Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research 33 (2006) 1226–1241

computers & operations research

www.elsevier.com/locate/cor

Capacitated facility location problem with general setup cost☆

Ling-Yun Wu[a,*], Xiang-Sun Zhang[a], Ju-Liang Zhang[b]

[a] *Institute of Applied Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China*

[b] *Department of Management Science and Engineering, School of Economics and Management, Tsinghua University, Beijing 100080, China*

Abstract

This paper presents an extension of the capacitated facility location problem (CFLP), in which the general setup cost functions and multiple facilities in one site are considered. The setup costs consist of a fixed term (site setup cost) plus a second term (facility setup costs). The facility setup cost functions are generally non-linear functions of the size of the facility in the same site. Two equivalent mixed integer linear programming (MIP) models are formulated for the problem and solved by general MIP solver. A Lagrangian heuristic algorithm (LHA) is also developed to find approximate solutions for this NP-hard problem. Extensive computational experiments are taken on randomly generated data and also well-known existing data (with some necessary modifications). The detailed results are provided and the heuristic algorithm is shown to be efficient.

Keywords: Facility location problem; Lagrangian heuristic; Mixed integer programming

1. Introduction

The facility location problem is a classical, combinatorial optimization problem to determine the number and locations of a set of facilities (warehouses, plants, machines, etc.) and assign customers to these in such a way that the total cost is minimized. Two types of costs are considered in the problem. A *setup cost* (facility cost) occurs while a facility is opened, and a *connection cost* occurs while a customer is

☆ Partially supported by the National Natural Science Foundation of China under Grant #70302003, and National Postdoctoral Foundation of China.

* Corresponding author. Fax: +86 10 62561963.

E-mail address: wlyun@amt.ac.cn (L.-Y. Wu).

0305-0548/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2004.09.012

assigned to the opened facility. For example, considering the locations of warehouses, the total costs may include the fixed costs associated with opening the warehouses and the transportation costs associated with shipping products from warehouses to customers. If an arbitrary number of customers can be connected to a facility, the problem is called uncapacitated facility location problem (UFLP). If each facility has a limit on the number of customers it can serve, it becomes a capacitated facility location problem (CFLP). Both UFLP and CFLP are NP-hard [1] and have been extensively studied. Lots of algorithms, exact and heuristic, have been developed in the past decades [2–11].

Gianpaolo Ghiani et al. [12] dealt with a facility location problem in which several identical facilities can be opened in one site. The classical CFLP heuristics may fail to find a good solution in such a situation due to the inherent difficulty to discriminate among the facilities located in the same site. They formulated the problem as a mixed integer linear programming and developed an efficient Lagrangian heuristic algorithm (LHA).

Hajiaghayi et al. [13] studied an UFLP with general setup cost functions. Their problem was motivated by an application in placing servers on the Internet. The setup cost of a server is a non-decreasing function of the number of clients connected to it. When the number of clients increase, the cost per client will decrease since they share some common expenses. Therefore, the setup cost function exhibits concavity due to economies of scale. This problem was formulated as a generalized UFLP.

The ideas in this paper are motivated by the works of Ghiani et al. and Hajiaghayi et al. Since several facilities are located in one site, it is not realistic to expect that these facilities have the same unit setup cost as the case where a single facility is placed. The facilities in the same site may share common expenses such as management costs, maintenance parts inventory costs. Therefore, the setup costs of a site and facilities at this site depend on the size (i.e., number) of the facilities opened at the site. There are several distinguishing features of the model in this paper: the setup costs consist of a fixed term plus a second term that depends on the size of the facility; the setup cost functions are generally non-linear and of the size of the facility; there are several types of facilities that can be located at the same sites.

There are many research works dealing with the facility location problem using concave or convex production cost functions [14–16]. In those models, the production cost is a concave/convex function of the total output of a site so that the production costs cannot be merged with other linear connection costs like it is done in the classical model. These models resemble the continuous version of the generalized facility location problem with concave or convex setup cost functions proposed in this paper. Actually, if the number of facilities is allowed to be a fraction, it always equals to the total output of the site due to non-decreasing setup cost functions. But in many, if not most, cases of real world, the size of facilities is multiple of some fixed values and represented by a discrete scale. On the other hand, the model in this paper can be treated as a facility location problem with staircase production cost functions.

In this paper, we define and study CFLP with general setup cost functions. In the problem, multiple facilities are located in several sites and provide services to customers. Two types of setup costs are considered: site setup costs and facility setup costs. The site setup cost is a fixed cost associated with opening a site and independent of the facilities located in it. The facility setup cost is a function of its size which is defined as the number of customers served by it in the uncapacitated problem or the number of facilities located in the site in the capacitated problem. The problem is formulated as two mixed integer linear programming (MIP) models and solved by general MIP solver. We also developed a heuristic algorithm based on Lagrangian relaxation to solve the problem. The computational results of the heuristic algorithm and the MIP solver are provided to show the efficiency of the proposed MIP formulations and heuristic algorithm.

This paper is organized as follows. In the next section, we firstly state the UFLP with general cost functions, and then derive the capacitated model. In Section 3, the problem is formulated as two MIP models. A LHA is developed in Section 4 to solve the problem. The computational experiments and results are reported in Section 5. Finally, conclusions are given in Section 6.

2. Generalized facility location problem

2.1. Uncapacitated model

Consider a set of facilities (servers) I and a set of customers (clients) J. Let $g_i(z)$ be a non-decreasing function for each facility i. The facility setup cost $g_i(z_i)$ occurs when facility i is opened with size z_i, i.e., z_i customers are served by it. The connection cost of assigning customer j to facility i is c_{ij}. z_i is a non-negative integer decision variable which denotes the number of customers of facility i. $z_i > 0$ if facility i is open, $z_i = 0$ otherwise; x_{ij} is a binary decision variable which takes the value 1 if the customer j is served by facility i, 0 otherwise.

The UFLP with general cost functions can be formulated as follows:

$$\text{(GUFLP)} \quad \text{Minimize} \quad \sum_{i\in I} g_i(z_i) + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{i\in I} x_{ij} = 1, \quad j \in J, \tag{2}$$

$$\sum_{j\in J} x_{ij} \leqslant z_i, \quad i \in I, \tag{3}$$

$$x_{ij} \in \{0, 1\}, \quad i \in I,\ j \in J, \tag{4}$$

$$z_i \geqslant 0 \text{ integer} \quad i \in I. \tag{5}$$

The setup cost function $g_i(z)$ can be any function. If the setup cost function is convex, i.e., $g_i(z+1) - g_i(z) \geqslant g_i(z) - g_i(z-1)$, the GUFLP can be solved in polynomial time [13]. But the general case is more complicated. Note if

$$g_i(z) = \begin{cases} f_i & \text{if } z > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where f_i is a fixed setup cost, the GUFLP is a classical UFLP which is an NP-hard problem. Therefore, the general case of GUFLP is also NP-hard. In practice, the setup cost function often exhibits concavity due to economies of scale.

