Ann Oper Res (2009) 172: 45–69
DOI 10.1007/s10479-008-0483-2

Fixed Charge Transportation Problems: a new heuristic approach based on Lagrangean relaxation and the solving of core problems

Jesús Sáez Aguado

Published online: 3 December 2008

Abstract In this paper the Fixed Charge Transportation Problem is considered. A new heuristic approach is proposed, based on the intensive use of Lagrangean relaxation techniques. The more novel aspects of this approach are new Lagrangean relaxation and decomposition methods, the consideration of several core problems, defined from the previously computed Lagrangean reduced costs, the heuristic selection of the most promising core problem and the final resort to enumeration by applying a branch and cut algorithm to the selected core problem. For problems with a small ratio of the average fixed cost to the average variable cost (lower than or equal to 25), the proposed method can obtain similar or better solutions than the state-of-art algorithms, such as the tabu search procedure and the parametric ghost image processes. For larger ratios (between 50 and 180), the quality of the obtained solutions could be considered to be halfway between both methods.

Keywords Transportation problem · Fixed costs · Heuristics · Integer programming · Lagrangean relaxation · Core problem

1 Introduction

In this paper the Fixed Charge Transportation Problem (FCTP) is considered. The problem is a variant of the well known Linear Transportation Problem (LTP), and occurs when both fixed and variable costs are simultaneously present. There is a wide variety of applications of the FCTP, mainly in areas such as distribution, transportation, scheduling and location (Adlakha and Kowalski 2003). Moreover, other applications of FCTP have been cited in such problems as allocation of vehicles (Stroup 1967), solid-waste management (Walker 1976), process selection (Hirsch and Dantzig 1968) and teacher assignment (Hultberg and Cardoso 1997).

J. Sáez Aguado (✉)
Departamento de Estadística e Investigación Operativa, Universidad de Valladolid,
C/ Prado de la Magdalena s/n, 47005, Valladolid, Spain
e-mail: jsaez@eio.uva.es

The FCTP is a special case of the fixed cost linear programming problem, already introduced in the origins of the Operations Research (Hirsch and Dantzig 1968). Also, it is a special case of the Fixed Cost Network Problem (FCNP), which has a central role in many distribution and network design problems (Nemhauser and Wolsey 1988). The introduction of fixed costs to linear programming, network or transportation problems renders the objective function concave and discontinuous in the origin and this implies that a local minimum for these problems need not be a global minimum (Hirsch and Dantzig 1968).

In practice, the above facts imply that fixed cost problems are much more difficult to solve than their corresponding linear versions. In fact, as shown in Guisewite and Pardalos (1990), most minimization network problems with strictly concave objective function, including that with fixed-charges, are NP-hard, even when the underlying graph is bipartite, as happens in FCTP. On the other hand, Klose (2008) shows that a particular case of FCTP, the Single Source FCTP, is NP-hard, which also proves the NP-hardness of FCTP.

Network design problems constitute one of the most important areas in combinatorial optimization. These problems have numerous practical applications, mostly encountered in telecommunication, transportation and supply chain network planning. Moreover, important classical optimization problems can be seen as particular cases. See Magnanti and Wong (1984), Balakrishnan et al. (1997) for excellent surveys and additional information on this class of problems, as well as the text book by Ghiani et al. (2004). There are different approaches to solve network design problems, although three ones have to be pointed up: Branch and Bound, Lagrangean relaxation and heuristics. For multi-commodity network design problems, Gendron et al. (1999) gives a survey of recent modeling and algorithmic results, concluding that a efficient solution procedure should combine the three mentioned approaches. For the case single-commodity FCNP, the following papers deal with these approaches: Hochbaum and Segev (1989), Ortega and Wolsey (2003), Cruz et al. (1998) and Kim and Hooker (2002). Next, we review these approaches in the context of FCTP.

There are numerous algorithms in the literature for the solution, either exact or approximate, of the FCTP. Given the great computational difficulty of this problem, many heuristic methods have been developed over several decades. Hirsch and Dantzig (1968) proved that there is an optimal solution of the fixed charge problems that is an extreme point of the feasible region, and therefore the search for the optimal solution may be restricted to the extreme points of this region. Many heuristic algorithms have been designed based on this property, some of them being variants of the local search methods, such as the methods of Steinberg (1970) and Walker (1976), and the most recent of Adlakha and Kowalski (2003).

Other heuristic methods that must be emphasized are the Lagrangean heuristic method of Wright et al. (1989, 1991), the tabu search heuristic procedure (TS) of Sun et al. (1998) and the recently proposed parametric ghost image processes (GIP) of Glover et al. (2005).

With respect to the exact algorithms, the most widely applied for the FCTP are the Branch and Bound (B&B) ones. In Bell et al. (1999), Lamar and Wallace (1997), Palekar et al. (1990), different penalties are specifically developed for the FCTP and used in the B&B tasks of fathoming and guiding separation.

It is commonly accepted in the literature on the FCTP that exact solution algorithms are not very useful in practice since, except for small dimension problems, the computation time required is usually excessive (Sun et al. 1998). The main reason for this is that the most commonly used relaxations taking part in B&B methods, such as Lagrangean and linear relaxations, are weak for the FCTP. For example, the bound provided by Lagrangean relaxation in Wright et al. (1989, 1991) does not improve that provided by the linear relaxation, as the special structure has the integrality property. In almost all references to the B&B algorithms cited above, the relaxation used is simply the linear relaxation, which is very weak for the FCTP.

In this paper, a new heuristic approach for FCTP is proposed, which combines, in different phases, Lagrangean relaxation, Branch and Bound and heuristics. The method is based on the solution of certain core problems containing only a subset of the set of all variables. The use of a core problem for reducing the size of the original problem has been done, with remarkable success, in some large scale combinatorial problems, such as the Set Covering Problem (Caprara et al. 1999), the Capacitated Facility Location Problem (Avella et al. 2005) and the P-Median Problem (Avella et al. 2007).

The algorithm consists of three phases. In phase 1 we apply either Lagrangean relaxation or Lagrangean decomposition to obtain the both a lower bound and the Lagrangean reduced costs of all variables. The relaxations used in this paper are new and they are stronger than those used in other papers such as Wright et al. (1989, 1991). No attempt is made in this phase to obtain good solutions, as the problem is still too difficult. In phase 2 we define, from the previously computed reduced costs, one or several core problems with the same structure as the original problem but fewer variables. Lagrangean relaxation or Lagrangean decomposition is once again applied to each core problem and the best heuristic solution attained in this phase is saved. Note that the solution in this phase is obtained by applying only Lagrangean methods. In phase 3 we resort to enumeration, applying a standard branch and cut algorithm, with limited time, to the core problem that produced the best solution in phase 2, thus improving the final solution.

This paper is organized as follows. In Sect. 2 we give the formulation of the problem and a overview of the proposed procedure. Section 3 develops the Lagrangean relaxation and Lagrangean decomposition algorithms which are the base of the overall procedure. In Sect. 4 the core problem is defined, and Sect. 5 presents the simple heuristics that will be applied at different points of the algorithm. Section 6 gives an overall view of the method. Section 7 gives details of the different parts of the algorithm proposed and analyzes its computational performance on 72 FCTP test problems available in the literature. Finally, Sect. 8 gives the conclusions and the future extensions.

2 Formulation and general framework

The mixed integer programming formulation for FCTP is well known (Balinski 1961):

$$\text{Min} \quad \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij} + f_{ij}y_{ij}) \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = s_i \quad i = 1, \dots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad j = 1, \dots, n \tag{3}$$

$$x_{ij} \le u_{ij}y_{ij} \quad i = 1, \dots, m, j = 1, \dots, n \tag{4}$$

$$x_{ij} \ge 0 \quad i = 1, \dots, m, j = 1, \dots, n \tag{5}$$

$$y_{ij} \in \{0, 1\} \quad i = 1, \dots, m, j = 1, \dots, n \tag{6}$$

with the following parameters and variables:

m number of origins or supply points
n number of destinations or demand points
s_i supply of origin i, for each $i = 1, \ldots, m$
d_j demand of destination j, for each $j = 1, \ldots, n$
c_{ij} cost of shipping each unit from origin i to destination j
f_{ij} fixed cost incurred whenever there is a shipment from origin i to destination j
x_{ij} amount shipped from origin i to destination j
y_{ij} $= 1$ if $x_{ij} > 0$, and $= 0$ otherwise

Bounds u_{ij} in (4) can either be explicit, giving us a Capacitated FCTP, or they can be taken as $u_{ij} = \min(s_i, d_j)$, giving us the standard FCTP (the only one dealt with in this paper), although the formulation and the methods used are valid in both cases. It is well known that a transportation problem can easily be converted into a balanced one that verifies $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$, so we suppose the data already verify this condition.

The proposed heuristic method is first outlined step by step, and then each step is described in more detail.

