International Journal of Management Science
Vol 5, No 2, November 1999

Generalized Multicommodity Distribution System Design

Ju-Long Chen
National Defense College, P.O. Box 90046-15,
Cheng-Ho, Taipei, Taiwan

James G. Morris
School of Business, UW-Madison, 975 University Avenue,
Madison, WI 53706-1323

Ki Hyoung Son
Dept. of Business Administration, Chonnam Nat'l University,
Kwangju, 500-757 Korea

(Received June 1999 ; revision received October 1999)

ABSTRACT

This paper generalizes the classic two-stage multicommodity distribution system design problem to the one that includes plant locations as well as distribution center locations. Accommodating plant location leads to subproblems which are mixed integer. Hence, no LP-type duals of the subproblems are available, and therefore standard Benders decomposition no longer applies. We develop new solution method which combines an integer L-shaped method with Benders decomposition to suit the purpose, and present the test results.

1. INTRODUCTION

Multicommodity distribution system considered in this paper can be summarized as follows. There are several commodities which can be produced at several potential plants. The demand of each commodity at each customer zone is known, and is satisfied by shipping via regional distribution center (abbreviated DC). The maximum annual output of each commodity from each plant is given. Also known are lower as well as upper bounds on the allowable total annual throughput for each DC. Potential sites for the plants and DCs are also given, but the particular locations to be used are to be selected within the model. The plant and DC costs are based on fixed plus linear variable costs. Transportation costs are also assumed to be linear. Thus the problem is to decide which plants and distribution centers to use, how much of each commodity is to be produced at each selected plant, what size of DC to have at each selected site, which customers are to be assigned to which DCs, and which plants are to supply which commodities to which DCs. This should be done so as to minimize total cost while honoring capacity constraints and satisfying customer demands. There may be additional configuration constraints on the plants and distribution centers.

Formulation of the proposed model incorporates the following notation:

C_{ijk}	transportation cost per cwt for commodity i from plant j to DC k,
x_{ijk}	the amount of commodity i supplied from plant j to DC k,
a_j	non-negative fixed cost of operating plant j per unit time,
b_j	binary variable that equals 1 if plant j is opened, and 0 otherwise,
q_{ij}	average variable cost per unit time to produce commodity i at plant j,
S_{ij}	amount of commodity i produced at plant j,
f_k	non-negative fixed cost of operating DC k per unit time,
z_k	binary variable that equals 1 if DC k is opened, and 0 otherwise,
v_k	average variable cost per cwt of operating DC k,
D_{il}	customer l's demand for commodity i,
y_{kl}	binary variable that equals 1 if the path from DC k to customer l is opened,
t_{ikl}	average unit cost of storing, processing and shipping commodity i from DC k to customer l,
CH_{ij}	the maximum output capacity for commodity i at plant j per unit time,
VH_k	the maximum total throughput at DC k per unit time,
VL_k	the maximum total throughput at DC k per unit time.

The problem can now be formulated as follows. It is understood that all summations run over the allowable combinations of the indices.

$$(P)\quad Min \quad \sum_k \left[f_k z_k + v_k \sum_{il} D_{il} y_{kl} \right] + \sum_{ikl} t_{ikl} D_{il} y_{kl} + \sum_j a_j b_j + \sum_{ij} q_{ij} s_{ij} + \sum_{ijk} c_{ijk} x_{ijk}$$

s.t.

$$\sum_k y_{kl} = 1, \qquad \text{for } all \; l \qquad (1)$$

$$VL_k z_k \le \sum_{il} D_{il} y_{kl} \le VH_k z_k, \qquad \text{for } all \; k \qquad (2)$$

$$\sum_k x_{ijk} \le s_{ij}, \qquad \text{for } all \; i \text{ and } j \qquad (3)$$

$$\sum_j x_{ijk} = \sum_l D_{il} y_{kl}, \qquad \text{for } all \; i \text{ and } k \qquad (4)$$

$$s_{ij} \le CH_{ij} b_j, \qquad \text{for } all \; i \text{ and } j \qquad (5)$$

