Computers & Industrial Engineering 61 (2011) 430–435

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A Lagrangian relaxation approach for a large scale new variant of capacitated clustering problem

Zhen Yang [a], Haoxun Chen [a], Feng Chu [b,*]

[a] Institut Charles Delaunay (ICD, FRE CNRS 2848), Industrial Systems Optimization Laboratory, University of Technology of Troyes, 12 Rue Marie Curie, BP 2060, Troyes 10010, France
[b] Laboratoire d'Informatique, Biologie Intégrative et Systèmes Complexes (IBISC), EA 4526, Université d'Evry Val d'Essonne, 40 Rue du Pelvoux, CE1455 Courcouronnes, 91020 Evry Cedex, France

ARTICLE INFO

Article history:
Available online 21 August 2010

Keywords:
Logistics
Facility location
Lagrangian relaxation
Heuristics

ABSTRACT

This paper studies a new variant of capacitated clustering problem (VCCP). In the VCCP, p facilities which procure a raw material from a set of suppliers are to be located among n potential sites ($n > p$) such that the total cost of assigning suppliers to the facilities and opening such facilities is minimized. Each supplier has a limited supply volume and each facility has a minimum supply requirement that must be satisfied by assigning enough suppliers to the facility. Each supplier can be assigned to at most one facility. When a supplier is assigned to a facility, the former will supply its all available volume to the latter. In order to solve the VCCP, a Lagrangian relaxation approach (LR) with two phases of dual optimization, the subgradient deflection in the first phase and the standard subgradient method in the second phase, is proposed. In the approach, the assignment constraints are relaxed. The resulting Lagrangian relaxed problem can be decomposed into a set of independent knapsack problems, which can be solved to optimality efficiently. At each Lagrangian iteration, a feasible solution is constructed from that of the Lagrangian relaxed problem by applying a greedy algorithm. Finally, the best feasible solution found so far is improved by a simple tabu search algorithm. Numerical tests on random instances show that the proposed LR can produce a tight lower bound and a high quality feasible solution for all instances with up to 4000 suppliers, 200 potential sites, and 100 plants to locate.

1. Introduction

The Capacitated Clustering Problem (CCP) is one of the most widely studied location problems with various applications, for example in vehicle routing (Koskosidis & Powell, 1992) and political districting (Bozkaya, Erkut, & Laporte, 2003), etc. It is to partition a set of n customers with given demand into p clusters with the maximum capacity constraint on each cluster. Every customer must be assigned to exactly one cluster. For each given cluster, a vertex is defined as the *cluster centre* whose customers are all customers in the cluster. The objective is to find p optimal clusters and their corresponding cluster centres so that the sum of the distances from the cluster centres to their customers is minimized.

The CCP is closely related to the Capacitated p-Median Problem (CPMP). If the sites of customers are used as the potential sites for cluster centres, the CCP reduces to the CPMP. The difference between the two problems is so small that the solution methodologies proposed for one problem can also be used for the other. The CCP and the CPMP can be further reduced to the Generalized Assignment Problem, if the cluster centres are fixed. As a consequence, the CCP and the CPMP are NP-hard.

Heuristic and metaheuristic approaches have been proposed for the CCP and the CPMP, for example, Lorena and Senne (2004) examined a column generation approach, the instances with up to 402 demand points and 40 clusters were solved. By using hybrid scatter search with path relinking, Diaz and Fernandez (2006) solved the instances with 737 demand points. Franca, Sosa, and Pureza (1999) presented a tabu search algorithm for the CPMP. A bionomic approach was proposed by Maniezzo, Mingozzi, and Baldaci (1998). Shieh and May (2001), whereas Lorena and Furtado (2001) developed a genetic algorithm to the problem. Simulated annealing and tabu search algorithm (Osman & Christofides, 1994), scatter search (Scheuerer & Wendolsky, 2006) and variable neighbourhood search (Fleszar & Hindi, 2008) were also proposed.

Exact methods, such as set partitioning approach (Baldacci, Hadjiconstantinou, Maniezzo, & Mingozzi, 2002), branch-and-price algorithm (Ceselli & Righini, 2005) were also developed for the CPMP. More recently, by using fenchel cutting planes in separating knapsack polytopes, Boccia, Sforza, Sterle, and Vasilyev (2008) proposed a cut and branch approach for the problem. To our best knowledge, up to now only a few instances of CPMP with up to 200 customers and 80 clusters have been solved to optimality in a reasonable time in the literature (Boccia et al., 2008).

* Corresponding author.
E-mail addresses: zhen.yang@utt.fr (Z. Yang), haoxun.chen@utt.fr (H. Chen), feng.chu@ibisc.univ-evry.fr (F. Chu).

