Lagrangian relaxation heuristics for the uncapacitated single-source multi-product facility location problem

A.li Mohammad Nezhad[a],

Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
a_mohammadnezhad@ie.sharif.edu

Hasan Manzour[*b]

School of Industrial and Systems Engineering, University of Oklahoma, United States

smh.manzour@ou.edu

Said. Salhi[c]

The Centre for Logistics and Heuristic Optimisation, Kent Business School, University of Kent at Canterbury, Canterbury, UK,
s.salhi@kent.ac.uk

Abstract

Facility location problem is one of the strategic logistical drivers within the supply chain which is a hard to solve optimization problem. In this study, we focus on the uncapacitated single-source multi-product production/distribution facility location problem with the presence of set-up cost. To efficiently tackle this decision problem, two lagrangian-based heuristics are proposed one of which incorporates integer cuts to strengthen the formulation. Local search operators are also embedded within these methods to improve the upper bounds as the search progresses. Three set of instances with various characteristics are generated and used to evaluate the performance of the proposed algorithms. Encouraging results are obtained when assessed against an ILP formulation using CPLEX. The latter is used for generating optimal solutions for small size instances and also as a means for producing upper and lower bounds for larger ones when restricted by a limited amount of execution time.

Keywords: location problems, Integer programming formulation, Lagrangian relaxation, Cutting plane, Local search.

* Corresponding author: smh.manzour@ou.edu (Hasan Manzour)

1. Introduction

Facility location problem (FLP) is concerned with where to locate a set of facilities and how to satisfy customers' demands from these open facilities so that the total cost which includes the facility set up cost as well as the transportation cost is minimized. The facility location problem has many applications in various areas such as distribution management, transportation, health and telecommunication networks design. The FLP and its variants have received a great deal of attention in the literature, for instance see the comprehensive edited books by Drezner and Hamacher (2002), and Nickel and Puerto (2005).

Lagrangian relaxation was first proposed by Held and Karp (1970, 1971) and proved to be successful at solving many classes of optimization problems. In brief, these types of heuristics are designed to take into account the advantages of both exact and heuristic methods. The idea is to relax the set of constraints that are known to make the problem hard to solve by adding these to the objective function with a penalty attached. The transformed problem then becomes easier to solve optimally for which its optimal objective function value is a lower bound for the original problem. A feasible solution of the original problem (an upper bound in case of a minimization problem) is derived using a usually quick heuristic method. The penalties are then adjusted and the process continues until the gap between the best lower and upper bounds is reasonably small. This approach, which fits into the class of mathematically-based heuristics (see Salhi (2006) for an overview on heuristic search) was applied successfully in solving several classes of FLPs.

We briefly mention those studies on location problems which are closely related to ours and for which Lagrangian relaxation (LR) was used. Capacitated FLP (CFLP) is a well-known variant of FLP where facilities have restricted capacity. A variant of CFLP is the single-source CFLP (SSCFLP) where each customer has to be served from one facility only. Different solution methods have been proposed to deal with this problem, some are based on LR, see Barcelo and Casanovas (1984); Klincewicz and Luss (1986); Beasley (1993); Sridharan (1993); Agar and Salhi (1998); Hindi and Pienkosz (1999); Rönnqvist et al. (1999), Holmberg et al. (1999), Cortinhal and Captivo (2003), and Chen and Ting (2008).

There are obviously other types of FLPs for which LR was also used and which are worth mentioning here. For instance, Pirkul and Jayaraman (1998) use LR to solve a multi-commodity, multi-plant capacitated FLP. The authors split the problem into two separate problems where each one was reduced to a continuous knapsack problem. Tragantalerngsak et al. (1997) develop LR for a two-echelon SSCFLP. Mazzola and Neebe (1999) propose an interesting hybrid approach that combines branch and bound with LR heuristic to solve a multi-product CFLP. Tragantalerngsak et al. (2000) use a similar hybrid that solves a two-echelon SSCFLP to optimality. Klose (2000) put forward a

Lagrangian relax-and-cut approach for the two-stage CFLP for which the relaxation is strengthened by adding valid inequalities. Shen (2005) formulates a general multi-commodity supply chain design as a nonlinear integer programming and exploits the structure of the problem to apply LR problem. Recently, Li et al. (2009) solve a CFLP with multi-commodity flow by integrating LR with Tabu Search. Lin (2009) proposes a hybrid heuristic of Lagrangian relaxation embedded with branch and bound to tackle a stochastic version of the single-source capacitated facility location problem with service level requirements.

The purpose of the study is twofold: (i) to investigate a new variant of FLP by considering single-source and multi-product which has many practical applications, and (ii) to develop LR heuristics that incorporate new effective cuts and local searches to make the general LR methodology more efficient for solving hard combinatorial problems in general and this class of location problems in particular.

The remainder of this paper is organized as follows. In Section 2, the problem definition with its mathematical formulation and possible applications are given. In Section 3, the proposed LR heuristics are presented followed by three local searches in Section 4. Our computational results are provided in Section 5 and finally in Section 6, our conclusions are summarized along with highlights of some research avenues.

2. Problem definition

Let $J = \{1,\ldots,n\}$ be the set of customers whose demands need to be satisfied by a subset of uncapacitated facilities chosen from the set of potential sites $I = \{1,\ldots,m\}$. Each customer requires a number of commodities taken from the set of product types $K = \{1,\ldots,p\}$ and each open facility can produce one product type only. The corresponding demand of each product type for a given customer must be assigned to only one facility. There is no restriction on the capacity of the facilities. There is a fixed setup cost for each facility as well as a variable production cost that depends on the type and the amount of product.

The objective is to minimize the total cost which includes the transportation cost, the production cost and the setup cost. We aim to determine (i) the set of facilities to be opened, (ii) the product type to be produced at each of the open facilities (one type only) and finally (iii) the set of customers to be served by an open facility for a given type of product.

The notations and the mathematical model are given below:

Parameters

q_{jk} Demand of product type *k* for customer *j*, $\forall j \in J, k \in K$

c_{ijk}	Transportation cost of product type k taken from facility i to customer j, $\forall i \in I,\ j \in J, k \in K$
f_{ik}	Set up cost of producing product type k at facility i, $\forall i \in I,\ k \in K$
p_{ik}	Unit production cost of product type k at facility i, $\forall i \in I,\ k \in K$

Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if customer } j \text{ receives product type } k \text{ from facility } i \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if product type } k \text{ is produced at facility } i \\ 0 & \text{otherwise} \end{cases}$$

Mathematical Formulation

(IP)

$$Z_{\mathrm{IP}} = \mathrm{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} \gamma_{ijk} x_{ijk} + \sum_{i=1}^{m} \sum_{k=1}^{p} f_{ik} y_{ik} \tag{1}$$

$s.t.$

$$\sum_{i=1}^{m} x_{ijk} = 1 \qquad \forall j = 1,\dots,n;\ \ k = 1,\dots,p \tag{2}$$

$$x_{ijk} \le y_{ik} \qquad \forall i = 1,\dots,m\ ;\ j = 1,\dots,n\ ;\ \ k = 1,\dots,p \tag{3}$$

$$\sum_{k=1}^{p} y_{ik} \le 1 \qquad \forall i = 1,\dots,m \tag{4}$$

$$x_{ijk}, y_{ik} \in \{0,1\} \qquad \forall i = 1,\dots,m;\ j = 1,\dots,n\ ;\ k = 1,\dots,p \tag{5}$$

where $\gamma_{ijk} = (c_{ijk} + p_{ik})q_{jk}$.

The objective function (1) minimizes the total cost, constraint set (2) ensures that each customer is served by one facility for each product type only, constraint set (3) indicates that the assignments are made to the open facilities only and constraint set (4) implies that at most one product type is produced at an open facility. Constraint set (5) denotes the binary nature of the decision variables.

The proposed problem is reducible to the uncapacitated FLP by considering a single product. The latter is known to be NP-hard (Nemhauser and Wolsey, 1999) and hence the proposed problem is also NP-hard.

The proposed model has many applications in manufacturing. For instance, a company may produce numerous types of products (e.g., high tech companies). However, to take advantages of the economy of scale and multiple benefits due to sole sourcing such as the required training, reductions in product variation, cost of quality, and fixed cost of machinery, the company has to centralize the production in a single facility. This strategic view point leads to the question of where to establish such a facility with specific requirement (a given product) and which incurs setup and production costs to produce a specific product. Thus, the objective is to find the location of all types of facilities such that the overall distribution cost, production cost and setup cost is minimized.

3. The LR heuristic

The main concept of Lagrangian relaxation is to identify the set of complicating constraints of a general integer program (i.e., those which increase the computational complexity of the solution approach) and to introduce them into the objective function in a Lagrangian fashion by attaching unit penalties to them so to guide the search toward reducing the amount of constraints violation. This transformation should be constructed to render the new problem easier to solve optimally and hence produce lower bounds (see the interesting seminal paper by Geoffrion (1974)). The penalties are adjusted based on the violation and the process is repeated until a suitable stopping criterion (for instance, when the gap between the best lower and upper bound is small, a negligible change in the solution configuration is detected, the maximum computing time is reached, among others.) is met.

