Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management
Kuala Lumpur, Malaysia, January 22 – 24, 2011

A Hybrid Approach for CPLP with Multicommodity Flow

S.A. MirHassani[1]
Faculty of Mathematics and Computer Science,
Amirkabir University of Technology, Tehran, Iran

F. Hooshmand Khaligh
Faculty of Mathematics and Computer Science,
Amirkabir University of Technology, Tehran, Iran

Abstract

We consider a capacitated plant location problem with multi-commodity flow on a two-level network including plants and transshipment points. The objective is to locate plants and determine commodities movement while satisfying customers' demand and minimizing costs. We present a new and effective hybrid method, composed from Lagrangean Relaxation and *UB* Creating Method to solve the problem. We introduce some criteria which help us to identify promising open plants. Based on these criteria, *UB* Creating Method is used to construct feasible solutions from the solutions at the lower bounds obtained by Lagrangean relaxation. The results show that for large size instances, the direct methods are become inefficient and failed to give good solutions in appropriate time; in contrast our method is able to give near-optimal solutions with acceptable runtimes, and optimality gaps are less than 0.70%.

Keywords
Plant location problem, Lagrangean relaxation, Weighted Dantzig-Wolf decomposition, sub-gradient method.

1. Introduction
Facility location has become a well-established research area within the field of operations research. In this paper a deterministic, static, two-level, multi-commodity, multi-source capacitated plant location problem will be considered. We study a network with two levels of facilities. In which location decision is considered in plants' level. For each customer, the demand for each commodity is known. We assume that each plant has a production capacity and also there is a capacity on each arc. The fixed cost of building every plant and the unit transshipment cost of each commodity on each arc are known. The aim is to determine where to locate plants and how to move flows that all customers' demands are satisfied and the total cost is minimized. This problem is called the capacitated plant location problem with multi-commodity flow (*CPLPMF*) which is NP-Hard. The key idea of our approach is to identify the promising open plants. *CPLPMF* was studied by Li et al. in [1]. They proposed a Lagrangean-based method that includes Lagrangean relaxation, Lagrangean heuristic and sub-gradient optimization to provide lower and upper bounds on problem objective value. Finally, they employed a Tabu Search to further improve upper bounds provided by the Lagrangean procedure. Ignacio et al. in [2] considered a two-level, single-source, single-commodity and static problem, in which location decision should be considered in both levels. They proposed a method, composed from Lagrangean relaxation, sub-gradient optimization and Tabu Search meta-heuristic approach. Keskin et al. in [3] considered a problem which has two levels of capacitated facilities (plants and Distribution centers (DCs)). Customers are to be served with different commodities from a number of plants through a number of intermediate *DCs*. It must be to locate exact *p DCs* and problem is multi- source. They presented an *MIP* model and provided meta-heuristic approaches, including Scatter Search and Tabu Search to solve it. Mohammadi Bidhandi et al. in [4] considered a three-level, multi-commodity and single-source facility location problem in which location decision is considered in any level. Each facility is capacitated and also there is an upper limit on each link. They proposed an *MIP* model and considered a modified version of Benders' decomposition to solve it. Thanh et al. in [5] proposed an *MIP* model for a dynamic, multi-level, multi-commodity production–distribution network with three levels composed of suppliers, plants and warehouses. Location decisions are considered in plants' level and warehouses' level. They solved their model with a standard solver.

[1] Corresponding Author E-mail: A_MirHassani@aut.ac.ir

The main contribution of this paper is to present a Lagrangean based heuristic method. We introduce some criteria which help us to identify promising open plants. Based on these criteria *UB* Creating Method is used to construct feasible solutions from the solutions at the lower bounds obtained by Lagrangean relaxation. Weighted Dantzig-Wolfe *(WDW)* decomposition approach and sub-gradient method are used for updating Lagrangean multipliers.
The rest of this paper is organized as follows. In section 2, we present an *MIP* model for *CPLPMF*. In section 3, we describe a heuristic method. Computational results and conclusions will be presented in section 4 and 5 respectively.

