4.1	tl	15.2	13,613	1.5	10	1,243	464	0.0	1,299	84	23	6	116
7	**14,551**	21.5	10.4	6.6	tl	16.0	14,269	1.9	11	1,507	562	0.0	8,219	617	160	6	808
8	**14,039**	23.1	10.1	6.2	tl	15.9	13,793	1.8	13	1,187	446	0.0	5,595	352	95	9	470
9	**14,079**	21.5	8.9	4.5	tl	16.7	13,890	1.3	7	1,164	424	0.0	1,171	73	16	3	94
10	**14,537**	21.4	7.9	4.2	tl	14.5	14,310	1.6	7	1,123	515	0.0	2,035	132	32	4	173
Avg		22.0	9.4	5.4		15.0		1.7	9	1,229	466	0.0	11,264	682	182	6	899
40 × 40																	
1	17,931	19.2	6.9	5.0	tl	9.7	17,653	1.5	19	1,482	577	0.3[a]	*78,465*	*6,612*	*2,335*	*11*	*9,256*
2	18,249	20.3	8.3	6.1	tl	13.7	17,860	2.1	20	1,691	592	0.8[b]	*79,711*	*7,021*	*2,158*	*14*	*9,510*
3	17,738	21.8	8.9	6.2	tl	12.8	17,375	2.0	18	1,587	562	0.4[c]	*73,490*	*6,482*	*2,097*	*12*	*8,888*
4	**17,911**	18.9	8.2	5.6	tl	12.5	17,691	1.2	28	1,621	492	0.0	3,041	265	66	22	364
5	**17,300**	20.6	9.1	6.7	tl	12.3	17,064	1.4	27	1,749	610	0.0	11,469	1,121	379	20	1,568
6	**17,815**	18.6	5.9	3.7	tl	14.3	17,567	1.4	33	1,449	519	0.0	27,391	2,231	708	28	3,065
7	17,727	21.4	9.2	7.0	tl	15.8	17,349	2.1	27	1,644	602	0.5[d]	*73,827*	*6,799*	*2,055*	*19*	*9,178*
8	**18,024**	18.4	7.6	5.4	tl	12.8	17,802	1.2	17	1,458	534	0.0	18,439	1,340	436	11	1,853
9	**18,428**	18.9	8.0	6.0	tl	13.4	18,175	1.4	20	1,766	574	0.0	12,297	1,136	289	15	1,489
10	**17,900**	18.3	7.0	4.9	tl	11.1	17,654	1.4	27	1,356	409	0.0	31,871	1,535	527	22	2,176
Avg		19.6	8.0	5.6		12.8		1.6	24	1,580	547	0.2	18,443	1,271	401	20	1,793

[a]Corresponding final best lower bound is 17,885. [b]Corresponding final best lower bound is 18,108. [c]Corresponding final best lower bound is 17,664. [d]Corresponding final best lower bound is 17,643.

Tables 5–10 compare B&P with CPLEX on Data Set 3. In addition to the columns of Tables 1–4, these latter tables also report, for both algorithms, the percentage gap left ("bLB") by the best lower bound computed at termination. Column "UB" indicates the best-known upper bound, obtained by either the B&P or the heuristic described in Buson et al. (2014) (the upper bound is in bold whenever it is the optimal solution cost). For B&P, columns "T_{LP}," "T_{Col}," and "T_{Cut}" report the time spent for solving the master problem, for generating columns, and for separating cuts, respectively. We do not report a further breakdown of the computing time spent for separating each of the six classes of valid inequalities considered because roughly 95% of the total computing time for separating cuts is spent for separating SC inequalities.

Agarwal and Aneja (2012) set a time limit of 10 minutes on Data Set 2 instances, whereas a time limit of three hours was set on CPLEX solving Data Set 3 instances. We report "tl" under column "T_{Tot}" whenever AA12 or CPLEX reaches its time limit. For B&P, values in italics in columns "Nds," "T_{LP}," "T_{Col}," "T_{Cut}," and "T_{Tot}" indicate a premature termination as a result of memory overflow.