Linear configuration constraints on y, z, and/or b (6)

$$b_j, z_k, \text{ and } y_{kl} \in \{0,1\}, \qquad \text{for } all \; j, k \text{ and } l \qquad (7)$$

$$s_{ij}, x_{ijk} \ge 0 \qquad \text{for } all \; i, j \text{ and } k \qquad (8)$$

Constraint (1) specifies that each customer demand must be served by only one DC. Besides keeping total throughput at DC k between minimum and maximum allowed, constraint (2) also enforces the correct logical relation between y and z. Constraint (3) is a supply constraint for each commodity at each plant, constraint (4) requires that each customer's demand must be fully satisfied, and constraint (5) is to ensure that no commodity is to be supplied from a plant not chosen for a plant location. Constraint (6) gives the model quite a lot of flexibility to incorporate many practical consideration. For instance, one may extend this model to allow expandable capacity limits on plants and DCs. The model makes two assumptions; single sourcing and origin forgetting policy.

Single sourcing policy by constraint (1) requires that no customer zone is allowed to deal with more than one DC. Geoffrion, Graves and Lee [9] reported that this assumption is usually justified in practice, and tends to reduce small shipment. Of course, relaxing this assumption converts each y_{kl} to a continuous variable, making the problem much easier to solve. Another feature of the model is origin-forgetting policy. Under this policy, commodities lose their identity when transferred through DC points. Geoffrion and Graves [8] adopted origin-remembering policy with quadruply subscripted transportation variables in their model, citing practical needs and advantages in some situation over origin-forgetting policy. Elson [6] used origin-forgetting policy with triply subscripted variables, employing separate transportation variables for plant-to-DC and DC-to-customer links. Others with origin-forgetting policy includes Ellwein [5] and Pooley [13]. Note that changing the policy to origin-remembering policy has no significant impact on the problem (P) except increase in the number of transportation variables. Other important works in facility location and distribution system design include Erlenkotter [7] and Bilde and Krarup [2] for simple uncapacitated location model, Christofides and Beasley [4] and Van Roy [14] for capacitated warehouse location model, Warszawski [18] and Laundy [12] for multicommodity location model, and Elson [6] for multicommodity capacitated facility location model. The structure of the paper is as follows. In Section 2, we briefly review the integer L-shaped method. Section 3 focuses on application of the integer L-shaped method to the problem (P), and develops how to generate continuous L-shaped cuts. The algorithm to solve the problem (P) will be proposed. Computational results is reported in Section 4.

2. THE INTEGER *L*-SHAPED METHOD

A deterministic large-scale mathematical programming can be solved by a variety of specialized approaches. These approaches typically use some forms of decomposition strategy such as Benders decomposition (Benders [1], Wets [16], Wets [17]), Dantzig -Wolfe decomposition (Dantzig and Wolfe [3]), and the integer L-shaped method (Laporte and Louveaux [10]). The scheme employed here is to combine the integer L-shaped method with Benders decomposition to solve the problem (P). Let us briefly summarize the integer L-shaped method. The object of the integer L-shaped algorithm is to provide a decomposition of problem with first-stage integer decision variables in a two-stage solution process. The main difference between the continuous L-shaped method by Van Slyke and Wets [15] and the integer L-shaped method is the type of decision variables in the second stage. The continuous L-shaped method restricts the second-stage variables to be continuous while the integer L-shaped method can accommodate second stage decision variables which are integer. Consider a two-stage mixed-integer linear programming problem with fixed recourse which has the following form:

$$
\begin{aligned}
Min \quad & Z = ag + qy + TQ(y) \\
\text{s.t.} \quad & Ag + By = b \\
& y \in Y \cap \{0,1\}
\end{aligned}
\tag{9}
$$

, where $TQ(y) = \min\{cx \mid Wx = h - Ty, \textit{ some } x \in \{0,1\}\}, a, c, b, q,$ and h are known vectors, $A, B, W,$ and T are known matrices, and Y is the set defined by (9). It is assumed that the problem is feasible and has a finite optimal value. Then, at a given stage, so-called current problem is formed as:

$$
\begin{aligned}
Min \quad & ag + qy + \theta \\
\text{s.t.} \quad & Ag + By = b \\
& D_k y \geq d_k, \qquad k = 1,2,\cdots,s, \qquad (10) \\
& E_k y + \theta \geq e_k, \qquad k = 1,2,\cdots,t, \qquad (11) \\
& y \geq 0, \quad \theta \in R
\end{aligned}
$$