2. Problem Formulation

Suppose, = (,) is a directed graph composed of a set of nodes and a set of directed arcs. The set of nodes is classified into three categories: = ∪ ∪ , where is the set of potential plants, ∈ ; is the set of transshipment points, , ∈ ; and is the set of customers, ∈ . There are three kinds of arcs (,),(,),(,) in the network that link a plant to a transshipment point, a transshipment point to another one, and a transshipment point to the customer, respectively. We assume that there are no arcs between customers or plants, also plants and customers are connected to at least one transshipment point. denotes the set of commodities type, ∈ . h , h , h denote the cost of moving a unit of commodity ∈ through arc (,),(,),(,) ∈ respectively. Also , , denote the capacity of arc (,),(,),(,) ∈ respectively. For each plant ∈ , and denote the production capacity and the fixed cost respectively, and finally denotes the demand for the commodity ∈ at customer ∈ . All parameters are assumed to be deterministic. For each plant ∈ , is a binary variable, which is equal to one, if plant is open; otherwise is zero. Also , , are continuous variables, which denote amount of the commodity ∈ moving on arc (,),(,),(,) ∈ respectively. So the *CPLPMF* is formulated as (*P1*). In this formulation, ( ) = ∑ ∈ ∑ ∈ and denotes the total demand of customers, and φ is a fixed constant ≥1. The objective function minimizes the total cost of transporting commodities and the cost of building plants. Constraints (2) prohibit flows moving from closed plants. Constraints (3) as well as constraints (2) stipulate that the total flows moving out of a plant must not exceed its capacity, but we use both these constraints to ease the Lagrangean relaxation detailed in the next section. Constraint (4) is a redundant constraint; adding this to the formulation sharpens the Lagrangean lower bounds. Our experiments have shown that $\varphi = 3$ is an appropriate choice. Constraints (5) indicate a conservation of flow at each transshipment point. Constraints (6) require that all customer demands be met. Constraints (7)–(9) are the arc capacity requirements. Variable non-negativity is imposed by constraints (10)–(12). Constraint (13) restricts the plant to be either open or closed.

*P*1:*CPLPMF problem*

$$Z_{CPLPMF} = Min \sum_{(i,q)\in A}\sum_{c\in C} h_{iq}^c x_{iq}^c + \sum_{(q,p)\in A}\sum_{c\in C} h_{qp}^c x_{qp}^c + \sum_{(q,j)\in A}\sum_{c\in C} h_{qj}^c x_{qj}^c + \sum_{i\in I} g_i y_i \quad (1)$$

$$s.t. \sum_{\{q\in Q|(i,q)\in A\}}\sum_{c\in C} x_{iq}^c \le s_i y_i \quad \forall i \in I \quad (2)$$

$$\sum_{\{q\in Q|(i,q)\in A\}}\sum_{c\in C} x_{iq}^c \le s_i \quad \forall i \in I \quad (3)$$

$$\sum_{i\in I} s_i y_i \ge \frac{D(J)}{\varphi} \quad (4)$$

$$\sum_{\{i\in I|(i,q)\in A\}} x_{iq}^c + \sum_{\{p\in Q|(p,q)\in A\}} x_{pq}^c - \sum_{\{j\in J|(q,j)\in A\}} x_{qj}^c - \sum_{\{p\in Q|(q,p)\in A\}} x_{qp}^c = 0 \quad \forall q \in Q \forall c \in C \quad (5)$$

$$\sum_{\{q\in Q|(q,j)\in A\}} x_{qj}^c = d_{jc} \quad \forall j \in J \forall c \in C \quad (6)$$

$$\sum_{c\in C} x_{iq}^c \le l_{iq} \quad \forall (i,q) \in A \quad (7)$$

$$\sum_{c\in C} x_{qp}^c \le l_{qp} \quad \forall (q,p) \in A \quad (8)$$

$$\sum_{c\in C} x_{qj}^c \le l_{qj} \quad \forall (q,j) \in A \quad (9)$$

$$x_{iq}^c \ge 0 \quad \forall (i,q) \in A \quad (10)$$

$$x_{qp}^c \ge 0 \quad \forall (q,p) \in A \quad (11)$$

$$x_{qj}^c \ge 0 \quad \forall (q,j) \in A \quad (12)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (13)$$

*P*2:*Lagrangean relaxation problem*$(LR(\lambda))$

$$Z_{LR}(\lambda) = Min \sum_{(i,q)\in A}\sum_{c\in C} (h_{iq}^c + \lambda_i) x_{iq}^c + \sum_{(q,p)\in A}\sum_{c\in C} h_{qp}^c x_{qp}^c + \sum_{(q,j)\in A}\sum_{c\in C} h_{qj}^c x_{qj}^c - \sum_{i\in I}(s_i\lambda_i - g_i) y_i$$

s.t. constraints (3) – (13)