0360-8352/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.07.021

In this paper, we study a new variant of CCP encountered in a green energy investment project of the Champagne–Ardenne region of France. In the project, the local government wants to build 4 plants in 6 potential sites to produce the ethanol from the wheat straw at the minimum total cost of opening plants and assigning suppliers to the plants. Considering the potential demand of ethanol and economies of scale of production, each plant is subject to a minimum capacity requirement whose corresponding wheat straw requirement must be satisfied by assigning enough suppliers to the plant, where each supplier is assigned to one plant at most and when a supplier is assigned to a facility, the former will supply its all available volume to the latter. Note that it is not necessary to set a maximum capacity limit on each plant when modeling this problem since the objective is to minimize the total cost. For an open plant, if its minimum capacity requirement has been attained, then assigning more suppliers to it will increase the cost. This case will never happen in the optimal solution of the problem. There are about 2000 suppliers and the total supply volume of all suppliers is greater than the total requirement of all plants. To our best knowledge, this oversupply condition has not received enough attention in facility location, since most studies focused on the distribution of final products, in which all customers' demands must be satisfied.

Compared with the CCP and the CPMP, the VCCP has the following new features: (1) Each open plant has a minimum capacity requirement that must be satisfied. (2) Unlike the CCP where each customer has to be assigned to one cluster centre, in the VCCP, each supplier can be assigned to at most one plant. These two features make our problem a variant of the CCP, denoted as VCCP in the following discussion. They cannot be transformed each other. A detailed discussion on this will be given in the next section.

If one plant is located ($p = 1$) at one potential site ($n = 1$), the VCCP reduces to a knapsack problem. As a result, the VCCP is NP-hard. Heuristics are therefore the only practical techniques for handling large scale instances of the problem. Lagrangian relaxation approaches have been successfully applied to facility location problems, such as the Capacitated Facility Location Problem (Cornuéjols, Sridharan, & Thizy, 1991), the Single Source Capacitated Facility Location Problem (Beasley, 1993; Pirkul, 1987), etc. A systematic survey can be found in Klose and Drexl (2005). A Lagrangian relaxation approach with two phases of dual optimization, the subgradient deflection method in the first phase and the standard subgradient method in the second phase, is developed in this paper to approximately solve the VCCP by relaxing the assignment constraints. In the approach, the best Lagrange multipliers and the best upper bound found in the first phase are used as inputs for the second phase. At each Lagrangian iteration, a feasible solution is constructed from the optimal solution of the Lagrangian relaxed problem by applying a greedy algorithm. Furthermore, at the end of the subgradient method, the best feasible solution found so far is improved by a simple tabu search algorithm.

The paper is organized as follows. In Section 2, the VCCP is formulated as a binary integer program. Section 3 presents the Lagrangian relaxation approach and Section 4 the method to solve the Lagrangian dual problem. The methods to construct feasible solutions are presented in Section 5. The computational results are given in Section 6. Finally, Section 7 draws a conclusion to this paper.

2. Problem formulation

Let S be a set of suppliers with positive supply volume sv_i and $P = \{1, \ldots, n\}$ denote a set of potential sites for plants (facilities) with an associated minimum capacity requirement CR_j and a fixed opening cost FP_j. c_{ij} denotes the cost of assigning supplier $i \in S$ to the plant located at site $j \in P$. We define a binary variable y_j, which takes 1 if a plant is located at site $j \in P$ and 0 otherwise. Binary variable x_{ij} equals 1 if supplier $i \in S$ is assigned to plant located at site $j \in P$ and 0 otherwise.

The VCCP can then be formulated as the following binary integer program:

$$\text{Model P}: z = \min \sum_{i \in S} \sum_{j \in P} c_{ij} \cdot x_{ij} + \sum_{j \in P} FP_j \cdot y_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in P} x_{ij} \leqslant 1, \quad \forall i \in S \tag{2}$$

$$\sum_{i \in S} sv_i \cdot x_{ij} \geqslant CR_j \cdot y_j, \quad \forall j \in P \tag{3}$$

$$\sum_{j \in P} y_j = p \tag{4}$$

$$x_{ij},\ y_j \in \{0,1\}, \quad \forall i \in S,\ \forall j \in P \tag{5}$$

The objective function (1) is to minimize the total cost of assigning suppliers to plants and establishing such plants. The assignment constraints (2) guarantee that each supplier is assigned to at most one plant. Minimum supply requirements are satisfied through constraints (3). Constraint (4) ensures that p plants are exactly located.

The VCCP cannot be transformed into the classical CCP. To prove this, we consider the extreme case with $p = 1$ and $n = 1$, that is, only one plant will be located at one potential site. The resulting VCCP reduces to a NP-hard knapsack problem after replacing x_{ij} in the above model P by its complementary variable $Z_{ij} = 1 - x_{ij}$. However, in the same extreme case, the CCP reduces to a trivial problem: all customers have no choice but to be assigned to the sole depot (plant). The trivial problem can be solved in a polynomial time. From the view point of computational complexity, we can conclude that the VCCP cannot be transformed into the CCP, unless the knapsack problem can be solved in a polynomial time, so the algorithms existing in the literature for the CCP cannot be applied to the VCCP.

