

Dynamic Mean Value Cross Decomposition Algorithm for Capacitated Facility Location Problems

Chulyeon KIM¹, Gyunghyun CHOI^{2*}, Sung-Seok KO³

¹*Department of Industrial Engineering, Hanyang University
17 Haengdang-dong, Seongdong-gu, Seoul 133-791, Korea*

²*Graduate School of Technology and Innovation Management, Hanyang University
17 Haengdang-dong, Seongdong-gu, Seoul 133-791, Korea*

³*Department of Industrial Engineering, Konkuk University
1 Hwayang-dong, Gwangjin-gu, Seoul 143-701, Korea
e-mail: art4life94@naver.com, ghchoi@hanyang.ac.kr, ssko@konkuk.ac.kr*

Received: October 2011; accepted: December 2012

Abstract. In this article, we propose a practical algorithm for capacitated facility location problems (CFLP). There are some approaches which can obtain primal solutions while simultaneously exploiting the primal structure and the dual structure. One of these approaches is the mean value cross decomposition (MVCD) method that ensures convergence without solving master problems. However, MVCD has been previously applied only to uncapacitated facility location problems (UFLP), due to the fact that the performance is highly dependent on the structure of the problem. The proposed algorithm, named the dynamic mean value cross decomposition algorithm (DMVCD), is effectively integrated with MVCD and cutting plane methods in order to tighten the bounds by reducing the duality gap. Computational results of various instances are also reported to verify the effectiveness and efficiency of DMVCD.

Key words: capacitated facility location problems, cross decomposition, mean value cross decomposition, primal recovery strategies, Lagrangian relaxation

1. Introduction

The facility location problem (FLP) is a problem to choose facility locations, such as industrial plants or warehouses, in order to minimize the cost to meet the product demands. In general, there are fixed costs for locating the facilities and delivery costs for distributing the products between facilities and customers.

Problem P_{CFL} , the mathematical formulation of the CFLP (Capacitated Facility Location Problem), can thus be formulated as a mixed integer program. To do this, we define a set of sites $I = \{1, 2, \dots, m\}$ where facilities can be located, and a set of customers $J = \{1, 2, \dots, n\}$ with a given demand for a single product. Let f_i be the given fixed cost

* Corresponding author. Tel.: +82-2-2220-0471.

of opening facility i and c_{ij} be the delivery cost from facility i to customer j . Also, let d_j be the demand of customer j and s_i be the maximum capacity of facility i . For decision variables, let the continuous variable x_{ij} be the fraction of customer j 's demand d_j delivered from facility i and the binary variable $y_i = 1$ if facility i is open and $y_i = 0$ otherwise. Then P_{CFL} is formalized as follows:

(P_{CFL})

$$v^* = \min \sum_{i=1}^m f_i y_i + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}, \quad (1)$$

subject to

$$\sum_{i=1}^m x_{ij} = 1, \quad \forall j, \quad (2)$$

$$x_{ij} \leq y_i, \quad \forall i, j, \quad (3)$$

$$\sum_{j=1}^n d_j x_{ij} \leq s_i y_i, \quad \forall i, \quad (4)$$

$$x_{ij} \geq 0, \quad \forall i, j, \quad (5)$$

$$y_i \in \{0, 1\}, \quad \forall i. \quad (6)$$

P_{CFL} is a problem of finding a subset of facilities to open and of deciding the ratio x_{ij} to be delivered in order to minimize the total costs, as a objective function (1), while satisfying all the constraints (2)–(6). Constraints (2) indicate all demands must be met, and constraints (3) require customers to be supplied only from open facilities. Finally, constraints (4) are the capacity constraints for which the total supply at a facility is limited by the maximum capacity of the facility. Note that the UFLP (Uncapacitated Facility Location Problem) does not include the constraints (4). However, it is well known that CFLP better reflects more realistic problems than UFLP, and both of them are NP-hard problems. A new extension of CFLP, called the ordered CFLP, is also introduced to deal with more realistic situations (Kalcsics *et al.*, 2010).

Many branch-and-bound algorithms have been developed in which several relaxations and decomposition schemes are used to find desired lower bounds. These kinds of branch-and-bound algorithms focus on improving bounds obtained by relaxations or decompositions and recovering feasible solutions from these bounds. In this context, the cutting plane methods grounded on the polyhedral theory are primarily used to improve the bounds in conjunction with valid inequalities. The polyhedral structure of CFLP and valid inequalities of network flow problems with capacitated fixed charges were studied in some researches (Padberg *et al.*, 1985; Wolsey, 1989; Leung and Magnanti, 1989; Aardal *et al.*, 1995). Based on these studies, various versions of the branch-and-cut algorithms were developed (Aardal, 1998a; Klose, 2000; Sridhar and Park, 2000; Ortega and Wolsey, 2003). In addition, Avella and Boccia (2009) developed a branch-and-cut-and-

price algorithm based on the mixed dicut inequalities, and Yang *et al.* (2012) presented a cut-and-solve based algorithm for the single-source capacitated facility location problem.

Another key method to guarantee promising error bounds is the Lagrangian heuristic algorithm on Lagrangian relaxation, mainly using subgradient optimization techniques to solve the Lagrangian dual problems. This method also requires relaxation techniques as well as some recovery schemes in order to produce feasible solutions. Klincewicz and Luss (1986) suggested a Lagrangian heuristic algorithm to relax the capacity constraints, while Lagrangian relaxation algorithms relaxing the demand constraints to create a number of knapsack problems were employed (Sridharan, 1993; Hindi and Pienkosz, 1999; Holmberg *et al.*, 1999; Barahona and Chudak, 2005). Beasley (1993) proposed a heuristic algorithm by relaxing both capacity constraints and demand constraints, and Agar and Salhi (1998) improved Beasley algorithm. For feasible solution recovery methods, a restricted neighborhood search and a repeated-matching algorithm were studied (Hindi and Pienkosz, 1999; Holmberg *et al.*, 1999) and a subgradient-based heuristic algorithm in conjunction with the volume algorithm and randomized rounding was developed (Barahona and Chudak, 2005). Also, Chen and Ting (2008) developed a hybrid algorithm which combined the Lagrangian heuristic and a multiple ant colony system in order to solve the single source capacitated facility location problem, and an add-drop-interchange heuristic presented by Sinha (2009) was used to improve a lower bound based on the LP-relaxation. Furthermore, Du *et al.* (2010) proposed an approximation algorithm for the k -level CFLP that unified and extended several existing facility location problems and Sun (2012) developed an effective tabu search heuristic procedure for CFLP, as compared to the Lagrangian heuristic method with improved subgradient scheme.

In addition, there are different approaches simultaneously considering the primal and the dual structures of the FLP problems. On one hand, the cross decomposition algorithm (Van Roy, 1983) is integrated with Benders decomposition and Lagrangian relaxation into a single framework. On the other hand, there is the mean value cross decomposition (MVCD) algorithm as a variant of cross decomposition (Holmberg, 1992). While most of the Lagrangian relaxation-based algorithms only give us lower bounds for the primal problems, these approaches provide good upper and lower bounds at the same time and generate various information to find feasible solutions. Cross decomposition has been successfully applied to CFLP (Van Roy, 1986), but it should incorporate a complicated procedure to solve master problems depending on the outcomes of convergence tests. At worst, hence, as the convergence test fails at each iteration, cross decomposition cannot be better than Benders or Dantzig–Wolfe decomposition. Indeed, the cross decomposition is designed to work well with few master problems, but it has been reported that the master problems have to be addressed frequently, which consequently makes the algorithm inefficient (Cornuéjols *et al.*, 1991; Wentges, 1996; Holmberg, 2001). Without the master problems, MVCD was proposed as a method which can efficiently obtain optimal solutions for linear programming problems by using only the subproblem structures (Holmberg, 1992; Holmberg, 1994). It was also shown that MVCD provided the same bounds as Lagrangian relaxation, which are better than those of LP-relaxation (Holmberg, 1997). Then, MVCD was extended for MIP (Mixed Integer Programming) problems and applied to UFLP

(Holmberg, 1997, 2001). In addition, it has been shown to be useful for convex nonlinear problems (Holmberg and Kiwiel, 2006). In this regard, MVCD has been considered as an efficient approach due to its simplified computational procedure than cross decomposition, guaranteeing relatively good lower bounds. Of course, cross decomposition is able to present better solutions than MVCD, but it is in general harder to quickly obtain good solutions within finite iterations, especially for large scale problems.