*P*3 :$(LR^1(\lambda))$

$$Z_{LR^1}(\lambda) = Max \sum_{i\in I}(s_i\lambda_i - g_i) y_i \quad st. \text{ Constraint s (4) and (13)}$$

*P*4: $(LR^2(\lambda))$

$$Z_{LR^2}(\lambda) = Min \sum_{(i,q)\in A}\sum_{c\in C} (h_{iq}^c + \lambda_i) x_{iq}^c + \sum_{(q,p)\in A}\sum_{c\in C} h_{qp}^c x_{qp}^c + \sum_{(q,j)\in A}\sum_{c\in C} h_{qj}^c x_{qj}^c$$

s.t.Constraints (3) and (5) – (12)

*P*5 : *Lagrangean Dual* Pr *oblem(LD)*

$$Z_{LD} = \underset{\lambda\ge 0}{Max} Z_{LR}(\lambda)$$

*P*6 : *Master problem*

$$Z_{LD} = \underset{u_0\in R, u\ge 0}{Max} u_0$$

$$s.t.\ u_0 \le \sum_{(i,q)\in A}\sum_{c\in C} (h_{iq}^c + u_i) x_{iq}^{c,t} + \sum_{(q,p)\in A}\sum_{c\in C} h_{qp}^c x_{qp}^{c,t} + \sum_{(q,j)\in A}\sum_{c\in C} h_{qj}^c x_{qj}^{c,t} - \sum_{i\in I}(s_i u_i - g_i) y_i^t \quad \forall t \in T_D$$

*P*7 : *Restricted Master Problem*(T_D^k)

$$Z_{RMP}(T_D^k) = \underset{u_0\in R, u\ge 0}{Max} u_0$$

$$s.t.\ u_0 \le \sum_{(i,q)\in A}\sum_{c\in C} (h_{iq}^c + u_i) x_{iq}^{c,t} + \sum_{(q,p)\in A}\sum_{c\in C} h_{qp}^c x_{qp}^{c,t} + \sum_{(q,j)\in A}\sum_{c\in C} h_{qj}^c x_{qj}^{c,t} - \sum_{i\in I}(s_i u_i - g_i) y_i^t \quad \forall t \in T_D^k$$

*P*9 :

$$Z_E(i_0) = Min \sum_{\{q\in Q|(i_0,q)\in A\}}\sum_{c\in C} h_{i_0q}^c x_{i_0q}^c + \sum_{(q,p)\in A}\sum_{c\in C} h_{qp}^c x_{qp}^c + \sum_{(q,j)\in A}\sum_{c\in C} h_{qj}^c x_{qj}^c + \sum_{j\in J}\sum_{c\in C} M.v_{jc}$$

$$s.t. \quad \sum_{\{q\in Q|(i_0,q)\in A\}}\sum_{c\in C} x_{i_0q}^c \le s_{i_0}$$

$$x_{i_0q}^c + \sum_{\{p\in Q|(p,q)\in A\}} x_{pq}^c - \sum_{\{j\in J|(q,j)\in A\}} x_{qj}^c - \sum_{\{p\in Q|(q,p)\in A\}} x_{qp}^c = 0 \quad \forall q \in Q \,|\, (i_0,q) \in A \ , \ \forall c \in C$$

$$\sum_{\{q\in Q|(q,j)\in A\}} x_{qj}^c + v_{jc} = d_{jc} \quad \forall j \in J \forall c \in C$$

$$\sum_{c\in C} x_{i_0q}^c \le l_{i_0q} \quad \forall (i_0,q) \in A$$

$$x_{i_0q}^c \ge 0 \quad \forall (i_0,q) \in A$$

$$v_{jc} \ge 0 \quad \forall j \in J, \forall c \in C \ \text{ and constraints (8),(9),(11),(12)}$$

3. Solution Method

We present a Lagrangean-based method to solve *CPLPMF*, and at any iteration, *UB* Creating Method is used to construct feasible solutions from the solutions at the lower bounds obtained by the relaxed problem. In section 3.1 and 3.2 we describe Lagrangean relaxation and *WDW* decomposition respectively, in section 3.3 we present preliminary steps of solution method, and main algorithm will be described in section 3.4.