Phase 1: Lagrangean relaxation. Apply either Lagrangean relaxation or Lagrangean decomposition to the original problem in order to obtain the Lagrangean reduced costs of all variables. No attempt is made in this phase to obtain good solutions, as the problem is still too difficult.

Phase 2: Core problem. Define, from the previously computed reduced costs, one or several core problems with the same structure as the original problem but fewer variables. Either Lagrangean relaxation or Lagrangean decomposition is once more applied to each core problem, with its corresponding heuristic solution, and the best solution attained in this phase is saved. Note that the solution in this phase is obtained by applying only Lagrangean methods. The algorithm used up to this phase we shall call Core 2.

The different core problems defined in this step depend on the values chosen for two parameters, namely K and F. K is the minimum number of indexes (i, j) we want in the core problem for any destination j, and F is a factor to compensate the fixed and variable costs. The choice of these parameters has a great effect on the final solution and will be examined in detail later.

Phase 3: Branch and Cut. Resort to enumeration by applying a branch and cut algorithm, with limited time, to the core problem that produced the best solution in phase 2, thus improving the final solution. The whole algorithm used up to this phase we shall call Core 3.

3 Lagrangean relaxation

In this section Lagrangean relaxation is applied to FCTP. Lagrangean relaxation is a standard procedure in Integer Programming. See Beasley (1993), Guignard (2003), Nemhauser and Wolsey (1988) for a complete description. The only references to the application of Lagrangean relaxation to FCTP are Wright et al. (1989, 1991).

3.1 Choosing underlaying relaxations

Immediately after, lower and upper bounds will be obtained by means of Lagrangean relaxation. The procedure requires the optimization on certain sets containing the variables of the problem. In this paper, we only consider sets containing separately the binary y and the continuous x variables. It is also possible to use sets containing simultaneously both types of variables, but this will not be done in this paper.

Let $\mathcal{F}$ be the feasible set for FCTP, defined by

$$\mathcal{F} = \{(x, y) \in R^{2mn} : (2) \text{ to } (6)\}$$

and let $Y_p = \text{proj}(\mathcal{F})$ be the projection of $\mathcal{F}$ onto the space of the y variables, that is

$$Y_p = \{y \in \{0, 1\}^{mn} : \text{there exists } x \in R^{mn} / (x, y) \in \mathcal{F}\}$$

Set Y_p is only defined implicitly, and is not easy to use in a subproblem since, as shown in Göthe-Lundgren and Larsson (1994), the solution of a linear optimization problem on Y_p requires the solution of a series of Set Covering problems. As this should be done every time the subproblem has to be solved, the approximation will be very inefficient. Instead, here we consider a series of relaxations of the set Y_p with an explicit representation, such that the optimization of the corresponding subproblems are easier. For posterior references we will use the following notations:

1. $Y_b = \{0, 1\}^{nm}$ is the binary set of dimension mn.
2. $Y_r = \{y \in \{0, 1\}^{nm} : \sum_{j=1}^{n} u_{ij} y_{ij} \geq s_i, i = 1, \ldots, m\}$ is the row knapsack set. Set Y_r is separable, and for every $i = 1, \ldots, m$ we have the knapsack set for row i

$$Y_i = \left\{ (y_{ij}) \in \{0, 1\}^n : \sum_{j=1}^{n} u_{ij} y_{ij} \geq s_i \right\} \tag{7}$$

3. $Y_c = \{y \in \{0, 1\}^{nm} : \sum_{i=1}^{n} u_{ij} y_{ij} \geq d_j, j = 1, \ldots, n\}$ is the column knapsack set. As in the previous case, set Y_c is separable and for every $j = 1, \ldots, n$

$$Y^j = \left\{ (y_{ij}) \in \{0, 1\}^m : \sum_{i=1}^{m} u_{ij} y_{ij} \geq d_j \right\} \tag{8}$$

is the knapsack set for the column j.

Each of the previous sets are relaxations of Y_p. In Nemhauser and Wolsey (1988) sets Y_r and Y_c were introduced for the FCTP, and their possible use in a fractional cutting plane algorithm is indicated. The following inclusions are verified:

$$Y_p \subseteq Y_r \subseteq Y_b \quad \text{and} \quad Y_p \subseteq Y_c \subseteq Y_b \tag{9}$$

The smaller the set over which the optimization is performed, the stronger the corresponding relaxation, yet the more difficult the solution of the subproblem is. Sets Y_r and Y_c give an adequate trade-off of these questions, and will be used intensively throughout the paper.

With respect to the continuous variables x, the feasible region in the variables $x \in R^{nm}$ given by constraints (2) to (5) will be denoted by X. So, a subproblem of linear optimization over X reduces to the well known Linear Transportation Problem (LTP).

We distinguish two cases, according to whether only one set is considered or two sets are simultaneously considered.

3.2 Using simple relaxations

Once a set Y, such as that of Sect. 3.1, has been chosen, the redundant constraints $y \in Y$ can be added to the formulation which is as follows:

$$\begin{aligned}
&\text{Minimize} && \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}y_{ij} \\
&\text{subject to} && x \in X && (10)\\
& && y \in Y && (11)\\
& && x_{ij} \le u_{ij}y_{ij} \quad \text{for all } i, j && (12)
\end{aligned}$$

Dualizing the linking constraints (12) with multipliers $\lambda_{ij} \ge 0$ the Lagrangean subproblem takes the form:

$$\begin{aligned}
(LR(\lambda)) \qquad &\text{Minimize} && \sum_{i=1}^{m}\sum_{j=1}^{n}\{c_{ij}x_{ij} + f_{ij}y_{ij} + \lambda_{ij}(x_{ij} - u_{ij}y_{ij})\} && (13)\\
&\text{subject to} && x \in X, y \in Y
\end{aligned}$$

This subproblem decomposes into two subproblems, one with variables x and constraints (10), and another with variables y and constraints (11):

$$\begin{aligned}
(LR_1(\lambda)) \qquad &\text{Minimize} && \sum_{i=1}^{m}\sum_{j=1}^{n}\{(c_{ij} + \lambda_{ij})x_{ij}\} && (14)\\
&\text{subject to} && x \in X
\end{aligned}$$

$$\begin{aligned}
(LR_2(\lambda)) \qquad &\text{Minimize} && \sum_{i=1}^{m}\sum_{j=1}^{n}\{(f_{ij} - \lambda_{ij}u_{ij})y_{ij}\} && (15)\\
&\text{subject to} && y \in Y
\end{aligned}$$

Subproblem $(LR_1(\lambda))$ is the well known linear Transportation Problem which can be solved by special methods (Nemhauser and Wolsey 1988). The form of subproblem $(LR_2(\lambda))$ depends on the chosen set Y. Generally, in most papers where Lagrangean relaxation is applied to FCTP or FCNP, the dualized constraints also are (12), but the y-subproblem is one trivial. For example, in Wright et al. (1989, 1991) and Cruz et al. (1998), Lagrangean relaxation is applied following the above scheme, but using $Y = Y_b$. In this case, subproblem $(RL_2(\lambda))$ is trivial and has the integrality property. This implies that the bound obtained solving the Lagrangean dual problem does not improve that obtained with the linear relaxation. In this paper we obtain stronger relaxations using sets strictly contained in Y_b, which does not have the integrality property. So, with the row knapsack structure Y_r, one has the subproblem:

$$(LR_2(\lambda)) \qquad \text{Minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n}\{(f_{ij} - \lambda_{ij}u_{ij})y_{ij}\}$$

$$\text{subject to} \quad \sum_{j=1}^{n} u_{ij}y_{ij} \geq s_i, \quad i = 1, \ldots, m$$

$$y_{ij} \in \{0, 1\}, \quad i = 1, \ldots, m, j = 1, \ldots, n$$

Here, the subproblem decomposes into m binary knapsack subproblems, each with n variables, and there are specific algorithms for their solution (Martello and Toth 1990).

Analogously, if we use the column knapsack set Y_c, the corresponding subproblem $(LR_2(\lambda))$ decomposes into n knapsack subproblems, each with m variables.

Given any set Y, such as that previously considered, we will denote by $LD(Y)$ the resulting Lagrangean dual, that is:

$$\max_{\lambda \geq 0}\{\nu(LR(\lambda))\}$$

where $\nu(P)$ indicates the optimal value of problem (P). From the relations in (9), the following inequalities are clear:

$$\nu(LP) = \nu(LD(Y_b) \leq \nu(LD(Y_r)) \leq z \tag{16}$$

where z is the optimal value of the problem and (LP) is the linear relaxation. Similar inequalities are verified if $\nu(LD(Y_r))$ is replaced by $\nu(LD(Y_c))$. As usual, the dual problem will be solved by a subgradient type algorithm.