In sum, based on the decomposition techniques, either the branch-and-bound, branch-and-cut, or heuristic methods have been widely used for CFLP. Though MVCD with the branch-and-bound methods has been examined in the literature, MVCD in conjunction with the cutting plane methods has been less studied.

Hence, we propose a specially designed algorithm that combines MVCD and the cutting plane methods in order to practically solve CFLP. This algorithm, named DMVCD (dynamic mean value cross decomposition), is a general application of MVCD in which cutting plane methods play an important role for convergence without solving master problems while reducing the duality gap.

The rest of this paper is organized as follows. In Section 2, we describe MVCD which is the basis of our algorithm. Our proposed algorithm is detailed in Section 3. Computational results are presented in Section 4. In the last section, we summarize our results and show a way forward based on the present study.

2. Mean Value Cross Decomposition Method

Algorithm MVCD is very similar to cross decomposition, in that both algorithms simultaneously exploit the primal and dual structures of problems (Holmberg, 1992, 1994). The main difference between MVCD and cross decomposition is that MVCD uses the mean value of previous dual subproblem solutions as the input value of the primal subproblem and vice versa, whereas the cross decomposition employs the last dual subproblem solutions directly to solve the primal subproblem and vice versa. Though cross decomposition has been widely applied to combinatorial optimizations, MVCD has a rather different advantage such as guaranteed convergence using only a simple procedure without addressing master problems, as a consequence, it is also very efficient.

MVCD is built on a suitable combination of the primal master problem and the dual master problem, both of which have the same optimal value. However, it is difficult to use MVCD in directly solving MIP because of the duality gap. In order to apply MVCD for MIP, we first need to construct primal problems and dual problems to have tight bounds. MVCD with this combination of problems produces the same bound as that of Lagrangian relaxation (Holmberg, 1997). If the dual subproblem has an integrality property, the same lower bound as that of Lagrangian relaxation can be obtained even if the dual subproblem is LP-relaxed. Based on this property, the MVCD-based branch-and-bound algorithm for UFLP was developed (Holmberg, 2005).

Problem P_{UFL} is defined as a problem with no capacity constraints (4) in P_{CFL} . Let MP be the primal master problem, MD be the dual master problem and v^* , v_P and v_D be the optimal values of P_{UFL} , MP and MD , respectively.

Let us define the bounded sets X, Y_{LP}, Y as $X = \{x_{ij}: \sum_{i=1}^m x_{ij} = 1, x_{ij} \geq 0, \forall i, j\}$, $Y_{LP} = \{y_i: 0 \leq y_i \leq 1, \forall i\}$ and $Y = \{y_i: y_i \in \{0, 1\}, \forall i\}$, respectively. We assume that for each $y_i \in Y_{LP}$ there exists some $x_{ij} \in X$ such that $x_{ij} \leq y_i$ for all i, j . Dual subproblems, denoted by USD_μ , are constructed via Lagrangian relaxation with respect to coupling constraints (3) with the dual variable $\mu \geq 0$. The surrogate constraint (7) is required in order to exist some $x_{ij} \in X$ satisfying $x_{ij} \leq y_i$ for $y_i \in Y_{LP} \forall i, j$. For fixed $y_i \in Y_{LP} \forall i$, let us define primal subproblems as USP_y with $\psi(y)$ of the optimal values.

(USD_μ)

$$\varphi(\mu) = \min \left(\sum_{i=1}^m f_i - \mu_i s_i \right) y_i + \left(\sum_{i=1}^m \sum_{j=1}^n c_{ij} + \mu_i d_j \right) x_{ij},$$

subject to

$$\begin{aligned} x &\in X, \\ y &\in Y, \\ \sum_i^m y_i &\geq 1. \end{aligned} \tag{7}$$

Then, we obviously have

$$\varphi(\mu) \leq v_D \leq v^* = v_P \leq \psi(y), \quad \forall \mu \geq 0, \forall y \in Y.$$

Now, let us define $\tilde{Y}_{LP} = \{y_i: 0 \leq y_i \leq 1, \sum_i^m y_i \geq 1, \forall i\}$, $\tilde{Y} = \{y_i \in \{0, 1\}: \sum_i^m y_i \geq 1, \forall i\}$ and \tilde{Y}_C as the convex hull of \tilde{Y} . Since constraint (7) defines the facet of Y_C , the convex hull of Y , we have $\tilde{Y}_C = Y_C = \{y_i: 0 \leq y_i \leq 1, \forall i\}$. Therefore, we can construct another convexified primal master problem (MPC) as follows:

$$(MPC) \quad v_C = \min_{y \in Y_C} \psi(y).$$

Note that $v_C = v_D$. In this case, MVCD performs on a combination of MPC and MD, which can guarantee the same bounds as $v_D \leq v^*$ of Lagrangian relaxation.

Let us denote MPL and MDL as LP-relaxation problems of MP and MD, respectively, and define $v_{PL} = \min_{y \in \tilde{Y}_{LP}} \psi(y)$ (MPL) and $v_{DL} = \max_{u \geq 0} \varphi_{LP}(u)$ (MDL). Here, $v_{PL} = v_{DL}$ holds because MDL is the Lagrangian dual of MPL. Then, we have another MVCD application that runs on a combination of MPL and MDL having $v_{PL} = v_{DL}$. In addition, there exists an integrality property in USD_μ by Definition 1.

DEFINITION 1. (See Geoffrion, 1974.) The dual subproblem is said to have the integrality property if $\varphi(u) = \varphi_{LP}(u), \forall u \geq 0$, i.e., if replacing Y with Y_{LP} in the dual subproblem has no effect on the objective function value.

As Problem USD_μ has the integrality property and if it has been LP-relaxed, the same bounds as those of MPC and MD are obtained; that is, $v_{DL} = v_D = v_C$ (Geoffrion, 1974).

However, it is not easy for MVCD to find the exact integer solutions because the duality gap between $v_{DL}(v_D)$ and v^* still exists, and MVCD has a feature of infinite asymptotic convergence to the optimal solution. Therefore, in order to overcome these difficulties and improve the efficiency of asymptotic convergence, the branch-and-bound and primal recovery heuristic techniques were used. In this light, several combinations with other techniques such as cross decomposition, primal-dual subgradient optimizations, and the variations of MVCD were studied in the MVCD framework (Holmberg, 2001, 2005).

In particular, MVCD is well applied to the special case of UFLP, the Lagrangian dual of which is separated into m parts with the only binary variable y and has the integrality property. In contrast, in the case of CFLP, no Lagrangian dual meets the integrality property. The Lagrangian dual divided respectively into x and y variables yields the lower dual performance with the large duality gap. Also, as is the case of UFLP, the closer one is to the optimal solutions, the slower is the convergence due to infinite asymptotic convergence of MVCD. For these reasons, there is no empirical study of MVCD with application to CFLP.