Due to the large scale of VCCP instances needed to be solved (the number of suppliers may be several thousands) and the state-of-the-art solution methods can only solve CCP instances of relative small sizes with no more than 400 customers to optimality in a reasonable time in the literature (Lorena & Senne, 2004), in this paper, we try to find a sharp lower bound and a high quality near-optimal solution (upper bound) for the VCCP by using a Lagrangian relaxation approach.

3. Lagrangian relaxation

Lagrangian relaxation approaches have been widely used to solve location problems. Surveys can be found in Cornuéjols et al. (1991), Beasley (1993) and Klose and Drexl (2005). In this paper, we relax the assignment constraints (2) and dualizing them into the objective function (1) by introducing a set of non-negative Lagrange multipliers $\lambda \in \mathbb{R}^{|S|}$, the Lagrangian relaxed problem $P_{LR}(\lambda)$ can be formulated as:

$$z_{LR}(\lambda) = \min \sum_{i \in S} \sum_{j \in P} c_{ij} \cdot x_{ij} + \sum_{j \in P} FP_j \cdot y_j + \sum_{i \in S} \lambda_i \cdot \left(\sum_{j \in P} x_{ij} - 1 \right)$$
$$\text{s.t.} \quad (3)-(5) \tag{6}$$

It is well known that $z_{LR}(\lambda)$ provides a lower bound for the VCCP for any $\lambda_i \geqslant 0$, $i \in S$ (Fisher, 2004).

To find the optimal solution of $P_{LR}(\lambda)$, we can initially set $y_j = 1$, then, $P_{LR}(\lambda)$ can be decomposed into $|P|$ independent 0-1 knapsack

problems (KPs). These KPs can be solved efficiently by using Pisinger's MINKNAP (Pisinger, 1999). The jth knapsack problem is:

$$z_{KP}^{j}(\lambda) = \min \sum_{i\in S} c_{ij}\cdot x_{ij} + FP_j + \sum_{i\in S}\lambda_i \cdot x_{ij} \quad \text{s.t.} \quad (3),\ (5) \tag{7}$$

Let π be a permutation of the numbers 1,2,…,$|P|$ such that

$$z_{KP}^{\pi(1)}(\lambda) \leqslant z_{KP}^{\pi(2)}(\lambda) \leqslant \cdots \leqslant z_{KP}^{\pi(|P|)}(\lambda)$$

A p plants have to be located, then the first p knapsack problems' solutions with the smallest values of $z_{KP}^{\pi(j)}(\lambda)$ make up of the optimal solution of $P_{LR}(\lambda)$. The optimal solution of the Lagrangian relaxed problem can thus be defined as: $y_{\pi(j)} := 1$, $\forall j \leqslant p$; $y_{\pi(j)} := 0$, $\forall j > p$; $x_{i,\ \pi(j)}$, $\forall j \leqslant p$, $\forall i \in S$ takes the solution value of the corresponding knapsack problem, and $x_{i,\pi(j)} := 0$, $\forall j > p$, $\forall i \in S$. The optimal objective value of $P_{LR}(\lambda)$ is:

$$z_{LR}(\lambda) = \sum_{i=1}^{p} z_{KP}^{\pi(i)}(\lambda) - \sum_{i\in S}\lambda_i \tag{8}$$

4. Resolution of the Lagrangian dual problem

To find the best lower bound, we need to solve the Lagrangian dual problem to determine the optimal Lagrange multipliers λ^*, which is defined as

$$z_D = \max_{\lambda \geqslant 0} z_{LR}(\lambda) \tag{9}$$

The subgradient method is often applied to solve the Lagrangian dual, which repeatedly solves the Lagrangian relaxed problem $P_{LR}(\lambda)$ and update the Lagrange multipliers based on the subgradient of $z_{LR}(\lambda)$ at the current value of λ. The iteration will be terminated until a stopping criterion is met. Let (x^k, y^k) denotes the optimal solution of Lagrangian relaxed problem at iteration k. The following two methods are used to update the Lagrange multipliers: *the standard subgradient method* and *the subgradient deflection method.*

4.1. Standard subgradient method

The multipliers for the next iteration λ^{k+1} are generated based on the current multipliers λ^k and the subgradient of $z_{LR}(\lambda)$ at $\lambda = \lambda^k$. The ith component of the multiplier vector λ^{k+1} is calculated as:

$$\lambda_i^{k+1} = \max\{0, \lambda_i^k + t_k \cdot g_i^k\}, \quad \forall i \in S \tag{10}$$

where t_k is the step size and $g_i^k = \sum_{j\in P} x_{ij}^k - 1$ the ith component of the subgradient of $z_{LR}(\lambda)$ at $\lambda = \lambda^k$, denoted by g^k.