, where D_k, E_k are vectors, and d_k and e_k are scalars to be generated. Constraints (10) are referred to as feasibility cuts and constraints (11) are as optimality cuts. A set of feasibility cuts is said to be valid if there exists some finite values s such that $y \in Y$ if and only if $\{D_k y \geq d_k, k = 1, 2, \cdots, s\}$. The method used to form the feasibility cuts is same as that for Benders decomposition. Let t be the finite number and let $\hat{Y} = \{y \mid y \in Y \cap \{0,1\}\}$. A set of t optimality cuts is said to be valid if for all $y \in \hat{Y}, (y, \theta) \in \{E_k y + \theta \geq e_k, k = 1,2,\cdots,t\}$ implies $\theta \geq TQ(y)$. In order to derive optimality cuts for a two-stage problem in which some first-stage decision variables are binary and some second-stage decision variables are binary, two assumptions are required.

Assumption 1. Given binary first-stage vector of decision variables y, the function $TQ(y)$ is computable from y.

Assumption 2. There exists a finite value L satisfying $L \leq \min_y \{TQ(y) \mid y \in \hat{Y}\}$.

Now let y^p be the p^{th} feasible solution generated by the integer L-shaped method, and $TQ(y^p)$ be the corresponding second-stage objective value. Further, let $kl \in S_p$ if $y_{kl}^p = 1$, and let $kl \notin S_p$ if $y_{kl}^p = 0$. Then the integer L-shaped optimality cut can be formed as

$$\theta \geq (TQ(y^p) - L)(\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl}) - (TQ(y^p) - L)(|S_p| - 1) + L$$

, where $|S_p|$ is the cardinality of set S_p. Note that the quantity $(\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl})$ is always less than or equal to $|S_p|$. It takes the value $|S_p|$ only when y is the p^{th} feasible solution. When $(\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl})$ is equal to $|S_p|$, the right-hand side takes the value $TQ(y^p)$. For other $y \in \hat{Y}$, the right-hand side is less than or equal to L. Therefore a set of valid cuts is obtained by imposing one such constraint for each first-stage feasible solution $y \in \hat{Y}$. Then, the integer L-shaped method yields an optimal solution of the original problem, if any, in a finite number of steps. This is basically because the existence of a valid set of feasibility cuts and a valid set of optimality cuts imply s and t in (10) and (11) are finite. For further details of the algorithm, we refer to Laporte and Louveaux [10] and Larporte, Louveaux, and Hamme [11].

3. APPLICATION OF THE INTEGER *L*-SHAPED AND THE CONTINUOUS *L*-SHAPED METHOD

The problem (P) is a large scale mixed integer problem, but has a special structure that enables it to be decomposed into distribution center location problem and plant location problem with multicommodity flow.

For fixed y and z, the problem (P) can be reduced to

$$(Q) \qquad Min \quad \sum_j a_j b_j + \sum_{ij} a_{ij} S_{ij} + \sum_{ijk} c_{ijk} x_{ijk}$$

$$\text{s.t.} \quad (3),(4),(5), \text{ and } (8)$$
$$\text{Linear configuration constraints on } b \qquad (12)$$
$$b_j \in \{0,1\} \qquad \text{for } all \ j$$

Hence problem (P) can be rewritten as an equivalent two-stage problem as below.

$$(OP) \qquad Min \quad \sum_k \left[f_k z_k + v_k \sum_{il} D_{il} y_{kl} \right] + \sum_{ikl} t_{ikl} D_{il} y_{kl} + Q(y)$$

s.t. (1), (2)

Linear configuration constrains on y, and/or z (13)