2.2. Capacitated model

The GUFLP only considers one type of facility at one site without capacity restriction. The natural extension of the problem is to allow multiple type facilities with and/or without capacity restriction. Consider locating a number of facilities with different types to several sites (locations). If one site is selected, a fixed setup cost occurs, which is independent of the type and number of facilities being installed in it. In addition to the site setup cost, there are facility setup costs which depend on the type and number of facilities placed in the site.

The sites are indexed by $i \in I$ and customers are indexed by $j \in J$. The facility types (demand types) are indexed by $k \in K$. There are various demands d_{jk} at each customer j which can be supplied by the type-k facility. For each type of facility, s_k is the amount of demand that can be satisfied by a single type-k facility. A fixed site setup cost f_i is associated with opening a site at i. A facility can only be located in an open site. The facility setup cost of the type-k facility in site i, g_{ik}, depends on the size (number) of this type facilities z_{ik}. The size of type-k facility in site i cannot exceed a limit u_{ik}. The connection cost of satisfying the type-k demand of customer j by the facilities in site i is c_{ijk}.

Let $y_i = 1$ if site i is opened, and $y_i = 0$ otherwise, z_{ik} be the number of type-k facilities installed in site i, and x_{ijk} be the proportion of type-k demand of customer j served by type-k facilities in site i. Then the generalized facility location problem can be formulated as follows:

$$\text{(GFLP)} \quad \text{Minimize} \quad \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{k\in K} g_{ik}(z_{ik}) + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}x_{ijk} \tag{7}$$

$$\text{subject to} \quad \sum_{i\in I} x_{ijk} = 1, \quad j \in J, k \in K, \tag{8}$$

$$\sum_{j\in J} d_{jk}x_{ijk} \leqslant s_k z_{ik}, \quad i \in I, k \in K, \tag{9}$$

$$z_{ik} \leqslant u_{ik} y_i, \quad i \in I, k \in K, \tag{10}$$

$$\sum_{i\in I} y_i \leqslant p, \tag{11}$$

$$x_{ijk} \geqslant 0, \quad i \in I, j \in J, k \in K, \tag{12}$$

$$z_{ik} \geqslant 0 \text{ integer} \quad i \in I, k \in K, \tag{13}$$

$$y_i \in \{0, 1\}, \quad i \in I. \tag{14}$$

The objective function (7) minimizes the total costs: the setup costs of opened sites, the setup costs of assigning facilities to opened sites, the costs of serving customers by the facilities in opened sites. Constraints (8) ensure that the demand of each customer is satisfied; constraints (9) express that the service provided by a site cannot exceed its capacity; constraints (10) state that no facility can be located in a closed site, and the number of facilities in an opened site cannot be larger than allowed; constraint (11) establishes that the total number of opened sites is at most p. Constraints (13) and (14) are the integrality constraints; constraints (12) give lower bounds of continuous variables x_{ijk}.

By allowing $d'_{jk} = d_{jk}/s_k$ and $s'_k = 1$, and substituting d_{jk}, s_k in constraints (9) with d'_{jk}, s'_k, respectively, the problem will be equivalent to the original one. Therefore, without loss of generality, we will take $s_k = 1$ for all $k \in K$ in the rest of this paper.

The GUFLP is a special case of GFLP with $|K| = 1$, $f_i = 0$ and $d_{jk} = 1$. Obviously, GFLP (even with convex or concave setup cost functions) is an NP-hard problem since the classical NP-hard problem CFLP is a special case of GFLP with $g_{ik} = 0$.

A large number of real-world situations can be satisfactorily modelled as GFLP. One of the examples is an application in the supply chain network design and is described as follows. The problem is to select a set of suppliers from an available supplier network of internationally located potential suppliers to satisfy the demand of a set of customers. In the classical supply chain network design problem, each potential supplier is considered as a facility with a fixed capacity and a fixed setup cost. However, in the real life the purchasing price often depends on the purchasing size. Each possible size yields a certain price and the price function is often a step function associated with economies of scale. Consequently, the total setup cost of selecting a supplier consists of a fixed part and a variable part which depends on the purchasing size.

This supply chain network design problem can be modelled as a GFLP by letting I be the set of potential suppliers; J the set of customers; K the type of products; f_i the fixed cost to develop business relationship with supplier i; $g_{ik}(z)$ the variable setup cost for purchasing type-k product from supplier i with capacity z; c_{ijk} the total costs (linear part of purchasing cost, transportation cost and taxes, etc.) for shipping unit type-k product from supplier i to customer j; d_{jk} the demand of type-k product at customer j; u_{ik} the maximal potential production size of the type-k product at supplier i; p the maximal number of suppliers the company would like to develop.

3. MIP formulations

Let

$$g_{ikl} = g_{ik}(l), \quad 1 \leqslant l \leqslant u_{ik}. \tag{15}$$

Define $z_{ikl} = 1$ if $l = z_{ik}$ and 0 otherwise. Then the GFLP can be formulated as a MIP problem:

$$\text{(GFLP-MIP1)} \quad \text{Minimize} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{k \in K} \sum_{1 \leqslant l \leqslant u_{ik}} g_{ikl} z_{ikl} + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \tag{16}$$

$$\text{subject to} \quad \sum_{i \in I} x_{ijk} = 1, \quad j \in J, k \in K, \tag{17}$$

$$\sum_{j \in J} d_{jk} x_{ijk} \leqslant \sum_{1 \leqslant l \leqslant u_{ik}} l z_{ikl}, \quad i \in I, k \in K, \tag{18}$$

$$\sum_{1 \leqslant l \leqslant u_{ik}} z_{ikl} \leqslant y_i, \quad i \in I, k \in K, \tag{19}$$

$$\sum_{i \in I} y_i \leqslant p, \tag{20}$$

$$x_{ijk} \geqslant 0, \quad i \in I, j \in J, k \in K, \tag{21}$$

$$z_{ikl} \in \{0, 1\}, \quad i \in I, k \in K, l = 1, 2, \ldots, u_{ik}, \tag{22}$$

$$y_i \in \{0, 1\}, \quad i \in I. \tag{23}$$

The objective function (16) and constraints (17), (18), (20) are equivalent to (7)–(9), (11), respectively. Constraints (19) state that facility can only be located in an opened site and only in one size among all choices. Constraints (23) and (22) are the integrality constraints; constraints (21) provide lower bounds of variables x_{ijk}.