The step size t_k is defined by:

$$t_k := \theta_k \left[z_{UB} - z_{LR}(\lambda^k)\right] / \|g^k\|^2 \tag{11}$$

where z_{UB} is an upper bound of the VCCP, $z_{LR}(\lambda^k)$ is the current dual value and $\theta_k \in (0,2)$ is a parameter. The parameter should be initialized (take a value in the interval (0,2)) and be halved if a given number of iterations (say 20–30) of the subgradient procedure have been performed without improving the lower bound.

The general procedure of the subgradient method can be summarized as follows:

Subgradient method

Step 0. Initiate $\lambda^0 := 0$, $\theta_0 := 1$, $k := 0$ and $z_{UB} = +\infty$;
Step 1. Solve the relaxed problem $P_{LR}(\lambda^k)$, construct a feasible solution and update the upper bound z_{UB} if necessary;
Step 2. Set step size t_k by $t_k := \theta_k[z_{UB} - z_{LR}(\lambda^k)]/\|g^k\|^2$ and update the Lagrange multipliers according to (10);
Step 3. If in a given number of iterations no improvement of lower bound is observed, halve $\theta_{k+1} := \theta/2$; Otherwise, set $\theta_{k+1} := \theta_k$;
Step 4. Check the stopping criterions:
(1) The lower bound is not improved for a given number of iterations, or
(2) The maximum number of iterations is reached.

If one of these criterions is met, stop and output the best dual value, the best feasible solution and the duality gap; Otherwise, $k := k+1$, return to Step 1;

4.2. Subgradient deflection

The subgradient deflection method differs from standard subgradient method in the definition of the moving direction of the Lagrange multipliers at each iteration. The deflection technique defines the moving direction as a combination of the previous direction and the current subgradient direction (Camerini, Fratta, & Maffioli, 1975). It is proved to significantly improve the convergence of the subgraditent method. At iteration k, the moving direction h^k, which replaces g^k in (10) and (11), is calculated by using the following formula (Avella, Boccia, Sforza, & Vasil'ev, 2009)

$$h^k = \frac{g^k + 0.1h^{k-2} + 0.3h^{k-1}}{1.4} \tag{12}$$

where h^{-2} and h^{-1} will be initialized as a $|S|$-dimensional zero vector, respectively.

Then, the Lagrange multipliers are updated as

$$\lambda_i^{k+1} = \max\{0, \lambda_i^k + t_k \cdot h_i^k\}, \quad \forall i \in S \tag{13}$$

The step size t_k is calculated by:

$$t_k := \theta_k \left[z_{UB} - z_{LR}(\lambda^k)\right] / \|h^k\|^2 \tag{14}$$

However, in our application, we use (11) rather than (14) to set the step size t_k for the subgradient deflection method, since numerical results show that the lower bound obtained by setting step size t_k using (11) is better than that of using (14).

The general procedure of the subgradient deflection method is the same as that of standard subgradient method except that the moving direction at iteration k, g^k, is replaced by h^k.

4.3. Two phases of dual optimization

If we simply apply standard subgradient method to solve the Lagrangian dual problem, we observe that during the iteration of the subgradient method the value of parameter θ becomes smaller and smaller. Finally, it tends to be zero. At this time, no further improvement of the lower bound can be achieved since the Lagrange multipliers have been stable. In order to jump out of this "dilemma", two phases of dual optimization are performed. In the first phase, the subgradient deflection is used to update the Lagrange multipliers. In the second phase, the subgradient method is used with parameter θ reset to its initial value. The best Lagrange multipliers and the best upper bound found in the first phase are used as inputs for the second phase.

The idea of combining subgradient deflection and standard subgradient is based on the following observations: (1) subgradient deflection can improve the convergence of subgradient algorithms (Avella et al., 2009). (2) Good initial Lagrange multipliers and different ways of updating the multipliers can augment the probability of finding a better lower bound. (3) Setting θ to its initial value may help to jump out the local optima of the Lagrangian dual.

5. Construction of feasible solutions

An upper bound is needed in calculating the step size t_k by (11) and by (14). Thus, feasible solutions of the VCCP should be constructed. As a common knowledge, the optimal solution of the current Lagrangian relaxed problem $P_{LR}(\lambda^k)$ may be infeasible, but it provides some useful information for constructing a feasible solution of the original problem: the locations of p open plants. In the solution of the current Lagrangian relaxed problem, if supplier i is assigned to exactly one open plant l, for example $x_{il}^k = 1$ and $\sum_{j\in P} x_{ij}^k = 1$, then we fix $x_{il} = 1$ in the feasible solution. Now, the remaining problem is to select enough suppliers from unassigned ones to satisfy the residual capacity requirements of all open plants. The generation of the feasible solution can be divided into two phases: the construction phase and the improvement phase.