3. Dynamic Mean Value Cross Decomposition for CFLP

In this section, we describe the dynamic mean value cross decomposition (DMVCD) method which is designed for practically applying the MVCD method to CFLP. As stated in the previous sections, the applications of MVCD are limited depending on the structures of the problems, in spite of its various advantages. In the case of CFLP, Lagrangian relaxation generally leads to a large duality gap. Consequently, the MVCD itself is not a useful method for this problem, and hence, the DMVCD algorithm proposed in this study employs the cutting plane methods to obtain tighter bounds. The main concept of DMVCD is to approximate the convex hull of P_{CFL} whenever no improvement takes place by considering some valid inequalities in the MVCD framework. This scheme reduces the duality gap and tightens the bounds of MVCD and is able to accelerate the asymptotic convergence. Most of the existing studies relevant to the cutting plane methods in conjunction with decomposition methods have only focused on strengthening the master problems by enhancing the cuts consisting of the master problems or adding strong valid inequalities to the master problems (Magnanti and Wong, 1981; Klose, 2000; Van Roy, 1986; Sridhar and Park, 2000). In contrast, we try to add some valid inequalities to the subproblems, which allows DMVCD to generate tighter bounds while preserving the efficient procedures of MVCD.

Our approach owes much to the MVCD-based branch-and-bound method (Holmberg, 2005). This study pays attention to issues such as heuristics for obtaining feasible integer solutions, tree search strategies and reoptimization after branching or adding valid inequalities and it has thus been successfully applied to UFLP. However, in the case of CFLP, some guidelines were established but no empirical study was presented (Holmberg, 2005). Our study focuses on developing a practical procedure for the addition of valid inequalities that ensures consecutive improvements of the bounds, and evaluating the convergence and effectiveness of the procedures through an empirical study for CFLP.

3.1. A Lagrangian Relaxation for DMVCD

Like MVCD, we first need to obtain the dual subproblem to be addressed by DMVCD. To do so, there are several options to relax the demand constraints (2), the variable upper bounds (3) and the capacity constraints (4) in P_{CFL} , mentioned in Introduction. It is known that it is more effective to relax the demand constraints (2) for Lagrangian heuristics as shown in the studies (Sridharan, 1993; Hindi and Pienkosz, 1999; Holmberg *et al.*, 1999, 2005; Barahona and Chudak, 2005), whereas cross decomposition forms the dual subproblems by relaxing the capacity constraints (4). However, the Lagrangian subproblems obtained by dualizing (2) or (4) are not able to be separated into x and y variable parts and have no integrality property. In other words, none of them is able to obtain a suitable primal master problem with the same objective value as that of the dual master problem.

In our approach, instead of using these Lagrangian relaxation schemes, we form the dual subproblem, denoted $CSD_{(\mu,v)}$, through a relaxation with regard to both constraints (3) and (4). Let μ and v be the dual variables corresponding to constraints (3) and (4), respectively. In addition, constraint (8) is required for the feasibility of the primal subproblems, denoted CSP_y , which is defined by fixing the primal variables $y_i \in Y_{LP} = \{y_i: 0 \leq y_i \leq 1, \sum_{i=1}^m s_i y_i \geq \sum_{j=1}^n d_j \forall i\}$ in P_{CFL} . In addition, the optimal value of CSP_y is defined as $\psi(y)$.

($CSD_{(\mu,v)}$)

$$\varphi(\mu, v) = \min \sum_{i=1}^m (f_i - \mu_i s_i - \sum_{j=1}^n v_{ij}) y_i + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} + \mu_i d_j + v_{ij}) x_{ij},$$

subject to

$$\begin{aligned} \sum_{i=1}^m x_{ij} &= 1, \quad \forall j, \\ \sum_{i=1}^m s_i y_i &\geq \sum_{j=1}^n d_j, \\ x_{ij} &\geq 0, \quad \forall i, j, \\ y_i &\in \{0, 1\}, \quad \forall i. \end{aligned} \tag{8}$$

Even though separability is obtained, constraint (8) is a knapsack constraint which is not able to define the facets of the convex hull of $\{y_i: 0 \leq y_i \leq 1\}$, so that $CSD_{(\mu,v)}$ no longer has the integrality property. Therefore, the LP-relaxation of $CSD_{(\mu,v)}$ cannot provide the same bounds as those of the Lagrangian relaxation. As a result, we adopt another MPC and MD in accordance with the convergence conditions of MVCD, both of which have CSP_y and $CSD_{(\mu,v)}$ as their subproblems. Here, MPC and MD are defined as follows:

(MPC)

$$v_C = \min_{y \in Y_C} \psi(y),$$

where Y_C is the convex hull of Y :

$$Y = \left\{ y_i \in \{0, 1\}: \sum_{i=1}^m s_i y_i \geq \sum_{j=1}^n d_j, \forall i \right\}, \tag{9}$$

(MD)

$$v_D = \max_{\mu \geq 0, v \geq 0} \varphi(\mu, v).$$

3.2. Cutting Plane Methods and Valid Inequalities in DMVCD

For convenience, the optimal objective values of P_{CFL} , MP, MD, and MPC defined in the previous Section 3.1 are repeated as v^* , v_P , v_D , and v_C , respectively. Again, let us duplicate MPL (LP-relaxations of MP) and MDL (LP-relaxations of MD), and their optimal values are $v_{PL} = \min_{y \in Y_{LP}} \psi(y)$ and $v_{DL} = \max_{u \geq 0} \varphi_{LP}(u)$, respectively. Then, the relation between these values holds true (see Fig. 1 below):

$$v_{DL} = v_{PL} \leq v_C = v_D \leq v^* = v_P.$$

Now, in order to examine another relation, we denote Z as the feasible set defined by constraints (2)–(6) in P_{CFL} and Z_{LR} as the feasible set of the LP-relaxation of P_{CFL} . Also, let c_k for $k \in K$ be a valid inequality violated by a point in Z_{LR} , where K is the index set of all valid inequalities for $\text{conv}(Z)$. Given $V_{\tilde{K}}$ as the feasible space obtained by intersecting c_k for all $k \in \tilde{K} \subseteq K$, we can define $Z_S = \{(x, y) \in V_{\tilde{K}} : y \text{ integer}\}$. Then, we can state another master problem as follows:

$$v_S = \min_{y \in \text{conv}(Z_S)} \psi(y).$$

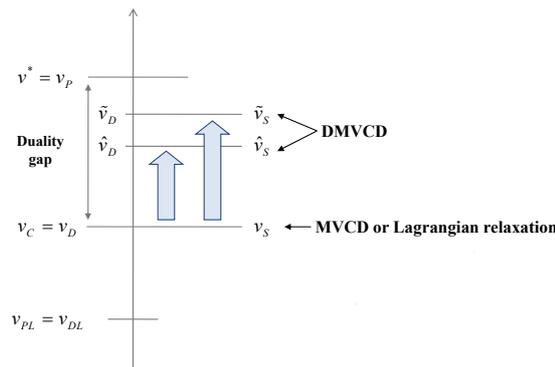


Fig. 1. The relations between various values.

Here, $\text{conv}(Z) \subseteq \text{conv}(Z_S) \subseteq Z_{LR}$ is valid, so that we have the following obvious result (see Fig. 1):

$$v_{PL} \leq v_S \leq v^*.$$

If we could find all valid inequalities in order that \tilde{K} is the same as K , we would have $v_S = v^*$. However, unfortunately, for NP-hard problems such as CFLP, it is almost impossible to obtain \tilde{K} that can completely describe $\text{conv}(Z)$.