An alternative MIP formulation is given by defining

$$g'_{ik1} = g_{ik}(1), \tag{24}$$

$$g'_{ikl} = g_{ik}(l) - g_{ik}(l-1), \quad 2 \leqslant l \leqslant u_{ik}, \tag{25}$$

i.e., g'_{ikl} is the increasing setup cost of the lth type-k facility in site i. Define $z_{ikl}=1$ if $l \leqslant z_{ik}$ and $z_{ikl}=0$ if $l > z_{ik}$. Then the GFLP can also be formulated as the following MIP problem:

$$
\begin{aligned}
\text{(GFLP-MIP2)} \quad \text{Minimize} \quad & \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{k\in K}\sum_{1\leqslant l\leqslant u_{ik}} g'_{ikl} z_{ikl} + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}x_{ijk} && (26)\\
\text{subject to} \quad & \sum_{i\in I} x_{ijk} = 1, \quad j\in J, k\in K, && (27)\\
& \sum_{j\in J} d_{jk}x_{ijk} \leqslant \sum_{1\leqslant l\leqslant u_{ik}} z_{ikl}, \quad i\in I, k\in K, && (28)\\
& z_{ikl}\leqslant y_i, \quad i\in I, k\in K, l=1,2,\ldots,u_{ik}, && (29)\\
& z_{ikl}\geqslant z_{ikl+1}, \quad i\in I, k\in K, l=1,2,\ldots,u_{ik}-1, && (30)\\
& \sum_{i\in I} y_i \leqslant p, && (31)\\
& x_{ijk}\geqslant 0, \quad i\in I, j\in J, k\in K, && (32)\\
& z_{ikl}\in\{0,1\}, \quad i\in I, k\in K, l=1,2,\ldots,u_{ik}, && (33)\\
& y_i\in\{0,1\}, \quad i\in I. && (34)
\end{aligned}
$$

The objective function (26) and constraints (27), (28), (31) are equivalent to (7) and (8), (9), (11), respectively. Constraints (29) state that facility can only be located in an opened site. Constraints (30) ensure that facilities are located in correct order, i.e., the l'th type-k facility can only be opened in a site after all lth type-k facilities with $l<l'$ are located. Constraints (34) and (33) are the integrality constraints; constraints (32) provide lower bounds of variables x_{ijk}. Note that in the case of convex setup cost functions, $g'_{ikl}\leqslant g'_{ikl+1}$. Therefore, constraints (30) are redundant and can be removed.

4. Lagrangian heuristic algorithm

The Lagrangian relaxation has been successfully used in various optimization problems as well as in CFLP computation [17–24]. Based on extensive comparison of several algorithms to solve CFLP, Cornuejelos et al. [25] suggested that the Lagrangian heuristic is an efficient algorithm to solve large-scale instances of the CFLP. The basic idea behind a Lagrangian heuristic is the computation of a lower bound through a Lagrangian relaxation. The procedure is usually encapsulated in a subgradient optimization algorithm, on the basis of generating a sequence of Lagrangian multipliers in order to achieve the highest lower bound as possible. At each iteration of the subgradient method, a feasible solution of the original problem can be constructed from the solution of Lagrangian relaxation by a heuristic algorithm. If this feasible solution is better than those found before, the upper bound is improved. The availability of both lower and upper bounds on the optimal objective function value is an attractive feature of the Lagrangian heuristic research. The relative gap between upper and lower bounds is typically used as a measure of the maximal error of the solution.

4.1. The lower bound estimation

We relax constraints (8) in GFLP with multipliers $\lambda_{jk},\ j \in J, k \in K$. In order to tighten the bound of relaxation, the following redundant constraints are imposed into the original model:

$$x_{ijk} \leqslant 1, \quad i \in I, j \in J, k \in K. \tag{35}$$

Then we get a Lagrangian relaxation of the GFLP:

$$\begin{aligned}\text{(LR)} \quad \text{Minimize} \sum_{i \in I} f_i y_i &+ \sum_{i \in I} \sum_{k \in K} g_{ik}(z_{ik}) + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \\ &+ \sum_{j \in J} \sum_{k \in K} \lambda_{jk} \left(\sum_{i \in I} x_{ijk} - 1 \right)\end{aligned} \tag{36}$$

subject to (9)–(14) and (35).

Leaving constraints (11) aside, we can decompose problem (LR) into $|I|$ subproblems, one for each site.

$$(\text{LR}_i) \quad \text{Minimize} \quad f_i y_i + \sum_{k \in K} g_{ik}(z_{ik}) + \sum_{j \in J} \sum_{k \in K} (c_{ijk} + \lambda_{jk}) x_{ijk} \tag{37}$$

subject to (9) (10) (12)–(14) and (35). For each subproblem, y_i is either equal to 0 or 1. If y_i is equal to 0, z_{ik} and x_{ijk} is equal to 0 for all i, j and k. Then the objective function (37) is equal to 0 too. If y_i is equal to 1, the subproblem (LR$_i$) can be further decomposed into $|K|$ subproblems, one for each facility type.

$$(\text{LR}_{ik}) \quad \text{Minimize} \quad g_{ik}(z_{ik}) + \sum_{j \in J} (c_{ijk} + \lambda_{jk}) x_{ijk} \tag{38}$$

$$\text{subject to} \quad \sum_{j \in J} d_{jk} x_{ijk} \leqslant z_{ik}, \tag{39}$$

$$0 \leqslant z_{ik} \leqslant u_{ik} \quad \text{integer}, \tag{40}$$

$$0 \leqslant x_{ijk} \leqslant 1, \quad j \in J. \tag{41}$$

For each fixed z_{ik}, the subproblem LR$_{ik}$ is a fractional knapsack problem that can be solved by the greedy algorithm. The demands of the customers are satisfied according to the non-decreasing order of negative $(c_{ijk} + \lambda_{jk})/d_{jk}$, until reaching the temporary capacity z_{ik}. When the temporary capacity is increased from 1 to u_{ik} in steps, the minimal value of the object function is recorded in Z^* and the best solutions are stored in x^*_{ijk} and z^*_{ik}. In detail, we can solve the subproblem LR$_{ik}$ by the following procedure. The variable τ records the remaining available capacity with the current value of z_{ik} and θ holds the unsatisfied demand of the current selected customer.

Step 1: Let L be a list of all $j \in J$ such that $(c_{ijk} + \lambda_{jk})/d_{jk} < 0$, and sorted in non-decreasing order. Set $x^*_{ijk} := 0$, $j \in J$, $z^*_{ik} := 0$ and $Z^* := 0$. Let $x_{ijk} := 0$, $j \in J$, $z_{ik} := 1$, $Z := 0$ and $\tau := 1$.

Step 2: If $L \neq \emptyset$, then select and remove the j with smallest value of $(c_{ijk} + \lambda_{jk})/d_{jk}$ from L, let $\theta := 1$; else go to Step 5.