Tables 1–4 clearly indicate that lower bound $z(\overline{\mathrm{LF1}})$ strongly dominates the lower bounds achieved at the root node by both CPLEX and AA12 and that B&P outperforms both CPLEX and AA12. Notice that B&P solves, in about a second, each of the three instances unsolved by AA12 within the imposed time limit of 600 seconds.

Tables 5–10 indicate that B&P outperforms CPLEX on Data Set 3 in terms of both the number and size of

Table 10 **Results on** Data Set 3 **with** $B=50$ **and** $\theta=0.5$

		CPLEX				B&P											
Inst	UB	LF0	LBr	bLB	T_{Tot}	LF1	$z(\overline{\mathrm{LF1}})$	$\overline{\mathrm{LF1}}$	$T_{\overline{\mathrm{LF1}}}$	Cols	Cuts	bLB	Nds	T_{LP}	T_{Col}	T_{Cut}	T_{Tot}
20 × 20																	
1	**13,335**	14.5	3.8	0.0	62	8.7	13,282	0.4	4	796	399	0.0	13	2	0	3	5
2	**12,947**	15.0	3.6	0.0	12	9.9	12,933	0.1	3	572	268	0.0	3	1	0	3	4
3	**13,936**	18.2	6.6	0.0	1,478	8.6	13,759	1.3	3	646	321	0.0	463	17	4	2	23
4	**13,191**	14.5	3.9	0.0	39	9.4	13,045	1.1	3	517	272	0.0	67	2	0	3	5
5	**13,759**	15.5	6.4	0.0	2,703	9.9	13,550	1.5	2	582	242	0.0	583	11	3	1	15
6	**13,774**	16.2	4.0	0.0	16	10.9	13,774	0.0	1	731	283	0.0	0	1	0	0	1
7	**13,184**	16.6	4.8	0.0	117	7.6	13,053	1.0	3	619	256	0.0	55	2	0	2	5
8	**13,134**	15.5	5.4	0.0	855	10.5	12,914	1.7	3	673	263	0.0	413	10	3	2	14
9	**12,811**	16.4	7.7	0.0	6,938	12.9	12,561	2.0	3	774	341	0.0	1,213	38	8	2	50
10	**13,218**	16.9	5.2	0.0	116	11.4	13,144	0.6	7	962	416	0.0	7	3	0	5	9
Avg		15.9	5.1	0.0	1,234	10.0		1.0	3	687	306	0.0	282	9	2	2	13
30 × 30																	
1	**18,298**	16.5	6.0	2.4	tl	10.9	18,084	1.2	9	1,072	431	0.0	2,187	106	31	6	148
2	**18,785**	14.6	5.9	2.6	tl	10.6	18,545	1.3	8	918	354	0.0	3,043	112	29	5	153
3	**18,844**	15.1	5.3	2.4	tl	9.0	18,650	1.0	6	982	353	0.0	2,131	84	25	3	115
4	**18,163**	18.1	7.0	3.7	tl	12.6	17,953	1.2	9	1,253	523	0.0	6,659	595	150	4	768
5	**18,628**	15.5	6.7	3.1	tl	11.5	18,355	1.5	12	1,186	486	0.0	8,349	499	134	9	665
6	**18,922**	15.9	6.0	2.4	tl	10.4	18,680	1.3	9	1,138	489	0.0	2,651	173	48	6	234
7	**18,353**	17.5	7.3	3.6	tl	11.2	18,109	1.3	5	976	373	0.0	7,637	354	102	2	474
8	**18,950**	15.8	6.3	3.1	tl	10.6	18,688	1.4	8	1,104	432	0.0	6,707	309	91	5	420
9	**18,036**	17.4	7.6	4.2	tl	11.1	17,832	1.1	9	1,049	416	0.0	3,037	160	41	6	213
10	**19,068**	14.6	6.2	2.2	tl	12.1	18,776	1.5	6	990	384	0.0	14,361	489	151	4	675
Avg		16.1	6.4	3.0		11.0		1.3	8	1,067	424	0.0	5,676	288	80	5	387
40 × 40																	
1	23,480	17.1	6.6	4.3	tl	11.1	23,088	1.7	25	1,552	634	0.3[a]	*73,335*	*8,056*	*2,293*	*19*	*10,704*
2	**23,282**	14.0	5.5	3.5	tl	9.5	23,070	0.9	19	1,441	522	0.0	5,979	458	135	14	629
3	**23,927**	12.6	4.7	2.8	tl	8.6	23,764	0.7	36	1,319	466	0.0	1,041	78	22	32	135
4	24,858	17.7	7.7	5.7	tl	10.8	24,480	1.5	16	1,525	599	0.4[b]	*83,855*	*7,589*	*2,094*	*8*	*10,027*
5	23,486	13.4	5.3	3.8	tl	10.2	23,118	1.6	35	1,437	535	0.3[c]	*84,591*	*6,604*	*2,017*	*31*	*9,007*
6	23,781	15.4	6.8	4.9	tl	8.9	23,398	1.6	18	1,508	574	0.5[d]	*76,138*	*6,954*	*2,419*	*12*	*9,678*
7	**23,914**	15.6	7.7	5.7	tl	11.2	23,684	1.0	30	1,773	613	0.0	34,591	3,256	1,032	23	4,460
8	23,804	14.6	6.4	4.4	tl	10.0	23,476	1.4	25	1,450	508	0.1[e]	*83,270*	*5,882*	*2,010*	*19*	*8,214*
9	23,989	16.9	7.1	5.0	tl	11.1	23,650	1.4	20	1,643	699	0.2[f]	*73,299*	*8,227*	*2,306*	*13*	*10,733*
10	23,950	15.0	6.3	4.4	tl	10.0	23,551	1.7	29	1,432	530	0.2[g]	*85,000*	*6,031*	*1,909*	*24*	*8,296*
Avg		15.2	6.4	4.5		10.2		1.3	25	1,508	568	0.2	13,870	1,264	396	23	1,741