5.1. Construction phase

The assignment of supplier i to an open plant j is evaluated by a weight function $f(i,j)$. For each open plant j, the difference between the second smallest and the smallest value of $f(i,j)$ is computed, and the plants are supplied (served) by suppliers in decreasing order of this difference. That is, for each open plant j, the *desirability* of assigning a supplier i to plant j is given by

$$\rho_j = \min_{s\neq i_j} f(s,j) - f(i_j,j)$$

where $i_j = \arg\min_i f(i,j)$. Since the locations of plants have been fixed and a part of capacity requirements of these open plants have been satisfied, we only need to consider those open plants and those suppliers that have not been assigned. We define the weight function as

(i) $f(i,j) = c_{ij}/sv_i$ or
(ii) $f(i,j) = c_{ij}$

similar with those proposed by Martello and Toth (1990) in solving the Generalized Assignment Problem. The motivation for choosing one of the weight functions (i) and (ii) is that it is desirable to assign a supplier to a plant that will result in the least (absolute or relative) cost. The greedy algorithm for constructing a feasible solution can now be stated as the following:

Greedy algorithm

Step 0: Set $y_j = y_j^k$ for all $j \in P$. If $\exists i \in S$, $l \in P$ such that $x_{il}^k = 1$ and $\sum_{j\in P} x_{ij}^k = 1$, then fix $x_{il} = 1$.

Step 1: Set $\widetilde{P} = \{j | y_j = 1, j \in P\}$, $\widetilde{S} = \{i | \sum_{j\in P} x_{ij} = 0, i \in S\}$ and $\widetilde{CR}_j = CR_j - \sum_{i\in S\setminus\widetilde{S}} sv_i \cdot x_{ij}$, $\forall j \in \widetilde{P}$. Update $\widetilde{P} := \left\{ j | \widetilde{CR}_j > 0, j \in \widetilde{P} \right\}$.

Step 2: Calculate $i_j = \arg\min_{i\in\widetilde{S}} f(i,j)$ and $\rho_j = \min_{s\in\widetilde{S},\ s\neq i_j} f(s,j) - f(i_j,j)$ for $j \in \widetilde{P}$.

Step 3: Calculate $\hat{j} = \arg\max_{j\in\widetilde{P}} \rho_j$, i.e., a supplier will be assigned to plant $\hat{j}$ next.

$$x_{i_{\hat{j}},\hat{j}} = 1$$
$$x_{i_{\hat{j}},j} = 0 \quad \forall j \in \widetilde{P}, j \neq \hat{j}$$
$$\widetilde{CR}_j := \widetilde{CR}_j - sv_{i_{\hat{j}}}$$
$$\widetilde{S} := \widetilde{S} \setminus \{i_{\hat{j}}\}$$

Step 4: If $\widetilde{CR}_j \leqslant 0$, then $\widetilde{P} := \widetilde{P} \setminus \{\hat{j}\}$. If $\widetilde{P} = \varnothing$: Stop, a feasible solution (x,y) of VCCP is found. If $\widetilde{P} \neq \varnothing$ and $\widetilde{S} = \varnothing$, then the greedy algorithm fails, Stop. Otherwise, return to Step 2.

5.2. Improvement phase – method 1

If the solution (x,y) obtained by the above greedy algorithm is feasible, then it can be improved in several ways. First of all, redundant suppliers can be deleted from open plants. We complete this task by solving a knapsack problem. For each open facility, for example plant k, let $\Omega = \{i | x_{ik} = 1,\ i \in S\}$, the knapsack problem can be formulated as

$$(KP) \quad \min \quad \sum_{i\in\Omega} c_{ik} s_i$$
$$\text{s.t.} \quad \sum_{i\in\Omega} sv_i \cdot s_i \geqslant CR_k$$
$$s_i \in \{0,1\} \quad \forall i \in \Omega$$

The optimal solution of KP, s^*, is used to update the upper bound solution. If $s_i^* = 1,\ i \in Q$, then set $x_{ik} = 1$; Otherwise, $x_{ik} = 0$. A new feasible solution with a smaller objective value is thus obtained. This improvement will be performed at each Lagrangian iteration.

5.3. Improvement phase – method 2

To further reduce the upper bound, a simple tabu search algorithm is applied to search a better feasible solution. We explore two types of neighborhood which are defined by the following two kinds of moves, respectively:

Move 1: Exchange one assigned supplier with one unassigned supplier while respecting the capacity constraint of each plant;
Move 2: Exchange two suppliers assigned to two different plants while respecting the capacities of both plants.

Given a feasible solution, we search its neighborhood. The most gainful move is performed and the move contrary to this gainful move is declared as tabu in the subsequent m iterations, where m is a given parameter. Update the solution and repeat the above procedure until a given maximal number of iterations is attained or the current solution has not been improved in a given number of iterations. Since the tabu search is time consuming, we only apply it to improve the best upper bound solution found during the Lagrangian iteration.