Our purpose is to obtain Z_S such that $v_C \leq v_S \leq v^* = v_P$ for better bounds than those of the Lagrangian relaxation or MVCD (see Fig. 1). However, note that the knapsack cover inequalities, which are well known as the single most effective class (Padberg *et al.*, 1985; Aardal, 1998a), are already implicitly included in both MPC and MD by defining Y_C with constraint (8). As a consequence, we can say that Z_S which is equivalent to Y_C is already held, namely $v_S = v_C$.

Now we will discuss how v_S can be improved to $\tilde{v}_S > v_S$.

Proposition 1. Assume that there exists a set G of points $(x, y) \in \{Y_C - \text{conv}(Z)\}$. Let c_k for $k \in \tilde{K} \subseteq K$ be a valid inequality violated by a point $(\hat{x}, \hat{y}) \in G$, and let $\tilde{Z}_S = Z_S \cap Y_C$. Then, $v_C < \tilde{v}_S \leq v^*$, where $\tilde{v}_S = \min_{y \in \text{conv}(\tilde{Z}_S)} \psi(y)$.

Proof. Since $\text{conv}(\tilde{Z}_S)$ is the convex hull of $\{Z_S \cap Y_C\}$ and $\text{conv}(Z) \subset \text{conv}(Z_S)$, obviously $\text{conv}(Z) \subseteq \text{conv}(\tilde{Z}_S) \subset Y_C$. Thus, $v_C < \tilde{v}_S \leq v^*$. □

Both a point in Y_C and a valid inequality c_k for $k \in K$ violated by that point are required for ensuring $\tilde{v}_S > v_S = v_C$ according to Proposition 1. Note that a convex combination of y -solutions of $\text{CSD}_{(\mu, v)}$, denoted by \bar{y} , always lies in Y_C and x can be obtained by solving $\text{CSP}_{\bar{y}}$. However, considering that solutions yielded always stay within Y_C and converge to a particular solution in Y_C , it is not required all valid inequalities for Z cutting off points in Y_C . Therefore, we will take the valid inequalities into consideration only in situations of no improvement. Consequently, in terms of the search space for identifying the valid inequalities, DMVCD becomes more efficient than the cutting plane methods based on LP-relaxation. To some extent, there are some valid inequalities to be considered in the previous studies (Padberg *et al.*, 1985; Aardal *et al.*, 1995; Aardal, 1998b), e.g., flow cover inequalities, effective capacity inequalities, submodular inequalities, and so on. Among these valid inequalities, we use a common inequalities set, the flow cover inequalities.

Now, we need to construct the dual subproblems, $\text{CSD}_{(\mu, v)}$, such that the dual optimal value is equal to \tilde{v}_S . Assume that a valid inequality (10) is violated by (x, \bar{y}) .

$$\sum_{ij} \alpha_{ij}^k x_{ij}^k + \sum_i \beta_i^k y_i^k \leq \gamma^k, \quad k \in \tilde{K} \cap K. \tag{10}$$

Then, we have two choices, namely either to keep a valid inequality (10) or to relax it.

Proposition 2. Let $\tilde{\varphi}(\mu, v)$ be the dual subproblem via a Lagrangian relaxation with respect to a valid inequality (10) and $\tilde{v}_D = \max_{\mu \geq 0, v \geq 0} \tilde{\varphi}(\mu, v)$. Then, $\tilde{v}_S = \tilde{v}_D$.

Proof. See Geoffrion (1974). \square

Through Propositions 1 and 2, we can choose a type of relaxation that ensures $\tilde{v}_S = \tilde{v}_D$, keeping the valid inequality. However, in order to implement DMVCD practically, a valid inequality with only y -variables, i.e., $\alpha_{ij} = 0, \forall i, j$ in (10), is available for this type of relaxation. Because the primal subproblem, $\text{CSP}_{\bar{y}}$, is defined by fixing only $y = \bar{y}$, except for x -variables in the DMVCD framework, keeping the valid inequality (10) may result in $\tilde{v}_S < \tilde{v}_D$. In other words, when only the y -variables exist in a valid inequality generated, we could keep it for convergent DMVCD. If both x and y variables appear in the valid inequality, we can obtain \hat{v}_D by relaxing it. Then, we can also obtain \hat{v}_S such that $v_C \leq \hat{v}_S = \hat{v}_D \leq \tilde{v}_S$, where \hat{v}_S is defined as follows:

$$\hat{v}_S = \min_{y \in \text{conv}(Y)} \hat{\psi}(y),$$

where $\hat{\psi}(y)$ is the optimal value of Problem CSP_y with the valid inequality as a constraint.

3.3. The Dynamic Mean Value Cross Decomposition Algorithm

Our DMVCD algorithm first attempts to reach a specific value, $v_C = v_D$, by repeating the following two steps:

- fix y at its mean value \bar{y} and solve the restricted primal subproblem $\text{CSP}_{\bar{y}}$ to get new values for μ, v , and
- fix μ, v at its mean value $\bar{\mu}, \bar{v}$ and solve the Lagrangian dual subproblem $\text{CSD}_{(\bar{\mu}, \bar{v})}$ to get new values for y .

In parallel with these basic steps, the additions of the flow cover inequalities with both x and y variables enhance the current bounds v_C to the bounds \hat{v}_S such that $v_C \leq \hat{v}_S = \hat{v}_D \leq \tilde{v}_S$ by dualizing these inequalities as described in Section 3.2. Furthermore, we carry out this procedure repetitively to make DMVCD convergent to a tight bound $v_C \ll \hat{v}_S \leq v^*$. In addition, we will exploit valid inequalities whenever no improvement takes place by taking advantage of a feature that \bar{y} always lies in the Y_C .

According to Holmberg (Holmberg, 2005), the effect of bad starting solutions remains for many iterations and might make MVCD inefficient. For this reason, some combinations of MVCD, subgradient optimization and ordinary cross decomposition were suggested in order to produce good starting solutions (Holmberg, 2005). Also, the earlier solutions accumulated as mean values have knock-on effects on converging to specific values. In other words, though they help to guarantee the convergence of our scheme, they might make our scheme more asymptotic. Thus, it is necessary to lessen the effect of the earlier solutions. To do this, DMVCD intends to reduce the solutions space approximately onto $\text{conv}(Z)$ by adding valid inequalities. Therefore, the earlier solutions that cannot validate the added inequalities barely affect the convergence to specific values. So, we attempt to accelerate the asymptotic convergence by restarting the solutions accumulating in the procedure of DMVCD whenever valid inequalities are added. In this sense, we do not

need to obtain the good starting solutions with additional techniques and get the starting solutions as the input values of the dual subproblem by simply using the dual-subgradient optimization for $\varphi_{LP}(\mu, v)$, the LP-relaxation of $\text{CSD}_{(\mu, v)}$.

Our proposed algorithm is summarized as follows:

Algorithm: Dynamic Mean Value Cross Decomposition

Initialization Get a start solution (μ^0, v^0) by using the subgradient optimization based on $\varphi_{LP}(\mu, v)$.

Set $t = 1, k = 1, v_U = \infty, v_L = -\infty$.

Let N_1 be a maximum iteration number and ϵ be a termination tolerance as the gap between optimal values of the primal and dual master problem.

Main Step Guaranteeing good bounds

Step (I) Dual subproblem phase

Let $\bar{\mu}^k = \frac{1}{k}\mu^{k-1} + \frac{k-1}{k}\bar{\mu}^{k-1}, \bar{v}^k = \frac{1}{k}v^{k-1} + \frac{k-1}{k}\bar{v}^{k-1}$.

Solve $\varphi(\bar{\mu}^k, \bar{v}^k)$ and get y^k .