3.3 Lagrangean decomposition

Adding the redundant constraints $y \in Y_r$ and $y \in Y_c$ to the formulation of FCTP, and making variable splitting, one obtains the reformulation (with shortened matrix notation):

$$\begin{aligned} \text{Minimize} \quad & cx + fy_1 \\ \text{subject to} \quad & x \in X \\ & y_1 \in Y_r \\ & y_2 \in Y_c \\ & x \leq uy_1 && (17) \\ & y_1 = y_2 && (18) \end{aligned}$$

Dualizing constraints (17) with non-negative multipliers $\lambda_{ij} \geq 0$, and constraints (18) with non-constrained multipliers μ_{ij}, for $i = 1, \ldots, m, j = 1, \ldots, n$, one obtains a Lagrangean subproblem $(LD(\lambda, \mu))$ that separates into three subproblems with variables $x \in X$, $y_1 \in Y_r$ and $y_2 \in Y_c$ respectively:

$$(LR_1(\lambda)) \quad \text{Minimize} \quad (c+\lambda)x \tag{19}$$
$$\text{subject to} \quad x \in X$$

$$(LR_3(\lambda,\mu)) \quad \text{Minimize} \quad (f+\mu-\lambda u)y_1 \tag{20}$$
$$\text{subject to} \quad y_1 \in Y_r$$

$$(LR_4(\lambda,\mu)) \quad \text{Minimize} \quad (-\mu)y_2 \tag{21}$$
$$\text{subject to} \quad y_2 \in Y_c$$

Here, the subproblem with variables y_1 decomposes into m knapsack subproblems, and that with variables y_2 separates into n knapsack subproblems, as in Sect. 3.

Denoting as $LD(Y_r, Y_c)$ the Lagrangean dual corresponding to this Lagrangean decomposition, it is well known (Guignard 2003) that its value is always better than or equal to that corresponding to the simple relaxations Y_r or Y_c, that is

$$\max(\nu(LD(Y_r)), \nu(LD(Y_c))) \le \nu(LD(Y_r, Y_c))$$

It is then possible to incorporate the information given by the two sets Y_r and Y_c to the Lagrangean scheme, though at the cost of having $2mn$ multipliers. As in the case of Lagrangean relaxation, the dual problem

$$\max_{\lambda\ge 0,\mu}\{\nu(LD(\lambda,\mu))\}$$

will be solved by a subgradient type algorithm.

4 The core problem

Once the dual problem has been solved in phase 1, we have available the optimal multipliers $\overline{\lambda}_{ij}$, in the case of Lagrangean relaxation, or $(\overline{\lambda}_{ij}, \overline{\mu}_{ij})$ in the case of Lagrangean decomposition. From these multipliers we are going to define a reduced core problem containing the variables we expect to be in the optimal solution. Whenever the variable x_{ij} is in the core problem, then the variable y_{ij} will also be present, so we will consider the pair (i, j), and the core problem will be defined by a set of pairs:

$$S \subset \{1,\dots,m\} \times \{1,\dots,n\}$$

If Lagrangean relaxation is applied in phase 1, from (14) and (15), the Lagrangean reduced costs for variables x_{ij} and y_{ij} are, respectively, $c_{ij} + \overline{\lambda}_{ij}$ and $f_{ij} - u_{ij}\overline{\lambda}_{ij}$. If Lagrangean decomposition is applied in phase 1, from (19), the Lagrangean reduced costs for variables x_{ij} is the same as in the previous case, but the variable y_{ij} has a reduced cost of $f_{ij} - u_{ij}\overline{\lambda}_{ij} + \overline{\mu}_{ij}$ in the first subproblem (20), and one of $-\overline{\mu}_{ij}$ in the second subproblem (21). In order to define a unique reduced cost for the variable y_{ij} there are several options, but if we add both costs the final reduced cost is $f_{ij} - u_{ij}\overline{\lambda}_{ij}$, the same as in the case of Lagrangean relaxation.

Now the Lagrangean reduced cost for the pair (i, j) is defined as

$$\pi_{ij} = f_{ij} - u_{ij}\overline{\lambda}_{ij} + F(c_{ij} + \overline{\lambda}_{ij})$$

where F is a factor for compensating the fixed and variable costs. Note that the multiplier $\overline{\mu}_{ij}$ does not appear in this formula. Now, for defining the index set S in the core, we have first to set a threshold π and then to select all pairs whose reduced cost does not exceed π, that is:

$$S = \{(i, j) : \pi_{ij} < \pi\}$$

The value of π is essential for the core problem to be effective, and has to be chosen so that the problem is feasible but does not contain too many variables. To this end, for each $j = 1, \ldots, n$ let k_j be the number of pairs (i, j) in S, and let $K = \min_{j=1,\ldots,n} k_j$, which is the minimum number of origins in the core for any destination. The following binary search algorithm finds the adequate value of π from a value for K introduced by the user. At any iteration lb and ub are, respectively, a lower and an upper bound of π.

Binary search algorithm

Input (π_{ij}) and K.
Output π.
Step 0 Initially set $lb = 0$ and $ub = \max\{\pi_{ij}\}$.
Step 1 Set $\overline{\pi} = \frac{1}{2}(lb + ub)$.
Step 2 For each $j = 1, \ldots, n$ let $S_j = \{i : \pi_{ij} < \overline{\pi}\}$, $k_j = |S_j|$ and $k\_\min = \min_{j=1,\ldots,n} k_j$.
Step 3 If $k\_\min = K$ stop, $\pi = \overline{\pi}$ is the final value. If $k\_\min < K$ set $lb = \overline{\pi}$ and go to Step 1. If $k\_\min > K$ set $ub = \overline{\pi}$ and go to Step 1.

The algorithm converges in a few iterations to a π value such that for any destination j the core problem contains at least K indexes $(i, j) \in S$. If K is too large, a slight reduction will be reached, whereas if K is too small, the core may not be feasible. We have experimented with different values for K and have found that the values from $K = 2$ to $K = 5$ are, generally, the most effective. Besides the value of K, the core problem depends on the value of F, the factor for compensating the fixed and the variable costs. The value or values taken by the parameters (K, F) has a great influence on the quality of the solution provided by the core problem. In Sect. 7 we deal with the question of how to choose the values for these parameters.

Summarizing, the index set of the core problem S is obtained with the following procedure of three steps:

1. From $\overline{\lambda}_{ij}$ and F, form the reduced cost of the pair (i, j)

$$\pi_{ij} = f_{ij} - u_{ij}\overline{\lambda}_{ij} + F(c_{ij} + \overline{\lambda}_{ij})$$

2. From (π_{ij}) and K calculate the threshold value π applying the binary search algorithm.
3. Set $S = \{(i, j) : \pi_{ij} < \pi\}$.

The core problem has the same structure as the original one, but in a sparse format. If we denote $S_i = \{j : (i, j) \in S\}$ for each $i = 1, \ldots, m$, and $S^j = \{i : (i, j) \in S\}$ for each $j = 1, \ldots, n$, the core problem is formulated as follows:

$$\text{Min} \quad \sum_{(i,j)\in S} (c_{ij}x_{ij} + f_{ij}y_{ij})$$

$$\text{s.t.} \quad \sum_{j\in S_i} x_{ij} = s_i \quad i = 1, \ldots, m \tag{22}$$

$$\sum_{i \in S^j} x_{ij} = d_j \quad j = 1, \ldots, n \tag{23}$$

$$x_{ij} \geq 0 \quad (i, j) \in S \tag{24}$$

$$x_{ij} \leq u_{ij} y_{ij} \quad (i, j) \in S \tag{25}$$

$$y_{ij} \in \{0, 1\} \quad (i, j) \in S \tag{26}$$

From this formulation, the core problem can be attacked by any exact algorithm as a branch and cut one, although, in spite of the reduction reached, on many occasions it is not possible to solve it optimally, and a heuristic method has to be used. In this paper we once more apply either Lagrangean relaxation or Lagrangean decomposition to the core problem to obtain the best heuristic solution in this phase 2. The relaxations and subproblems needed in this approach are analogous to those studied in Sect. 3 for the complete problem. So, for the continuous variables, the set

$$X(S) = \{x \in R^{|S|} : x \text{ verify } (22), (23), (24)\} \tag{27}$$

corresponds to the feasible set for a sparse $LTP(S)$. For the binary variables we consider the row knapsack set given by

$$Y_r(S) = \left\{ y \in \{0, 1\}^{|S|} : \sum_{j \in S_i} u_{ij} y_{ij} \geq s_i, i = 1, \ldots, m \right\} \tag{28}$$

and the column knapsack set given by:

$$Y_c(S) = \left\{ y \in \{0, 1\}^{|S|} : \sum_{i \in S^j} u_{ij} y_{ij} \geq d_j, j = 1, \ldots, n \right\} \tag{29}$$

Due to the sparsity of these sets, the solution of the Lagrangean relaxation or Lagrangean decomposition is much quicker than for the complete problem.