3.1. Lagrangean Relaxation of *CPLPMF*

Lagrangean relaxation is a well-known technique for solving mathematical programs and provides good bounds on the optimal value of objective function. For a good survey of this technique, we refer readers to [6]. We dualize constraints (2) using Lagrangean multipliers ≥ 0, $\in$ and reach *(P2)*. This problem can be decomposed into two sub-problems: (3) and (4). (3) is a 0–1 knapsack problem and (*P4*) is similar to a transportation problem. Both of them are solvable in a reasonable time. For each vector λ we solve (3) and (4) independently, then we get to objective value of (2) as ( ) = ( ) + ( ). In order to update Lagrangean multipliers, we primarily apply the sub-gradient method to accelerate Lagrangean procedure, but in the next iterations we use *WDW* decomposition approach. *WDW* decomposition was introduced by Wentges in [7], also Klose in [8] applied it to solve a location problem. We use it for two reasons: good stopping criteria and less-oscillating lower bound.

3.2 Applying the *WDW* Decomposition to *CPLPMF*

Suppose = {(,)| (,) (2)} and suppose {(,): $\in$ } is a finite set of extreme points of the convex hull of *F*. The best lower bound for *CPLPMF* that can be obtained via Lagrangean relaxation is the optimal value of the Lagrangean Dual problem (*LD*) (see *(P5)*). According to (5), we have:

$$Z_{LD} = \underset{u\geq 0}{Max}\left\{ \underset{(x,y)\in F}{Min}\left\{ \sum_{(i,q)\in A}\sum_{c\in C}(h^c_{iq}+u_i)x^c_{iq} + \sum_{(q,p)\in A}\sum_{c\in C}h^c_{qp}x^c_{qp} + \sum_{(q,j)\in A}\sum_{c\in C}h^c_{qj}x^c_{qj} - \sum_{i\in I}(s_iu_i - g_i)y_i \right\}\right\} = \underset{u\geq 0}{Max}\left\{ \underset{t\in T_D}{Min}\left\{ \sum_{(i,q)\in A}\sum_{c\in C}(h^c_{iq}+u_i)x^{c,t}_{iq} + \sum_{(q,p)\in A}\sum_{c\in C}h^c_{qp}x^{c,t}_{qp} + \sum_{(q,j)\in A}\sum_{c\in C}h^c_{qj}x^{c,t}_{qj} - \sum_{i\in I}(s_iu_i - g_i)y^t_i \right\}\right\}$$

So we get to (*P6*) which is called Master Problem and its constraints are called dual cuts. In each iteration *k*, we assume that $\subset$ denotes a set of indexes of generated extreme points (,) found so far. So *(P7)* is called Restricted Master Problem corresponding to . The *WDW* decomposition procedure consists of successively and repeatedly solving Lagrangean sub-problems, *(P3)* and *(P4)*, which generates new extreme points of the convex hull of *F* and new dual cuts for *(P7)*. Let denote the optimal solution of the Restricted Master Problem corresponding to , and denoting the multipliers corresponding to the best lower bound found so far. Wentges in [7] proposed using = — + ——— , with the weight = ¯, ——— . Where ¯ is a positive constant and *Num* denotes the number of improvements in the lower bound. We start with = {0} where (,) is a feasible solution of *CPLPMF*, to guarantee that the Restricted Master Problem is bounded. The procedure stops after a finite number of iterations when all dual cuts needed for solution of the master problem (*P6*) are generated.

3.3. Preliminary Steps in Heuristic Method

Before implementing the original algorithm, we should follow some preliminary steps. In the other words we must calculate some parameters such as , , , , , , , for each $\in$ and the value of , (,) and . In this section we introduce these parameters, calculate them, and describe their justification.

•**Step 1:** we add constraints (14) and (15) to *(P1)*, and denote this new problem with *(P8)*.

$$\sum_{c\in C} x^c_{iq} \leq l_{iq}\, y_i \quad \forall (i,q)\in A \qquad (14) \qquad , \qquad \sum_{(i,q)\in A} l_{iq}\, y_i \geq D(J) \qquad (15)$$

We solve the linear relaxation of *(P8)* and let , $\in$ denote the corresponding solution to variable and ° denote the corresponding optimal objective function value. Constraints (14) and (15) strengthen linear relaxation bound and cause the value of e near the optimal solution to *CPLPMF*. We observe that more often a plant with larger , has a greater chance of being open in an optimal solution to *CPLPMF*. Thus, if < we prefer plant to be open. So, we organize plants by defining an integer value vector such that < if < and = if = , ≠ . The values start at 1 and increase consecutively.