6. Numerical results

The proposed Lagrangian relaxation approach has been tested on random instances which are generated with reference to the facility location problem in the green energy investment project of the Champagne–Ardenne region of France and the way used in Cornuéjols et al. (1991) for the CFLP. The instances contain four subsets of problems which differ in the ratio r, which takes 40%, 60%, 80%, 95%, respectively, where r is defined as total minimum capacity requirement of all plants over the total supply volume of all suppliers. For each r, we generate and test instances of 10 different sizes of $n \times p \times |S|$: $10 \times 5 \times 50$, $20 \times 5 \times 50$, $20 \times 10 \times 100$, $50 \times 10 \times 200$, $50 \times 25 \times 200$, $100 \times 25 \times 500$, $100 \times 50 \times 500$, $200 \times 50 \times 1000$, $200 \times 50 \times 4000$, $200 \times 100 \times 4000$, respectively. Taking 200 and 100 as the maximum number of potential sites and sites to locate, respectively, is just for evaluating the performance of our LR approach for large instances, a real location problem may have less number of potential sites and less number of sites to locate. Taking 4000 as the maximum number of suppliers is motivated by a real location problem which has more than 2000 suppliers. For each combination of r and problem size, five instances are generated. These instances are generated as follows:

(1) The geographical positions of the potential sites of plants and the customers are randomly and uniformly generated in a unit square. The cost of assigning each supplier to each plant is then computed by multiplying their Euclidean distance by 10.
(2) The supply volume of each supplier is randomly and uniformly generated from the interval (Bozkaya et al., 2003).
(3) Since the plants to locate have different capacity requirements, the p plants that have small capacity requirements tend to be opened. To increase the complexity of the generated instances, we assume that all plants have the same capacity requirements. The capacity requirement CR of each plant is generated by the formula $CR = r \cdot \sum_{i \in S} s v_i / p$.

The fixed opening cost of each plant is proportional to its capacity requirement, which is randomly generated according to the formula $FP_j = U[0, 90] + U[100, 110]\sqrt{CR}$, where $U[\cdot,\cdot]$ denotes a random variable with uniform distribution.

Our algorithm was coded by using Visual C++ 6.0. The numerical tests were performed on a PC with Pentium (R) 1.73 GHz and 1.24 GB RAM. The maximum number of iterations of the subgradient method and the subgradient deflection method is set to 400. If no improvement of lower bound in 30 iterations is observed, the parameter θ is halved. The algorithm is terminated if no improvement of lower bound is observed in 60 iterations. In the tabu search for improvement of the best feasible solution, the length of tabu list is set to 20. The maximum number of iterations of the tabu search is set to 200. The outcome of the computational experience is reported in Tables 1–4. We compare the performance of two subgradient strategies: the standard subgradient method only and the two-phase dual optimization that combines standard subgradient method and subgradient deflection. For each strategy, column *No TS* reports the average duality gap of the tested instances obtained without tabu search, whereas column Ave, Max and Time (in seconds) report the average, maximal duality gap and the average computational time (in seconds) of the tested instances obtained with tabu search, respectively.

For each set of instances (for a given r value), the average duality gap corresponding to each method is given in the last line of the corresponding table. The maximal duality gaps of this set of instances obtained by the standard subgradient or the two-phase method are indicated by the values in **boldface**.

From Tables 1–4, we can observe that the average and the maximal duality gaps have little increment with the value of parameter r. That is because the higher the value of parameter r, the bigger the minimum capacity requirement of each plant is. Consequently, more suppliers have to be assigned to plants and thus increase the difficulty of the instance. For the same r, we find that our algorithm is insensitive to the problem size.

For all randomly generated instances, the proposed Lagrangian relaxation approach with two phases of dual optimization can produce a feasible solution which deviates from the best lower bound by 0.51% on average and 2% in the worst case. The method is very effective in closing the duality gap especially the maximal duality gap compared with the method only using the standard subgradient only. The proposed tabu search algorithm is also effective in reducing the upper bound, which in turn contributes to the improvement of the performance of the subgradient method in the second phase of the dual optimization and thus leads to a tight lower bound.

The computational complexity of the both LR approaches can be estimated as the number of LR iterations multiplied by the computational complexity of all knapsack subproblems plus the computational complexity of feasible solution construction and improvement. As we know, the computational complexity of a knapsack problem is $O(m \times b)$, where m is the number of items and b is the value of the right-hand side constant of the knapsack constraint. For our jth knapsack subproblem ($j \in P$), m is the

Table 1
Results for randomly generated problems with $r = 40\%$.