5 Simple heuristics

In the method we are proposing, either Lagrangean relaxation or Lagrangean decomposition is applied to the complete original problem in phase 1, and to each core problem in phase 2. However, no attempt is made to obtain good solutions in phase 1, as the problem is still too difficult. We limit ourselves to the search for good solutions in phase 2. During the solution of the relaxations of the previous sections on each core problem, certain solutions $x \in X(S)$ are obtained as solutions of their respective subproblems. From these solutions, and sometimes solving another additional subproblem, it is easy to obtain a complete feasible solution of the problem. So, we will refer to a *Lagrangean heuristic* as the best solution obtained while solving a Lagrangean relaxation or decomposition.

Given any solution $x \in X(S)$ it is enough to take $y_{ij} = 1$ if $x_{ij} > 0$ and $y_{ij} = 0$ otherwise, for each $(i, j) \in S$, so that (x, y) is a feasible solution for the core problem, and is thus also feasible for the original problem. We call this the *simple heuristic*, and it is the most usual way of finding a feasible solution from the different heuristic methods commented in Sect. 1. Vector $x \in X(S)$, from which the solution is found, can be either that obtained in the solution of the Lagrangean subproblem (14), or that obtained in the solution of any linear transportation problem with constraint set $x \in X(S)$. So, *Balinski's heuristic* (Balinski 1961) is obtained by first solving the *LTP*(S) with objective coefficients: $\lambda_{ij} = c_{ij} + f_{ij}/u_{ij}, (i, j) \in S$ and then completing the solution in the previously commented form. Balinski's heuristic was used to initialize the upper bound in the Lagrangean calculations.

On the other hand, we have found the following *improved heuristic* specially useful. It is applied along the subgradient algorithm, and takes elements of other known heuristics such as the dynamic slope scaling procedure (Kim and Pardalos 1999) and the parametric ghost image processes (Glover et al. 2005). Let $x \in X(S)$ be the solution at the current iteration, let $\overline{x}_{ij}$ be the maximum value attained by the variable x_{ij} until this iteration, and consider the *LTP*(S) with objective coefficients for $(i, j) \in S$ given by:

$$\lambda_{ij} = \begin{cases} c_{ij} + f_{ij}/x_{ij} & \text{if } x_{ij} > 0 \\ c_{ij} + f_{ij}/\overline{x}_{ij} & \text{if } x_{ij} = 0, \overline{x}_{ij} > 0 \\ \infty & \text{if } x_{ij} = 0, \overline{x}_{ij} = 0 \end{cases} \tag{30}$$

Observe that the objective in this problem is composed, in the first case, of the variable cost c_{ij} plus the fixed cost prorated with the current activity level f_{ij}/x_{ij}, when $x_{ij} > 0$. In the second case, we prorate the fixed cost with the maximum activity level until now. In the third case, the null activity level does not justify the fixed cost in any way. Thus, both the variable and the fixed costs are included in the new marginal cost, depending on the current and maximum activity level. Once an optimal solution $x' \in X(S)$ of the corresponding LPT has been found, it is completed with a solution (x', y') of FCTP.

From any feasible solution, obtained with any of the previous methods, it is possible to apply a *local search heuristic*, such as that commented in Sect. 1, to improve the solution. In this paper we apply this procedure once for each core problem.

6 Overall procedure

Besides the algorithmic parameters and the choices necessary for applying Lagrangean relaxation in the phases one and two (namely, the type of relaxation and the controls of the subgradient algorithm), there are other important parameters that are decisive for the final result. These are the number of core problems that we are going to examine (N), the values of parameters K and F for defining each of these problems, and the time spent in branch and bound enumeration in phase 3 (T_BB). We will take the same value of K for all problems, and different values $F_1, \ldots, F_N$ for parameter F. The choice of these parameters has a great effect on the final solution and will be examined in detail later. The procedure outputs the best solution found (x^*, y^*) and its objective function value Z^*.

The overall procedure can be summarized as follows:

Input: $K, N, F_1, \ldots, F_N, T\_BB$.
Output: $(x^*, y^*), Z^*$.
Phase 1: Lagrangean relaxation.

1.1 Calculate the Balinski's heuristic, solving the corresponding problem *LTP*(λ) stated in Sect. 5, and initialize (x^*, y^*) and Z^*.
1.2 Starting with the upper bound $UB = Z^*$ apply either Lagrangean relaxation or Lagrangean decomposition to the original problem, to obtain the optimal multipliers $\overline{\lambda}_{ij}$, in addition of the lower bound *LB*.

Phase 2: Core problem.

2.1 For $k = 1, \ldots, N$ do:
  2.1.1 Define, from the factor F_k, the reduced cost of pair (i, j) as $\pi_{ij}^k = f_{ij} - u_{ij}\overline{\lambda}_{ij} + F_k(c_{ij} + \overline{\lambda}_{ij})$.
  2.1.2 Apply the Binary search algorithm to find the threshold value π^k.
  2.1.3 Define the Core problem associated to the set of pairs: $S^k = \{(i, j) : \pi_{ij}^k < \pi^k\}$.
  2.1.4 Apply either Lagrangean relaxation or Lagrangean decomposition to the Core problem to obtain the heuristic solution (x^k, y^k) and its objective function value z^k.
  2.1.5 Apply the local search heuristic to the present solution, and rename it as (x^k, y^k) and z^k.
  2.1.6 If $z^k < z^*$ then actualize $z^* = z^k$ and $(x^*, y^*) = (x^k, y^k)$.
2.2 Save the Core problem that produced the best solution.

Phase 3: Branch and Cut.

3.1 Apply to the Core problem saved in step 2.2 the standard Branch a Cut algorithm starting with solution (x^*, y^*), and with a limit time of *T_BB*. Rename again the final found solution as (x^*, y^*) and its objective function value as z^*.
3.2 Calculate the final gap $100(z^* - LB)/z^*$.

7 Computational results

7.1 Introduction

In this section we report the results of applying the procedures of the previous sections to different FCTP problems. The objective of the experiments is to compare the effectiveness, both in the quality of solutions and the elapsed time, of the following procedures:

1. TS from Sun et al. (1998).
2. Parametric GIP from Glover et al. (2005).
3. Core 2 and Core 3.

Additionally, some problems are solved with Cplex. The Cplex solver is presently one of the most efficient implementations of B&C algorithms, and it includes the most recent developments in Integer Programming, such as different cutting planes methods, heuristics and preprocessing techniques, in addition to sophisticated rules for guiding the enumeration tree. Although it is a general purpose algorithm, its efficiency for solving different types of problems of IP is well known (Atamturk and Savelsbergh 2005), so the comparison of new procedures with Cplex can be interesting.

All algorithms needed for running Core 2 and Core 3 were programmed in C, compiled with Microsoft Visual C++.NET version 7.1, and run in a PC Pentium 3000 with 1 GB of RAM. For the solving of the different problems and subproblems of LP and IP the CPX library of solver Cplex, version 9.1, was used. The 0-1 knapsack subproblems arising in all algorithms were solved with the subroutine MT1 of Martello and Toth (1990). Finally, every LTP appearing in the Lagrangean algorithms and the different heuristics were solved with the specialized library CPXNET of Cplex. As all these problems have the same constraint set but different objectives, functions of the library CPXNET can be used to efficiently reoptimize each new problem.

Table 1 Problem characteristics

Type	Range of variable costs		Range of fixed costs		Ratio
	inf	sup	inf	sup	
A	3	8	50	200	22.72
B	3	8	100	400	45.25
C	3	8	200	800	90.90
D	3	8	400	1,600	181.81

Until very recently, the TS method of Sun et al. (1998) could be considered as the most efficient method to attack FCTP problems. The appearance of the Parametric GIP method of Glover et al. (2005) and its superior computational performance makes it currently the most efficient presently. Both methods are based on different algorithmic strategies and use local search intensively. The results corresponding to the TS and GIP methods are extracted from Glover et al. (2005), and were obtained in a Dell, Latitude Laptop, 1 GHz, with 256K Cache running on Windows 2000 operating system.

The treated problems and some details of the considered algorithms will be described in the following sub-sections. Finally, a comparison of the different methods will be made.

7.2 Test problems

In order to study the effectiveness of the method we have used the same comprehensive FCTP test bed as in Sun et al. (1998), Glover et al. (2005). The setbed consists of eight problem types, from letter A to letter H, each in seven problem sizes. For the same problem size, the ratio of the average fixed cost to the average variable cost rises from A to H. For each type and size, 15 totally dense problems were randomly generated, so the set contains 840 problems. A complete description of how the data were generated can be found in Sun et al. (1998), Glover et al. (2005).

To begin with, it is necessary to say that the proposed method is only suitable for problems with a ratio that is not excessively high, in our case for type A to D problems. For larger ratios, we would need to examine, in phase 2, too many values of parameters K and F, and Core 2 would be very inefficient. Furthermore, for large ratios, the enumeration in phase 3 attains too little improvement, and Core 3 would also be very inefficient. In Table 1 the main data used for the generation of these problems are given.