[a]Corresponding final best lower bound is 23,413. [b]Corresponding final best lower bound is 24,752. [c]Corresponding final best lower bound is 23,423. [d]Corresponding final best lower bound is 23,659. [e]Corresponding final best lower bound is 23,791. [f]Corresponding final best lower bound is 23,937. [g]Corresponding final best lower bound is 23,893.

Table 11 Data Set 3: **Average Number of Cuts of Each Class Added and Columns Generated in the Search Tree**

			Cuts							Columns		
B	θ	n	CGD-CGU	LCGD	FSB	GUBC	CPL	SC	All	Root	All	Node
20	0.0	30	86	65	33	7	1	8	199	682	23,695	95
		50	139	90	42	7	2	12	291	1,317	334,445	174
		70	192	96	77	9	3	22	398	2,051	797,245	260
20	0.2	30	87	66	29	7	1	9	198	666	22,334	103
		50	143	96	53	10	1	12	315	1,300	209,754	181
		70	200	100	54	7	1	11	373	2,106	44,36,324	263
20	0.5	30	84	77	38	12	1	7	220	689	10,782	101
		50	136	116	50	13	2	12	329	1,280	1,232,773	162
		70	197	133	73	15	1	16	434	2,008	3,652,300	238
50	0.0	20	110	105	65	5	1	5	290	662	126,793	99
		30	172	136	89	4	2	5	408	1,114	769,318	155
		40	222	174	109	7	2	7	521	1,495	14,331,386	203
50	0.2	20	109	104	56	7	1	4	280	649	39,633	95
		30	184	168	101	8	2	3	466	1,229	1,818,438	151
		40	227	177	127	10	2	5	547	1,580	9,877,285	218
50	0.5	20	101	107	66	21	3	8	306	687	29,407	89
		30	160	153	92	13	2	5	424	1,067	846,571	152
		40	217	189	137	12	4	9	568	1,508	14,700,302	226