Step 3: If $d_{jk}\theta < \tau$, let $\Delta x_{ijk} := \theta$, $\theta := 0$ and $\tau := \tau - d_{jk}\Delta x_{ijk}$; else let $\Delta x_{ijk} := \tau/d_{jk}$, $\theta := \theta - \Delta x_{ijk}$ and $\tau := 0$.

Step 4: Let $x_{ijk} := x_{ijk} + \Delta x_{ijk}$ and $Z := Z + (c_{ijk} + \lambda_{jk})\Delta x_{ijk}$. If $\tau > 0$ go to Step 2.

Step 5: If $Z + g_{ik}(z_{ik}) < Z^*$, update $x^*_{ijk} := x_{ijk}$, $z^*_{ik} := z_{ik}$ and $Z^* := Z + g_{ik}(z_{ik})$. If $z_{ik} \geqslant u_{ik}$ or $L = \emptyset$, go to Step 7.

Step 6: Set $z_{ik} := z_{ik} + 1$ and let $\tau := 1$. If $\theta = 0$, go to Step 2; else go to Step 3.

Step 7: STOP. x^*_{ijk} and z^*_{ik} are the optimal solution and the minimal objective function value is Z^*.

Denote Z^*_{ik} as the minimal objective function value of the subproblem (LR$_{ik}$), then the minimal objective function value of the subproblem (LR$_i$) is

$$Z^*_i := \min\left\{ f_i + \sum_{k\in K} Z^*_{ik}, 0 \right\}. \tag{42}$$

Constraints (11) then can be enforced by choosing the p best (minimal) solutions obtained after solving the $|I|$ subproblems (LR$_i$). The solution to the Lagrangian relaxation (LR) provides a lower bound to the original problem.

4.2. The upper bound computation

By solving the Lagrangian relaxation problem, we obtain a set of open sites I^* and the numbers of each type facility in these sites, z^*_{ik}. Then we solve the following linear programming problem to find a feasible solution of the GFLP.

$$\text{(GFLP-FS}(I^*, z^*_{ik})) \quad \text{Minimize} \quad \sum_{i\in I^*}\sum_{j\in J}\sum_{k\in K} c_{ijk}x_{ijk} \tag{43}$$

$$\text{subject to} \quad \sum_{i\in I^*} x_{ijk} = 1, \quad j \in J, k \in K, \tag{44}$$

$$\sum_{j\in J} d_{jk}x_{ijk} \leqslant z^*_{ik}, \quad i \in I^*, k \in K, \tag{45}$$

$$x_{ijk} \geqslant 0, \quad i \in I^*, j \in J, k \in K. \tag{46}$$

The optimal solution found is an upper bound on the original problem. If it is smaller than the existing upper bound, the upper bound is improved.

The solution I^* and z^*_{ik} through solving the LR problem may not always be feasible to the original problem. In other words, the total capacity defined by the set of open sites I^* and the numbers of each type facility in these sites z^*_{ik} is smaller than the total demand. Then before the GFLP-FS problem is solved by using the linear programming solver of CPLEX, a simple greedy heuristic is applied to adjust the I^* and z^*_{ik} so that all demands can be satisfied. Suppose that the total capacity of type-k facility is insufficient. This greedy heuristic simply calculates the values

$$\sigma_{ik} := \begin{cases} g_{ik}(z^*_{ik}+1) & \text{if } y^*_i = 1, \\ g_{ik}(z^*_{ik}+1) + f_i & \text{if } y^*_i = 0, \end{cases} \tag{47}$$

for the site i with $z^*_{ik} < u_{ik}$, and select the smallest one, say i'. Then increase $z^*_{i'k}$ by one. If $y^*_{i'} = 0$, let $y^*_{i'} = 1$. The procedure is repeated until the total capacity of type-k facility is large enough.

4.3. Subgradient algorithm

After the LR problem is solved in each iteration, we need to update the Lagrangian multipliers. There are many methods to solve the Lagrangian dual problem, i.e., to find the optimal Lagrangian multipliers. We implemented the classical (simple) subgradient method [26,28] and the r-algorithm proposed by Shor [27].

The classical subgradient method is as follows. First, the subgradients are calculated:

$$\psi_{jk} := \sum_{i \in I} x_{ijk} - 1, \quad j \in J, k \in K. \tag{48}$$

Then a step size μ is defined by

$$\mu := \frac{\sigma(UB - LB)}{\sum_{j,k} \psi_{jk}^2}, \tag{49}$$

where *UB* is the objective value of the best-known feasible solution and *LB* is the best lower bound found by solving the Lagrangian relaxation problem. σ is a scalar with initial value between 0 and 2, and reduced by a factor whenever the solution of Lagrangian relaxation problem fails to increase in a specified number of iterations. The classical subgradient method updates the Lagrangian multipliers λ_{jk} by

$$\lambda_{jk} := \lambda_{jk} + \psi_{jk} * \mu. \tag{50}$$

The r-algorithm is a modified subgradient method which employs a space dilation strategy in the direction of the difference between two successive subgradients and is recognized as being one of the most effective procedures for solving non-differentiable optimization problems. For the details of r-algorithm, we refer the interested readers to [27]. Extensive computational experiments show that the r-algorithm converges much faster than the classical subgradient method and obtains better quality solution in this problem.

The complete proposed LHA then is illustrated as follows:

Step 1: Set $LB := -\infty$, $UB := +\infty$, $GAP := +\infty$. Let $t := 0$. Initialize the Lagrangian multipliers by setting $\lambda_{jk}^t := 0$, $j \in J, k \in K$. Given termination criteria *MAXITER* and ε.

Step 2: Solve LR($\boldsymbol{\lambda}^t$). Denote the optimal objective value as *LR*. If $LR > LB$, then set $LB := LR$.

Step 3: Get the sets of open facilities I^* and the size of each type facility in these sites, z_{ik}^*, from the optimal solution of LR($\boldsymbol{\lambda}^t$). Solve the corresponding linear programming problem (GFLP-FS(I^*, z_{ik}^*)) to obtain a feasible solution, and denote the optimal objective value as *FS*. If $FS < UB$, then set $UB := FS$. If $(UB - LB)/LB \leqslant GAP$, then set $GAP := (UB - LB)/LB$.

Step 4: If either $t > MAXITER$ or $GAP < \varepsilon$ then *STOP*.

Step 5: Update the Lagrangian multipliers. Set $t := t + 1$. Return to Step 2.