In our formulation, the variable upper bound constraint as defined by the constraint set (3) proves to be a facet for the convex hull of the feasible region of (IP), see (Nemhauser and Wolsey, 1999). This constraint set (3) links the **x** and **y** variables and hence can be considered to be a set of complicating constraints that could contribute in making the original problem harder to solve. In addition, this set of constraints (3), if relaxed, also has the advantage to leave the remaining problem with a structure which happens to be easier to exploit and hence to solve.

Based on the above reasoning, the constraint set (3) is dualized to provide a lower bound. A corresponding upper bound is then derived by exploiting the problem structure through the resolution of a sub-problem of (IP). In other words, at each iteration of the Lagrangian relaxation, lower and upper bounds are concurrently generated for (IP). The construction of efficient cuts and the adaptation of well-known local searches are then embedded into the search to enhance the overall efficiency of the proposed LR approaches.

3.1. Lagrangian relaxation heuristic 1 (LR1)

The Lagrangian relaxation problem can be expressed as follows:

$$Z_{\mathrm{LR}}(\boldsymbol{\alpha}) = \mathrm{Min}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}(\gamma_{ijk} + \alpha_{ijk})x_{ijk} + \sum_{i=1}^{m}\sum_{k=1}^{p}(f_{ik} - \sum_{j=1}^{n}\alpha_{ijk})y_{ik}$$

$$s.t.$$

$$\sum_{i=1}^{m} x_{ijk} = 1 \qquad \forall j = 1,...,n;\ \ k = 1,...,p$$

$$\sum_{k=1}^{p} y_{ik} \le 1 \qquad \forall i = 1,...,m$$

$$x_{ijk}, y_{ik} \in \{0,1\} \qquad \forall i = 1,...,m;\ \ j = 1,...,n;\ \ k = 1,...,p$$

where $\boldsymbol{\alpha}$ stands for the array of Lagrange multipliers.

The above formulation can be decomposed into two separable and easier to solve sub-problems, which we refer to as *Sub1* and *Sub2*, respectively. The first sub-problem, *Sub1,* which is expressed in terms of the $\mathbf{x}$ binary variables only, is given below:

(Sub1)

$$Z_{\mathrm{Sub1}}(\boldsymbol{\alpha}) = \mathrm{Min}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}(\gamma_{ijk} + \alpha_{ijk})x_{ijk}$$

$$s.t.$$

$$\sum_{i=1}^{m} x_{ijk} = 1 \qquad \forall j = 1,...,n;\ \ k = 1,...,p$$

$$x_{ijk} \in \{0,1\} \qquad \forall i = 1,...,m;\ \ j = 1,...,n;\ \ k = 1,...,p$$

This formulation postulates that for some given products, the customers are to be assigned to the set of potential sites regardless of whether or not a potential site contains an open facility. In addition, *Sub1* does not restrict each facility to produce one type of product only. As a result, an optimal solution of this sub-problem may violate the constraints sets (3) and (4) leading to an improper pattern of distribution. *Sub1* consists of np multiple choice problems that can be solved optimally (Guignard, 2003) using a simple inspection procedure. Here, it suffices to compare the objective coefficients for $j = 1,...,n$ and $k = 1,...,p$ leading to $\hat{x}_{\bar{i}jk} = 1$ where $\bar{i} = \operatorname*{argmin}_{i \in I}\{\gamma_{ijk} + \alpha_{ijk}\}$ for $j = 1,...,n$ and $k = 1,...,p$.

The second sub-problem (*Sub2*), which is defined in terms of the $\mathbf{y}$ binary variables only, is given as follows:

(Sub2)

$$Z_{\text{Sub2}}(\boldsymbol{\alpha}) = \text{Min}\sum_{i=1}^{m}\sum_{k=1}^{p}(f_{ik} - \sum_{j=1}^{n}\alpha_{ijk})y_{ik}$$

$s.t.$

$$\sum_{k=1}^{p} y_{ik} \le 1 \qquad \forall i = 1,\ldots,m$$

$$y_{ik} \in \{0,1\} \qquad \forall i = 1,\ldots,m; \quad k = 1,\ldots,p$$

Sub2 deals with the assignment of the open facilities to the products provided that the constraint set (4) is satisfied. Nevertheless, the coverage of all the product types is not necessarily guaranteed unless $f_{ik} - \sum_{j=1}^{n}\alpha_{ijk} < 0$ holds true for $i = 1,\ldots,m$ and $k = 1,\ldots,p$. In other words, the optimal solution for this sub-problem may include a given product type that happens not to be served by any open facility which is obviously an infeasible solution for (IP). Such a violation can be overcome by introducing the following set of redundant constraints:

$$x_{ijk} \le y_{ik} \qquad \forall \begin{cases} i = 1,\ldots,m \\ j = 1,\ldots,n \\ k = 1,\ldots,p \end{cases} \Rightarrow \sum_{i=1}^{m} x_{ijk} \le \sum_{i=1}^{m} y_{ik}$$

$$\Rightarrow \sum_{i=1}^{m} y_{ik} \ge 1 \qquad \forall k = 1,\ldots,p$$

To be more specific, the linear combination of the constraint set (3), which can be considered as a surrogate set of constraints, derives a set of valid inequalities, cutting away those solutions of *Sub2* which are infeasible for (IP). As such, the revised sub-problem, including the valid inequalities, is expressed as follows:

(Re - Sub2)

$$Z_{\text{Re-Sub2}}(\boldsymbol{\alpha}) = \text{Min}\sum_{i=1}^{m}\sum_{k=1}^{p}(f_{ik} - \sum_{j=1}^{n}\alpha_{ijk})y_{ik}$$

$s.t.$

$$\sum_{k=1}^{p} y_{ik} \le 1 \qquad \forall i = 1,\ldots,m$$

$$\sum_{i=1}^{m} y_{ik} \ge 1 \qquad \forall k = 1,\ldots,p \qquad (6)$$

$$y_{ik} \in \{0,1\} \qquad \forall i = 1,\ldots,m; \quad k = 1,\ldots,p$$

Re-Sub2 is introduced to generate feasible solutions $\mathbf{y}$ for (IP) given that all the products are now guaranteed to be produced by at least one open facility. More importantly, this revised sub-problem also results in generating tighter lower bounds for (IP). Note that *Re-Sub2* has similarities with the assignment problem with the important addition that the coefficient matrix induced by the set of constraints (6), matches the Total Unimodularity properties as explained in Nemhauser and Wolsey (1999). As a consequence, *Re-Sub2* satisfies the sufficient conditions of a sharp integer program where the linear programming relaxation offers systematically integral solutions as well. In other words, we can solve this sub-problem optimally and efficiently just as a pure linear programming problem.

As mentioned above, solving the second Lagrangian sub-problem, *Re-Sub2*, leads to a feasible solution $\mathbf{y}$. The following integer program then obtains the best feasible solution $\mathbf{x}$ corresponding to the decision variable $\mathbf{y}$:

(FP)

$$Z_{\text{FP}} = \text{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} \gamma_{ijk} x_{ijk} + \sum_{i=1}^{m} \sum_{k=1}^{p} f_{ik} \bar{y}_{ik}$$

$$s.t.$$

$$\sum_{i=1}^{m} x_{ijk} = 1 \qquad \forall j = 1,...,n; \quad k = 1,...,p$$

$$x_{ijk} \leq \bar{y}_{ik} \qquad \forall i = 1,...,m; \; j = 1,...,n; \; k = 1,...,p$$

$$x_{ijk} \in \{0,1\} \qquad \forall i = 1,...,m; \; j = 1,...,n; \; k = 1,...,p$$

where $\bar{\mathbf{y}}$ stands for the optimal solution of *Re-Sub2*. This ensures the satisfaction of constraint set (3). Note that the feasibility problem (FP) is the same as (IP) with the exception that $\bar{\mathbf{y}}$ is replaced by $\mathbf{y}$. Note that the optimal solution $\bar{\mathbf{x}}$ can be easily found by setting $\bar{x}_{\bar{i}jk} = 1$ for $\bar{i} = \underset{i \in I_k}{\text{argmin}} \{\gamma_{ijk}\}$ and $I_k = \{i \,|\, \bar{y}_{ik} = 1, \quad i = 1,...,m\}$ for $k = 1,..., p$.

Here, the assignment of the products to the open facilities is provided by the Lagrangian relaxation while solving (FP) which provides the best assignment of the customers to these open facilities.

