PERGAMON

Computers & Operations Research 28 (2001) 411–427

computers & operations research

www.elsevier.com/locate/dsw

An algorithm for the capacitated, multi-commodity multi-period facility location problem

Cem Canel[a,*], Basheer M. Khumawala[b], Japhett Law[c], Anthony Loh[d]

[a]*Department of Production and Decision Sciences, University of North Carolina at Wilmington, Wilmington, NC 28403-3297, USA*

[b]*Department of Decision and Information Sciences, University of Houston, Houston, TX 77204-6282, USA*

[c]*Department of Decision Sciences and Managerial Economics, Chinese University of Hong-Kong, Shatin, N.T., Hong Kong*

[d]*Department of Industrial Engineering, University of Houston, Houston, TX 77204, USA*

Received 1 April 1998; received in revised form 1 August 1999

Abstract

There are substantial number of exact and heuristic solution methods proposed for solving the facilities location problems. This paper develops an algorithm to solve the capacitated, multi-commodity, multi-period (dynamic), multi-stage facility location problem. The literature on such composite facility location problem is still sparse. The proposed algorithm consists of two parts: in the first part branch and bound is used to generate a list of candidate solutions for each period and then dynamic programming is used to find the optimal sequence of configurations over the multi-period planning horizon. Bounds commonly known in the location literature as delta and omega are used extensively to effectively reduce the total number of transshipment subproblems needed to be solved. The proposed algorithm is particularly effective when the facility reopening and closing costs are relatively significant in the multi-period problem. An example problem is included to illustrate the proposed solution procedure.

Scope and purpose

The literature on the facilities location problem is extensive with a wide variety of solution methods for addressing these problems. In this paper, an algorithm is proposed to solve the capacitated, multi-commodity, multi-period (dynamic), multi-stage facility location problem. The literature on such composite facility location problem is still sparse. We develop the decision rules to be employed in this algorithm and describe their rationale. Through an illustrative example, we offer insights on the solution algorithm and how it can be implemented to solve the composite problem. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Facility location; Branch and bound algorithm

* Corresponding author. Tel.: + 910-962-3816; fax: + 910-962-3815.

E-mail address: canelc@uncwil.edu (C. Canel).

0305-0548/00/$ - see front matter
PII: S0305-0548(99)00126-4

1. Introduction

In many industrial applications concerning capacity planning, different commodities are manufactured and delivered directly from the factories, at discounted bulk delivery rates, or via warehouses or distribution centers (henceforth referred to as facilities) to satisfy the customer demands [1]. A majority of these problems seek to minimize the total distribution costs: both inbound (from factories to facilities) and outbound costs (from facilities to demand points) plus the facility operating costs. Since the factories are assumed to be preexisting, the problem then becomes one of selecting the facility locations from among all pre-selected potential sites such that the total distribution cost is minimized. Problems of this type have been known as multi-stage facility location problems. Once the planning horizon is extended to more than a single period, the location problem is dynamic (in contrast to a static problem which has only one period). The dynamic facility location problem is generally regarded as the essence of capacity expansion, and is very significant for companies facing changing market conditions or expecting relevant costs to change over time. This problem involves decisions on how many facilities to open, where to locate them, and how to allocate the production of each facility, in each period, to a list of known demand centers. These decisions are made for a predetermined planning horizon in which the variables included in the problem can change. The objective is to locate facilities in response to expected changes in demand from customers over a planning horizon of T periods.

The literature on dynamic facility location problem (DFLP) is substantial. Ballou [2] was the first to address the dynamic location problem and suggested a heuristic algorithm for its solution. A number of exact as well as heuristic solution methods have been proposed in earlier papers. Erlenkotter [3], Aikens [4], Krarup et al. [5], Francis et al. [6] and Brandeau and Chiu [7] provided comparative studies and surveys of these methods. Ballou [2], Wesolowsky [8], Wesolowsky and Truscott [9], Warzawski [10], Sweeney and Tatham [11] and Hormozi and Khumawala [12] integrated mixed-integer programming and dynamic programming for solving the DFLP. In these studies, first single-period solutions are generated for each period and then dynamic programming is used to find the optimal sequence for the complete planning horizon. Van Roy and Erlenkotter [13] developed a dual approach for the uncapacitated DFLP. Shulman [14] described an optimization algorithm for solving the dynamic capacitated facility location problem based on the Lagrangean relaxation technique. Kelly and Marucheck [15] proposed a method based on extensions of simplification rules found in Khumawala's [16,17] single-period model, and then applied Bender's decomposition, by partitioning the time horizon into shorter intervals. The use of simplification rules to establish bounds for facility location problems has been successfully applied since Efroymson and Ray [18], and later improved by Khumawala [16]. Khumawala also coined the terms "delta" and "omega" (D&O) for the simplification rules to fix a facility to be "open" or "closed", respectively. Since then, the D&O rules have been successfully applied to various extensions of the facility location problem [16,17,19–22].