•**Step 2:** We observe that = ⌊∑ $\in$ ⌋ is a good lower bound on the number of open plants in the optimal solution of *CPLPMF,* where ⌊ ⌋ returns the largest integer value ≤ x. More often the number of open plants in the optimal solution of *CPLPMF* is equal to *OP* or *OP+1*, which shows very good estimation.

•**Step 3:** We now compare plants from another point of view. Suppose we want the total demand to be satisfied by only plant and we pay penalties for unsatisfied demand. Therefore, we solve problem *(P9)*. In this problem, is a slack variable denoting unsatisfied demand of customer $\in$ from commodity $\in$ and penalized by a large positive value, say *M*. Let = ( ) − ∑ $\in$ ∑ $\in$. − ; ∀ $\in$. In fact, might be seen as the amount of saving due to opening plant . We observe that a plant with larger more often has a greater chance of

being open in an optimal solution of *CPLPMF*. Thus, if $<$, we prefer plant to be open. So, we organize plants by defining an integer value vector such that $<$ if $<$ and $=$ if $=$, $\neq$. The values start at 1 and increase consecutively.

Also let $=$ $( ) - \sum_{\in} \sum_{\in}$. $+$ $( ) \sum_{(,)\in} \sum_{\in}$. In fact, might be seen as unit cost of satisfied demand by plant . We observe that a plant with larger more often has a greater chance of being closed in an optimal solution of *CPLPMF*. Thus, if $<$,we prefer plant to be closed.

•**Step 4:** In this step, we introduce some criteria which help us to identify promising open plants. For each plant $\in$, we have three different criteria, and denote them by ; $= 1,2,3$. The definitions of them are as follows: $=$, $=$, $=(+)/2$. Now we calculate feasible solutions based on these criteria. Suppose we want to generate a feasible solution based on criterion , we set y=0 and implement *procedure(1).*

$$procedure\,(1): \quad While\left(\sum_{i\in I} y_i < OP \;\; or \;\; \sum_{i\in I} y_i s_i < D(J) \, or \sum_{(i,q)\in A} y_i l_{iq} < D(J)\right) \;\; select \;\; a\, closed \;\; plant \;\; i_0 \;\; with \;\; the \;\; biggest \;\; Cr_i^{n_0} \;\; and \;\; set \;\; y_{i_0} = 1.$$

In procedure (1) we open plants until the total capacity of open plants is greater than total demand, total capacity of the arcs that extract from open plants is greater than total demand, and total number of open plants is greater or equal to the *OP*. Then we solve the transportation problem corresponding to open plants. If it was feasible then we get to a feasible solution for *CPLPMF*; otherwise, we open more plants based on criterion and resolve it. The process is repeated for all criteria; therefore, three feasible solutions will be generated. We denote the value of the decision variable corresponding to the best feasible solution by (,) and corresponding objective value by *FUB*.

•**Step 5:** This step has a considerable role in decreasing computational time and helps to avoid extra computations in the original algorithm. Suppose we have a specific binary assignment for all , $\in$ and let *UB* be a known upper bound on the optimum objective value of the *CPLPMF*. We want to know whether this binary assignment leads to a feasible solution better than *UB*. To answer this question, we must substitute the binary assignments of , $\in$ in *(P1)* and solve the resulting transportation problem. But to avoid extra computation, we use the penalty method due to [9]. First we solve the linear relaxation of *CPLPMF* (call it (*P10)*) and let be the optimum objective value of *(P10)*. Suppose be the increase in resulting from imposing a specific binary assignment for each , $\in$, then the associated optimum objective value is $+$. Now the specific binary assignment resulting in may be discarded if $+ \geq$. In the above outline, one notice that can be determined exactly by solving the corresponding transportation problem, but this may be extremely costly from a computational standpoint. So we approximate the value of by and the specific binary assignment can be discarded if $+ \geq$. Thus for each $\in$, we calculate two penalty values , which are the increase in resulting from imposing $= 1$ and $= 0$, respectively. Now, suppose that we have a specific binary assignment for all , $\in$; we approximate penalty associated with the current binary assignment by $= \sum_{\in} ( ) + ((1 -))$.