If $\varphi(\bar{\mu}^k, \bar{v}^k) \geq v_L$, set $v_L = \varphi(\bar{\mu}^k, \bar{v}^k)$.

If $(v_U - v_L)/v_U \times 100 \leq \epsilon$ or $t \geq N_1$, then TERMINATE.

Step (II) Primal subproblem phase

Let $\bar{y}^k = \frac{1}{k}y^k + \frac{k-1}{k}\bar{y}^{k-1}$.

Solve $\psi(\bar{y}^k)$ and get μ^k, v^k and x^k .

Set $k \leftarrow k + 1, t \leftarrow t + 1$.

If $\psi(\bar{y}^k) \leq v_U$, set $v_U = \psi(\bar{y}^k), y^{best} = \bar{y}^k$ and $x^{best} = x^k$.

Otherwise, go to Step (III).

If $(v_U - v_L)/v_U \times 100 \leq \epsilon$ or $t \geq N_1$, then TERMINATE.

Step (III) Addition of valid inequalities

Identify valid inequalities violated by (y^{best}, x^{best}) and add them to CSP_y and dualize them to $\text{CSD}_{(\mu, v)}$.

Reset $k = 1$, and $\bar{y}^k = 0, \mu^k = \bar{\mu}^k, v^k = \bar{v}^k$.

Go to step (I).

4. Computational Results

We performed several computational tests suitable for the research goals, applying DMVCD to both known benchmark test problems as well as to proper modifications of those problems. In detail, we compared DMVCD with both MVCD and CPLEX. Most of Lagrangian heuristic methods concern the lower bound computation. To do so, they generally employed the Lagrangian relaxation that provides the same bounds as MVCD, which also makes us to assess DMVCD against MVCD. In addition, CPLEX was included in the computational tests, which is a widely used commercial optimization tool for solving MIP, because it solves MIP using a very general and robust algorithm based on branch-and-cut,

which can also generate certain cuts in order to make the relaxation a better approximation of the original MIP. Our comparisons were made on quality (i.e., bounds and time to solve) of the solutions generated in a rather different size of the CFLP problems.

The proposed algorithm (DMVCD) was implemented in Visual C++ and run on an IBM PC with 32GB RAM, an Intel Xeon processor of 2.53 GHz and a 64-bit Windows operating system. The transportation problems (the primal subproblems) and the binary knapsack problems (the dual subproblems), both of which were added by flow cover inequalities, were solved by the dual simplex and MIP Optimizer contained in CPLEX 12.1. In addition, we took advantage of CPLEX Concert Technology in order to repeatedly call out CPLEX as a subproblem solver in the proposed algorithm procedure.

For the computational tests below, we define some measures as follows:

Gap: $\frac{|v_U - v_L|}{v_U} \times 100$ (unit: %),

cpu: the computational time spent on solving P_{CFL} (unit: seconds),

D_{opt} : $(1 - \frac{|v^* - v_L|}{v^*}) \times 100$, where v^* is the known optimal value (unit: %),

P_{opt} : $(1 - \frac{|v^* - v_U|}{v^*}) \times 100$, where v^* is the known optimal value (unit: %).

Here, the upper bound v_U and lower bound v_L used to evaluate Gap (%) represent \hat{v}_C (\tilde{v}_C) and \hat{v}_D (\tilde{v}_D) for DMVCD, v_C and v_D for MVCD. Also, D_{opt} and P_{opt} represent the dual and primal optimality, respectively.

The main objective of the present study focuses on developing some implementation schemes of MVCD available for CFLP. Thus, we first want to see whether the proposed algorithm really works on CFLP, comparing its performance to that of MVCD. To this end, our first computational test runs were performed on the Beasley's test problems that are available from "OR-Library: Distributing test problems by electronic mail". Table 1 shows the results. Note that the test runs were limited to only 150 iterations and $\epsilon \leq 0.2$. There are 4 columns in Table 1, namely "instance", "DMVCD", "MVCD", and "improvement" that mean, respectively, the problem sizes and names, results of DMVCD and MVCD, and the improvements achieved from DMVCD compared to MVCD. As shown in Table 1, for smaller size problems up to 25×50 , where the first number is the number of potential facilities and the second is the number of customers, DMVCD achieves a 99% optimality while keeping the small primal-dual gap of about 1% and especially satisfying $\epsilon \leq 0.2$ for cap41, cap42, cap43, and cap44. In contrast, MVCD yields an optimality from 96.39% to 98.35% with the gap of about 3% to problems such as cap81, cap82, cap83, cap91 and cap92. Thus, it is thought that better solutions can be generated from Lagrangian dual in conjunction with valid inequalities than the solutions that are taken from only Lagrangian dual, even though the valid inequalities are implicitly imposed.

For instances with a size of 50×50 with the large duality gap, the optimality of MVCD was 80.90% to 88.31% except that of cap131. However, we can see more improvement of DMVCD over MVCD, which points out that DMVCD performs better on problems with a large duality gap.

There are 6 instances of cap71, cap101, cap102, cap103, cap104, and cap131 with no improvement of DMVCD, meaning no valid inequalities were generated. These instances were made up of facilities that have the same constant capacity as total demands given and fixed charges with a large variance, so that very few facilities are mostly opened.

Table 1
Comparison of DMVCD and MVCD ($N_1 = 150, \epsilon \leq 0.2$).

Instance	DMVCD			MVCD			Improvement		
	Gap	cpu	D _{opt}	Gap	cpu	D _{opt}	Gap	D _{opt}	
16 × 50	cap41	0.20	2.61	99.84	0.43	3.60	99.67	0.24	0.17
	cap42	0.20	2.34	99.83	0.40	3.40	99.67	0.20	0.16
	cap43	0.20	0.90	99.80	0.40	3.39	99.67	0.20	0.13
	cap44	0.20	0.73	99.80	0.37	3.51	99.69	0.17	0.11
	cap51	0.37	3.68	99.68	0.70	3.34	99.44	0.33	0.24
	cap61	0.47	3.46	99.71	1.46	3.53	98.99	0.99	0.72
	cap62	0.45	3.48	99.75	1.34	3.34	99.26	0.89	0.49
	cap63	0.82	3.67	99.23	1.25	3.45	99.15	0.43	0.08
cap71	0.38	3.45	99.75	0.38	3.42	99.75	0.00	0.00	
25 × 50	cap81	1.22	6.26	99.11	3.95	5.34	96.39	2.73	2.72
	cap82	0.62	6.86	99.38	3.57	5.37	96.98	2.95	2.40
	cap83	1.03	6.24	99.51	3.27	5.43	97.33	2.23	2.18
	cap91	0.85	5.27	99.39	2.89	5.51	98.06	2.05	1.33
	cap92	0.77	5.66	99.46	2.61	5.30	98.35	1.84	1.12
	cap101	1.09	5.26	99.37	1.09	5.41	99.37	0.00	0.00
	cap102	0.92	5.51	99.43	0.92	5.34	99.43	0.00	0.00
	cap103	0.82	5.23	99.44	0.82	5.30	99.44	0.00	0.00
cap104	0.67	5.27	99.74	0.67	5.34	99.74	0.00	0.00	
50 × 50	cap111	8.78	10.58	94.24	24.65	10.94	85.44	15.87	8.80
	cap112	0.98	10.98	99.60	27.42	10.75	85.19	26.43	14.41
	cap113	0.82	10.89	99.29	25.57	10.78	86.28	24.75	13.01
	cap114	1.26	11.11	99.08	23.56	10.50	86.86	22.31	12.22
	cap121	14.80	10.30	90.04	29.04	10.67	80.90	14.24	9.14
	cap122	13.79	10.56	90.71	31.87	10.87	82.23	18.08	8.48
	cap123	13.06	10.61	91.09	31.67	10.64	86.85	18.61	4.24
	cap124	12.26	10.75	91.59	29.92	10.62	88.31	17.66	3.29
	cap131	2.03	10.23	99.14	2.03	10.34	99.14	0.00	0.00

Table 2
Primal and dual end-tail behaviors of DMVCD.