7.3 Some details of phases one to three

In phase 1 it is necessary to apply either Lagrangean relaxation or Lagrangean decomposition to the original problem. Moreover, in the first case, any of the sets Y_b, Y_r or Y_c could be used. The objective in this phase is not to obtain good lower and upper bounds, but to obtain Lagrangean reduced costs that allow to define an effective core problem. Here we have observed a strong association between obtaining a better lower bound and obtaining Lagrangean reduced costs that lead to a better core problem. In fact, in preliminary experimentation, we have verified that, in general, it is better to apply Lagrangean decomposition than simple Lagrangean relaxation, and within this method it is better to use sets Y_r or Y_c than Y_b. This question will be clarified in the following subsection.

The maximization of the Lagrangean function $v(LD(\lambda, \mu))$ was carried out by a subgradient algorithm, following the usual rule of updating (Beasley 1993; Guignard 2003).

In the case of Lagrangean decomposition, if (λ^k, μ^k) is the vector of multipliers in the iteration k, x^k is an optimal solution of the subproblem $(LR1(\lambda^k))$, y_1^k of the subproblem $LR3(\lambda^k, \mu^k)$ and y_2^k of the subproblem $LR4(\lambda^k, \mu^k)$, then the subgradient vector decomposes $\gamma^k = (\zeta^k, \eta^k) \in R^{2mn}$ where $\zeta_{ij}^k = x_{ij}^k - u_{ij} y_{ij}^k$ and $\eta_{ij}^k = x_{ij}^k - u_{ij} y_{ij}^k$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. The multiplier vectors in the following iteration are:

$$\lambda_{ij}^{k+1} = \max\left(0, \lambda_{ij}^k + \theta \frac{\overline{L} - L(\lambda^k, \mu^k)}{\|\gamma^k\|^2} \zeta_{ij}^k\right)$$

$$\mu_{ij}^{k+1} = \lambda_{ij}^k + \theta \frac{\overline{L} - L(\lambda^k, \mu^k)}{\|\gamma^k\|^2} \eta_{ij}^k$$

where $\overline{L}$ is an upper bound for $L(\lambda, \mu)$, that was taken as the value of the best solution found until that moment. Parameter θ was taken as equal to 2 for the first iteration, and halved whenever, after P subgradient iterations, the function $L(\lambda, \mu)$ did not improve more than a certain tolerance. The algorithm finishes when either the parameter θ becomes less than another tolerance, or the number of iterations exceeds a maximum allowed. The value taken by the parameter P is important for the efficiency of the method, and in this phase it is necessary to take high values such as $P = 60$, which was the value used.

With respect to the resolution of the successive subproblems the following process was used: first, Balinski's heuristic is applied to initialize the upper bound $\overline{L}$ and define an initial problem LTP. Then, every time the Lagrangean subproblem $(RL1(\lambda, x))$ has to be solved, it is enough to change the objective of the LPT and reoptimize it to obtain the new solution and the new heuristic. In this way, the computational cost of the successive reoptimizations is very low. In this phase we do not apply either the improved heuristic or the local search heuristic.

In phase 2, for each chosen value of the parameters K and F, a core problem is defined and a Lagrangean decomposition algorithm is applied to it. The subgradient algorithm is almost identical to that in phase 1, except for the subproblems corresponding to sets (27), (28) and (29) of Sect. 4, and the application of the improved heuristic at each subgradient iteration and the local search heuristic to the final feasible solution.

During phase 2, which is essentially a heuristic phase, we apply two types of heuristics. Firstly, at each iteration of the subgradient algorithm the subproblem (14) has to be solved, and this yields directly the simple heuristic, as stated in Sect. 5. Additionally, the improved heuristic can be obtained solving the new $LTP(\lambda)$ given by (30). Also, the local search heuristic could be apply in any point of the subgradient algorithm, but this should be time consuming. We limit its application a one only time, for each core problem that is being examined. The impact of these heuristics in the final solution is always low, that is, these heuristics improves only lightly the solution obtained from the simple Lagrangean heuristic.

In phase 3 we load the core problem that produced the best heuristic solution and apply the standard branch and cut algorithm implemented in Cplex 9.1. Most parameters and controls of the branch and cut algorithm were left in their default values, except for the use of the `CPXcopymipstart` routine to initialize the branch and cut search from the final solution of phase 2, and for the maximum time limit that was set at a short value.

7.4 Initial experiments

In this subsection we describe some initial computational experiments that try to justify the algorithms Core 2 and Core 3. At first place, it is necessary to specify what relaxation will

Table 2 Results of the phase 1 for the different relaxations

Problem	$\nu(LD(Y_b))$	t_b	$\nu(LD(Y_r))$	t_r	$\nu(LD(Y_c))$	t_c	$\nu(LD(Y_r, Y_c))$	t_{rc}
N3004	163,610.04	0.97	164,290.84	1.80	164,519.13	1.50	164,525.30	2.09
N3009	163,351.05	1.16	164,025.33	2.37	164,526.64	1.59	164,529.63	2.97
N300E	165,262.08	1.02	165,989.71	1.83	166,780.52	1.59	166,780.14	2.91
N3104	171,174.80	1.09	173,131.50	2.20	173,758.48	1.55	173,762.79	3.14
N3109	170,074.54	1.14	171,448.54	2.25	172,568.28	1.59	172,576.49	2.75
N310E	171,573.97	1.22	173,067.83	2.05	174,679.14	1.61	174,686.26	2.98
N3204	186,506.14	1.22	189,127.19	2.23	191,331.47	1.63	191,336.09	2.92
N3209	183,002.44	1.24	185,946.56	2.13	188,521.90	1.63	188,533.52	2.92
N320E	183,673.36	1.30	186,496.01	1.97	190,334.23	1.67	190,352.27	3.08
N3304	212,966.62	1.36	218,114.20	2.50	223,155.28	1.70	223,156.28	2.16
N3309	208,040.92	1.38	213,287.33	2.11	219,799.72	1.70	219,809.49	2.27
N330E	206,036.61	1.33	211,912.61	2.19	220,324.35	1.70	220,354.09	2.36

to be chosen both in the phase 1 and in the phase 2. For the first phase, we need to compare the strength of the four considered relaxations, and their computational time. To this end, we have selected, inside the problems solved in Glover et al. (2005), twelve difficult problems, three of each type, and we have applied Lagrangean relaxation, based in each of the sets Y_b, Y_r and Y_c, as well as Lagrangean decomposition, based on (Y_r, Y_c). Table 2 gives, for each relaxation, the corresponding lower bound and computation time.

We observe in Table 2 that, as expected, the bound $LD(Y_r, Y_c)$ is the stronger, although $\nu(LD(Y_c))$ is always near enough. Moreover, given that the computation time with this relaxation is only lightly larger than with the others, we finally, we decide for Lagrangean decomposition for the first phase, although Lagrangean relaxation based on Y_c could have been another good option. For the second phase, we have to pay attention to the heuristic solution of the core problem obtained for each of the four possible relaxations. Here, there is not any relaxation that produces always better solutions, although in extensive preliminary computational tests, we have found that Lagrangean decomposition based on (Y_r, Y_c) yields more frequently the better solutions, followed by Lagrangean relaxation based on Y_c. Finally, we decide again for Lagrangean decomposition based on (Y_r, Y_c) for the second phase, although Lagrangean relaxation based on Y_c produces rather similar, although lightly inferior, results.

Another initial question is to check how much reduction is attained in the core problem. This depends, mainly, on the value of K, whereas it is very insensitive to the value of F. To illustrate this question, we have selected four problems of dimensions 50×100, one for each type, and for a fixed value of F, say $F = 5$, we have calculated the number of variables of the core problem for different values of the parameter K. The following table gives, for each problem and each value of K, the number of variables of the core problem and, in the two last rows, the average number in the four problems and the percentage over the possible total number (10,000).

For small values of K the reduction is considerable, and predictably the core problem will be more easily solved than the original problem. The reduction is very similar for the rest of the problems with identical dimensions, although it is smaller for problems of smaller dimensions.