$z_k, y_{kl} \in \{0,1\}$ for all k and l

For notational simplicity, constraint (12) and (13) are not spelled out in detail. The only requirement is that they be linear and do not involve x or s in (12), and no b, s, or x variables in (13). These configuration constraints can make problem (OP) or (Q) infeasible. For the sake of simplicity, however, we drop the configuration constraints in the following discussion. It is also assumed that $\sum_j CH_{ij} \geq \sum_l D_{il}$ for each commodity i to prevent problem (Q) from being infeasible. Before going further, let us first rewrite the problem (Q) in a condensed form. Nothing that $\sum_k x_{ijk} = s_{ij}$, one can remove the variable s from the model and write a condensed model, which can be solved very efficiently. Combining (3) and (5), the problem (Q) can be rewritten as follows;

$$(EQ) \qquad Min \quad \sum_j a_j b_j + \sum_{ijk} C_{ijk} x_{ijk}$$

s.t (4)

$\sum_k x_{ijk} \leq CH_{ij} b_j$, for all i and j

$b_j \in \{0,1\}$, for all i

$x_{ijk} \geq 0$, for all i, j, and k

where $C_{ijk} = q_{ij} + c_{ijk}$ for all i, j, and k. The problem (EQ) is equivalent to (Q) and can replace (Q) for further discussion. Note that the problem (OP) can be decomposed into two partitions. The first is so-called current problem in the proposed problem, to be defined later as problem (CP). It is a distribution center location problem with deterministic demand. The second partition, to be called problem (EQ) is a plant location problem with multicommodity flow transportation problem constraints. Since the problem (EQ) is a mixed integer linear problem, standard Benders optimality cut can not be formed for problem (OP). The strategy, therefore, employed here is to use the integer L-shaped method with the Benders decomposition to solve problem (OP). Note that problem (OP) satisfies the definitions and assumptions of the integer L-shaped method, since the first-partition of problem (OP) is a pure binary problem and the second-partition, problem (EQ), is a mixed-integer programming problem. Further, problem (EQ) is computable from y, and there is a lower bound L that can be easily obtained. Hence, the integer L-shaped algorithm can be used to solve problem (OP).

To derive (CP), let us first define the current problem (OCP) of (OP) at iteration P as follows.

$$(OCP) \qquad Min \quad \sum_k \left[f_k z_k + v_k \sum_{il} D_{il} y_{kl} \right] + \sum_{ikl} t_{ikl} D_{il} y_{kl} + \theta$$

$$\begin{aligned} \text{s.t.} \quad & (1), (2) & & \\ & y_{kl} \le z_k, & & \text{for } all \ \ k \text{ and } l \\ & & & \theta + E_p y \ge e_p, \ p = 1,2,\cdots,P \qquad (14) \\ & \theta \ge L, & & \qquad (15) \\ & z_k, y_{kl} \in \{0,1\}, & & \text{for } all \ \ k \text{ and } l \end{aligned}$$

, where E_p is a vector, e_p is a scalar and p is an index denothing the p^{th} iteration. Constraint (14) is an optimality cut produced by integer L-shaped method. Let us first derive valid optimality cuts. Let y^p be the value which corresponds to a value in z, say z^p, such that (y^p, z^p) is the p^{th} feasible solution generated for problem (OCP). Let S_p be a set of indices such that $kl \in S_p$ if y_{kl}^p=1 and $kl \notin S_p$ if y_{kl}^p=0. $EQ(y^p)$ is optimal objective value of the problem (EQ) with y^p. Define optimality cut as

$$\theta \ge (EQ(y^p) - L)(\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl}) - (EQ(y^p) - L)(|S_p| - 1) + L$$

The set of optimality cuts defined for all feasible solutions generated for problem (OP) in y is a valid set of optimality cuts. This is because for all feasible solutions of problem (OP) in y, when $y = y^p$, the right-hand side of the cut is equal to $EQ(y^p)$ since the quantity $\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl}$ is equal to $|S_p|$. When $y \ne y^p$, one can easily verify that the right-hand side of the cut is less than or equal to L. In addition, the number of possible feasible values of y for problem (OP) is finite. Therefore, $\theta \ge EQ(y)$ for all feasible y when all the cuts constructed from all possible values of feasible y are added to the problem (OCP).