3.1. Subgradient optimization

Subgradient optimization is a commonly used method to update the Lagrange multipliers. In fact, subgradient optimization can be considered as an adapted version of the gradient method. Here, the subgradients are used instead of the gradients where a subgradient direction is obtained by minimizing the dual function. The theoretical convergence properties of the subgradient method are given by Held et al. (1974).

Step 0 Initialize the parameters ($\boldsymbol{\alpha}^0$, λ_0, L_{Count}, L_{Max}, ε, $Maxiter$ and $Maxtime$). Set $Z_{LBD} \leftarrow -\infty$ and $Z_{UBD} \leftarrow \infty$.

Step 1 (a) Solve Sub1 and output $\hat{\mathbf{x}}^t$ and $Z_{\text{sub1}}(\boldsymbol{\alpha}^t) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}(\gamma_{ijk} + \alpha_{ijk}^t)\hat{x}_{ijk}^t$

(b) Solve Re-Sub2 and output $\bar{\mathbf{y}}^t$ and $Z_{\text{Re-Sub2}}(\boldsymbol{\alpha}^t) = \sum_{i=1}^{m}\sum_{k=1}^{p}(f_{ik} - \sum_{j=1}^{n}\alpha_{ijk}^t)\bar{y}_{ik}^t$.

(c) Set $Z_{\text{LR}}(\boldsymbol{\alpha}^t) = Z_{\text{Sub1}}(\boldsymbol{\alpha}^t) + Z_{\text{Re-Sub2}}(\boldsymbol{\alpha}^t)$.

If $Z_{\text{LR}}(\boldsymbol{\alpha}^t) \le Z_{LBD}$ set $L_{Count} = L_{Count} + 1$,

Otherwise set $L_{Count} = 0$.

(d) $Z_{LBD} = \max\{Z_{LBD}, Z_{\text{LR}}(\boldsymbol{\alpha}^t)\}$ as the best lower bound.

Step 2 (a) Solve FP and output $\bar{\mathbf{x}}^t$ and $Z_{\text{FP}}^t = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\gamma_{ijk}\bar{x}_{ijk}^t + \sum_{i=1}^{m}\sum_{k=1}^{p}f_{ik}\bar{y}_{ik}^t$

(b) Set $Z_{UBD} = \min\{Z_{UBD}, Z_{\text{FP}}^t\}$ as the best upper bound.

Step 3 If $t \ge Maxiter$ or $TotalCPU \ge Maxtime$ or $\frac{Z_{UBD} - Z_{LBD}}{Z_{UBD}} \le \varepsilon$, stop and output $(\bar{\mathbf{x}}^t, \bar{\mathbf{y}}^t)$ as the final solution, otherwise, go to Step 4.

Step 4 (a) Update the Lagrange multipliers using Eqs (7) to (9).

(b) If $L_{Count} \ge L_{Max}$ set $\lambda_{t+1} = \lambda_t / 2$ and $L_{Count} = 0$, otherwise set $\lambda_{t+1} = \lambda_t$.

(c) Set $t \leftarrow t + 1$ and return to Step 1.

Fig. 1. The overall framework of the proposed algorithm (LR1).

Regarding the dualized constraint set (constraint set (3)), the subgradient of the dual function with respect to the Lagrange multipliers α_{ijk}^t is represented as follows:

$$g_{ijk}^t = x_{ijk}^t - y_{ik}^t \quad \forall i = 1,\dots,m;\ j = 1,\dots,n;\ k = 1,\dots,p \tag{7}$$

According to Fisher's formula (1981), the Lagrange multipliers are then updated as follows:

$$\alpha_{ijk}^{t+1} = \text{Max}\{\alpha_{ijk}^t + S^t g_{ijk}^t, 0\} \quad \forall i = 1,\dots,m;\ j = 1,\dots,n;\ k = 1,\dots,p \tag{8}$$

where

$$S^t = \frac{\lambda_t (Z_{UBD} - Z_{LBD})}{\left\| \mathbf{g}^t \right\|^2} \tag{9}$$

Z_{UBD} and Z_{LBD} denote the best upper and lower bounds for (IP) that are obtained by the LR heuristic respectively, λ_t refers to a control parameter with $0 < \lambda_t \leq 2$ and t stands for the iteration number. The step by step of our LR heuristic, which we call *LR1*, is outlined in Fig. 1.

In this study, λ_t is halved whenever the LR heuristic fails to improve the lower bound for a number of consecutive iterations (i.e., $L_{Count} \geq L_{Max}$ where L_{Count} refers to the number of consecutive iterations without improvement in the lower bound and L_{Max} represents the maximum allowed number of consecutive iterations without improvement).

3.3. Lagrangian relaxation heuristic 2 (LR2)

The Lagrangian relaxation attempts to reduce the violation of constraint set (3) through updating the Lagrange multipliers. Nevertheless, this set of constraints links the variables $\mathbf{x}$ and $\mathbf{y}$. Therefore, such a correspondence becomes hard to maintain when dualizing this set of constraints. To overcome this limitation, we intend to tighten the polyhedron of the Lagrangian relaxation invoking the integer cuts which are usually known as the canonical cuts.

Consider the alternative formulation of the feasibility problem which we call (AFP) for short:

(AFP)

$$Z_{\mathrm{FP}} = \mathrm{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} \gamma_{ijk} x_{ijk} + \sum_{i=1}^{m} \sum_{k=1}^{p} f_{ik} y_{ik}$$

s.t.

$$\sum_{i=1}^{m} x_{ijk} = 1 \qquad \forall j = 1,\ldots,n; \quad k = 1,\ldots,p$$

$$x_{ijk} \leq y_{ik} \qquad \forall i = 1,\ldots,m; \; j = 1,\ldots,n; \; k = 1,\ldots,p$$

$$\sum_{(i,k)\in B^t} y_{ik} - \sum_{(i,k)\notin B^t} y_{ik} = \left| B^t \right|$$

$$x_{ijk} \in \{0,1\} \qquad \forall i = 1,\ldots,m; \; j = 1,\ldots,n \; ; \; k = 1,\ldots,p$$

Where $B^t = \left\{ (i,k) \middle| y_{ik}^t = 1,\ i = 1,\ldots,m,\ k = 1,\ldots,p \right\}$ represents the set of pairs (open facility, chosen product) at iteration t of the Lagrangian heuristic, and $|.|$ denotes the cardinality of the set. According to Balas and Jeroslow (1972), the canonical cut $\sum_{(i,k)\in B^t} y_{ik} - \sum_{(i,k)\notin B^t} y_{ik} \leq \left| B^t \right| \text{-}1$ removes a portion of the integer feasible region of (IP). To be more specific, the canonical cut

$\sum_{(i,k)\in B^t} y_{ik} - \sum_{(i,k)\notin B^t} y_{ik} \le \left|B^t\right| - 1$ eliminates all integer solutions $(\mathbf{x},\mathbf{y})$ for which $y_{ik}=1$ holds true when $(i,k)\in B^t$ for $i=1,\ldots,m$ and $k=1,\ldots,p$.

The Lagrangian relaxation is defined on two separate blocks where one uses the $\mathbf{x}$ binary variables and the other the $\mathbf{y}$ binary variables. The sub-problem defined on the $\mathbf{x}$ binary variables is expressed as before whereas the second sub-problem which includes the canonical cuts $\sum_{(i,k)\in B^t} y_{ik} - \sum_{(i,k)\notin B^t} y_{ik} \le \left|B^t\right| - 1$ can now be represented as follows:

$$
\begin{aligned}
&(\text{Tight-Sub2})\\
&Z_{\text{Tight-Sub2}}(\boldsymbol{\alpha}^t) = \text{Min} \sum_{i=1}^{m}\sum_{k=1}^{p}\Big(f_{ik} - \sum_{j=1}^{n}\alpha_{ijk}^{t}\Big)y_{ik}\\
&\quad s.t.\\
&\sum_{k=1}^{p} y_{ik} \le 1 \qquad \forall i=1,\ldots,m\\
&\sum_{i=1}^{m} y_{ik} \ge 1 \qquad \forall k=1,\ldots,p\\
&\sum_{(i,k)\in B^t} y_{ik} - \sum_{(i,k)\notin B^t} y_{ik} \le \left|B^{t'}\right| - 1 \qquad t'=1,\ldots,t-1\\
&y_{ik}\in\{0,1\} \qquad \forall i=1,\ldots,m;\ \ k=1,\ldots,p
\end{aligned}
$$

Note that the canonical constraints, in spite of tightening the Lagrangian feasible region, do also make the sub problem *Tight-Sub2* harder to solve as the Lagrangian heuristic proceeds. To reduce such a burden, we apply the non-accumulating method proposed by Gzara and Erkut (2009) in which only one canonical cut is kept at each iteration. In other words, at the first iteration of the LR heuristic, *Re-Sub2* is solved without considering any canonical cut. From iteration t onward, we set $B^t = \left\{(i,k)\,\middle|\, y_{ik}^t = 1,\ i=1,\ldots,m,\ k=1,\ldots,p\right\}$ if $Z_{\text{FP}}^t < Z_{UBD}$ holds true, and $B^t = B^{t-1}$ otherwise. We can say that *Tight-Sub2* keeps a typical canonical cut until the Lagrangian heuristic succeeds in improving the upper bound.