There have been several studies on multi-commodity facility location problem. Warszawski [10] proposed a branch and bound algorithm and a heuristic solution procedure for solving the multi-commodity facility location problem. Geoffrion and Graves [1] provided a solution procedure based on Bender's decomposition and applied it to a real situation for a major food company. Neebe and Khumawala [21] used the delta, omega simplification rules adjusted for the multi-commodity case. Barnhart and Sheffi [23] presented a primal–dual, heuristic solution approach for large-scale multi-commodity network flow problems. Crainic [24] described a dual-ascent-based approach for solving simple multi-commodity location problems with balancing requirements. Aggarwal [25] proposed a general heuristic procedure for multi-commodity integer flows which can be utilized for solving multi-commodity facility location problems. Lee [26] developed a general model for a capacitated facility location problem which incorporates the multi-product, multi-type facility and proposed an optimal solution algorithm based on Bender's decomposition technique. Lee [27,28] extended a standard capacitated facility location problem to a generalization of multi-product, multi-type capacitated facility location problem with a choice of facility type and presented an effective algorithm based on cross decomposition. The algorithm unifies Bender's decomposition and Lagrangean relaxation into a single framework. Pirkul [29] developed an efficient heuristic procedure for solving the multi-commodity, multi-plant capacitated facility location problem.

All these papers have attempted to investigate the facility location problem from two perspectives; either as a multi-period or as a multi-commodity facility location problem. In this paper, we propose to solve the more comprehensive facility location problem: capacitated, multi-commodity, multi-period (dynamic), multi-stage facility location problem. It is based on the solution framework of Sweeney and Tatham [11], together with the use of D&O rules to effectively reduce the total number of transshipment subproblems that need to be solved. A branch and bound approach is adopted to generate the list of candidate solutions for each period, as opposed to the specialized Bender's partitioning procedure with canonical cuts in Sweeney and Tatham [11].

The rest of this paper is organized as follows. In Section 2 we present the mathematical formulation of the capacitated, multi-stage, multi-commodity dynamic facility location problem. Section 3 provides the development of the algorithm for solving this problem. The fourth section presents an illustrated application, while the last section presents conclusions and suggestions for further research.

2. Problem formulation

Let I, J, N, M, and T be the set of indices for the factories, facilities, customers, commodities, and time periods respectively. The capacitated, multi-stage, multi-commodity, dynamic facility location problem (CMDLP) can be formulated as follows:

$$\text{Min} \quad \sum_{t\in T}\left[\sum_{m\in M}\left(\sum_{i\in I}\sum_{j\in J} C_{ijmt}X_{ijmt} + \sum_{i\in I}\sum_{n\in N} C_{inmt}X_{inmt} + \sum_{j\in J}\sum_{n\in N} C_{jnmt}X_{jnmt}\right)\right.$$

$$\left. + \sum_{j\in J}\left(F_{jt}Y_{jt} + a_{jt}Y_{jt}(1 - Y_{j,t-1}) + b_{jt}Y_{jt}(1 - Y_{j,t+1})\right)\right]$$

s.t.

$$\sum_{j\in J} X_{ijmt} + \sum_{n\in N} X_{inmt} \leqslant g_{imt} \quad \forall i\in I,\ m\in M,\ t\in T, \tag{1}$$

$$\sum_{m\in M}\left(r_m \sum_{n\in N} X_{jnmt}\right) \leqslant h_{jt}Y_{jt} \quad \forall j\in J,\ t\in T, \tag{2}$$

$$\sum_{i\in I} X_{ijmt} = \sum_{n\in N} X_{jnmt} \quad \forall j\in J,\ m\in M,\ t\in T, \tag{3}$$

$$\sum_{i\in I} X_{inmt} + \sum_{j\in J} X_{jnmt} \geqslant D_{nmt} \quad \forall n\in N,\ m\in M,\ t\in T, \tag{4}$$

$$U_{ijmt} \geqslant X_{ijmt} \geqslant 0 \quad \forall i\in I,\ j\in J,\ m\in M,\ t\in T, \tag{5}$$

$$U_{inmt} \geqslant X_{inmt} \geqslant 0 \quad \forall i\in I,\ j\in J,\ m\in M,\ t\in T, \tag{6}$$