Table 3 Core size

Problem	Number of columns of the core for $K =$						
	2	3	4	5	8	14	20
N3004	1,338	1,680	1,792	2,078	2,934	4,458	5,618
N3104	1,380	1,704	2,088	2,444	3,128	4,512	5,862
N3204	1,432	1,682	1,926	2,324	2,932	4,818	6,002
N3304	1,136	1,778	1,972	2,420	2,910	4,948	5,956
Average	1,321.5	1,711.0	1,944.5	2,316.5	2,976.0	4,684.0	5,859.0
Percentage	13.21	17.11	19.44	23.16	29.76	46.84	58.59

Table 4 Initial results for some easy problems

Problem		Cplex 9.1		Core 2 ($K = 3$, $F = 5$)		Core 3 ($K = 3$, $F = 5$)	
Name	Size	z^1	t^1	z^2	t^2	z^3	t^3
N104	10 × 10	40,258	0.08	40,258	0.09	40,258	0.98
N107	10 × 10	42,029	0.19	42,130	0.19	42,129	0.20
N204	15 × 15	54,502	0.89	54,604	0.16	54,604	0.20
N207	10 × 15	53,596	2.16	53,610	0.23	53,596	0.44
N304	10 × 20	56,366	0.33	56,366	0.17	56,366	0.20
N307	10 × 20	49,742	1.30	49,767	0.19	49,742	1.02
N504	10 × 30	57,130	93.44	57,152	0.23	57,130	5.53
N507	10 × 30	52,903	12.81	52,918	0.31	52,903	0.72

The next question we have to answer is how effective the core problem will be for small values of K. As another initial experiment, we have selected a set of easy problems, extracted from the set of easy problems solved in Glover et al. (2005), for which Cplex 9.1 can find the optimal solution, and we have first solved the original problem with Cplex 9.1. Then we have applied Core 2 for fixed values of K and F, say $K = 3$ and $F = 5$, and finally we have solved the core problem again with Cplex 9.1 (Core 3). The main results, summarized in Table 4, show the problem name and size, the optimal objective function value (z^1) and the CPU time (t^1) for Cplex 9.1, the optimal objective function value (z^2) and the CPU time (t^2) for Core 2, and analogous values (z^3) and the CPU time (t^3) for Core 3, in the last two columns.

As can be observed in Table 4, for all problems except for N204, the solution of the core problem coincides with the solution of the original problem. This indicates that the core problem includes the variables that generally participate in the optimal solution. Furthermore, for the more difficult problems, N504 and N507, the solution of the core problem is obtained much quicker than the solution of the original problem.

The following initial question is to check the performance of the algorithm for more difficult problems, for which the solution to optimality of neither the original problem nor the core problem is possible. In this point there is a great difference in the situation for other problems where a core problem has been used, such as the p-median or capacitated facility location problems. For these problems, once the core problem is defined, its complete resolution is possible by branch and cut (Avella et al. 2005, 2007) even for large scale problems.

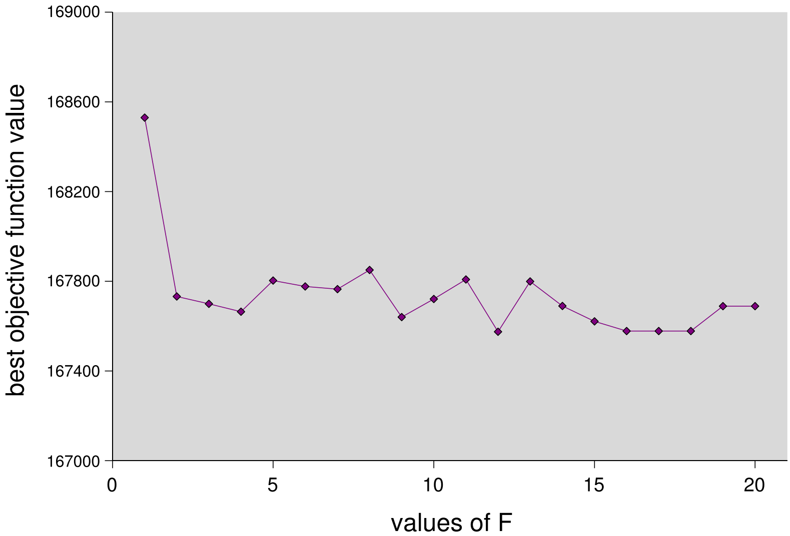

Fig. 1 Best objective function value for different values of F

For FCTP, in many cases, the core problem is too difficult to be solved to optimality, and a heuristic approach is required. In this paper we have again applied a Lagrangean heuristic although, in principle, we have different approaches depending on both the number of core problems we want to explore in phase 2 and the time we want to devote to branch and bound enumeration in phase 3. We have grouped all possibilities in three approaches or options. In the first approach, we first define the core problem for a single value of K and F, that is $N = 1$, apply the fast Lagrangean heuristic (Core 2), and then apply the branch and cut algorithm to the resulting core problem (Core 3). In the second approach, we first explore the core problem for $N = 10$ values of K and F, applying the fast Lagrangean heuristic to each one and saving the best solution obtained during the process, and then apply Core 3 to the core problem that produced the best solution. In the third approach, we explore the core problem for $N = 20$ values of K and F, applying the fast Lagrangean heuristic to each one and saving the best solution obtained during the process.

The values given to the parameters K and F influence both the computational time and the solution quality. In extensive computational experience, we have observed that for problem types A and B the value $K = 5$ produces the better solution most frequently, while for types C and D this happens for the value $K = 2$. These are the final values for parameter K. With respect to the value of the parameter F, it is difficult to anticipate what the best one is because near values can produce solutions of substantially different quality. To illustrate the difficulties in the choice of the value for F, the next figure shows a typical picture of the value of the found solution for different values of F (in this case from 1 to 20 in increments of 1). We have found no procedure for determining an adequate value for F except for the solving of the core problem for different values of F.

From our computational experience, the most we can predict is a range of values for F where the best solution is most frequently found. For example, for problem types A and B the range from 1 to 10 is very frequent, whereas for problem types C and D the range from 20 to 40 is frequent enough. In Table 5 we summarize the values taken by the different parameters for the three algorithmic options mentioned above. As a notation matter, $F = 21 \ldots 40$, $I = 2$

Table 5 Values of parameters for the three algorithmic options

	Option 1 ($N = 1$)	Option 2 ($N = 10$)	Option 3 ($N = 20$)
Types A and B	$K = 5$	$K = 5$	$K = 5$
	$F = 5$	$F = 1 \ldots 10$	$F = 1 \ldots 10$
	$T\_BB = 60$	$I = 1$	$I = 0.5$
		$T\_BB = 30$	$T\_BB = 0$
Types C and D	$K = 2$	$K = 2$	$K = 2$
	$F = 30$	$F = 21 \ldots 40$	$F = 21 \ldots 40$
	$T\_BB = 60$	$I = 2$	$I = 1$
		$T\_BB = 30$	$T\_BB = 0$

indicates that F ranges from 21 to 40 in increments of 2, and T_BB shows the seconds spent in branch and bound enumeration in Core 3.

In order to explore the above possibilities, in the third initial experiment we have selected, inside the problems solved in Glover et al. (2005), twelve difficult problems, three of each type, and we have applied Core 2 and Core 3 for the three sets of selected parameters. In this way, the total elapsed time with all methods is roughly the same. The results, summarized in Table 6 in the Appendix, show, for each option, the best objective function value found and the execution time for Core 2, z^2 and t^2 respectively, and analogously for Core 3, z^3 and t^3. For each problem, we use the symbol * to indicate the option where the best solution was obtained.

We observe in Table 6, firstly, that with Option 1 the solutions are always worse than with the other options. Option 2 is very advantageous for problem types A and B, that is, for small ratios. For these problems, to examine 10 core problems and carry out branch and bound enumeration during 30 seconds yields better solutions than either examining only one core problem and carrying out branch and bound enumeration during 60 seconds, or examining 20 core problems and not carrying out branch and bound enumeration. Instead, for problems of types C and D, Option 3 is clearly the best, because with branch and bound enumeration, the improvement of the solution of Core 2 is null in many cases.

7.5 Complete results

In this subsection we make the comparisons between the proposed algorithms Core 2 and Core 3, and the TS method of Sun et al. (1998) and the parametric GIP of Glover et al. (2005), on a total of 72 problems, including the subset of 12 easy problems and the 60 large and difficult problems of types A to D, all solved in Glover et al. (2005). For type A and B problems and the easy problems, we run Core 2 and Core 3 with the above Option 2 and parameters $K = 5$, $F = 1 \ldots 10$, $I = 1$ and $T\_BB = 60$. For type C and D problems we run Core 2 with Option 3 and parameters $K = 2$, $F = 21 \ldots 40$, $I = 1$ and $T\_BB = 0$.

We begin with the subset of easy problems. For these problems it is still interesting to compare the different algorithms with Cplex. Table 7 in the Appendix presents the results for these 12 problems. The data for columns corresponding to Cplex 9.0, TS and GIP are extracted from Glover et al. (2005), and were obtained with a Dell, Latitude Laptop, Pentium III, 1 GHz, with 256K Cache running on Windows 2000 operating system. Cplex was executed with a limit time of 11,000 seconds.

From Table 7 we observe that Core 3 obtains the same objective function value as TS for five problems and better ones for seven. Comparing with GIP, Core 3 obtains the same objective function value as GIP for seven problems, better ones for three and worse for two. Comparing with Cplex, for the problems for which Cplex can obtain an optimal solution, Core 3 reaches the same value, and for the rest Cplex obtains a better solution than Core 3 for two problems and worse for the other two. We can conclude that, for these types of problems, Core 3 is competitive enough. On the other hand, the performance of Core 2 is only similar to that of TS but not that of the other methods. This indicates that for this type of problem the branch and bound enumeration carried out in Core 3 is essential, although it has to be applied on the reduced core problem obtained in Core 2.