Instance	$N_1 = 250$		$N_1 = 300$		$N_1 = 350$		$N_1 = 500$		
	P _{opt}	D _{opt}							
50 × 50	cap111	97.78	96.81	98.05	97.46	98.32	97.83	98.85	98.46
	cap121	96.83	93.82	97.37	94.84	97.63	95.71	98.08	97.26
	cap122	96.75	94.58	97.23	95.56	97.62	96.20	98.25	97.43
	cap123	97.34	94.49	97.70	95.50	97.73	96.51	97.88	96.70
cap124	97.63	94.74	97.70	95.98	97.91	96.25	97.91	96.25	

In effect, we can say that DMVCD performs well enough on CFLP and shows excellence on problems with a large duality gap.

Although we see that DMVCD generates sufficient bounds, we want to investigate its end-tail behavior in order to confirm that it practically converges to the near optimal values. In so doing, we enlarge the maximum number of iterations from 150 to 500 as shown in Table 2. This experiment was only carried out where there was in a stark con-

trast in terms of Gap (i.e., cap111, cap121, cap122, cap123, and cap124). In this table, we see that both the primal and dual optimality gradually increase and finally converge to some accumulation points, even though the asymptotic behavior still occurs as conjectured. According to our convergent procedures of DMVCD, $v_C \ll \hat{v}_S = \hat{v}_D \leq v^*$ holds. However, we observed that \hat{v}_S are mostly greater than v^* . It seems that the behaviors of upper bounds (\hat{v}_S) are caused by approximating the solution space to $\text{conv}(Z)$. From this experiment, it is certain that the convergence of DMVCD also becomes more asymptotic as the bounds get closer to optimal values, even though DMVCD is more efficient than MVCD.

In addition to Tables 1 and 2, in order to ascertain the excellence of DMVCD for a large-size CFLP and computational efficiency, we modified Beasley's large scale problems with a size of 100×1000 since the original large scale problems are made up facilities that have the large constant capacities to demands and fixed charges with a large variance so that there are few trade-offs between the fixed charges and the delivery costs when some facilities are opened. Also the fixed charges of the facilities are very expensive, so the delivery costs appear to have little effect on the objective values. Thus, we modified Beasley's problems in two new types of problem. The first was obtained by reducing the constant capacity of each facility by the tightness of $\sum_i s_i / \sum_j d_j$. For the second type, we randomly generated the capacities of the facilities according to $s_i \sim U[\sum_i s_i / m - \alpha, \sum_i s_i / m + \beta]$, $\forall i$ for appropriately chosen values of $\alpha, \beta \geq 0$. In this test, we evenly used the tightness between 3.00 and 6.00 and set α, β as 1,000. For both types, the fixed charges were adjusted proportionally to the capacities, and the original costs and demands were used without modifications. The test problem instances are divided into 'capC' and 'capR' representing problems with constant capacities and problems with randomly generated capacities, respectively. Nine instances for each 'capC' and 'capR' were generated.

Table 3 shows how well the large scale CFLP is solved by DMVCD, MVCD, and CPLEX, given $N_1 = 150$ and $\epsilon \leq 0.2$. Here CPLEX equally employed MIP Optimizer in version 12.1. Note that the CPLEX MIP Optimizer solves MIP problems based on the branch-and-cut approach, which is almost tuned in by its default parameter settings (refer to <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>). Also we set 2,500 seconds as a termination time limit for CPLEX. For conducting cost-effectiveness evaluations, the column "cost_{eff}", expressed as $(\text{cpu}_{\text{DMVCD}} / \text{cpu}_{\text{CPLEX}}) \times 100(\%)$, is used and indicates how quickly DMVCD reaches the optimal or feasible solutions produced by CPLEX. Therefore, the (near) optimal values of CPLEX are used, as v^* , to calculate the dual optimality, namely D_{opt} . In this context, we can say that DMVCD is cost-effective if DMVCD produces good optimality on the condition of a lower cost_{eff}. In addition, the column "status" in Table 3 indicates whether each instance is exactly solved by CPLEX within the given time limit.

In the instances of capC1–capC9 and capR3–capR9, DMVCD reached an optimality of 97% with dramatic improvements from MVCD during a similar run time to that by MVCD. DMVCD is far superior to MVCD in terms of both the quality of the bounds provided and

Table 3
Performance and cost-effectiveness of DMVCD for large scale problems with a size of 100×1000
($N_1 = 150, \epsilon \leq 2$).

Instance	DMVCD			MVCD			Improvement		CPLEX		
	Gap	D _{opt}	cpu	Gap	D _{opt}	cpu	Gap	D _{opt}	cpu	Status	cost _{eff}
capC1	2.50	97.85	439.11	26.49	81.15	426.27	24.00	16.69	2500.80	Feasible	17.6%
capC2	2.70	97.62	439.73	26.27	81.08	428.55	23.56	16.54	2500.52	Feasible	17.6%
capC3	1.98	97.96	343.03	22.56	84.58	429.38	20.57	13.38	2500.45	Feasible	13.7%
capC4	4.26	96.35	442.98	20.67	85.99	431.14	16.41	10.36	2345.59	Optimal	18.9%
capC5	4.78	96.36	570.68	20.90	85.00	632.47	16.12	11.36	2153.36	Optimal	26.5%
capC6	1.98	98.10	514.94	31.96	79.33	514.94	29.98	18.77	1370.05	Optimal	37.6%
capC7	3.44	97.38	444.93	18.12	88.52	433.89	14.67	8.86	894.17	Optimal	49.8%
capC8	1.99	98.18	301.55	22.05	87.07	553.63	20.07	11.12	1220.76	Optimal	24.7%
capC9	6.07	96.27	572.34	18.16	90.28	540.42	12.09	5.98	1375.55	Optimal	41.6%
capR1	9.69	92.59	529.51	9.69	92.59	530.98	0.00	0.00	2500.54	Feasible	21.2%
capR2	6.20	95.27	527.31	6.20	95.27	567.78	0.00	0.00	2500.57	Feasible	21.1%
capR3	3.44	97.38	444.93	18.12	88.52	433.89	14.67	8.86	884.17	Optimal	50.3%
capR4	3.70	96.65	538.99	22.42	86.65	547.72	18.72	10.00	1272.96	Optimal	42.3%
capR5	2.67	97.16	568.17	38.82	75.77	568.17	36.15	21.39	2452.93	Optimal	23.2%
capR6	2.04	90.68	539.80	29.45	74.46	539.80	27.41	16.21	1572.76	Optimal	34.3%
capR7	6.22	95.69	567.86	14.11	91.58	546.87	7.88	4.11	2500.57	Feasible	22.7%
capR8	2.36	97.68	646.32	26.90	84.46	646.32	24.53	13.22	2500.57	Feasible	25.8%
capR9	2.07	97.83	595.11	30.71	79.51	595.11	28.64	18.32	1580.21	Optimal	37.7%

the cost-effectiveness in these instances. Furthermore, DMVCD provided lower bounds at approximately 97% of the (near) optimal values of CPLEX, consuming only about 20% to 50% of the computational time of CPLEX. Especially for capC3, DMVCD satisfied the termination rule, $\epsilon \leq 2$, with a cost_{eff} of only 13.7%, whereas CPLEX was not able to solve capC3 within the time limit.