To complete the discussion, we need to compute a value for L in (15). Let G be the feasible region of problem (OP). Then for all feasible solutions of problem (OP) in y and z, L must be less than or equal to $\min_{y \in G} EQ(y)$. Define L_1 as below.

$$L_1 = Min \quad \sum_j a_j b_j$$

$$\text{s.t.} \quad \sum_j CH_{ij} b_j \geq \sum_l D_{il}, \qquad \text{for } all \ \ i$$
$$b_j \in \{0,1\}. \qquad \text{for } all \ \ j$$

Then L_1 is the minimum of total fixed cost of opening plants to supply total demands for all commodities.

Now let $C_i^* = \min_{jk} C_{ijk}$. Then for any x, $\sum_{ijk} C_{ijk} x_{ijk} \geq \sum_i C_i^* \sum_{jk} x_{ijk}$, and for any feasible solution x, $\sum_j x_{ijk} = \sum_l D_{il} y_{kl}$. Therefore, defining

$$L_2 = \sum_i C_i^* \sum_l D_{il} = \sum_i C_i^* \sum_l D_{il} \sum_k y_{kl} = \sum_i C_i^* \sum_{jk} x_{ijk} \leq \sum_{ijk} C_{ijk} x_{ijk}$$

and letting $L = L_1 + L_2$ can serve the purpose. Combining the results, we rewrite master problem (CP) of (OP) as follows.

$$(CP) \qquad Min \quad \sum_k \left[f_k z_k + v_k \sum_{il} D_{il} y_{kl} \right] + \sum_{ikl} t_{ikl} D_{il} y_{kl} + \theta$$

$$\text{s.t.} \quad (1), (2),$$
$$y_{kl} \leq z_k, \qquad \text{for } all \ \ k \text{ and } l$$
$$\theta \geq (EQ(y^p) - L)\left(\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl}\right) - (EQ(y^p) - L)(|S_p| - 1) + L,$$
$$p = 1, 2, \cdots, p$$
$$\theta \geq L,$$
$$z_k, y_{kl} \in \{0,1\}. \qquad \text{for } all \ \ k \text{ and } l$$

Since the number of possible values of feasible solution of problem (OP) is finite, when integer L-shaped optimality cuts are constructed for all feasible solutions of problem (OP) in y, and added to problem (CP), the validity of the set of optimality cuts implies that the optimal objective value of problem (OP) is equal to problem (P). With integer L-shaped cut alone, however problem (CP) will converge very slowly. This is because the optimality cut constructed by y^p only ensures that $\theta \geq EQ(y^p)$, where $p = 1,2,\cdots, P$ and y^p is the optimal solution in y of problem (CP) at the p^{th} iteration. For y other than y^p, these cuts only make $\theta \geq L - \delta$ with $\delta \geq 0$. Hence, one may not be able to obtain good lower bound at the early stage of the iteration. One way to provide richer information on y is to include continuous L-shaped cuts in (CP), derived from LP relaxation of $EQ(y^p)$. Let $R(y^p)$ be an LP relaxation of $EQ(y^p)$. The LP dual $\overline{R(y^p)}$ of problem $R(y^p)$ can be stated as:

$$\overline{R(y^p)} \qquad Max \quad \sum_{ikl} D_{il} y_{kl}^p v_{ik} - \sum_j \pi_j$$

$$\begin{aligned} \text{s.t.} \quad & -\mu_{ij} + v_{ik} \leq C_{ijk}, && \text{for } all \ \ i,\ j, \text{and } k \\ & -\pi + CH_{ij}\mu_{ij} \leq a_j, && \text{for } all \ \ j \\ & \mu \geq 0, \pi \geq 0, \text{ and } v \ \ free. \end{aligned}$$

Hence, the continuous L-shaped cut takes the following form.