$n \times p \times \lvert S\rvert$	Standard subgradient method				Two-phase			
	NoTS	Ave	Max	Time (s)	NoTS	Ave	Max	Time (s)
$10 \times 5 \times 50$	1.210	0.301	1.203	3.78	0.908	0.276	**0.805**	4.73
$20 \times 5 \times 50$	0.781	0.198	0.344	3.37	0.562	0.151	0.259	4.84
$20 \times 10 \times 100$	0.092	0.079	0.150	7.21	0.083	0.079	0.150	7.71
$50 \times 10 \times 200$	0.663	0.223	0.469	7.78	0.596	0.175	0.332	9.02
$50 \times 25 \times 200$	0.128	0.075	0.109	14.33	0.125	0.075	0.109	16.42
$100 \times 25 \times 500$	0.374	0.147	0.232	18.73	0.330	0.100	0.121	22.00
$100 \times 50 \times 500$	0.485	0.083	0.152	21.48	0.369	0.082	0.149	25.55
$200 \times 50 \times 1000$	1.001	0.335	0.799	30.01	0.867	0.176	0.344	41.91
$200 \times 50 \times 4000$	1.495	0.423	**1.227**	150.33	1.101	0.382	0.569	243.26
$200 \times 100 \times 4000$	0.739	0.254	0.584	559.43	0.425	0.131	0.153	814.29
Average	0.700	0.212			0.534	0.163		

Table 2
Results for randomly generated problems with $r = 60\%$.

$n \times p \times \lvert S\rvert$	Standard subgradient method				Two-phase			
	NoTS	Ave	Max	Time (s)	NoTS	Ave	Max	Time (s)
$10 \times 5 \times 50$	0.697	0.546	**2.433**	1.88	0.653	0.285	**0.997**	2.84
$20 \times 5 \times 50$	0.992	0.189	0.886	2.21	0.829	0.141	0.419	3.55
$20 \times 10 \times 100$	0.677	0.522	1.238	3.93	0.524	0.348	0.865	6.69
$50 \times 10 \times 200$	0.331	0.343	0.637	4.62	0.236	0.113	0.206	7.02
$50 \times 25 \times 200$	0.752	0.201	0.420	6.38	0.427	0.130	0.317	11.58
$100 \times 25 \times 500$	0.703	0.242	0.433	9.88	0.688	0.241	0.402	17.84
$100 \times 50 \times 500$	0.779	0.155	0.304	13.53	0.701	0.132	0.198	20.30
$200 \times 50 \times 1000$	0.469	0.222	1.037	20.55	0.384	0.104	0.166	35.57
$200 \times 50 \times 4000$	0.296	0.279	1.338	256.47	0.231	0.111	0.145	465.55
$200 \times 100 \times 4000$	1.575	0.684	0.721	350.48	0.942	0.587	0.669	567.65
Average	0.727	0.338			0.562	0.219		

Table 3
Results for randomly generated problems with $r = 80\%$.

$n \times p \times \lvert S\rvert$	Standard subgradient method				Two-phase			
	NoTS	Ave	Max	Time (s)	NoTS	Ave	Max	Time (s)
$10 \times 5 \times 50$	1.497	0.467	0.922	2.98	1.022	0.431	0.846	4.32
$20 \times 5 \times 50$	1.543	0.387	1.432	1.88	1.331	0.379	1.167	2.98
$20 \times 10 \times 100$	2.688	0.881	1.139	4.30	2.010	0.761	0.951	6.30
$50 \times 10 \times 200$	1.941	0.682	**2.031**	4.42	1.341	0.562	**1.542**	6.77
$50 \times 25 \times 200$	2.055	0.562	0.771	5.87	1.678	0.441	0.514	8.56
$100 \times 25 \times 500$	1.485	0.331	0.428	9.90	1.120	0.331	0.428	14.70
$100 \times 50 \times 500$	1.777	0.534	0.891	9.11	1.560	0.478	0.723	13.92
$200 \times 50 \times 1000$	3.001	0.342	0.533	20.18	2.327	0.254	0.435	33.95
$200 \times 50 \times 4000$	1.992	0.099	0.369	420.11	1.372	0.087	0.244	668.72
$200 \times 100 \times 4000$	2.101	0.351	0.521	433.69	1.491	0.282	0.423	686.23
Average	2.008	0.464			1.525	0.400		

Table 4
Results for randomly generated problems with $r = 95\%$.

$n \times p \times \lvert S\rvert$	Standard subgradient method				Two-phase			
	NoTS	Ave	Max	Time (s)	NoTS	Ave	Max	Time (s)
$10 \times 5 \times 50$	4.205	1.352	2.512	0.78	3.241	1.050	**1.955**	1.28
$20 \times 5 \times 50$	3.545	1.391	2.447	0.45	2.350	0.968	1.056	0.91
$20 \times 10 \times 100$	3.898	1.764	2.004	2.01	3.172	1.632	1.898	4.06
$50 \times 10 \times 200$	4.321	1.673	2.549	2.60	3.232	1.464	1.902	4.50
$50 \times 25 \times 200$	4.835	1.879	2.673	4.98	4.147	1.476	1.892	9.27
$100 \times 25 \times 500$	4.031	2.003	**2.845**	8.48	3.465	1.459	1.846	15.72
$100 \times 50 \times 500$	5.523	1.579	2.335	12.41	4.989	1.367	1.806	22.73
$200 \times 50 \times 1000$	2.648	1.384	1.993	66.59	2.012	1.209	1.361	119.45
$200 \times 50 \times 4000$	5.221	1.726	2.515	721.28	4.239	1.300	1.952	1412.97
$200 \times 100 \times 4000$	3.003	1.023	1.421	973.39	2.561	0.782	0.873	1806.39
Average	4.123	1.577			3.341	1.27		