problems solved to optimality and the total computing time. Whereas B&P is able to solve 29 out of 30 instances with 70 origins and 70 destinations with $B = 20$, CPLEX can only solve 18 out of 30 instances with 30 origins and 30 destinations but fails in solving all other instances with 50 and 70 origins and destinations. When B is increased to 50, instances become more difficult for both CPLEX and B&P. CPLEX is able to solve 21 out of 30 instances with 20 origins and 20 destinations, whereas B&P can solve all instances with 20 or 30 origins and destinations but can solve only 14 of the 30 instances with 40 origins and 40 destinations.

The better performance of B&P is mainly due to the quality of the achieved lower bound $z(\overline{\mathrm{LF1}})$, which is far superior to lower bound LBr of CPLEX. At the same time, the performance of B&P slightly deteriorates when $B = 50$ because the average lower bound provided by $\overline{\mathrm{LF1}}$ is lower, and thus, the number of nodes and the total computing time increase. No significant conclusions can be drawn by analyzing the performance of the B&P for different values of parameter θ.

For each set of 10 instances identified by the values of B, θ, and n of Data Set 3, Table 11 reports the average number of cuts separated for each class ("CGD-CGU," "LCGD," "FSB," "GUBC," "CPL," and "SC") and in total ("All") under the heading "Cuts." The last three columns of Table 11 report the average number of columns generated at the root node ("Root"), in the entire search tree ("All"), and on average at each node ("Node") under the heading "Columns."

Table 12 shows the gaps left, on instances of Data Set 1, when adding only a class of valid inequalities

Table 12 **Gap Left on** Data Set 1 **with Individual Classes of Cuts**

	CPLEX		B&P							
Inst	LF0	LBr	LF1	CGD-CGU	LCGD	FSB	GUBC	CPL	SC	LBr
4×64	0.75	0.15	0.20	0.05	0.19	0.05	0.05	0.20	0.04	0.00
8×32	5.89	1.37	3.03	0.08	1.92	0.00	0.51	3.03	1.18	0.00
10×10a	16.41	0.53	10.01	3.80	4.07	2.74	0.60	9.74	1.60	0.00
10×10b	14.94	2.12	8.36	0.94	2.41	0.78	2.44	8.36	2.51	0.00
10×10c	13.86	4.00	8.86	1.81	6.06	2.72	3.90	8.86	5.74	0.81
10×12	10.57	1.25	5.56	0.37	0.15	1.58	0.48	5.27	0.99	0.00
10×26	9.65	3.98	4.71	1.78	2.74	1.62	2.04	4.71	3.28	0.63
12×12	20.25	6.77	12.48	4.63	4.19	3.58	3.32	12.48	6.59	1.35
12×21	13.81	5.68	8.52	1.61	3.82	2.40	3.63	8.19	5.38	1.09
13×13	17.22	6.40	12.27	5.20	6.06	6.21	3.91	11.96	8.46	1.38
14×18	18.72	9.00	11.85	3.96	4.18	4.47	5.25	11.85	8.73	1.40
16×16	18.47	6.85	11.85	6.23	6.15	5.60	6.09	11.56	7.35	1.83
17×17	11.43	1.75	6.92	2.26	0.87	2.18	1.24	6.92	2.77	0.00
Avg	13.23	3.83	8.05	2.52	3.29	2.61	2.57	7.93	4.20	0.65

Table 13 Number of Cuts Added of Each Class and Gaps Left After Adding Each Class of Cuts on Data Set 1