$$\theta \geq \sum_{ikl} D_{il} y_{kl} v_{ik}^p - \sum_j \pi_j^p,$$

where v_{ik}^p and π_j^p are corresponding extreme point solution. Adding this cut, the master problem (CP) at the P^{th} iteration now becomes

$$(CP) \qquad Min \quad \sum_k \left[f_k z_k + v_k \sum_{il} D_{il} y_{kl} \right] + \sum_{ikl} t_{ikl} D_{il} y_{kl} + \theta$$

$$\begin{aligned} \text{s.t.} \quad & (1), (2), \\ & y_{kl} \leq z_k, && \text{for } all \ \ k \text{ and } l \\ & \theta \geq (EQ(y^p) - L)\Big(\sum_{kl \in S_p} y_{kl} - \sum_{kl \notin S_p} y_{kl}\Big) - (EQ(y^p) - L)(|S_p| - 1) + L, \end{aligned}$$

$$
\begin{aligned}
& & p=1,2,\cdots,P & \\
&\theta \geq \sum_{ikl} D_{il} y_{kl} v_{ik}^{p} - \sum_{j} \pi_{j}^{p}, & p=1,2,\cdots,P & \quad (16)\\
&\theta \geq L, & & \\
&z_k, y_{kl} \in \{0,1\}. & \text{for } all \quad k \text{ and } l &
\end{aligned}
$$

The algorithm is now presented below.

Algorithm

Step 0. Set $p = 0$, $UB = \infty$, choose $\varepsilon \geq 0$, compute L, and store associated b as the incumbent best plant location solution found.

Step 1. Solve the current problem (CP) by any integer programming code. If the current problem (CP) is infeasible, stop. The original problem is infeasible. Otherwise, let (z^p, y^p, θ^p) be an optimal solution, and let Z^p be the optimal value.

Step 2. If $Z^p + \varepsilon \geq UB$, stop. An ε-optimal solution has been found. Otherwise, compute $EQ(y^p)$ and let $zq^p = Z^p + EQ(y^p) - \theta^p$. If $zq^p < UB$, let $UB = zq^p$ and store (z^p, y^p) as the incumbent best DC solution.

Step 3. If $\theta^p + \varepsilon \geq EQ(y^p)$, stop. The global optimal solution is reached. Otherwise, add two new optimality cuts; one by the integer L-shaped cut, and the other by continuous L-shaped method. Set $p = p+1$ and return to Step 1.

Solving problem (EQ) is not an easy task since it is a large scale mixed integer programming problem. But for given y^p, it is a classical capacitated plant location problem. Furthermore, for fixed b_j, it can be separated into as many independent classical transportation problems as the number of commodities dealing with. Let SP_i denote the i^{th} transportation problem associated with commodity i. Then, SP_i is an LP, and standard Benders decomposition scheme can be employed to solve (EQ). We used Benders decomposition in our computational experiment summarized in section 4. Detailed procedure is not shown here, however, since it is a standard procedure. One final comment follows. One can improve the speed of convergence using improved continuous L-shaped cuts. To generate improved cut, suppose that we have optimal solution of $EQ(y)$ at H^{th} iteration. Then Benders master problem (NMP) of (EQ) with integrality relaxed is as below.

$$(NMP) \quad Min \quad \sum_j a_j b_j + y^0$$

$$\text{s.t.} \quad \sum_j CH_{ij} b_j \geq \sum_l D_{il}, \qquad \text{for } all \ i$$

$$y^0 + \sum_{ij} U_{ij}^h CH_{ij} b_j \geq -\sum_{ik} \lambda_{ik}^h \sum_l D_{il} y_{kl}^{cp}, \qquad h = 1,2,\cdots,H$$

$$\sum_j a_j b_j \geq L_1,$$

$$b_j \in [0,1], \qquad \text{for } all \ j$$

, where L_1 is the value obtained earlier. Note that (NMP) provides a lower bound on the optimal objective value of problem (EQ). The dual of (NMP) takes the following form:

$$Max \quad \sum_{il} D_{il} \alpha_i - \sum_{ikh} \lambda_{ik}^h \sum_l D_{il} y_{kl}^{cp} \beta_h + \sum_j \gamma_j + L_1 \eta$$

$$\text{s.t.} \quad \sum_i CH_{ij} \alpha_i + \sum_{ih} u_{ij}^h CH_{ij} \beta_h + \gamma_j + a_j \eta \leq a_j, \qquad \text{for } all \ j$$

$$\sum_h \beta_h \leq 1,$$

$$\alpha_i, \eta, \beta_h \geq 0, \text{ and } r_j \leq 0.$$

Since the feasible region of the dual of (NMP) is independent of y and b one can construct continuous cut based on the optimal extreme point of the dual as follows.

$$\theta \geq \sum_{il} D_{il} \alpha_i - \sum_{ikh} \gamma_{ik}^h \sum_l D_{il} y_{kl} \beta_h + \sum_j \gamma_j + L_1 \eta \qquad (17)$$

One can replace (16) with (17).