5. Computational results

Extensive computational experiments were taken to test the proposed MIP formulations and LHA. The algorithm was implemented in C++ and used the LP solver of CPLEX 7.1 to solve the subproblem GFLP-FS in order to find a feasible solution in each iteration. In the preliminary experiments, we implemented two methods to solve the Lagrangian dual problem: the classical subgradient method [26] and the r-algorithm proposed by Shor [27]. The r-algorithm performed better both in terms of solution quality and run time, therefore, it was used in the final experiments. We also solved the equivalent MIP formulations by using the MIP solver of CPLEX 7.1, which is one of the best commercial MIP solvers based on branching and bounding technique. The parameters of CPLEX solver were set at their defaults except that the presolve, the cuts generation and the periodic heuristic were turned off, i.e., only the pure branch and bound algorithm were adopted. We also tried different branching strategies, but they are not so good as the strategy automatically determined by CPLEX. All experiments were run on a 2.5 GHz Pentium 4 processor using Microsoft Visual C++ compiler 6.

5.1. Experiments on randomly generated data

Two sets of test problems using concave and convex setup cost functions are randomly generated by systematically simulating the real-world situations. The fixed site setup costs f_i are generated from a uniform distribution between 50 and 100. The zth type-k facility setup cost in site i uses the following function:

$$(b_k + v_{ik})(\rho_{ik})^{\log_2 z}, \tag{51}$$

where b_k is the basic unit setup cost of type-k facility and is generated from a uniform distribution between 10 and 20, v_{ik} is the deviation of the basic unit setup cost of type-k facility at site i and generated from a uniform distribution between 0 and 5, and ρ_{ik} is the changing rate of unit setup cost of type-k facility at site i when the size of facility is doubled. $\rho_{ik} < 1$ results in concave setup cost function while $\rho_{ik} > 1$ leads to convex setup cost function. The two sets of test problems in our experiments are generated by using $\rho_{ik} = 0.9$ (concave) and 1.1 (convex), respectively.

The locations of site and customer are randomly generated in a plane. Then the cost efficiencies c_{ijk} are generated by the Euclidean distance between location i and j and multiplied by a factor associated with each type facility. The demands at each customer are drawn from a uniform distribution between 50 and 100, then are adjusted according to the total capacity so that the problem is feasible.

The extensive comparison of the results on the problems with concave and convex setup cost functions obtained by the LHA and the MIP solver of CPLEX are reported in Tables 1 and 2, respectively. The gap between the best feasible solution and the lower bound found by heuristic algorithm is used to judge the quality of the solution. The maximal iteration number of the LHA is 1000. The algorithm stops if a feasible solution with the gap smaller than 0.1% is found or the maximal iteration number is reached. The CPLEX algorithm uses the same stop criteria of the solution quality or terminates if the run time exceeds 2000 seconds.

The Lagrangian heuristic reports very good solutions on all instances in terms of solution quality and the computing times. The gap of solutions is smaller than 1% on all the instances with concave setup cost functions and 3% on all the convex instances. Among the results of three methods, Lagrangian heuristic obtains the best solutions on 11 instances (marked by bold font in the tables) while the solutions of other instances are also very close to the best solutions found by the other methods. The relative deviations from the best found solutions are within 2% on all instances.

The Lagrangian heuristic provides better lower bound on almost all instances, compared with the best one obtained from the LP relaxations at the partial tree that is explored by CPLEX. The GAP reported by CPLEX is often worse than that by Lagrangian heuristic, even if the solution found by CPLEX is better.

Table 1
Comparison on GFLP with concave setup cost functions

NS	NF	NC	MS	MF	Lagrangian Heuristic						CPLEX - MIP1						CPLEX - MIP2					
					Iter	Time	Solution	LB	GAP	DIV	Iter	Time	Solution	LB	GAP	DIV	Iter	Time	Solution	LB	GAP	DIV
10	5	50	5	10	1000	5.719	4326.11	4298.41	0.64	0.58	1409200	2000	**4301.22**	4294.02	0.17	0.00	442500	1223	4303.98	4299.67	0.10	0.06
10	5	50	5	20	1000	11.515	8061.34	8033.65	0.34	0.15	436600	602[a]	**8049.65**	7964.18	1.06	0.00	538600	2000	8189.50	7996.31	2.36	1.74
10	5	50	5	50	215	2.11	**18857.95**	18841.46	0.09	0.00	361900	690[a]	18858.40	18581.19	1.47	0.00	293000	2000	19096.49	18671.44	2.23	1.26
10	5	100	5	10	1000	15.047	4103.95	4092.43	0.28	0.25	411600	1307	**4093.76**	4089.66	0.10	0.00	105000	615	4094.34	4090.24	0.10	0.01
10	5	100	5	20	1000	24.875	7685.38	7662.03	0.30	0.22	244500	611[a]	**7668.36**	7588.63	1.04	0.00	253200	2000	7844.80	7601.96	3.10	2.30
10	5	100	5	50	149	3.625	**17884.64**	17870.51	0.08	0.00	224800	676[a]	17884.99	17620.56	1.48	0.00	167900	2000	18189.70	17680.21	2.80	1.71
10	10	50	5	10	1000	22.985	8009.70	7981.56	0.35	0.23	871600	2000	**7990.99**	7890.91	1.25	0.00	341800	2000	8595.21	7883.33	8.28	7.56
10	10	50	5	20	1000	26.859	15209.68	15148.79	0.40	0.20	199600	547[a]	**15179.96**	14916.40	1.74	0.00	395600	2000	16501.72	14898.83	9.71	8.71
10	10	50	5	50	234	5.406	**35877.77**	35843.46	0.10	0.00	180000	602[a]	35929.03	35102.91	2.30	0.14	106800	2000	38471.79	34727.69	9.73	7.23
10	10	100	5	10	1000	76.406	7953.03	7876.90	0.96	0.82	396300	2000	**7888.44**	7763.67	1.58	0.00	101600	2000	8114.81	7777.76	4.15	2.87
10	10	100	5	20	1000	74.437	14963.24	14910.72	0.35	0.20	118700	755[a]	**14933.74**	14590.41	2.30	0.00	75900	2000	15476.62	14672.39	5.20	3.64
10	10	100	5	50	1000	75.266	35261.37	35213.26	0.14	0.07	112800	863[a]	**35237.93**	34419.76	2.32	0.00	87200	2000	35944.59	34396.92	4.31	2.01
20	5	50	5	10	1000	5.219	4147.05	4108.04	0.94	0.92	435200	1204	**4109.31**	4105.20	0.10	0.00	307800	2000	4124.42	4101.17	0.56	0.37
20	5	50	5	20	1000	6.922	7695.33	7670.11	0.33	0.20	315100	916[a]	**7679.76**	7621.89	0.75	0.00	269800	2000	7896.11	7600.79	3.74	2.82
20	5	50	5	50	1000	7.125	15458.07	15440.39	0.11	0.01	272100	1074[a]	**15456.53**	15277.22	1.16	0.00	32600	2000	19327.38	14820.18	23.32	25.04
20	5	100	5	10	1000	28.047	4115.40	4097.77	0.43	0.30	226100	1134[a]	**4103.10**	4067.01	0.88	0.00	87600	2000	4196.87	3967.50	5.47	2.29
20	5	100	5	20	1000	26.187	7608.70	7595.56	0.17	0.02	198800	1077[a]	**7607.20**	7497.80	1.44	0.00	47700	2000	9443.62	7043.74	25.41	24.14
20	5	100	5	50	171	3.813	**17568.76**	17551.60	0.10	0.00	239100	1616[a]	17612.70	17228.17	2.18	0.25	14700	2000	19295.29	15853.48	17.84	9.83
20	10	50	5	10	1000	26.532	7030.79	6979.99	0.72	0.42	158200	832[a]	**7001.69**	6834.48	2.39	0.00	152600	2000	8499.25	6845.34	19.46	21.39
20	10	50	5	20	1000	28.719	13129.78	13093.90	0.27	0.18	147700	974[a]	**13105.96**	12818.41	2.19	0.00	36400	2000	15365.52	12740.98	17.08	17.24
20	10	50	5	50	1000	27.531	30700.30	30627.43	0.24	0.16	142500	1495[a]	**30650.71**	29788.94	2.81	0.00	6400	2000	35076.23	26800.37	23.59	14.44
20	10	100	5	10	1000	79.218	7757.08	7696.17	0.79	0.48	109200	1558[a]	**7719.76**	7572.78	1.90	0.00	21000	2000	9810.38	7342.52	25.16	27.08
20	10	100	5	20	1000	76.156	14615.07	14563.13	0.36	0.06	97900	1408[a]	**14605.61**	14268.79	2.31	0.00	12000	2000	17411.00	13835.85	20.53	19.21
20	10	100	5	50	1000	76.375	**34382.09**	34317.69	0.19	0.00	94600	1779[a]	34408.44	33477.92	2.70	0.08	6400	2000	41248.24	29826.14	27.69	19.97