A note for capR6 is further needed here. It has an optimality of 90.68% and a Gap of 2.04%. This can be seen that the upper bound of DMVCD rapidly converged to the bound such \hat{v}_S as $v_D \leq \hat{v}_S = \hat{v}_D \leq v^*$ before the valid inequalities were sufficiently generated. Thus, more valid inequalities are needed in order to improve both the upper and lower bounds simultaneously.

As for the instances such as capR1 and capR2, there were no differences between DMVCD and MVCD. This means that no flow cover inequalities were generated, since the structures of the problems are not suitable for DMVCD. As a result, DMVCD only obtained the same level as MVCD.

Also, Table 4 shows a new comparison of DMVCD and an up-to-date heuristic algorithm. As a benchmark, we employed the tabu search (TS) procedure proposed by Sun (2012), which is more effective than the LRISS method (Lorena and Senne, 1999), an improved subgradient-based Lagrangian heuristic method. The program code for TS was obtained from the original author, Sun (2012). For the same problems used in Table 3, we compared the performance and cost-effectiveness between DMVCD and TS. The experiment was done on a 64-bit LINUX computer that

Table 4
Comparison of DMVCD and TS.

Instance	DMVCD		TS					
			P_1		P_2		P_3	
	P _{opt}	cpu						
capC1	99.65	146.34	99.36 (0.29)*	154.62	99.36 (0.29)*	191.08	99.22 (0.43)*	177.31
capC2	99.67	146.28	99.40 (0.27)*	160.29	99.40 (0.27)*	196.61	99.41 (0.26)*	181.99
capC3	99.94	125.36	99.18 (0.76)*	178.59	99.26 (0.68)*	219.71	99.10 (0.84)*	196.35
capC4	99.36	147.70	99.09 (0.27)*	168.16	99.09 (0.27)*	207.84	99.05 (0.31)*	202.65
capC5	98.81	144.24	99.47 (-0.66)	160.86	99.58 (-0.77)	204.19	99.32 (-0.51)	171.63
capC6	99.92	141.84	99.22 (0.70)	131.94	99.22 (0.70)*	159.83	99.22 (0.70)*	159.35
capC7	99.15	144.25	99.42 (-0.27)	176.94	99.42 (-0.27)	216.53	98.97 (0.18)*	167.93
capC8	99.82	133.73	99.64 (0.18)	113.31	99.64 (0.18)*	136.14	99.35 (0.47)	119.88
capC9	97.51	138.53	98.66 (-1.15)**	113.22	98.66 (-1.15)	145.30	98.71 (-1.20)	152.30
capR1	97.48	146.67	85.84 (11.64)*	149.47	85.84 (11.64)*	182.52	85.89 (11.59)*	178.22
capR2	98.44	144.99	99.09 (-0.65)**	135.50	99.09 (-0.65)	172.09	99.31 (-0.87)	155.73
capR3	99.15	141.97	99.32 (-0.17)	174.65	99.32 (-0.17)	216.59	98.97 (0.18)*	166.03
capR4	99.64	143.54	99.63 (0.01)	100.59	99.70 (-0.06)**	125.84	99.55 (0.09)	123.85
capR5	99.82	165.67	99.45 (0.37)	113.47	99.45 (0.37)	138.80	99.62 (0.20)	124.37
capR6	92.56	164.05	93.08 (-0.52)**	107.13	93.08 (-0.52)**	131.64	93.17 (-0.61)**	126.49
capR7	97.96	164.27	99.38 (-1.42)**	114.86	99.38 (-1.42)**	147.97	99.33 (-1.37)**	125.56
capR8	99.96	252.00	99.11 (0.85)	101.68	99.25 (0.71)	128.54	99.23 (0.73)	131.93
capR9	99.89	154.92	99.28 (0.61)	101.50	99.28 (0.61)	123.01	99.05 (0.84)	110.90

(Value) = P_{opt} in DMVCD - P_{opt} in TS.

** DMVCD has better P_{opt} than TS in a less time taken.

** TS has better P_{opt} than DMVCD in a less time taken.

has two AMD Opteron Dual Core 2.6 GHz processors and 4 GB RAM. Note that DMVCD was also run on the same computing environment with maximum 150 iterations and $\epsilon = 0.2$. Let $P = \{\alpha_1, \alpha_2, C, \bar{m}, [l_0^l, l_0^u], [l_1^l, l_1^u]\}$ as a set of parameters of TS. To specify the set of parameters for TS, we considered the following parameter sets: (i) $P_1 = \{2, 1, 40, 40, [10, 15], [10, 15]\}$, (ii) $P_2 = \{2, 1, 50, 40, [10, 15], [10, 15]\}$, and (iii) $P_3 = \{2, 1, 50, 40, [10, 25], [10, 25]\}$. These parameter sets were intentionally considered to make the search of TS more intensive and extensive as suggested by Sun (2012).

In Table 4, P_{opt} was calculated by using the optimal (or feasible) objective values of CPLEX in Table 3. The positive values in parenthesis in Table 4 mean that DMVCD produces the better solutions than TS and the negative values are the opposite case. Even though TS is very effective against the well-known most effective Lagrangian heuristic procedure for the CFLP in Sun (Sun, 2012), DMVCD is quite comparable to TS. In detail, we observed that DMVCD resulted in better performance for capC problems, whereas TS was better fit to capR problems. Strikingly interesting is, as shown in CapR1, DMVCD guaranteed better bounds rather than TS, even at the worst case of TS.

In conclusion, all experimental results clearly indicate DMVCD performs well for CFLP against other algorithms.

5. Conclusions and Further Research

In this article, we proposed the Dynamic Mean Value Cross Decomposition (DMVCD), which is effectively integrated with MVCD and the cutting plane methods in order to solve CFLP. In the combinatorial optimizations, although MVCD is an attractive method in that it can effectively provide the information of solutions and the same bounds as those of the Lagrangian relaxation, it has not been widely applied to other problems except UFLP, due to the fact that the performance is highly dependent on the structure of the problem. The present study showed that DMVCD ensures better bounds than those of Lagrangian relaxation or MVCD. Furthermore, the easy-to-solve structures remain intact as in MVCD and a scheme to remove the earlier solutions to decrease the rate of convergence is additionally included. For these reasons, it is expected that DMVCD is able to save computational efforts, even though an additional separation heuristic procedure is required to identify the valid inequalities. In addition, the computational results for CFLP demonstrated the effectiveness and efficiency of our proposed algorithm.

Even though the benefits of DMVCD are evident, it is impossible to obtain the exact optimal integer solutions because the asymptotic convergence and the duality gap still remain. Therefore, based on the DMVCD method, we need to develop a way to recover integer solutions, hopefully close to the optimal solutions. It is thought that a combination with the MVCD-based branch-and-bound method (Holmberg, 2005) may be a good alternative in that both DMVCD and MVCD-based branch-and-bound are complementary, as mentioned in Section 3. A study on this topic is planned for the near future. In addition to this combination, a technique to adjust the convex combination weights to accumulate solutions is needed, e.g., a dually-weighted method (Wentges, 2001) and GWMVC (general weighted mean value cross decomposition) (Holmberg, 2005), in order to accelerate the rate of convergence. From our computational results, we found that the flow cover valid inequalities were not well generated depending on the problem structure, despite the existence of a gap between the upper and lower bounds. Therefore, for the robustness of DMVCD, more empirical studies with flow cover inequalities as well as effective capacity inequalities and submodular inequalities are needed. Through these further studies, it will be possible to develop a method to adaptively use the valid inequalities relevant to the structures of the problems, which makes DMVCD more effective and robust.