Next, the results for the difficult problems of types A to D are presented in Tables 8 to 11 in the Appendix. For these problems the results from Cplex are not included because of its inferior performance. Again the data for the columns corresponding to TS and GIP are extracted from Glover et al. (2005).

For problem types A and B, Core 3 obtains a better objective function value than TS for 23 out of the 30 problems, and worse for the remaining 7. Compared with GIP, Core 3 obtains a better objective function value than GIP for 8 out of the 30 problems, and worse for the remaining 22. Core 2 obtains a better objective function value than TS for 19 out of the 30 problems, and worse for the remaining 11, whereas it obtains worse solutions than GIP for all 30 problems. As in the case of the easy problems, the branch and bound enumeration carried out in Core 3 is essential to improve the solutions obtained in Core 2. We can conclude that for problems of type A and B, Core 3 is still competitive enough, as it outperforms TS, and for some problems it obtains solutions of higher quality than GIP, although overall GIP is still the best method.

For problem types C and D, Core 2 obtains a better objective function value than TS for 18 out of the 30 problems, and worse for the remaining 12. Compared with GIP, Core 2 obtains a worse objective function value than GIP for all problems. For problems of types C and D, the performance of Core 2 is similar or slightly superior to that of TS, and worse than that of GIP in all problems.

With respect to the computation times, it is necessary to make some observations. Firstly, our results and those of Glover et al. were obtained in different machines and, therefore, their respective computation times are difficult to compare. A PC Pentium IV 3 GHz can be several times faster than a PC Pentium III 1 GHz, but this ratio is unknown. Even after looking at Dongarra (2008), it is difficult to deduce the value of this ratio, although it could be between 3 and 5. Then, the shown computation times have to be analysed with caution. Secondly, our method has an additional feature not present in the other ones, as it provides a lower bound allowing us to calculate the final gap. This fact influences the final computation times.

8 Summary and concluding remarks

In this paper the Fixed Charge Transportation Problem (FCTP) is considered and a new heuristic approach is presented, which is composed of three components or phases: Lagrangean relaxation, core problem and branch and bound enumeration. From its formulation as an MIP problem, and in the context of Lagrangean Relaxation, different special relaxations are considered, which lead to two types of subproblems: continuous transportation problems and binary knapsack problems. Although the use of Lagrangean relaxation is not novel for FCTP problems, we do not know the use of strong knapsack sets for FCTP. Using these structures, several Lagrangean relaxation and Lagrangean decomposition algorithms have been designed. The purpose of solving these relaxations is not properly the calculation of good bounds but the calculation of good reduced costs for all pairs of variables (x_{ij}, y_{ij}). This allows us to define, in phase 2, different core problems for each value of two parameters. As the core problems are, in general, too difficult to solve to optimality we have to apply a heuristic approach, and in this paper we again apply the Lagrangean relaxation algorithm which provides, besides the lower bound, a heuristic solution. Finally, in phase 3, we can resort to branch and bound enumeration and, in some cases, the solution of phase 2 can be improved.

To evaluate the effectiveness of the proposed algorithms, 72 test problems from the literature were considered and the efficiency of the algorithms was compared with that of two state of the art methods: the tabu search procedure of Sun et al. (1998) and the parametric ghost image processes of Glover et al. (2005). For problems with small ratios, specifically for ratios lower than or equal to 25, the proposed method can obtain similar or better solutions than the two state-of-art algorithms. For ratios between 50 and 180, the algorithm obtains solutions with better quality than the tabu search, although worse than the parametric ghost image processes. Its performance could be considered to be halfway between both methods for the type of problems considered.

Some additional questions not treated in this paper, and several possible extensions can be the following:

1. Stronger Lagrangean relaxations based on new binary sets Y, or even mixed sets including both types of variables.
2. Enhancements in the branch and cut phase, such as use of specific valid inequalities, use of linear programming based heuristics, or use of penalties for the branching variable.
3. Extensions to FCNP and other multicommodity capacitated fixed charge network design, as those dealt with in Crainic et al. (2004).
4. More intensive use of the local search heuristics, or even inclusion of strategies of tabu search and parametric ghost image processes.

Acknowledgement The author wish to thank the anonymous referees for their useful suggestions and comments that improved the paper.

Appendix: Results for some difficult problems

Table 6 Initial results for some difficult problems

Problem	Option 1				Option 2				Option 3	
	z^2	t^2	z^3	t^3	z^2	t^2	z^3	t^3	z^2	t^2
N3004	167,799	3.67	167,451	64.11	167,593	20.94	167,394*	51.70	167,542	37.33
N3009	167,605	4.63	167,171	65.67	167,320	36.09	167,162*	67.03	167,320	66.28
N300E	173,250	5.41	173,126	65.95	169,958	33.94	169,609*	64.84	169,940	62.45
N3104	180,322	3.11	179,776	64.36	179,609*	21.52	179,609*	51.81	179,609*	50.16
N3109	178,069	5.72	178,000	66.03	178,069	37.30	178,000	67.61	177,949*	73.05
N310E	184,193	4.61	183,823	64.92	183,987	30,50	183,624	60.80	183,986	56.13
N3204	202,189	4.24	202,189	64.39	202,189	20.59	202,189	50.77	201,408*	32.78
N3209	199,799	3.75	199,799	63.91	198,497	27.36	198,497*	57.48	198,497*	48.89
N320E	205,521	3.39	204,367	64.06	204,538	28.86	204,085	59.67	201,444*	51.16
N3304	246,367	3.81	246,637	63.92	246,285	19.58	246,285	49.70	243,803*	37.36
N3309	241,622	6.11	240,679	67.03	240,493	44.00	240,493	74.25	239,293*	61.50
N330E	245,335	4.28	245,098	64.52	243,854	32.69	243,516	62.98	242,794	60.34*

Table 7 Results for easy problems

Problem	Cplex 9.0		TS		GIP		Core 2		Core 3	
name	z_Cplex	t_Cplex	z_TS	t_TS	z_GIP	t_GIP	z^2	t^2	z^3	t^3
N104	40,258	0.11	40,258	0.03	40,258	0.26	40,258	0.06	40,258	0.36
N107	42,029	0.33	42,029	0.04	42,029	0.18	42,030	0.20	42,029	0.27
N204	54,502	2.70	54,502	0.11	54,502	1.12	54,578	4.17	54,502	5.44
N207	53,596	5.37	53,601	0.13	53,601	1.06	53,610	0.34	53,596	0.81
N304	56,366	1.00	56,391	0.13	56,366	0.78	56,366	0.33	56,366	0.44
N307	49,742	4.61	49,742	0.12	49,742	0.89	49,756	0.39	49,742	2.61
N504	57,130	237.33	57,130	0.28	57,130	2.86	57,152	0.61	57,130	6.75
N507	52,903	43.57	52,977	0.24	52,903	1.39	52,918	0.66	52,903	1.45
N1004	163,599	11,000	163,793	5.48	163,585	22.95	163,787	3.25	163,692	63.33
N1007	162,300	11,000	162,313	6.75	162,237	35.43	162,453	2.48	162,234	62.56
N2004	104,046	11,000	104.193	15.87	104,001	39.97	104,129	13.58	104,031	73.63
N2007	104,147	11,000	104,341	13.08	104,256	27.23	104,313	10.61	104,254	70.75

Table 8 Results for type A difficult problems

Problem	Core 2		Core 3		TS		GIP	
	z^2	t^2	z^3	t^3	z_ts	t_ts	z_gip	t_gip
N3000	168,212	21.11	167,957	81.36	168,460	33.44	168,057	193.23
N3001	166,806	24.03	166,684	84.73	166,930	40.38	166,678	178.12
N3002	168,145	26.05	168,000	86.83	167,888	37.31	167,919	195.38
N3003	168,680	22.27	168,431	82.33	168,847	27.38	168,434	144.31
N3004	167,593	20.94	167,244	81.20	167,581	33.08	167,275	194.51
N3005	168,041	28.63	167,658	89.17	168,251	37.84	167,639	195.59
N3006	165,943	25.28	165,747	85.81	166,287	31.56	165,862	179.85
N3007	167,515	27.03	167,349	87.72	167,845	32.85	167,364	196.17
N3008	165,968	34.69	165,888	95.56	165,944	28.34	165,576	187.98
N3009	167,320	34.55	167,080	95.81	167,206	30.79	167,193	110.50
N300A	167,568	24.00	167,436	84.97	167,895	29.54	167,358	142.26
N300B	168,727	35.48	168,633	95.81	168,807	35.68	168,504	175.73
N300C	165,734	59.41	165,698	119.95	165,765	32.11	165,295	192.61
N300D	166,446	35.25	166,371	100.50	166,295	37.89	166,217	204.43
N300E	169,958	32.73	169,490	93.52	169,865	28.90	169,375	155.39