NS—the number of sites (location); NF—the numberof facility types; NC—the number of customers; Iter—the iteration number; Solution—the best solution found by the algorithm; GAP(%)—(upper bound - lower bound)/lower bound $\times 100$; MS—the maximal number of desired sites; MF—the maximal capacity of facilities; Time—run time (in seconds); LB—the lower bound found by the algorithm; DIV(%)—the relative deviation from the best solution of three algorithms.

[a]These instances terminate earlier than desired due to out of memory.

Table 2
Comparison on GFLP with convex setup cost functions

NS	NF	NC	MS	MF	Lagrangian heuristic						CPLEX - MIP1						CPLEX - MIP2					
					Iter	Time	Solution	LB	GAP	DIV	Iter	Time	Solution	LB	GAP	DIV	Iter	Time	Solution	LB	GAP	DIV
10	5	50	5	10	1000	8.203	4350.04	4296.76	1.22	1.01	32758	38	**4306.52**	4302.98	0.08	0.00	1403	10	4306.93	4303.49	0.08	0.01
10	5	50	5	20	1000	5.125	8489.49	8469.17	0.24	0.14	122000	147	8480.39	8471.90	0.10	0.03	2730	34	**8477.58**	8475.37	0.03	0.00
10	5	50	5	50	1000	5.797	20303.11	20241.72	0.30	0.28	351700	523[a]	21019.57	20237.92	3.72	3.81	1480	154	**20247.18**	20241.39	0.03	0.00
10	5	100	5	10	1000	15.813	5489.30	5411.35	1.42	1.38	4400	13	**5414.53**	5409.09	0.10	0.00	454	9	5414.81	5410.72	0.08	0.01
10	5	100	5	20	1000	19.422	10757.99	10735.41	0.21	0.18	158104	359	**10738.76**	10733.16	0.05	0.00	300	24	10739.88	10689.54	0.47	0.01
80	5	100	5	50	1000	17.219	27403.04	27366.29	0.13	0.12	233400	577[a]	27459.37	27360.75	0.36	0.33	1447	251	**27369.67**	27361.51	0.03	0.00
10	10	50	5	10	1000	24.985	8943.46	8887.25	0.63	0.57	612037	1221	**8892.73**	8890.09	0.03	0.00	1817	38	8894.04	8889.23	0.05	0.01
10	10	50	5	20	1000	25.5	18009.00	17895.78	0.63	0.57	197300	535[a]	18025.56	17868.23	0.87	0.66	26393	448	**17906.71**	17902.84	0.02	0.00
10	10	50	5	50	1000	25.234	46181.01	46120.50	0.13	0.10	188600	620[a]	46236.49	46090.51	0.32	0.22	17181	1035	**46134.99**	46127.68	0.02	0.00
10	10	100	5	10	1000	77.765	8428.93	8356.12	0.86	0.75	389100	2000	8585.12	8353.33	2.70	2.62	17000	250	**8366.08**	8357.71	0.10	0.00
10	10	100	5	20	1000	62.812	16934.33	16868.97	0.39	0.37	119600	716[a]	16929.83	16864.70	0.38	0.34	5147	256	**16871.74**	16866.00	0.03	0.00
10	10	100	5	50	1000	72.563	43690.88	43642.51	0.11	0.09	112500	803[a]	43817.71	43638.96	0.41	0.38	444	501	**43652.28**	43638.02	0.03	0.00
20	5	50	5	10	1000	9.187	4603.17	4517.37	1.86	1.73	23500	83	4525.26	4520.74	0.10	0.00	7287	126	**4525.04**	4522.95	0.05	0.00
20	5	50	5	20	1000	10.516	8995.19	8966.27	0.32	0.22	293500	936[a]	9251.15	8962.38	3.12	3.07	25376	1472	**8975.58**	8967.45	0.09	0.00
20	5	50	5	50	1000	6.453	23058.01	22981.52	0.33	0.04	226300	2000	**23048.63**	22984.28	0.28	0.00	6600	2000	23657.06	22533.42	4.75	2.64
20	5	100	5	10	1000	29.328	4675.87	4570.74	2.25	1.99	171396	934	**4584.86**	4583.76	0.02	0.00	8900	432	4586.73	4582.06	0.10	0.04
20	5	100	5	20	1000	30.828	9256.57	9076.32	1.95	1.23	62600	2000	**9144.40**	9081.99	0.68	0.00	12200	2000	9322.15	8539.77	8.39	1.94
20	5	100	5	50	1000	25.547	23807.70	23208.58	2.52	1.60	22000	2000	24241.05	21315.25	12.07	3.45	3400	2000	**23433.74**	21865.15	6.69	0.00
20	10	50	5	10	1000	28.484	**8068.68**	8009.36	0.74	0.00	159200	1488[a]	8493.95	8003.31	5.78	5.27	20300	2000	8111.51	7976.04	1.67	0.53
20	10	50	5	20	1000	29	**16260.84**	16204.31	0.35	0.00	153300	2000	16328.64	16203.61	0.77	0.42	6400	2000	17285.99	15302.43	11.47	6.30
20	10	50	5	50	1000	25.735	**42196.50**	42045.67	0.36	0.00	10800	2000	44479.09	38445.72	13.56	5.41	2100	2000	44390.65	38434.74	13.42	5.20
20	10	100	5	10	1000	77.938	**8235.99**	8129.91	1.29	0.00	103900	1780[a]	8313.73	8127.79	2.24	0.94	7700	2000	8381.29	7908.96	5.64	1.76
20	10	100	5	20	1000	75.594	**16420.30**	16361.09	0.36	0.00	9600	2000	17753.85	15362.26	13.47	8.12	6100	2000	16722.63	15447.97	7.62	1.84
20	10	100	5	50	1000	69.032	**42413.82**	42278.47	0.32	0.00	5600	2000	44811.38	37865.27	15.50	5.65	1300	2000	45342.24	38655.02	14.75	6.90