		CGD-CGU		LCGD		FSB		GUBC		CPL		SC	
Inst	LF1	#	Gap	#	Gap	#	Gap	#	Gap	#	Gap	#	Gap
4 × 64	0.20	15	0.05	2	0.05	33	0.05	4	0.05	0	0.05	2	0.00
8 × 32	3.03	79	0.08	8	0.00	0	0.00	0	0.00	0	0.00	0	0.00
10 × 10a	10.01	21	3.80	19	1.07	3	0.80	1	0.00	0	0.00	0	0.00
10 × 10b	8.36	22	0.94	3	0.00	0	0.00	0	0.00	0	0.00	0	0.00
10 × 10c	8.86	19	1.81	15	1.63	18	1.63	1	1.63	0	1.63	9	0.81
10 × 12	5.56	21	0.37	2	0.00	0	0.00	0	0.00	0	0.00	0	0.00
10 × 26	4.71	92	1.78	93	0.96	45	0.89	11	0.66	0	0.66	3	0.63
12 × 12	12.48	41	4.63	40	1.96	32	1.88	17	1.62	1	1.44	9	1.35
12 × 21	8.52	78	1.61	34	1.15	10	1.15	4	1.12	0	1.12	2	1.09
13 × 13	12.27	38	5.20	52	2.43	18	1.97	10	1.78	6	1.54	7	1.38
14 × 18	11.85	76	3.96	76	1.75	26	1.48	6	1.43	1	1.40	2	1.40
16 × 16	11.85	61	6.23	60	4.08	33	3.17	29	2.72	4	2.64	13	1.83
17 × 17	6.92	73	2.26	101	0.29	37	0.15	16	0.00	0	0.00	1	0.00
Avg	8.05	49	2.52	39	1.18	20	1.01	8	0.85	1	0.81	4	0.65

and compares them with the lower bounds of CPLEX and B&P reported in Table 1. The table indicates that, by using CGD-CGU or FSB or GUBC inequalities alone, the average lower bound is less than 3% from optimality, whereas CPL inequalities alone are the least effective cuts. Moreover, the lower bounds achieved with CGD-CGU, LCGD, FSB, and GUBC inequalities alone are better than those achieved by CPLEX at the root node.

Table 13 shows the number of cuts ("#") of each class of valid inequalities and the gap left ("Gap") after adding them on Data Set 1 instances while executing the bounding procedure described in §5.1. Even though inequalities LCGD alone provide worse lower bounds than those yielded with CGD-CGU (see Table 12), when LCGD inequalities are added along with CGD-CGU, the lower bounds considerably increase. Moreover, even though CPL inequalities are the least effective class of cuts when considered alone (see Table 12), they slightly increase the lower bounds when separated along with the other classes of cuts.

7. Conclusions

In this paper, we have described a new exact branch-and price algorithm for solving both the FCTP and the PFCTP based on a new integer programming formulation involving an exponential number of binary variables. Each variable represents a feasible supply pattern from sources to sinks, and the new formulation requires choosing a supply pattern for each source in order to satisfy all sink demands. We have shown that the lower bound provided by the linear relaxation of the new formulation is stronger than the one achieved by the linear relaxation of the standard mixed integer formulation. We have investigated several classes of valid inequalities to improve the lower bound. Such inequalities are either new ones introduced in this paper or adaptations to the new formulation of valid inequalities proposed in the literature. We have described a column-and-cut generation method to compute a valid lower bound that is used by the exact branch-and-price algorithm to solve the FCTP. Extensive computational results on both instances from the literature and on new randomly generated instances indicate that the proposed method outperforms both the exact method recently proposed by Agarwal and Aneja (2012) and the latest version of CPLEX. The proposed algorithm is able to solve both FCTP and PFCTP instances involving up to 70 sources and 70 sinks.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/mnsc.2014.1947.

Acknowledgments

The authors gratefully acknowledge the two anonymous referees for their valuable comments that contributed to improve the quality of the paper.