Table 9 Results for type B difficult problems

Problem	Core 2		Core 3		TS		GIP	
	z^2	t^2	z^3	t^3	z_ts	t_ts	z_gip	t_gip
N3100	179,084	14.27	178,927	74.70	179,672	43.89	179,019	192.27
N3101	178,318	30.05	177,761	91.41	178,518	34.88	177,861	190.51
N3102	179,271	36.41	179,271	96.84	179,021	33.77	179,007	190.77
N3103	179,369	40.14	179,231	100.55	179,278	31.33	179,017	212.18
N3104	179,676	22.81	179,676	83.09	179,828	42.36	179,230	211.28
N3105	178,841	25.44	178,764	85.67	178,714	42.55	178,160	147.71
N3106	177,116	25,61	176,720	86.02	177,304	46.67	176,546	183.65
N3107	178,441	39.03	178,271	99.38	178,567	30.25	177,904	206.68
N3108	176,955	45.48	176,863	105.84	176,540	36.27	176,266	185.68
N3109	178,069	35.33	177,843	95.63	178,077	38.18	177,599	134.07
N310A	179,400	30.14	179,258	90.42	179,432	32.80	178,703	188.21
N310B	180,168	46.52	179,895	108.42	180,020	34.38	179,647	120.21
N310C	176,452	40.67	176,202	101.02	176,106	36.55	175,850	162.26
N310D	177,943	41.30	177,797	103.58	178,287	38.10	177,328	205.80
N310E	183,987	30.56	183,774	90.88	180,273	47.55	179,763	190.15

Table 10 Results for type C difficult problems

Problem	Core2		TS		GIP	
	z^2	t^2	z_ts	t_ts	z_gip	t_gip
N3200	200,670	68.53	201,441	31.67	199,611	185.97
N3201	199,797	75.72	199,720	49.61	198,843	206.12
N3202	201,593	66.81	201,728	33.54	199,986	195.94
N3203	200,055	45.30	200,648	34.54	199,338	220,96
N3204	201,408	72.44	201,748	42.21	201,089	184.59
N3205	199,960	75.46	199,576	54.21	198,764	210.42
N3206	198,189	41.53	198,305	43.47	197,383	134.98
N3207	199,096	46.14	200,195	39.88	198,006	208.88
N3208	197,292	41.22	197,043	42.55	196,558	209.44
N3209	198,497	48.89	199,160	29.43	198,262	215.64
N320A	201,108	58.97	201,041	35.33	197,924	201.10
N320B	201,813	46.98	202,682	29.29	201,108	187.57
N320C	197,334	43.97	198,738	36.73	196,264	210.73
N320D	199,184	51.66	199,738	46.42	198,158	189.05
N320E	201,444	51.34	201,583	38.14	200,178	172.56

Table 11 Results for type D difficult problems

Problem	Core 2		TS		GIP	
	z^2	t^2	z_ts	t_ts	z_gip	t_gip
N3300	240,597	66.30	240,209	34.80	239,115	210.02
N3301	240,410	94.73	241,428	33.86	238,570	159,56
N3302	242,310	110.91	240,555	39.57	239,876	197.97
N3303	239,015	68.02	237,274	40.18	237,204	157.40
N3304	243,803	72.13	243,778	40.03	241,295	208.32
N3305	240,647	77.91	241,594	36.97	237,920	187.67
N3306	237,434	64.72	237,461	47.34	236,061	197.88
N3307	237,611	86.75	238,483	33.18	236,150	210.07
N3308	236,530	63.36	236,800	33.91	234,479	159.81
N3309	239,293	105.31	238,961	34.61	238,233	197.42
N330A	242,812	64.06	242,350	38.09	242,000	187.10
N330B	242,830	76.30	243,341	36.71	241,009	205.16
N330C	237,050	63.83	237,911	30.90	235,173	201.58
N330D	239,881	48.09	237,071	42.69	236,002	188.27
N330E	243,626	104.77	241,727	44.41	238,434	197.21

References

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

Atamturk, A., & Savelsbergh, M. W. (2005). Integer programming software systems. *Annals of Operations Research*, *140*, 67–124.

Avella, P., Boccia, M., Sforza, A., & Vasil'ev, I. (2008). An effective heuristic for large-scale capacitated facility location problems. *Journal of Heuristics*. doi:10.1007/s10732-008-9078-y.

Avella, P., Sassano, A., & Vasil'ev, I. (2007). Computational study of large-scale p-Median problems. *Mathematical Programming*, *109*, 89–114.

Balakrishnan, A., Magnanti, T. L., & Mirchandani, P. (1997). Network design. In M. Dell'Amico, F. Maffioli, & S. Martello (Eds.), *Annotated bibliographies in combinatorial optimization*. New York: Wiley.

Balinski, M. L. (1961). Fixed-cost transportation problems. *Naval Research Logistics Quarterly*, *8*, 41–54.

Beasley, J. E. (1993). Lagrangean relaxation. In C. R. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems* (pp. 243–298). Oxford: Blackwell.

Bell, G. J., Lamar, B., & Wallace, C. A. (1999). Capacity improvement, penalties, and the fixed charge transportation problem. *Naval Research Logistics Quarterly*, *46*(4), 341–355.

Caprara, A., Fischetti, M., & Toth, P. (1999). A heuristic method for the Set Covering Problem. *Operations Research*, *47*, 730–743.

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

Cruz, F. R. B., MacGregor Smith, J., & Mateus, G. R. (1998). Solving to optimality the uncapacitated fixed-charge network flow problem. *Computers & Operations Research*, *25*(1), 67–81.

Dongarra, J. J. (2008). Performance of various computers using standard linear equations software. http://www.netlib.org/benchmark/performance.ps.

Gendron, B., Crainic, T. G., & Frangioni, A. (1999). In P. Soriano & B. Sansò (Eds.), *Multicommodity capacitated network design, telecommunications network planning* (pp. 1–19). Norwell: Kluwer Academic.

Ghiani, G., Laporte, G., & Musmanno, R. (2004). *Introduction to logistics systems planning and control*. New York: Wiley.

Glover, F., Amini, M., & Kochenberger, G. (2005). Parametric ghost image processes for fixed-charge problems: a study of transportation networks. *Journal of Heuristics*, *11*, 307–336.

Göthe-Lundgren, M., & Larsson, T. (1994). A set covering reformulation of the fixed charge transportation problem. *Discrete Applied Mathematics*, *48*, 245–259.

Guignard, M. (2003). Lagrangean relaxation. *TOP*, *11*, 151–228.

Guisewite, G. M., & Pardalos, P. M. (1990). Minimum concave-cost network flow problems: applications, complexity, and algorithms. *Annals of Operations Research*, *25*, 75–100.

Hirsch, W. M., & Dantzig, G. B. (1968). The fixed charge problem. *Naval Research Logistics Quarterly*, *15*, 413–424.

Hochbaum, D. S., & Segev, A. (1989). Analysis of a flow problem with fixed charges. *Networks*, *19*, 291–312.

Hultberg, T. H., & Cardoso, D. M. (1997). The teacher assignment problem: A special case of the fixed charge transportation problem. *European Journal of Operational Research*, *101*, 463–473.

Kim, H. J., & Hooker, J. N. (2002). Solving fixed-charge network flow problems with a hybrid optimization and constraint programming approach. *Annals of Operations Research*, *115*, 95–124.

Kim, D., & Pardalos, P. M. (1999). A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, *24*, 195–203.

Klose, A. (2008). Algorithms for solving single-sink fixed-charge transportation problem. *Computers & Operations Research*, *35*, 2079–2092.

Lamar, B. W., & Wallace, C. (1997). Revised-modified penalties for fixed charge transportation problems. *Management Science*, *43*, 1431–1436.

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

Martello, S., & Toth, P. (1990). *Knapsack problems*. New York: Wiley.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization*. New York: Wiley.

Ortega, F., & Wolsey, L. A. (2003). A Branch-and-Cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. *Networks*, *41*(3), 143–158.

Palekar, U. S., Karwan, M. H., & Zionts, S. (1990). A branch-and-bound method for the fixed charge transportation problem. *Management Science*, *36*, 1092–1105.

Steinberg, D. I. (1970). The fixed charge problem. *Naval Research Logistics Quarterly*, *17*, 217–236.

Stroup, J. W. (1967). Allocation of launch vehicles to space missions: a fixed-cost transportation problem. *Operations Research*, *15*, 1157–1163.

Sun, M., Aronson, J. E., McKeown, P. G., & Drinka, D. (1998). A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operational Research*, *106*, 441–456.

Walker, W. E. (1976). A heuristic adjacent extreme point algorithm for the fixed charge problem. *Management Science*, *22*, 587–596.

Wright, D. D., & Haehling von Lanzenauer, C. (1989). Solving the fixed charge problem with Lagrangian relaxation and cost allocation heuristics. *European Journal of Operational Research*, *42*, 305–312.

Wright, D. D., & Haehling von Lanzenauer, C., (1991). COAL: A new heuristic approach for solving the fixed charge problem—computational results. *European Journal of Operational Research*, *52*, 235–246.