3.4. Main Algorithm

Phase 1: Initialization

Let *k* be the iteration counter; *MaxIt*, the upper limit on number of iterations; *H*, the upper limit on the sub-gradient iterations; $\varepsilon>0$, the desired tolerance; > 0, the sub-gradient tolerance; > 0, sub-gradient step size, *LB* the best lower bound acquired by Lagrangean procedure*; UB,* the best upper bound; *Num*, the number of improvements in lower bound, and $\bar{}$, a positive constant. Also for every $\in$ let be Lagrangean multiplier at iteration *k* and the Lagrangean multiplier corresponding to the best lower bound found so far. Also similar to section 3.2 suppose that *t* denotes the index of extreme point (,) and $\subset$ denotes a set of indexes of the extreme points (,) generated so far. Implement preliminary steps and initialize *k:*=1, *MaxIt:*=500, *H:*=20, ε:=0.05, $:= 30$, $:= 2$, *LB:*=0, *UB:=FUB*, $:=$, *Num:*=0, $\bar{}:= 10$, $:= \{0\}$, φ*:*=3, $:= 0$, $:= 0$. Then go to phase 2.

Phase 2: Dual Cut Generation

Solve problem (2) for $=$ and let (,) be the optimal solution of it. Construct the dual cut corresponding to (,) and set $:=$ $\cup\{ \}$. If $( ) >$ then set $:=$ $( )$, $:=$, *Num:=Num+1*. Else set $:=$ $/ 2$. Go to phase 3.

Phase 3: Feasible Solution Generation

Based on the current solution of Lagrangean sub-problems (that is (,)), the *UB* Creating Method is invoked to derive a feasible solution. If this method improves *UB*, then update *UB*. Then go to Phase 4.

Phase 4: Updating Lagrangean Multipliers

WDW decomposition procedure in the primal iterations leads to weak lower bounds very far from the best lower bound that can be reached. This fact causes the Lagrangean procedure to terminate very late; therefore, in order to

update Lagrangean multipliers, at the beginning and while $\frac{\ \ ^{\circ}}{\ } \times 100 <$ and $<$, apply the sub-gradient method; otherwise apply *WDW* decomposition approach.

If $\frac{}{} \times 100 <$ or if $=$, stop the procedure. Otherwise let $:= \ + 1$ and go to phase 2.

Notification: In order to update Lagrangean multipliers with sub-gradient method, Because $^{\circ}$ is a good lower bound on the optimal objective value of *CPLPMF*, we use following formulation: $:= \ 0, \ +$,

where $= \sum_{(,) \in} \sum_{\in} \ ' -$ $\forall \in$ and $= \frac{\ ^{\circ}}{| |}$ $| | \neq 0$, 1 $| | = 0$.

Also in order to update Lagrangean multiplier with *WDW* decomposition we solve *(P7)* according to and suppose $(\ , \ )$ to be its optimal solution, then we set: $:= \frac{}{} + \frac{}{}$; where $= \ \bar{} , \frac{}{}$.

UB Creating Method

The aim is to convert the current solution of Lagrangean relaxation sub-problems to a feasible solution. In fact, solving the problem (*P3*) for $=$ has given us a specific binary assignment for all $, \in$. Based on this binary assignment, some plants are open and others are closed. First we try to reduce the number of open plants to $-$. Therefore, if the number of open plants is greater than $-$, we close redundant open plants based on . Thus, consider the set of open plants corresponding to the solution of (*P3*) and sort open plants in non-increasing order of parameter . Then select open plant with the largest and set $= 0$. Repeat this procedure until the number of open plants is $-$. In the following we use this modified binary vector y and construct three upper bounds. These upper bounds are constructed by means of $= 1,2,3$. We note that our preliminary experiments show $= \ \{2, \lfloor \ /5 \rfloor \}$ is an appropriate value to evaluate the number of plants that must be closed.