(AFP) successively generates a feasible integer solution for (IP) as a potential upper bound within the LR heuristic. This solution is considered as an incumbent if it improves the upper bound (see Fig. 1). Unfortunately, the possibility of cycling may also occur as an already found integer feasible solution may be revisited after a certain number of iterations before optimality is reached. The main advantage of the canonical cut is to avoid such a repetition as it discards the current incumbent solution along with all other inferior feasible solutions. Specifically, when $|B^t| = m$ holds true, it implies that all the facilities would contribute to make the products. Hence, due to the fact that the facilities can produce one type of product only, the canonical cut can be replaced by $\sum_{(i,k)\in B^t} y_{ik} \leq |B^t| - 1$.

In summary, the LR heuristic, which we call *LR2*, adheres to the framework elaborated in Fig. 1 except that *Tight-Sub2* is used instead of *Re-Sub2*.

Illustrative Example-

A small example with 3 potential sites, 2 customers and 2 types of products is given in the appendix to illustrate the above LR heuristic. For simplicity we provide one full cycle of the method and the optimal solution, see Appendix A for details.

4. Strengthening the upper bounds within the search

Local search methods have been widely used to improve the quality of the solutions, see for instance Agar and Salhi (1998), and Ahuja et al. (2004). In this section, we develop three local searches that are integrated into the proposed LR heuristic. These are used to improve the upper bounds by exploring the neighborhood of these obtained solutions. These local searches are the modified version of some well-known classical local searches (see, Cortinhal and Captivo (2003)) wherein multi products are taken into account. Here, a composite heuristic consisting of these three local searches is devised. In Figs 2 to 4, the location of the facilities and customers are shown by squares and black points respectively, and each color is represented by one type of products.

(i) In the first local search, 'swap', an open facility, $i1$, is removed and a closed facility, $i2$, with the same product type is opened. Obviously, all products assigned to facility $i1$ should be assigned to facility $i2$. All closed facilities are tested one at a time and the one (if any) with the least cost is chosen to replace one of the open facilities, see Fig. 2. The complexity of this local search is $O(m_0(m-m_0))$ where m_o represents the number of open facilities.

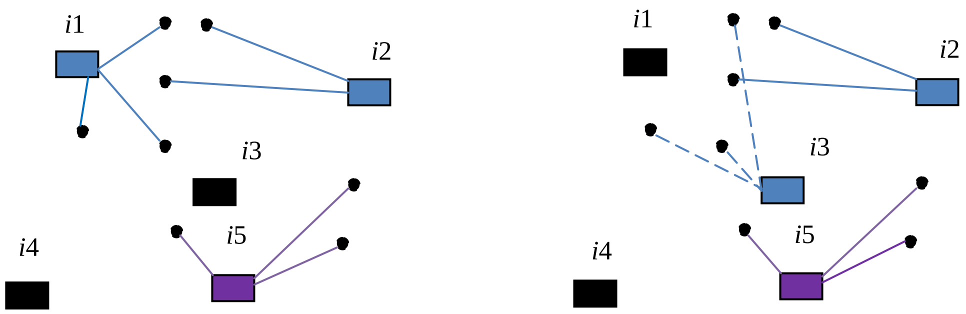

Fig. 2. The swap move.

(ii) The second local search is ‘exchange’. Here we assume that product type $k1$ is produced in facility $i1$ and product $k2$ is produced in facility $i2$. Then, we assign all customers from facility $i1$ to facility $i2$ and vice versa. To maintain feasibility, facility $i1$ should produce product $k2$ and vice versa, see Fig. 3. This is tested for all combinations leading to a time complexity for this local search of $O(m_0^2)$.

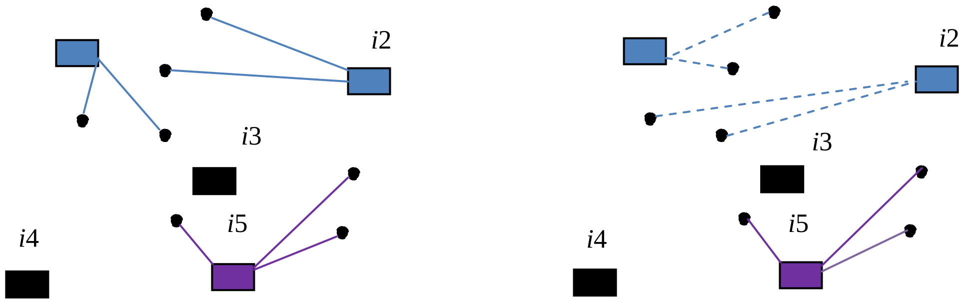

Fig. 3. The exchange move.

(i) In the third local search known as ‘add’, a closed facility is opened, and the best type of products (if any) which can be produced in this new open facility is selected. Moreover, the corresponding demand of those customers that are closer to this new facility are assigned accordingly, see Fig. 4. Here, all closed facilities as well as all types of products are tested to find the best alternative for the opening of the new facility. The complexity of this local search is $O(np(m-m_0))$.

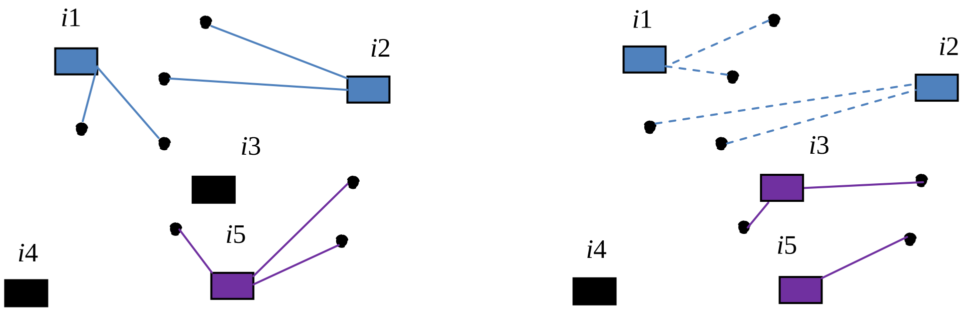

Fig. 4. The add move.

Based on this composite heuristic, when a better solution is found by a given local search, this is considered as the incumbent solution and the same local search is repeated until there is no improvement where the next local search starts. After applying all the three local searches, the resulting y is then fed into the feasibility problem (FP) to determine the best corresponding $\mathbf{x}$. The cycle reverses back to the first local search and the process continues till there is no improvement after one full cycle of the application of all the three local searches. In order to speed up the process, this composite heuristic is used when a better upper bound is obtained in Step 2b of Fig. 1 only.

5. Computational experiments

To the best of our knowledge, there are no instances publicly available for the proposed problem. We therefore base the construction of our data sets on modifying instances found in related studies. We generated 3 data sets which are used as a platform to assess the performance of our LR heuristics. We refer to these sets as three classes where the first one has 17 instances and the remaining two consist of 15 instances each. These well-known data sets are available at mpi-inf (2013).

The proposed algorithms were coded in MATLAB 7 and the programs were run on a Core 2 Due @2.4 GHZ Notebook with 2 GB RAM. In order to solve Re-Sub2, and Tight-Sub2, we employed the dual simplex and the branch and cut algorithm of CPLEX with all default options for linear programming and integer programming methods. In our experiments we set the values of the following parameters: $\lambda_0 = 2$, $\boldsymbol{\alpha}^0 = 0$, $\varepsilon = 0.0001$, $Maxtime = 1500$seconds of CPU time, $Maxiter = 1000$ iterations and $L_{Max} = 50$.

We solved the original problem IP by the branch and cut algorithm of CPLEX with automatic cut generation capability. To accelerate the computational performance of CPLEX, the barrier algorithm was chosen to solve the linear programming relaxation at the root node instead of the dual simplex algorithm. Based on our experimental results, the barrier algorithm has a much better performance when compared to the dual simplex algorithm especially for large scale location type instances. A reduction of approximately tenfold in computation time was observed when solving the initial relaxation. For CPLEX, we used default options but to expedite the performance, the optimality gap (ε) is set to 0.0001 and the maximum solution time is set to 2000 seconds.