number of suppliers and $b = \sum_{i \in S} sv_i - CR_j$. The computational complexity of the greedy algorithm for feasible solution construction is $O(|S|^2|P|)$, whereas the computational complexity of solution improvement is $O(p \times m \times b)$ and for method 1 and $O(m^2)$ multiplied by the number of local search iterations for method 2, respectively.

Compared with the LR approach using the standard subgradient method for its dual optimization, the LR approach with two phases of dual optimization costs 30–100% more computational time but provides better solutions for all tested instances. This increase of computational time can be explained as the second algorithm does two times (phases) of dual optimization. Since the computational time of the second algorithm is still acceptable for large instances, it would be better to use the algorithm for real problems.

7. Conclusions

In this paper, we have studied a new realistic variant of capacitated clustering problem encountered in up-stream supply chain network design and proposed a Lagrangian relaxation approach with two phases of dual optimization for its resolution. Numerical tests on 200 random instances have proved the effectiveness of the two-phase method. Tight lower bounds and high quality feasible solutions have been obtained for all tested instances.

References

Avella, P., Boccia, M., Sforza, A., & Vasil'ev, I. (2009). An effective heuristic for large-scale capacitated facility location problems. *Heuristics, 15*(6), 597–615.

Baldacci, R., Hadjiconstantinou, E., Maniezzo, V., & Mingozzi, A. (2002). A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research, 29*(4), 365–383.

Beasley, J. E. (1993). Lagrangian heuristics for location problems. *European Journal of Operational Research, 65*(3), 383–399.

Boccia, M., Sforza, A., Sterle, C., & Vasilyev, I. (2008). A cut and branch approach for the capacitated *p*-median problem based on fenchel cutting planes. *Journal of Mathematical Modelling and Algorithms, 7*(1), 43–58.

Bozkaya, B., Erkut, E., & Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research, 144*(1), 12–26.

Camerini, P. M., Fratta, L., & Maffioli, F. (1975). On improving relaxation methods by modified gradient techniques. *Mathematical Programming Study, 3*, 26–34.

Ceselli, A., & Righini, G. (2005). A branch-and-price algorithm for the capacitated *p*-median problem. *Networks, 45*(3), 125–142.

Cornuéjols, G., Sridharan, R., & Thizy, J. M. (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research, 50*, 280–297.

Diaz, J. A., & Fernandez, E. (2006). Hybrid scatter search and path relinking for the capacitated *p*-median problem. *European Journal of Operational Research, 169*, 570–585.

Fisher, M. L. (2004). The Lagrangian relaxation method for solving integer programming problems. *Management Science, 50*, 1861–1871.

Fleszar, K., & Hindi, K. S. (2008). An effective VNS for the capacitated *p*-median problem. *European Journal of Operational Research, 191*, 612–622.

Franca, P. M., Sosa, N. M., & Pureza, V. (1999). An adaptive tabu search algorithm for the capacitated *p*-median problem. *International Transactions in Operations Research, 6*, 665–678.

Klose, A., & Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research, 162*, 4–29.

Koskosidis, Y. A., & Powell, W. B. (1992). Clustering algorithms for consolidation of customer orders into vehicle shipments. *Transportation Research B, 26*(5), 365–379.

Lorena, L. A. N., & Furtado, J. C. (2001). Constructive genetic algorithm for clustering problems. *Evolutionary Computation, 9*(3), 309–327.

Lorena, L. A. N., & Senne, E. L. F. (2004). A column generation approach to capacitated *p*-median problems. *Computer & Operations Research, 31*, 863–876.

Maniezzo, V., Mingozzi, A., & Baldaci, R. (1998). A bionomic approach to the capacitated *p*-median problem. *Journal of Heuristics, 4*, 263–280.

Martello, S., & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations*. New York: John Wiley & Sons Ltd..

Osman, I., & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operations Research, 1*(3), 317–336.

Pirkul, H. (1987). Efficient algorithm for the capacitated concentrator location problem. *Computers and Operations Research, 14*, 197–208.

Pisinger, D. (1999). An exact algorithm for large multiple knapsack problems. *European Journal of Operational Research, 3*, 528–541.

Scheuerer, S., & Wendolsky, R. (2006). A scatter search heuristic for the capacitated clustering problem. *European Journal of Operational Research, 169*, 533–547.

Shieh, H. M., & May, M. D. (2001). Solving the capacitated clustering problem with genetic algorithms. *Journal of the Chinese Institute of Industrial Engineers, 18*(3), 1–12.