Constructing upper bound based on

Binary assignment may be infeasible for the transportation problem, so we implement procedure 1 as we described it in preliminary step 4. Then we use this modified binary assignment to construct an upper bound for *CPLPMF*. In order to construct feasible solution, we should solve transportation problem corresponding to open plants (call it (*P11*). But to avoid extra computation, as we noted in step 5, we check whether binary assignment gives a feasible solution for *CPLPMF* better than *UB*; therefore, we calculate penalty corresponding to this binary assignment, then we set isFeasible:=0 and implement procedure (2).

procedure (2):

$While\big(IsFeasible = 0 \ and \ number \ of \ open \ plants \leq OP + 3 \ and \ Z_{LPR} + \hat{\delta} < UB\big)do$

$solve\ (P11); and\ if\ (P11)\ was\ feasible\ then\ set\ IsFeasible = 1\ \ else\ select\ a\ closed\ plant\ i_0\ which\ has\ the\ biggest\ Cr_{i_0}^{n_0}\ and\ set\ y_{i_0} = 1\ and\ update\ the\ value\ of\ \hat{\delta}.$

In this procedure we solve *(P11)*; if it was feasible, we get to a feasible solution; otherwise, we open a new plant based on and set $= 1$ and calculate the value of corresponding to this modified binary assignment, if $+ \ <$ we solve *(P11)* again. We do this process until *(P11)* is feasible or the number of open plants is greater than *OP+3* or $+ \ \geq$. We observe that when the number of open plants becomes greater than *OP+3*, continuing procedure (2) cannot lead to a good feasible solution; so we stop procedure (2). After implementing this procedure, if acquired solution was better than *UB*, we update *UB*; and in the case of infeasibility, we discard the current binary assignment . We repeat this process for all criteria to improve *UB*, if possible.

4. Computational Results

The proposed algorithm was coded in AIMMS, and CPLEX solver was used to solve the problems in the heuristic method. Our tests were conducted in a Pentium IV, 2 GHz computer with 512 MB of RAM. We generated 180 test instances similar to [1], to evaluate the performance of the heuristic method in terms of running time and solution quality. We compared the upper bound acquired by the heuristic method with the optimal solution of the direct method. In the direct method we used CPLEX solver. Some of these results have been presented in Table 1. We generated 5 instances for each dataset in this table, and the representations of each column are as follows: |*I*|: number of potential plants ; |*Q*|: number of transshipment points; |*J*|: number of customers; |*C*|: number of commodities; *BVar*: number of binary variables; *CVar*: number of continuous variables; *Cons*: number of constraints; *DM-CPU-Time(sec)*: the average time required by the direct method; *HM-CPU-Time(sec)*: the average time required by the heuristic method; *Gap(Opt,UB)*: the gap between optimal solution of direct method and *UB*; *Gap(LB*,UB)*: the gap between *and *UB*. In fact, we have two different lower bounds on the optimal objective function value of *CPLPMF*: *LB* and $^{\circ}$. So we defined $^{*} = \ \{ \ ^{\circ}, \ \}$ and used the gap between *and *UB* to evaluate the heuristic algorithm as a stand-alone method. So we define:

$$(\ , \ ) = \frac{}{} \times 100 \text{ and } (\ ^{*}, \ ) = \frac{\ ^{*}}{\ ^{*}} \times 100 .$$

Two basic parameters for generating instances are plant tightness (*TI*) and density of network ( ). The definition of them is: $= \sum_{\in} / ( )$ and $= \frac{}{| | | | | |}$ respectively. For all datasets given in Table 1, ρ is fixed at 5 and TI is fixed at 2. The direct method is become inefficient when the problem was sufficiently large; thus, a time limit of 18,000 seconds was set for it. So in Table 1, an asterisk (*) indicates that the direct method failed to give an exact solution for any test instance within the time limit; therefore, we could not compute the (,) for the corresponding dataset and reported only the running time of the heuristic method and (*,). According to Table 1, (,) range between 0.00% and 0.17% on average, and maximum optimality gap over all instances is no greater than 0.36%. So, it is clear that the upper bound acquired by the heuristic method is very close to optima. Average gaps between *LB** and upper bounds range between 2.33% and 3.26%. The computation time required by the direct method, increases rapidly when more potential plants are involved but the time taken by the heuristic method does not increase significantly and is reasonable. Except instances which have been presented in Table 1, we tested more instances in two other cases too, in the first ρ is fixed at 5 and *TI* is fixed at 3 and in the second ρ is fixed at 10 and *TI* is fixed at 2. On the whole, we tested 180 instances. Of these, the direct method was able to obtain optimal solution in the time limitation for 130 instances; thus, for these 130 instances we could compare the solutions of the heuristic method with optimal solutions, we obtained less than 0.70% optimality gap with acceptable runtimes, even for large problems, and in 49.23% the heuristic method reached to optimal solution.