NS—the number of sites (location); NF—the number of facility types; NC—the numberof customers; Iter—the iteration number; Solution—the best solution found by the algorithm; GAP(%)—(upper bound - lower bound)/lower bound ×100; MS—the maximal number of desired sites; MF—the maximal number of desired sites; MF—the maximal capacity of facilities; Time—run time (in seconds); LB—the lower bound found by the algorithm; DIV(%)—the relative deviation from the best solution of three algorithms.

[a]These instances terminate earlier than desired due to out of memory.

This is due to the poor lower bound of linear relaxation used by the CPLEX. On the other hand, the small GAP reported by Lagrangian heuristic shows that the lower bound of Lagrangian relaxation for GFLP is quite tight in most instances.

Another observation is that the GAP obtained by CPLEX using both MIP formulation seems larger on the concave problems than on the convex problems. This is consistent with the fact that in the uncapacitated case the problem with convex setup costs can be solved in polynomial time. But, interestingly, the results of Lagrangian heuristic do not show this difference, which may deserve further research.

As observed, in terms of solution quality, CPLEX solver using MIP formulation 1 performs better than that using MIP 2 on the concave cases, and vise versa on the convex cases. CPLEX using MIP 1 provides the best solutions on 26 instances while CPLEX using MIP 2 only wins on 11 instances. But it seems that CPLEX using MIP 1 consumed much more computer memory than CPLEX using MIP 2. In all 48 instances, CPLEX using MIP 1 terminates on 28 instances due to out of memory before the stop criteria is satisfied. Despite the best performance of CPLEX using MIP 2 on the most convex problems, its overall performance is the poorest. The solution qualities of CPLEX using MIP 2 on the concave problems are worse than that of the other two methods, both in terms of the reported GAP and the relative deviation from the best solutions.

The size of problems used in our experiments are still modest. The larger the size of problem, the worse the MIP solver performs. The results of large size convex instances are evidential. Even for the small size problem, the time taken by MIP solver to obtain the same quality solution is 10 to 100 times slower than the Lagrangian heuristic. Therefore, Lagrangian heuristic is more practical when only near-optimal solutions are required. Note that the comparison between Lagrangian heuristic and CPLEX is not entirely fair since the former is an approximation algorithm and the latter is used to obtain exact optimal solutions. The options of CPLEX are not the best since the results of CPLEX provided here are mainly for measuring the performance of two MIP formulations.

5.2. Experiments on existing data

Another set of test problems was generated from the well-known data set used by Beasley [3]. The general setup costs had not been considered before for the CFLP. Beasley's data set was modified in the following way. The number of facility type $|K| = 1$ and the cardinality constraints were released by set p equal to the number of warehouse $|I|$. Each site (warehouse) consists of 10 facilities with equal capacities, i.e., the size of warehouse can be selected from 10 levels. The capacity of the warehouse, the demands of customers and the supply costs were scaled in order to see that the highest capacity of the warehouse is equal to 10. Then the facility setup costs g_i are added to Beasley's data set. The zth facility setup cost of the warehouse i is generated by $0.9^{\log_2 z}$ and then scaled so that the total facility setup cost $g_i(10) = \sum_{z=1}^{10} 0.9^{\log_2 z}$ is 20% of the original setup cost. The new fixed site setup cost of warehouse i, f_i is reduced to 80% of the original setup cost.

The computational results of the Lagrangian heuristic on the modified Beasley's data are given in Table 3. The Lagrangian heuristic reports very good solutions on almost all instances in terms of solution quality and the computing times. The gap of solutions are smaller than 1% on many instances except for seven instances, and 4% on all instances. The computing times are less than 4 s on all data.

It is interesting to compare the results of the modified problems with the results of the original warehouse location problems. The results of the original problems are provided in the right most column of Table 3. Except for few instances, the number of open warehouses in the modified problems are greater than

Table 3
The results on Beasley's data

Problem	NW	Iter	Time	Solution	LB	GAP	NOW	NOW-B
IV-1	16	1000	1.000	1045025.21	1023474.93	2.06	12	13
-2	16	1000	1.406	1101096.82	1088408.57	1.15	13	12
-3	16	1000	2.953	1160228.74	1142606.40	1.52	13	12
-4	16	1000	1.125	1240913.03	1216124.58	2.00	12	12
V-1	16	1000	0.750	1017596.41	1016125.91	0.14	9	8
VI-1	16	223	0.672	923506.42	922764.19	0.08	12	11
-2	16	239	0.625	966540.29	965615.40	0.10	10	9
-3	16	238	0.609	1001700.74	1000828.11	0.09	8	7
-4	16	310	0.703	1040945.33	1039926.30	0.10	5	5
VII-1	16	218	0.500	919806.62	919195.31	0.07	12	11
-2	16	231	0.453	960354.50	959836.16	0.05	10	9
-3	16	237	0.453	994350.58	993865.39	0.05	8	5
-4	16	255	0.515	1024396.91	1023805.64	0.06	4	4
VIII-1	25	1000	1.235	832822.68	829518.75	0.40	18	17
-2	25	1000	1.469	907702.82	903325.19	0.48	16	14
-3	25	1000	1.625	975262.35	966758.71	0.87	15	14
-4	25	1000	3.469	1089292.83	1061400.31	2.56	14	13
IX-1	25	245	0.922	781786.33	781498.02	0.04	17	15
-2	25	1000	1.109	839901.15	839044.12	0.10	12	11
-3	25	1000	0.703	882372.86	880998.80	0.16	9	8
-4	25	1000	0.922	939154.48	928444.44	1.14	8	7
X-1	25	216	0.672	777660.95	776961.37	0.09	17	15
-2	25	220	0.688	833200.64	832480.05	0.09	12	11
-3	25	202	0.500	873546.80	872854.33	0.08	9	8
-4	25	187	0.360	912019.74	911363.41	0.07	6	4
XI-1	50	432	1.579	819738.07	819212.09	0.06	17	17
-2	50	578	1.985	892772.27	891995.42	0.09	16	15
-3	50	1000	1.656	961589.70	952415.30	0.95	15	14
-4	50	1000	2.641	1081939.23	1048834.62	3.06	12	13
XII-1	50	243	1.015	778970.04	778760.67	0.03	16	15
-2	50	670	0.937	836616.62	835759.63	0.10	12	11
-3	50	337	1.266	879181.79	878417.18	0.09	9	9
-4	50	1000	1.203	935278.02	928365.13	0.74	9	7
XIII-1	50	221	0.765	774823.52	774566.87	0.03	16	15
-2	50	237	0.750	829363.30	829363.30	0.00	12	11
-3	50	277	0.922	870946.99	870844.39	0.01	10	8
-4	50	270	0.750	912019.74	911994.41	0.00	6	4