References

Adlakha V, Kowalski K (2003) A simple heuristic for solving small fixed-charge transportation problems. *Omega* 31(3):205–211.

Agarwal Y (2013) Private communication by email. February 6.

Agarwal Y, Aneja Y (2012) Fixed-charge transportation problem: Facets of the projection polyhedron. *Oper. Res.* 60(3):638–654.

Barr RS, Glover F, Klingman D (1981) A new optimization method for large scale fixed charge transportation problems. *Oper. Res.* 29(3):448–463.

Buson E, Roberti R, Toth P (2014) A reduced-cost iterated local search heuristic for the fixed-charge transportation problem. *Oper. Res.* 62(5):1095–1106.

Cabot AV, Erenguc SS (1984) Some branch and bound procedures for fixed cost transportation problems. *Naval Res. Logist. Quart.* 31(1):145–154.

Cabot AV, Erenguc SS (1986) Improved penalties for fixed cost linear programs using Lagrangean relaxation. *Management Sci.* 32(1):856–869.

Driebeck NJ (1966) An algorithm for the solution of mixed integer programming problems. *Management Sci.* 12(7):576–587.

Fisk J, McKeown PG (1979) The pure fixed charge transportation problem. *Naval Res. Logist. Quart.* 26(4):631–641.

Göthe-Lundgren M, Larsson T (1994) A set covering reformulation of the pure fixed charge transportation problem. *Discrete Appl. Math.* 48(3):245–259.

Gray P (1971) Exact solution of the fixed-charge transportation problem. *Oper. Res.* 19(6):1529–1538.

Gu Z, Nemhauser GL, Savelsbergh MWP (1998) Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS J. Comput.* 10(4):427–437.

Gu Z, Nemhauser GL, Savelsbergh MWP (1999) Lifted cover inequalities for 0-1 integer programs: Complexity. *INFORMS J. Comput.* 11(1):117–123.

Hirsch WM, Dantzig GB (1954) Notes on linear programming: Part XIX—The fixed charge problem. RAND Research Memorandum 1383, RAND Corporation, Santa Monica, CA.

Hirsch WM, Dantzig GB (1968) The fixed charge problem. *Naval Res. Logist. Quart.* 15(3):413–424.

Hultberg TH, Cardoso DM (1997) The teacher assignment problem: A special case of the fixed charge transportation problem. *Eur. J. Oper. Res.* 101(3):463–473.

Kennington J, Unger E (1976) A new branch-and-bound algorithm for the fixed-charge transportation problem. *Management Sci.* 22(10):1116–1126.

Lamar BW, Wallace CA (1997) Revised-modified penalties for fixed charge transportation problems. *Management Sci.* 43(10): 1431–1436.

Murty KG (1968) Solving the fixed charge problem by ranking the extreme points. *Oper. Res.* 16(2):268–279.

Nemhauser GL, Wolsey LA (1988) *Integer and Combinatorial Optimization* (Wiley-Interscience, New York).

Ortega F, Wolsey LA (2003) A branch-and-cut algorithm for the single commodity, uncapacitated, fixed-charge network flow problem. *Networks* 41(3):143–158.

Palekar US, Karwan MH, Zionts S (1990) A branch-and-bound method for the fixed charge transportation problem. *Management Sci.* 36(9):1092–1105.

Rardin RL, Wolsey LA (1993) Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *Eur. J. Oper. Res.* 71(1): 95–109.

Stroup JW (1967) Allocation of launch vehicles to space missions: A fixed-cost transportation problem. *Oper. Res.* 15(6): 1157–1163.

Tomlin JA (1971) An improved branch-and-bound method for integer programming. *Oper. Res.* 19(4):1070–1075.

Walker WE (1976) A heuristic adjacent extreme point algorithm for the fixed charge problem. *Management Sci.* 22(5):587–596.

Wolsey LA (1998) *Integer Programming* (Wiley-Interscience, New York).