5.1. Generation of the data sets

Class I test instances

In class I instances, the number of potential facilities, the number of customers with their respective coordinates and demand of customers were extracted from the well-known data sets provided by Barreto et al. (2007). These are originally used for location-routing problems where d_j represents the demand of customer *j* and $dist_{ij}$ indicates the Euclidian distance between facility *i* and customer *j*. There are 17 instances varying in size from 21 to 117 customers, 5 to 15 potential facilities and 2 to 12 products. We have generated the necessary data for the proposed model using the following formulas:

$$c_{ijk} = 0.001\rho_k dist_{ij} \quad \forall i = 1,\ldots,m;\ j = 1,\ldots,n;\ k = 1,\ldots,p$$

$$\alpha_{ik} = \left\lceil \frac{i+k}{\sqrt{(i^2+k^2)}} \log(ik) \right\rceil \quad \forall i = 1,\ldots,m;\ k = 1,\ldots,p$$

$$\rho_k = 1 + 0.01k\log(k) \text{ for } k = 1,\ldots,p \ \ (k \text{ is in radian})$$
$$f_{ik} = \lceil F(1+\alpha_{ik}) \rceil \quad \forall i = 1,\ldots,m;\ \ k = 1,\ldots,p$$
$$p_{ik} = \lceil S(1+\alpha_{ik}) \rceil \quad \forall i = 1,\ldots,m;\ \ k = 1,\ldots,p$$
$$q_{jk} = \lceil d_j(1+\cos(jk)) \rceil \quad \forall j = 1,\ldots,n\ ;\ \ k = 1,\ldots,p$$

where

$$F = 10F_0\ ,\ S = 2F_0 \text{ and } F_0 = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} c_{ijk}}{mnp} \ .$$

Class II test instances

With regard to class II, we have modified two well-known sets of instances originally constructed for the uncapacitated FLP. The first one is the data sets presented by Bilde and Krarup (1977). Here, the fixed cost values vary from 1000 to 10000. For most problems, the fixed costs are set to unity for all facilities whereas the connection costs are randomly and uniformly generated from a chosen range. In the second one, the instances are generated by Kochetov and Ivanenko (2005) and known as the Euclidian instances. Here, the customer points are randomly chosen in a square with a side of 7000. There are 15 instances having 80 to 100 customers, 30 to 100 potential sites and 15 to 40 products. The opening costs are denoted by $F_i = 3000$ for $i = 1,\ldots,m$. The other parameters are generated based on the following formulas

$$\rho_k = 0.0\log(k) \quad \forall k = 1,\ldots,p \ \ (k \text{ in radian})$$
$$c_{ijk} = 8\rho_k dist_{ij} \quad \forall i = 1,\ldots,m\ ;\ \ j = 1,\ldots,n;\ \ k = 1,\ldots,p$$
$$f_{ik} = \eta\lceil 0.1F_i \rceil \left|\cos(i\sqrt{\pi}/k)\right| \quad \forall i = 1,\ldots,m;\ \ k = 1,\ldots,p$$
$$p_{ik} = \lceil 0.1\nu F_i \rceil \left|\sin(i\sqrt{\pi}/k)\right| \quad \forall i = 1,\ldots,m\ ;\ \ k = 1,\ldots,p$$
$$q_{jk} = 8(5+\cos(jk)) \quad \forall j = 1,\ldots,n;\ \ k = 1,\ldots p$$

where (η, ν) are non-negative control parameters related to the proportions of fixed cost, production cost and transportation cost.

Through our empirical examination, we found that both the fixed cost and the production cost have a substantial impact on the complexity of the problem. This observation has inspired us to generate other instances by testing several values of η and ν. In this class, we set η and ν in a well chosen range so to produce relatively more difficult instances as will be shown by their respective high optimality gap values.

Class III test instances

In this class, we modified the 15 instances of class II to assess the impact of the parameters on the optimality gap (i.e., GAP), and to find out whether or not LR2 is more effective than LR1. These generated instances are made available at CLHO (2013).

$$f_{ik} = \eta F_0 \quad \forall i = 1,\ldots,m;\ k = 1,\ldots,p$$
$$p_{ik} = \nu F_0 \quad \forall i = 1,\ldots,m;\ \ k = 1,\ldots,p$$
$$q_{jk} = 0.01\sqrt{12(15+\sin(jk))} \quad \forall j = 1,\ldots,n;\ k = 1,\ldots,p$$

where $\quad F_0 = \dfrac{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} c_{ijk}}{mnp}$

5.2. Computational Results

Let LB (LR1) and UB (LR1) denote the lower and upper bounds of the first Lagrangian relaxation (*LR1*) respectively whereas LB (LR2) and UB (LR2) refer to *LR2*. In a similar way, LB (CPLEX) and UB (CPLEX) represent the lower and upper bounds found by CPLEX. Similarly, LB (*LR1+*), UB (*LR1+*), LB (*LR2+*) and UB (*LR2+*) represent the lower and upper bounds for *LR1* and *LR2* with the addition of local search. The best upper bound among all the proposed methods including CPLEX is denoted by UB (BEST). For small size instances this refers to the optimal solution found by CPLEX.

Let GAP(*X*) represents the gap in (%) for method *X* between its best upper and lower bounds denoted by UB(*X*) and LB(*X*) respectively (*X* refers to CPLEX, *LR*, or *LR+* where *LR* denotes either *LR1* or *LR2*).

$$\mathrm{GAP}(X) = 100\frac{\mathrm{UB}(X) - \mathrm{LB}(X)}{\mathrm{UB}(X)}$$

Similarly Dev(*X*) represents the deviation in % between the upper bound found by method *X* and the overall best upper bound UB (BEST). This is defined as

$$\mathrm{Dev}(X) = 100\frac{\mathrm{UB}(X) - \mathrm{UB(BEST)}}{\mathrm{UB(BEST)}}$$

Tables 1-3 summarize the computational results for the three classes but detailed information is available based on request. In each table, the last five rows show the number of best solutions, the average of each column, the standard deviation, the number of solutions with a deviation less than 0.1%, and the number of solutions with a deviation of less than 1%.

Class I results

In Table 1, we have compared the results of Lagrangian heuristics as well as CPLEX for class I. The results indicate that CPLEX reaches the optimal solution in all instances with an average time of 1.32 seconds. This shows that CPLEX is efficient to handle this set of small size instances. Regarding *LR1* with and without local searches, the average optimality gaps are 0.021 and 0.022 only with an average deviation of 0.001 for both. With regard to *LR2* and *LR2* with local search, the average optimality gaps are 0.024 and 0.022 but the average deviations are only 0.001 and 0.000 respectively showing that *LR2* with local search achieved optimality in all the 17 instances except in one instance where a negligible deviation is observed (i.e., instance # 10). This reinforces the idea that incorporating local searches within *LR* does improve the upper bound. It was also observed that those good quality solutions can be generated at the beginning of the search which could be used as a guide for controlling the number of iterations if necessary. In summary, the performance of *LR2* with local search is superior in comparison to the other *LR* heuristics. However, for these instances though it obtains the optimal solutions, it requires relatively more CPU time than CPLEX. This feature of CPLEX, as will be demonstrated in the next two classes, will not be maintained.

Class II results

The obtained results in Table 2 show that CPLEX is not capable of dealing with this set of instances in a reasonable amount of computation time. This deficiency is reflected by the weak upper bound found in some cases though the lower bound of CPLEX remains relatively tighter than ours. The average optimality gap and deviation of CPLEX are 7.45, and 0.76 respectively. *LR1* without and with local searches produce average optimality gaps of 8.38 and 8.25 with corresponding average deviations of 1.05 and 0.77, respectively. Concerning *LR2* without and with local search, the average optimality gaps are 8.44 and 8.21, and the average deviations are 1.1 and 0.74, respectively. It is shown that the increase in the number of products drastically impacts the practical difficulty in solving the problem. This is illustrated by the sharp increase in the optimality gap. As in class I, the performance of *LR2* with local search outperforms other heuristic methods in terms of the average optimality gap as well as the percentage deviation. It can be observed that this method also outperforms CPLEX in terms of average deviation but CPLEX has a better performance in terms of average optimality gap which is mainly due to the tighter lower bound of CPLEX. In this class, we can observe not only a sharp increase in the computation time required by CPLEX but also a deterioration in the quality of its upper bound.

Class III results

We have presented the class III instances to assess the impact of the parameters on the level of difficulty in solving the problem. The obtained results in Table 3 indicate that the average optimality gap of CPLEX is 8.90 with a corresponding average deviation of 3.81, respectively. The average optimality gaps of *LR1* without and with local searches are 7.77 and 6.24 and the average deviations are 2.12 and 0.16. Similarly *LR2,* without and with local search, produces average optimality gaps of 7.69 and 6.79, and average deviations of 2.02 and 0.84, respectively.

This class underlines the fact that the classical formulation using CPLEX becomes less and less appropriate when solving difficult instances especially in terms of determining good upper bounds. In some cases, CPLEX cannot even reach a feasible solution within the maximum allowed time of 2000 seconds. Again here, the results reinforce the idea that incorporating local searches can have a positive impact on improving the quality of the solutions in terms of optimality gap as well as the percentage deviations. All the *LR* heuristics outperform CPLEX in terms of both the average optimality gap and the average deviation. *LR2* provides better results than *LR1* indicating that the canonical cuts are effective. Nevertheless, according to these results, *LR1* combined with the local search has shown to be slightly better than the other proposed *LR* heuristics in terms of both optimality gap and average deviation. However, *LR2* with local search still dominates CPLEX and the other LR heuristics in terms of the number of best upper bounds.