Finally, DMVCD can be extended to large scale optimization problems as an efficient approximate algorithm to guarantee acceptable bounds. However, unfortunately, DMVCD cannot be applied to large scale optimization problems with an ill-defined polyhedral structure. For this case, RLT (Reformulation-Linearization Techniques) would be a good alternative approach.

Acknowledgement. The authors are very grateful to Professor Minghe Sun in the Department of Management Science and Statistics, College of Business, at University of Texas at San Antonio, San Antonio, TX 78249-0632, USA, for his generosity and kindness in letting the authors use his C code with detailed explanation in the computational experiments. Also, the authors gratefully acknowledge the help of Professor Hokyoung B. Ryu in

the Graduate School of Technology and Innovation Management at Hanyang University, Seoul, Korea, to review and correct this article in English.

References

- Aardal, K. (1998a). Capacitated facility location: separation algorithms and computational experience. *Mathematical Programming*, 81(2), 149–175.
- Aardal, K. (1998b). Reformulation of capacitated facility location problems: how redundant information can help. *Annals of Operations Research*, 82, 289–308.
- Aardal, K., Pochet, Y., Wolsey, L.A. (1995). Capacitated facility location: valid inequalities and facets. *Mathematics of Operations Research*, 20(3), 562–582.
- 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(10), 1072–1084.
- Avella, P., Boccia, M. (2009). A cutting plane algorithm for the capacitated facility location problem. *Computational Optimization and Applications*, 43(1), 39–65.
- Barahona, F., Chudak, F.A. (2005). Near-optimal solutions to large-scale facility location problems. *Discrete Optimization*, 2(1), 35–50.
- Beasley, J.E. (1993). Lagrangean heuristics for location problems. *European Journal of Operational Research*, 65(3), 383–399.
- Chen, C.H. and 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*, 44C(6), 1099–1122.
- Cornuéjols, G., Sridharan, R., M., T.J. (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research*, 50(3), 280–297.
- Du, D., Wang, X., Xu, D. (2010). An approximation algorithm for the k -level capacitated facility location problem. *Journal of Combinatorial Optimization*, 20(4), 361–368.
- Geoffrion, A.M. (1974). *Lagrangean Relaxation for Integer Programming*, Vol. 2. Springer, Berlin.
- 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(3), 268–274.
- Holmberg, K. (1992). Linear mean value cross decomposition: a generalization of the Kornai–Liptak method. *European Journal of Operational Research*, 62(1), 55–73.
- Holmberg, K. (1994). A convergence proof for linear mean value cross decomposition. *Mathematical Methods of Operations Research*, 39(2), 157–186.
- Holmberg, K. (1997). Mean value cross decomposition applied to integer programming problems. *European Journal of Operational Research*, 97(1), 124–138.
- Holmberg, K. (2001). Experiments with primal-dual decomposition and subgradient methods for the uncapacitated facility location problem. *Optimization*, 49(5), 495–516.
- Holmberg, K. (2005). *Mean value cross decomposition based branch-and-bound for mixed integer programming problems*. Technical report, Department of Mathematics, Linköping Institute of Technology, Sweden.
- Holmberg, K., Kiwiel, K.C. (2006). Mean value cross decomposition for nonlinear convex problems. *Optimization Methods and Software*, 21(3), 401–417.
- 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(3), 544–559.
- Kalcsics, J., Nickel, S., Puerto, J. (2010). The ordered capacitated facility location problem. *TOP*, 18(1), 203–222.
- 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(5), 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(2), 408–421.
- Leung, J.M.Y. and Magnanti, T.L. (1989). Valid inequalities and facets of the capacitated plant location problem. *Mathematical Programming*, 44(1), 271–291.
- Lorena, L.A., Senne, E.L.F. (1999). Improving traditional subgradient scheme for lagrangean relaxation: an application to location problems. *International Journal of Mathematical Algorithms*, 1, 133–151.

- Magnanti, T.L., Wong, R.T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3), 464–484.
- Ortega, F., Wolsey, L.A. (2003). A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. *Networks*, 41(3), 143–158.
- Padberg, M.W., Van Roy, T.J., Wolsey, L.A. (1985). Valid linear inequalities for fixed charge problems. *Operations Research*, 33(4), 842–861.
- Sinha, P. (2009). Performance of an add-drop-interchange heuristic for the capacitated facility location problem. *International Journal of Applied Management Science*, 1(4), 388–400.
- Sridhar, V., Park, J.S. (2000). Benders-and-cut algorithm for fixed-charge capacitated network design problem. *European Journal of Operational Research*, 125(3), 622–632.
- Sridharan, R. (1993). A Lagrangian heuristic for the capacitated plant location problem with single source constraints. *European Journal of Operational Research*, 66(3), 305–312.
- Sun, M. (2012). A tabu search heuristic procedure for the capacitated facility location problem. *Journal of Heuristics*, 18(1), 91–18.
- Van Roy, T. (1983). Cross decomposition for mixed integer programming. *Mathematical Programming*, 25(1), 46–63.
- Van Roy, T.J. (1986). A cross decomposition algorithm for capacitated facility location. *Operations Research*, 34(1), 145–163.
- Wentges, P. (1996). Accelerating benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research*, 44(2), 267–290.
- Wentges, P. (2001). *On Cross Decomposition for Mixed-Integer Programming*. Physica-Verlag, New York.
- Wolsey, L.A. (1989). Submodularity and valid inequalities in capacitated fixed charge networks. *Operations Research Letters*, 8(3), 119–124.
- Yang, Z., Chu, F., Chen, H. (2012). A cut-and-solve based algorithm for the single-source capacitated facility location problem. *European Journal of Operational Research*, 221(3), 521–532.

C. Kim is a PhD candidate in the department of Industrial Engineering at Hanyang University, Seoul, Korea. The main research interest is to develop efficient algorithms to solve combinatorial optimization problems, in particular with the use of approximation methods in the decomposition framework.

G. Choi is Dean of Graduate School of Technology and Innovation Management and professor of Industrial Engineering at Hanyang University. He holds MS and PhD in Industrial and Systems Engineering from Virginia Tech., USA. Current research interests include operations research, technology management, and innovation management.

S.-S. Ko is an associate professor in the department of industrial engineering at Konkuk university, Seoul, Korea. He received his PhD in 2003 and MS in 1999 from the School of Industrial and Systems Engineering at the Georgia Institute of Technology and a BS in Industrial Engineering from Hanyang University, Seoul. His research interests are in operations research, production and inventory control, parallel processing, collaboration systems and financial engineering.

Dinaminio vidutinės vertės kryžminio skaidymo algoritmas apribotos talpos objektų vietos parinkimo uždaviniams

Chulyeon KIM, Gyunghyun CHOI, Sung-Seok KO

Šiame straipsnyje pasiūlytas praktinis algoritmas apribotos talpos objektų vietos parinkimo uždaviniams spręsti. Yra keletas būdų, skirtų pirminiam sprendiniui rasti naudojantis pirmine ir antrine struktūromis. Vienas iš šių būdų yra vidutinės vertės kryžminio skaidymo metodas. Anksčiau šis metodas buvo taikytas tik neapribotos talpos objektų vietos parinkimo uždaviniams, nes pajėgumas labai priklauso nuo uždavinio struktūros. Siūlomas algoritmas, vadinamas dinaminio vidutinės vertės kryžminio skaidymo algoritmu, yra efektyvi vidutinės vertės kryžminio skaidymo ir pjaunančiųjų plokštumų metodų integracija. Pateikiami įvairių uždavinių sprendimo pasiūlytu algoritmu rezultatai.