Table 1 : Comparsion of Heuristic Method and direct method in terms of running time and solution quality (five instances for each data set)

TI=2 and ?=5

Data set	*|I|*	*|Q|*	*|J|*	*|C|*	*BVar*	*CVar*	*Cons*	*DM-CPU-Time*	*HM-CPU-Time*	*Gap(LB*,UB)%*	*Gap(Opt,UB)%*		
								(sec)	*(sec)*	*Ave*	*Min*	*Ave*	*Max*
1	10	15	20	30	10	6750	1296	4.48	11.84	2.96	0.00	0.02	0.06
2	10	15	25	30	10	7500	1471	5.44	13.46	2.72	0.00	0.06	0.29
3	20	30	50	30	20	15000	2941	119.21	71.40	2.48	0.00	0.00	0.00
4	20	30	65	30	20	17250	3466	123.98	79.24	2.43	0.00	0.09	0.28
5	30	45	60	30	30	20250	3886	967.24	182.66	3.09	0.00	0.13	0.31
6	30	45	70	30	30	21750	4236	1398.32	192.83	2.73	0.00	0.09	0.28
7	40	60	80	30	40	27000	5181	3759.32	418.81	2.65	0.00	0.14	0.24
8	40	60	100	30	40	30000	5881	4439.29	442.86	2.33	0.04	0.14	0.28
9	50	75	100	30	50	33750	6476	7111.10	842.03	3.20	0.00	0.10	0.21
10	50	75	125	30	50	37500	7351	8381.59	945.81	3.26	0.00	0.17	0.36
11	60	90	150	30	60	45000	8821	18000*	1296.84	3.09			
12	70	105	175	30	70	52500	10291	18000*	1416.40	3.11			

5. Conclusion

In this paper we presented a hybrid method, composed from Lagrangean relaxation and *UB* Creating Method to solve *CPLPMF*. *UB* Creating Method intends to identify promising open plants and modify the solution of a relaxed problem into a good feasible solution. We used sub-gradient optimization and *WDW* decomposition approaches for updating Lagrangean multipliers. According to computational results, this method is able to give near-optimal solutions within acceptable runtimes even for large instances and optimality gaps are less than 0.70%.

It is expected that the ideas mentioned here can be applied to solve other discrete location problems with different properties; therefore, it would be interesting to consider the application of these ideas in other problems and industrial applications.

References

[1]. Li, J., Chu, F., and Prins, C., 2009, "Lower and upper bounds for a capacitated plant location problem with multi commodity flow," Computers & Operations Research, 36(11), 3019-3030.

[2]. Ignacio, A.A., Ferreira Filho, V.J., and Galvão, R.D., 2008, "Lower and upper bounds for a two-level hierarchical location problem in computer networks," Computers & Operations Research, 35(6), 1982-1998.

[3]. Keskin, B.B., and Üster, H., 2007, "Meta-Heuristic approaches with memory and evolution for a multi-product production/distribution system design problem," European Journal Of Operational Research, 182(2), 663-682.

[4]. Mohammadi Bidhandi, H., Mohd.yusuff, R., Megat Ahamd, M.M., and Abu Bakar, M.R., 2009, "Development of a new approach for deterministic supply chain network design," European Journal of Operational Research, 198(1), 121-128.

[5]. Thanh, P.N., Bostel, N., and Péton, O., 2008, "A dynamic model for facility location in the design of complex supply chains," International Journal of Production Economics, 113(2), 678-693.

[6]. Guignard, M., 2003, "Lagrangean Relaxation," Top, 11(2), 151-228.

[7]. Wentges, P., 1997, "Weighted Dantzig-Wolfe decomposition for linear mixed-integer programming," International Transactions in Operational Research, 4(2), 151-162.

[8]. Klose, A., 2000, "A Lagrangean relax-and-cut approach for the two-stage capacitated facility location problem," European Journal of Operational Research, 126(2), 408-421.

[9]. Driebeek, N.J., 1966, "An algorithm for the solution of mixed integer programming problems," Management Science, 12(7), 576-587.