Another observation is that the decrease in the variance of the fixed cost parameters leads to generating powerful cuts for *LR2* as this aims to avoid exploring already considered facilities due to the canonical cuts. This is mainly because *Re-Sub2* is converted equivalently to an integer program in which the Lagrange multipliers guide the search toward the neighborhood of the optimal solution. Such a case may exert a repetition after a number of successive iterations.

Also, it can be noted that the duality gap increases when the values of the parameters ν and η are reduced. On the other hand when ν and η increase, the fixed cost becomes relatively high and as a result the number of open facilities will systematically and relatively be smaller and hence easier to find optimally. It can also be noted that by decreasing the production costs, the number of product types increases which makes it harder for the search to reach the optimal product assignment.

Table 1
The computational results for class I instances.

				CPLEX +		LR1			LR1+			LR2			LR2+		
#	n	m	P	UB	CPU(s)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)
1	21	5	2	**48760.68**	0.05	0.010	0.000	1.04	0.010	0.000	0.51	0.009	0.000	0.97	0.009	0.000	0.67
2	22	5	3	**22440.72**	0.19	0.010	0.000	1.25	0.010	0.000	1.19	0.009	0.000	0.94	0.010	0.000	1.69
3	29	5	4	**55761.30**	0.08	0.008	0.000	0.79	0.010	0.000	1.82	0.008	0.000	0.86	0.010	0.000	1.84
4	32	5	5	**114349.69**	0.17	0.008	0.000	16.82	0.010	0.000	19.23	0.008	0.000	11.88	0.010	0.000	11.43
5	36	5	5	**3582.29**	0.22	0.010	0.000	1.96	0.010	0.000	3.97	0.009	0.000	3.54	0.009	0.000	4.27
6	50	5	5	**3161.75**	0.26	0.009	0.000	2.51	0.010	0.000	4.56	0.010	0.000	3.47	0.010	0.000	4.12
7	75	10	8	**12075.28**	2.02	0.009	0.000	15.02	0.010	0.000	28.63	0.010	0.000	23.67	0.013	0.000	66.08
8	100	10	9	**13135.73**	3.01	0.009	0.000	39.49	0.010	0.000	48.71	0.009	0.000	17.90	0.010	0.000	66.72
9	12	2	2	**524.43**	0.01	0.007	0.000	0.49	0.007	0.000	0.32	0.000	0.000	0.18	0.000	0.000	0.10
10	55	15	12	**13143.04**	2.36	0.004	0.001	50.66	0.023	0.019	63.95	0.007	0.002	49.91	0.010	0.001	24.67
11	85	7	6	**10128.93**	0.70	0.009	0.000	5.43	0.010	0.000	6.12	0.009	0.000	4.53	0.010	0.000	13.71
12	318	4	4	**31146511.76**	0.90	0.005	0.000	6.21	0.005	0.000	6.24	0.003	0.000	3.99	0.003	0.000	4.02
13	27	5	2	**26699.36**	0.11	0.010	0.000	1.30	0.010	0.000	1.03	0.010	0.000	0.46	0.010	0.000	0.38
14	34	8	6	**89316.89**	1.21	0.010	0.000	14.95	0.010	0.000	49.60	0.010	0.000	27.37	0.010	0.000	50.92
15	88	8	8	**380294636.11**	1.38	0.092	0.000	71.68	0.119	0.000	67.88	0.118	0.000	84.82	0.082	0.000	109.40
16	150	10	10	**5312709655.55**	4.25	0.134	0.023	130.12	0.088	0.000	154.01	0.148	0.010	175.52	0.121	0.000	180.80
17	117	14	12	**24546.82**	5.49	0.017	0.000	149.73	0.030	0.000	147.72	0.037	0.000	225.01	0.044	0.000	176.56
# Best solutions				**17**			15			16			15			16	
Ave.					1.32	0.021	0.001	29.97	0.022	0.001	35.62	0.024	0.001	37.35	0.022	0.000	42.20
Std.					1.62	0.035	0.005	46.2	0.031	0.004	49.25	0.041	0.002	65.78	0.032	0.000	60.18
Prob. (0.1%) *							17			17			17			17	
Prob. (1%) **							17			17			17			17	

+ CPLEX guarantees the optimal solution for all instances.

* The number of instances whose deviations dip below 0.1%.

Table 2
The computational results for class II instances.

#	n	m	p	η	ν	UB(BEST)	LB(BEST)	CPLEX GAP	CPLEX Dev	CPLEX CPU(s)	LR1 GAP	LR1 Dev	LR1 CPU(s)	LR1+ GAP	LR1+ Dev	LR1+ CPU(s)	LR2 GAP	LR2 Dev	LR2 CPU(s)	LR2+ GAP	LR2+ Dev	LR2+ CPU(s)
B1.2	100	50	20	4	5	**4646350.02**	**4204368.18**	9.69	0.19	2000	10.15	0.00	270.08	10.34	0.19	330.25	10.15	0.00	322.00	10.34	0.19	324.84
B1.4	100	50	25	1	4	**7252438.01**	**6659568.29**	10.16	2.21	2000	9.65	0.43	384.94	8.98	0.00	406.13	9.65	0.43	387.84	8.98	0.00	416.88
B1.6	100	50	30	0	3	**10338547.73**	**9853587.03**	4.78	0.09	2000	5.23	0.00	394.12	6.15	0.85	496.57	5.99	0.57	476.32	5.94	0.74	457.40
C1.1	100	50	25	3	4	**6751291.49**	**5645216.66**	16.38	0.00	2000	20.38	3.92	283.55	20.28	3.38	316.39	20.38	3.92	397.16	20.28	3.38	437.35
C1.3	100	50	30	0.3	3	**9865435.37**	**8413468.82**	15.76	1.23	2000	15.76	0.17	373.13	15.89	0.00	440.81	15.76	0.17	407.82	15.89	0.00	457.09
C1.5	100	50	35	0	2	**13434197.00**	**12129724.10**	10.19	0.54	2000	12.09	1.52	419.43	11.54	0.40	477.22	12.13	1.59	477.17	11.13	0.00	603.1
D1.1	80	30	27	5	2	**8740157.655**	**8466238.63**	3.18	0.05	2000	4.21	0.10	298.60	4.11	0.00	305.99	4.21	0.10	320.66	4.11	0.00	326
D5.5	80	30	28	2	1	**17434782.03**	**17434782.03**	0.00	0.00	33.43	0.36	0.06	245.54	0.63	0.11	270.41	0.50	0.12	303.28	0.63	0.11	327.7
D10.10	80	30	29	0.01	1	**48574841.16**	**48574841.16**	0.00	0.00	31.12	0.10	0.01	292.62	0.53	0.21	280.78	0.11	0.01	349.07	0.53	0.21	367.85
E1.2	100	50	30	4	5	**9779846.02**	**7815656.61**	24.18	5.40	2000	21.39	0.00	399.11	22.67	1.38	424.01	21.39	0.00	432.22	22.67	1.38	519.180
E5.4	100	50	35	0.2	4	**14071557.74**	**13014296.45**	8.62	1.22	2000	8.18	0.00	427.78	8.41	0.30	459.22	8.18	0.00	523.48	8.41	0.30	574.380
E9.8	100	50	40	0.01	3	**21850868.66**	**21523697.46**	1.50	0.00	2000	2.42	0.67	645.38	2.40	0.55	659.30	2.42	0.67	534.09	2.40	0.55	626.38
511EuclS	100	100	25	5	4	**28650738.26**	**27544949.18**	3.86	0.00	2000	9.72	6.13	596.24	7.42	3.39	730.00	9.72	6.13	711.47	7.42	3.39	847.48
1511EuclS	100	100	20	0.8	3	**18276778.59**	**17734127.61**	3.48	0.53	2000	5.72	2.64	431.63	3.38	0.00	391.65	5.72	2.64	407.24	3.38	0.00	562.77
2511EuclS	100	100	15	0.2	2	**10754899.72**	**10754464.04**	0.00	0.00	209.312	0.29	0.09	329.22	1.05	0.80	386.41	0.29	0.09	362.47	1.05	0.80	380.78
# Best solutions								12	6		2	4		0	4		3	3		0	5	
Ave.								7.45	0.76	1618.26	8.38	1.05	386.09	8.25	0.77	425.01	8.44	1.10	427.49	8.21	0.74	481.95
Std.								7.10	1.43	791.22	6.84	1.82	113.41	6.95	1.13	130.87	6.81	1.80	106.14	6.94	1.14	143.70
Prob. (0.1%) *									8			8			4			6			5	
Prob. (1%) **									11			11			12			11			12	

** The number of instances whose deviations dip below 1%.

* The number of instances whose deviations dip below 0.1%.