Iter—the iteration number; time—run time (in seconds); Solution—the best solution found by the algorithm; LB—the lower bound found by the algorithm; GAP(%)—(upper bound−lower bound)/lower bound$\times$100; NW—Number of potential warehouses (sites); NOW—Number of open warehouses (sites) in the results of this paper; NOW-B—Number of open warehouses (sites) in the results of Beasley's paper.

or equal to that in the original problems. Further experiments show that this argument is true for all problems if exact optimal solutions are obtained. The reason is obviously that smaller size warehouses reduce the setup costs when various size levels are considered instead of fixed size. Extensive experiments with different fixed setup costs were taken. The smaller the proportion of the fixed site setup costs to the variant facility setup costs, the bigger the number of open warehouses in the solutions.

6. Conclusion

In this paper, we have considered the facility location problem which allows multiple facilities in the same site. The distinguishing feature of our model is its explicit consideration of non-fixed setup cost that depends on the size of the facility. The other important feature is that there are multiple types of facilities that can be located at different sites. The problem is modelled as a non-linear integer programming. Two equivalent mixed integer linear programming formulations are presented and solved by the general MIP solver. The computational experiments show that the first of the MIP formulations works better for concave setup functions than the second, while the second works better for convex functions than the first. Since the problem is NP-hard and even small size problem cannot be solved by using general MIP solver in short time, a heuristic based on Lagrangian relaxation is proposed in this paper. Experiments show that the Lagrangian heuristic is a good approximation approach for large-scale GFLP. We observed in the computational experiments that the lower bound obtained by Lagrangian relaxation approximates the optimal solution better than LP relaxation. Therefore, the Lagrangian relaxation can also be used to develop a good branch and bound algorithm for GFLP problem if exact optimal solution is desired.

Acknowledgements

The authors are grateful to the anonymous referees for comments and helping to improve the presentation of the earlier version of the paper.

References

[1] Mirchandani PB, Francis RL. Discrete location theory. New York: Wiley; 1990.
[2] Akinc U, Khumawala B. An efficient branch and bound algorithm for the capacitated warehouse location problem. Management Science 1997;23(6):585–94.
[3] Beasley JE. An algorithm for solving large capacitated warehouse location problems. European Journal of Operational Research 1988;33(3):314–25.
[4] Erlenkotter D. A dual-based procedure for uncapacitated facility location. Operations Research 1978;26:992–1009.
[5] Galvao RD, Raggi LA. A method for solving to optimality uncapacitated location problems. Annals of Operations Research 1989;18:225–44.
[6] Jacobsen SK. Heuristics for the capacitated plant location model. European Journal of Operational Research 1983;12: 253–61.
[7] Khumawala BM. An efficient branch and bound algorithm for the capacitated warehouse location problem. Management Science 1972;18(12):B718–31.
[8] Korkel M. On the exact solution of large-scale simple plant location problems. European Journal of Operational Research 1989;39:157–73.
[9] Krarup J, Pruzan PM. The simple plant location problem: survey and synthesis. European Journal of Operational Research 1983;12(1):36–81.
[10] Roy TJV. A cross decomposition algorithm for capacitated facility location. Operations Research 1986;34(1):145–63.
[11] Sridharan R. The capacitated plant location problem. European Journal of Operational Research 1995;87:203–13.

[12] Ghiani G, Guerriero F, Musmanno R. The capacitated plant location problem with multiple facilities in the same site. Computers & Operations Research 2002;29(13):1903–12.
[13] Hajiaghayi MT, Mahdian M, Mirrokni VS. The facility location problem with general cost function. Working paper, Massachusetts Institute of Technology, 2003.
[14] Feldman E, Lehrer F, Ray T. Warehouse location under continuous economies of scale. Management Science 1966;12(3):670–84.
[15] Harkness J, Revelle C. Facility location with increasing production costs. European Journal of Operational Research 2003;145(1):1–13.
[16] Soland R. Facility location with concave costs. Operations Research 1974;23(2):373–82.
[17] Barcelo J, Casanova J. A heuristic Lagrangean algorithm for the capacitated plant location problem. European Journal of Operational Research 1984;15:212–26.
[18] Barcelo J, Fernandez E, Jörnsten KO. Computational results from a new Lagrangean relaxation algorithm for the capacitated plant location problem. European Journal of Operational Research 1991;53(1):38–45.
[19] Beasley JE. Lagrangean heuristics for location problems. European Journal of Operational Research 1993;65:383–99.
[20] Christofides N, Beasley JE. Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. European Journal of Operational Research 1983;12(1):19–28.
[21] Geoffrion AM, McBride R. Lagrangian relaxation applied to capacitated facility location problem. AIIE Transactions 1978;10(1):40–7.
[22] Ghiani G, Grandinetti L, Guerriero F, Musmanno R. A Lagrangean heuristic for the plant location problem with multiple facilities in the same site. Optimization Methods and Software 2002;17(6):1059–76.
[23] Guignard M. A Lagrangean dual ascent algorithm for simple plant location problems. European Journal of Operational Research 1988;35:193–200.
[24] Guignard M, Kim S. Lagrangian decomposition: a model yielding stronger lagrangian bounds. Mathematical Programming 1987;39(2):215–28.
[25] Cornujelos G, Sridharan R, Thizy JM. A comparison of heuristics and relaxations for the capacitated plant location problem. European Journal of Operational Research 1991;50(3):280–97.
[26] Fisher ML. The Lagrangian relaxation method for solving integer programming problems. Management Science 1981;32:1019–36.
[27] Shor NZ. Minimization methods for non-differentiable functions. Berlin: Springer; 1985.
[28] Fisher ML. An applications oriented guide to Lagrangian relaxation. Interfaces 1985;15:10–21.