4. COMPUTATIONAL RESULTS

The proposed algorithm was implemented in C on a SUN work station under UNIX system. The algorithm uses CPLEX Mixed-Integer Callable Library to solve the current problems, master problems of (EQ), and independent transportation problems. Problem files was first constructed in LINGO and then converted to MPS format to fit into CPLEX. The optimal solutions obtained were verified by CPLEX. The results were compared with those obtained by CPLEX. As for the test problem, no standard test data are available for this generalized multicommodity distribution system design problem. We, therefore, have generated the problems from U.S. road map by randomly selecting 70 cities for possible distribution locations and 40 cities for customer zones. Then, 15 cities for possible plant sites were also selected at random from distribution center locations. The distances between plant sites and DCs and between DCs and customer zones were calculated. The transportation cost file was generated assuming $0.5 per mile for one gross cwt. The customer demand data file for each commodity was generated according to a uniform distribution with a specified range. The same method was used for generating the fixed cost data file, the variable operating cost data file for each DC, the maximum capacity limit data file on DCs and plants with minimum of zero for simplicity, and the production cost data file for each plant. The number of plant sites, DC sites, customer zones, commodities, and the actual number of variables for each test problems are shown in Table 1. For all testing problems, we adopted a dense network approach in which all possible paths between the DC sites and customer zones are included. This may not be necessary in a practical setting since some path variables can be dropped from consideration through pre-processing. Note that the proposed algorithm is not very much sensitive to the increase in the number of DC sites, plants, or commodities. Table 2 shows computational results for 3 test problems. Suboptimality tolerance was set to 0.001 for Test1 and Test3, and 0.003 for Test2. The proposed algorithm tends to perform only slightly better than CPLEX in terms of speed. But, when the quality of the upper bound is concerned, it provides clear advantage over CPLEX. We have also studied the impact of changes in transportation costs, fixed costs, and capacity limits respectively. The results are summarized in Table 3 and Table 4. Last two digits in test problem names represent inflation factors. For instance, test1t12 has transportation costs of 1.2 times those used in the basis problem test1, and test2f05 has fixed costs of .5 times those used in test2. For test1cp, test2cp, and test3cp, maximum capacity limit of 1.1 times those used in the basis problem were assigned. From Table 3, we find impact of changes in relative magnitude of transportation costs or fixed costs is problem specific, and that no regularity can be established to predict how the solution times would be affected with respect to those changes. The same goes to the changes in capacity limits. Table 4 shows that loosening capacity limits has mixed effect in terms of CPU time. We also tested the algorithm with improved L-shaped continuous cut (14) added to (CP), and found that improved cut can significantly improve the speed of convergence as shown in Table 5. One final remark follows. In order to verify that the proposed algorithm can solve a larger size problem than CPLEX does, we constructed a problem with 15 plant sites, 20 DC sites, 25 customer zones, and 50 commodities. It has 535 integer variables and 15000 continuous variables. The ε-optimal solution with $\varepsilon = 0.001$ was not reported after 2 hours by CPLEX whereas proposed algorithm reported the solution in 254.1 seconds.