** The number of instances whose deviations dip below 1%.

Table 3
The computational results for class III instances.

								CPLEX			LR1			LR1+			LR2			LR2+		
#	N	M	p	η	ν	UB(BEST)	LB(BEST)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)	GAP	Dev	CPU(s)
B1.2	100	50	30	2	1	80752.62	73469.17	10.01	1.10	2000	9.65	0.36	322.76	9.31	0.00	437.80	9.90	0.64	368.38	9.81	0.30	425.84
B1.2	100	50	30	2	0.8	73074.29	65431.66	11.10	0.72	2000	11.93	0.96	349.89	11.12	0.37	399.34	10.88	0.04	388.77	11.07	0.00	403.07
B1.2	100	50	30	2	0.4	56723.46	49356.62	14.20	1.41	2000	14.61	1.35	353.07	14.32	0.94	346.99	13.70	0.00	350.78	14.36	1.01	421.09
C1.3	100	50	30	5	10	451939.79	446545.10	-	-	2000	1.36	0.09	293.97	1.24	0.00	370.21	1.34	0.10	365.81	1.27	0.00	444.76
C1.3	100	50	30	5	1	92852.55	87457.87	-	-	2000	6.42	0.46	285.62	6.03	0.00	356.26	6.64	0.62	372.00	6.04	0.00	360.32
C1.3	100	50	30	5	0.1	56943.82	51549.14	-	-	2000	10.45	0.77	338.27	10.07	0.00	335.27	10.43	0.74	380.01	9.84	0.00	371.58
D10.10	80	30	25	0.1	20	517245.50	515108.76	0.48	0.07	2000	0.53	0.04	164.25	0.49	0.03	168.91	0.49	0.03	178.70	0.46	0.00	205.28
D10.10	80	30	25	0.1	1	47142.96	44984.60	5.18	0.63	2000	5.44	0.28	183.60	5.55	0.14	176.82	5.38	0.28	226.99	5.12	0.00	225.27
D10.10	80	30	25	0.1	0.05	23441.75	21517.07	10.10	2.10	2000	10.30	1.01	175.64	10.16	0.00	173.70	11.33	1.34	198.06	10.97	1.06	203.36
E1.2	100	50	25	10	5	210948.44	210174.66	8.71	9.14	2000	0.48	0.05	312.36	0.46	0.00	340.18	0.51	0.04	377.07	0.73	0.00	360.95
E1.2	100	50	25	1	5	183596.21	177656.94	3.62	0.40	2000	3.65	0.32	379.90	3.56	0.27	389.15	3.88	0.57	356.86	3.29	0.00	329.48
E1.2	100	50	25	0.1	5	179858.91	173421.51	3.71	0.14	2000	3.68	0.00	358.12	3.72	0.04	479.93	3.69	0.01	319.74	3.93	0.19	420.11
1511EuclS	100	100	20	20	0.1	396809.87	396809.87	0.00	0.00	173.83	0.79	0.02	390.89	0.92	0.00	348.16	0.12	0.04	383.59	0.26	0.00	408.94
1511EuclS	100	100	20	1	0.1	135738.68	124200.30	22.05	17.38	2000	15.25	7.76	399.10	9.28	0.65	563.93	14.69	6.93	499.89	8.69	0.00	519.14
1511EuclS	100	100	20	0.05	0.1	79008.11	73320.75	17.64	12.68	2000	21.99	18.26	412.96	7.40	0.00	584.45	22.41	18.85	501.77	15.96	10.07	508.41
# Best solutions								1	1		1	1		6	7		3	1		5	7	
Ave.								8.90+	3.81+	1878.25	7.77	2.12	314.69	6.24	0.16	364.74	7.69	2.02	351.23	6.79	0.84	373.84
Std.								6.8	5.88	471.51	6.41	4.86	81.28	4.41	0.28	125.54	6.41	4.97	92.32	5.16	2.57	98.301
Prob. (0.1%)*									2			5			10			7			10	
Prob. (1%) **									6			11			15			12			12	

+ The average is given for the instances with known upper bound.

- Indicates that no integer solution was found within 2000 seconds.

* The number of instances whose deviations dip below 0.1%.

** The number of instances whose deviations dip below 1%.

6. Conclusions and Suggestions

In this paper, we investigated the uncapacitated single-source multi-product production/distribution facility location problem with setup cost. Two *LR* heuristics are developed and strengthened by the introduction of new cuts to improve the bounds. In addition, we incorporated some local searches to improve the upper bounds which in turn speed up the search process. The proposed algorithms are tested on a large set of instances with various characteristics. Computational results indicate that the proposed *LR* heuristics are capable of dealing with this difficult location problem efficiently. Encouraging results are obtained when tested against ILP formulation using CPLEX. Our numerical results suggest that the inclusion of the canonical cuts improve the bounds on *LR2* though not always tight enough to dominate *LR1*. This study reinforces the idea that *LR* based heuristics can be a powerful tool for tackling hard combinatorial problems in general and complex variants of FLPs in particular. As a by-product of this study, our approaches also provide, for this particular class of FLP, competitive results for a large set of instances well suited for future benchmarking purposes.

The following research avenues are, in our view, worth highlighting. It can be noted that further reduction in computational time could be achieved in the implementation of our local searches if reduced neighborhoods were used instead of the entire neighborhoods. The extraction of strong valid integer cuts, derived from the problem formulation, within the Lagrangian relaxation can also be an interesting research direction. One exciting research avenue would be to design and analyze a hybrid search where meta-heuristics such as variable neighborhood search or *GRASP* are used as our local search operators within the *LR* approaches. As these powerful meta-heuristics usually require relatively more execution time, identifying when to use them and for how long can be one of the questions that need answering. In terms of related location problems, the proposed approaches can be extended to cater for the case where the facilities have a limited capacity or when the firm is restricted by budget constraints.

Acknowledgments- We would like to thank the referees and the editor for their constructive comments that improved both the content as well as the presentation of the paper. This research has been in part financed by the Algerian Ministry of Education (Sciences Fondamentales) under the research project PNR 8/U160/64.

References

Agar, M.C., Salhi, S., 1998. Lagrangean heuristics applied to a variety of large capacitated plant location problems. The Journal of the Operational Research Society 49, 1072-1084.

Ahuja, R.K., Orlin, J.B., Pallottino, S., Scaparra, M.P., Scutellà, M.G., 2004. A multi-exchange heuristic for the single-source capacitated facility location problem. Management Science 50, 749-760.

Balas, E., Jeroslow, R., 1972. Canonical cuts on the unit hypercube. SIAM Journal on Applied Mathematics 23, 61-69.

Barcelo, J., Casanovas, J., 1984. A heuristic Lagrangean algorithm for the capacitated plant location problem. European Journal of Operational Research 15, 212-226.

Barreto, S., Ferreira, S., Paixao, J., Santos, B.S., 2007. Using clustering analysis in a capacitated location-routing problem. European Journal of Operational Research 179, 968-977.

Beasley, J.E., 1993. Lagrangean heuristics for location problems. European Journal of Operational Research 65, 383-399.

Bilde, O., Krarup, J., 1977. Sharp lower bounds and efficient algorithms for the simple plant location problem. Annals of Discrete Mathematics 1, 79-97.

Chen, C.-H., Ting, C.-J., 2008. Combining Lagrangian heuristic and ant colony system to solve the single source capacitated facility location problem. Transportation Research Part E: Logistics and Transportation Review, 44(6), 1099–1122.

CLHO (2013) centre for Logistics & heuristic optimisation, http://www.kent.ac.uk/kbs/research/research-centres/clho/datasets.html.

Cortinhal, M.J., Captivo, M.E., 2003. Upper and lower bounds for the single source capacitated location problem. European Journal of Operational Research 151, 333-351.

Drezner, Z., Hamacher, H., 2002. Facility location: applications and theory Springer Verlag.

Fisher, M.L., 1981. The Lagrangian relaxation method for solving integer programming problems. Management Science 50, 1861-1871.

Geoffrion, A.M., 1974. Lagrangean relaxation for integer programming. Mathematical Programming Study 2, 82-114.

Guignard, M., 2003. Lagrangean relaxation. Top 11, 151-200.

Gzara, F., Erkut, E., 2009. A Lagrangian relaxation approach to large-scale flow interception problems. European Journal of Operational Research 198, 405-411.

Held, M., Karp, R.M., 1970. The traveling salesman problem and minimum spanning trees. Operations Research 18, 1138-1162.

Held, M., Karp, R.M., 1971. The traveling salesman problem and minimum spanning trees: Part II. Mathematical Programming 1, 6-25.

Held, M., Wolf, P., Crowder, H.D., 1974. Validation of subgradient optimization Mathematical Programming 6, 62-88.