Table 1. Description of the test problems

	No. of Plant sites	No. of DC sites	No. of customers	No. of Commodities	No. of integer v.	No. of continuous v.	No. of Constraints
Test1	5	10	15	20	165	1000	485
Test2	10	15	20	20	325	3000	750
Test3	15	20	25	20	535	6000	965

Table 2. Copmarison of the test results

	Solver	Optimal Value or Upper Bound	No. of main iteration	Time in second
Test 1	CPLEX	3909049	1169	29.52
	Proposed	3907337	4	21.80
Test 2	CPLEX	3925030	2113	76.18
	Proposed	3924174	6	69.3
Test 3	CPLEX	7197526	4119	193.86
	Proposed	7181037	5	120.90

Table 3. Effect of change in transportation costs

Problem ID	No. of main iteration	time in second	Problem ID	No. of main iteration	Time in Second
Test1t08	3	17.4	Test1f05	4	14.7
Test1t12	8	45.7	Test1f15	3	21.9
Test2t08	9	137.2	Test2f05	44	50.3
Test2t12	5	55.2	Test2f15	6	3247.4
Test3t08	4	99.0	Test3f05	3	59.8
Test3t12	6	138.1	Test315	4	57.4

Table 4. Effect of change in capacity limits

Problem ID	No. of main iteration	Time in second
Test 1	4	21.8
Test1cp	7	32.9
Test 2	6	69.3
Test2cp	5	65.0
Test 3	5	120.9
Test3cp	4	68.0

Table 5. Computational results with improved continuous cuts

	Suboptimality tolerance	Optimal Objective value	Time	Time with old cuts
Test1	0.001	3907337	16.0	21.8
Test2	0.002	3924174	140.8	193.6
Test3	0.001	7181037	69.5	120.9
Test2f15	0.003	3972948	1537.5	3247.4

REFERENCES

[1] Benders, J.F., "Partitioning procedures for solving mixed-variables programming problems," Numerische Mathematik 4 (1962), 238-252.

[2] Bilde, O. and Krarup, J., "Sharp lower bound for the simple location problem," Ann. of Discrete Math. 1 (1977), 79-97.

[3] Dantzig, G.B. and Wolfe, P., "Decomposition principle for linear programs," Oper. Res. 8 (1960), 101-111.

[4] Christofides, N. and Beasley, J.E., "Extension to a Lagtangean relaxation approach for the capacitated warehouse location problem," E.J.O.R., 12 (1983), 19-28.

[5] Ellwein, L.B. and Gray, P., "Solving fixed charge location-allocation problems with capacity and configuration constraints, AIIE Transactions, III, 4 (1971), 290-298.

[6] Elson, D.G., "Site location via mixed-integer programming," Operational Research Quarterly, 23 (1972), 31-43.

[7] Erlenkotter, D., "A dual based procedure for uncapacitated facility location," Oper. Res. 26 (1978), 992-1009.

[8] Geoffrion, A.M. and Graves, G.W., "Multicommodity distribution system design by Benders decomposition," Mgt. Sci. 20, 5 (1974), 822-843.

[9] Geoffrion, A.M., Graves, G.W., and Lee, S.J., "A management support system for distribution planning," INFOR 20 (1982), 287-314.

[10] Laporte, G. and Louveaux, F.V., "The integer L-shaped method for stochastic integer programs with complete recourse," Oper. Res. Letters 13 (1993), 133-142.

[11] Larporte, G., Louveaux, F.V., and Hamme, L.V., "Exact solution to a location problem with stochastic demands," Transportation Sci. 28, 2 (1994), 95-103.

[12] Laundy, R.S., "A tree search algorithm for the multi-commodity location problem," E.J.O.R. 20 (1994), 344-351.

[13] Pooley, J., "Integrated production and distribution facility planning at Ault Foods," Interface 24, 4 (1994), 113-121.

[14] Van Roy, T.J., "A cross decomposition algorithm for capacitated facility location," Oper. Res. 34 (1986), 145-163.

[15] Van Slyke, R. and Wets, R.J-B., "*L*-shaped linear programmings with application to optimal control and stochastic optimization," SIAM Journal on Applied Mathematics 17 (1969), 638-663.

[16] Wets, R.J-B., "Stochastic programs with fixed recourse: the equivalent deterministic program," SIAM Review 16 (1974), 309-339.

[17] ______, "Stochastic programming: Solution techniques and approximation schemes," in: A. Bachem, M. Grotschel and B. Korte(eds.), *Mathematical Programming: The State of the Art*, Springer-Verlag, Berlin, 1983, 566-603.

[18] Warszawski, A., "Multi-dimensional location problems," Oper. Res. Quart. 24 (1973), 165-179.