Hindi, K.S., Pienkosz, K., 1999. Efficient solution of large scale, single-source, capacitated plant location problems. The Journal of the Operational Research Society 50, 268-274.

Holmberg, K., Ronnqvist, M., Yuan, D., 1999. An exact algorithm for the capacitated facility location problems with single sourcing. European Journal of Operational Research 113, 544-559.

Klincewicz, J.G., Luss, H., 1986. A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints. The Journal of the Operational Research Society 37, 495-500.

Klose, A., 2000. A Lagrangean relax-and-cut approach for the two-stage capacitated facility location problem. European Journal of Operational Research 126, 408-421.

Kochetov, Y., Ivanenko, D., 2005. Computationally difficult instances for the uncapacitated facility location problem. Metaheuristics: Progress as Real Problem Solvers, 351-367.

Li, J., Chu, F., Prins, C., 2009. Lower and upper bounds for a capacitated plant location problem with multicommodity flow. Computers & Operations Research 36, 3019-3030.

Lin, C.K.Y., 2009. Stochastic single-source capacitated facility location model with service level requirements. International Journal of Production Economics 117, 439-451.

Mazzola, J.B., Neebe, A.W., 1999. Lagrangean-relaxation-based solution procedures for a multiproduct capacitated facility location problem with choice of facility type. European Journal of Operational Research 115, 285-299.

mpi-inf (2013), Max-Planck-Institut für Informatik http://www.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/packages.html.

Nemhauser, G.L., Wolsey, L.A., 1999. Integer and combinatorial optimization. Wiley, New York.

Nickel, S., Puerto, J., 2005. Location theory: a unified approach. Springer.

Pirkul, H., Jayaraman, V., 1998. A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution. Computers and Operations Research 25, 869-878.

Rönnqvist, M., Tragantalerngsak, S., Holt, J., 1999. A repeated matching heuristic for the single-source capacitated facility location problem. European Journal of Operational Research 116, 51-68.

Salhi, S., 2006. Heuristic Search: The Science of Tomorrow. OR48 Keynote Papers, (S.Salhi, Ed).

Shen, Z.J.M., 2005. A Multi-commodity Supply Chain Design Problem. IIE Transactions 37, 753-762.

Sridharan, R., 1993. A Lagrangian heuristic for the capacitated plant location problem with single source constraints. European Journal of Operational Research 66, 305-312.

Tragantalerngsak, S., Holt, J., Rönnqvist, M., 1997. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. European Journal of Operational Research 102, 611-625.

Tragantalerngsak, S., Holt, J., Rönnqvist, M., 2000. An exact method for the two-echelon, single-source, capacitated facility location problem. European Journal of Operational Research 123, 473-489.

Appendix A. A simple illustrative example using LR

The following small example is used to illustrate the use of our LR algorithm. We consider 3 potential sites, 2 customers and 2 product types whose parameters are given below:

$m=3$, $n=p=2$; $f_{11}=f_{12}=f_{21}=f_{22}=f_{31}=f_{32}=300$

$p_{11}=9.2,\ p_{12}=8.7,\ p_{21}=9.6,\ p_{22}=8.7,\ p_{31}=8.3,\ p_{32}=6.9$

$q_{11}=25,\ q_{12}=26,\ q_{21}=10,\ q_{22}=5$

$c_{111}=5.2,\ c_{112}=9,\ c_{121}=8.9,\ c_{122}=6.79$

$c_{211}=19.7,\ c_{212}=12.5,\ c_{221}=11.5,\ c_{222}=15.5$

$c_{311}=17.8,\ c_{312}=19.4,\ c_{321}=10.4,\ c_{322}=6.5$

The LR algorithm

Step 0: Initialization

Set $\boldsymbol{\alpha}^0=0$, $\lambda_0=2$, $\varepsilon=0.0001$, $\delta=0.00001$, $Z_{LBD}\leftarrow-\infty$ and $Z_{UBD}\leftarrow\infty$ and $t\leftarrow 0$.

Step 1

a) Solve the first sub-problem (*Sub1*):

$$\begin{aligned}&\text{Min}\sum_{j=1}^{2}\sum_{k=1}^{2}\gamma_{1jk}x_{1jk}+\gamma_{2jk}x_{2jk}+\gamma_{3jk}x_{3jk}\\&\quad s.t.\\&x_{1jk}+x_{2jk}+x_{3jk}=1 \qquad j=1,.2\quad k=1,.2\\&x_{ijk}\in\{0,1\} \qquad i=1,.2,3;\ j=1,2\ ;\ k=1,2\end{aligned}$$

According to the *multiple choice* approach, the optimum solution is: $Z_{\text{Sub1}}(0)=107.602$

$\hat{x}^0_{111}=1,\ \hat{x}^0_{112}=1,\ \hat{x}^0_{121}=1,\ \hat{x}^0_{122}=0 \qquad \hat{x}^0_{211}=\hat{x}^0_{212}=\hat{x}^0_{221}=\hat{x}^0_{222}=0 \qquad \hat{x}^0_{311}=\hat{x}^0_{312}=\hat{x}^0_{321}=0,\hat{x}^0_{322}=1$

b) Solve the second sub-problem (*Re-Sub2*):

$$\begin{aligned}&\text{Min}\sum_{i=1}^{m}f_{i1}y_{i1}+f_{i2}y_{i2}\\&\quad s.t.\\&y_{i1}+y_{i2}\le 1 \qquad i=1,.2,3\\&y_{1k}+y_{2k}\ge 1 \qquad k=1,.2\\&y_{ik}\in\{0,1\} \qquad \forall i=1,.2,3;\ k=1,.2\end{aligned}$$

The optimal solution is equal to $Z_{\text{Re-Sub2}}(0)=600$ and $\bar{y}^0_{11}=\bar{y}^0_{12}=\bar{y}^0_{21}=\bar{y}^0_{32}=0,\ \bar{y}^0_{22}=\bar{y}^0_{31}=1$

c) At this point, the Lagrangian relaxation obtains $Z_{\mathrm{LR}}(0) = 1076.02 + 600 = 1676.02$ as the lower bound:

d) $Z_{LBD} = \max\{Z_{LBD}, 1676.02\} = 1676.02$

Step 2

a) To obtain a feasible integer solution and thus an upper bound, given $\bar{\mathbf{y}}^0$, tackle the feasibility problem *FP*:

$$
\begin{aligned}
&\mathrm{Min} \sum_{j=1}^{2}\sum_{k=1}^{2} \gamma_{1jk} x_{1jk} + \gamma_{2jk} x_{2jk} + \gamma_{3jk} x_{3jk} + f_{22} + f_{31} \\
&\quad s.t. \\
&x_{1jk} + x_{2jk} + x_{3jk} = 1 \qquad j = 1,.2 \quad k = 1,.2 \\
&x_{1j1} \le 0,\ x_{1j2} \le 0,\ x_{2j1} \le 0,\ x_{2j2} \le 1,\ x_{3j1} \le 1,\ x_{3j2} \le 0 \quad j = 1,2 \\
&x_{ijk} \in \{0,1\} \qquad i = 1,2,3;\ \ j = 1,2\ ;\ \ k = 1,2
\end{aligned}
$$

Using the *multiple choice* approach, the optimum solution of the feasibility problem is $Z_{\mathrm{FP}}(\bar{y}^0) = 2117.21$ and $\bar{x}^0_{111} = \bar{x}^0_{112} = \bar{x}^0_{121} = \bar{x}^0_{122} = 0$; $\bar{x}^0_{211} = \bar{x}^0_{221} = 0,\ \bar{x}^0_{212} = \bar{x}^0_{222} = 1$; $\bar{x}^0_{311} = \bar{x}^0_{321} = 1,\ \bar{x}^0_{312} = \bar{x}^0_{322} = 0$

b) The upper bound value is modified as $Z_{UBD} = \min\{Z_{UBD}, 2117.21\} = 2117.21$.

Step 3

In this step, the stopping criterion is checked. Since $\dfrac{Z_{UBD} - Z_{LBD}}{Z_{UBD}} = 0.2 > 0.0001$ and $\lambda_0 = 2 > 0.00001$ holds, the algorithm continues and Step 4 follows.

Step 4

Update the Lagrange multipliers using Eqs. (9) to (11) as follows:

$\alpha^1_{111} = \alpha^1_{112} = \alpha^1_{121} = 110.3,\ \alpha^1_{122} = 0$; $\alpha^1_{211} = \alpha^1_{212} = \alpha^1_{221} = \alpha^1_{222} = 0$;

$\alpha^1_{311} = \alpha^1_{312} = \alpha^1_{321} = 0,\ \alpha^1_{322} = 110.3$

Set $\lambda_1 = 2/1.01 = 1.98,\ t \leftarrow t+1$ and go back to Step 1.

The process continues for 7 iterations where the optimum solution is obtained with $Z_{UBD} = 1819.08$ and $Z_{LBD} = 1819.08$.