$$U_{jnmt} \geqslant X_{jnmt} \geqslant 0 \quad \forall i\in I,\ j\in J,\ m\in M,\ t\in T, \tag{7}$$

$$Y_{jt} \in \{0, 1\} \quad \forall w\in W,\ t\in T, \tag{8}$$

where X_{ijmt} is the quantity of commodity m shipped from factory i to facility j in period t, X_{inmt} the quantity of commodity m shipped from factory i to customer n in period t, X_{jnmt} the quantity of commodity m shipped from facility j to customer n in period t, Y_{jt} the indicator variable for facility status (0 = closed, 1 = open), C_{ijmt} the unit shipment cost of commodity m from factory i to facility j in period t, C_{inmt} the unit shipment cost of commodity m from factory i to customer n in period t, C_{jnmt} the unit shipment cost of commodity m from facility j to customer n in period t, F_{jt} the fixed cost of operating facility j in time t, a_{jt} the reopening cost of facility j in time t, b_{jt} the closing cost of facility j in time t, g_{imt} the production capacity of factory i for commodity m in time t, h_{jt} the capacity of facility j in time t, D_{nmt} the demand from customer n for commodity m in time t, U_{ijmt} the capacity of transportation routes from factory i to facility j for commodity m in time t, U_{inmt} the capacity of transportation routes from factory i to customer n for commodity m in time t, U_{jnmt} the capacity of transportation routes from facility j to customer n for commodity m in time t, and r_m the commodity m weighing factor.

In the above formulation, the objective function minimizes total transportation costs and facility opening costs for a multi-commodity, multi-stage dynamic facility location model, with the addition of the reopening and closing penalties (here-to-fore referred to as dynamic costs) related to the dynamic nature of the problem. Constraint set (1) stipulates that all shipments from a factory to a facility and a customer must not exceed its capacity, and (2) prohibits deliveries to be made from closed facilities. Constraint set (3) indicates a conservation of flow at each facility, while (4) requires that all customer demands must be met. An upper bound and non-negativity on each shipment is imposed by constraint sets (5)–(7). Constraint set (8) on Y_{jt} restricts every facility to be either open or closed.

3. The algorithm

The proposed algorithm is segmented into three phases. Phase I is the dynamic cycle, where an attempt is made to identify facilities that should be opened or closed in the optimal solution to the CMDLP. In phase II, the optimal static solutions for each time period are obtained, and a list of candidate static facility configurations are generated using a branch and bound procedure. Both phases are based on the application of the D&O rules to effectively minimize the list of candidate configurations needed to be evaluated in phase III, where dynamic programming is used to obtain the optimal solution to CMDLP.

In the following discussion, K_{ot}, K_{ct}, and K_{ft} stand for the set of facilities that are open, closed, and free (not yet fixed) in period t, respectively. $Z(K_t)$ denotes the optimal objective function value for the static problem of CMDLP in period t, with K_{ot} being the set of open facilities in the optimal static solution. These static problems are transshipment problems. The algorithm begins with all facilities being in K_{ft}, while K_{ot} and K_{ct} are empty.

3.1. Phase I — the dynamic cycle

The objective of this phase is to determine the facilities which should be open or closed in the optimal solution to CMDLP. Two simplification rules, the dynamic delta and the dynamic omega rules, conceptually similar to those developed by Akinc and Khumawala [30] for the capacitated static problems are used to achieve this objective.

3.1.1. Dynamic delta (DD) rule

The minimum savings in opening facility j at time t, DD_{jt}, given that j is in K_{ft}, can be expressed as:

$$DD_{jt} = Z(K_{ot} + K_{ft} - j) - Z(K_{ot} + K_{ft}), \tag{9}$$

whereas the upper bound to the dynamic cost in opening j in period t is

$$\begin{aligned}\text{Max}[a_{jt}(1 - Y_{j,t-1}) + b_{jt}(1 - Y_{j,t+1})] - \text{Min}[b_{j,t-1} Y_{j,t-1} + a_{j,t+1} Y_{j,t+1}] \\ = a_{jt} + b_{jt} - \text{Min}[Y_{j,t-1}(a_{jt} + b_{j,t-1}) + Y_{j,t+1}(a_{j,t+1} + b_{jt})],\end{aligned} \tag{10}$$

where the maximization and minimization are defined over $Y_{j,t-1}$ in $K_{f,t-1}$, and $Y_{j,t+1}$ in $K_{f,t+1}$. Therefore, if the minimum savings is not less than the dynamic cost to open facility j in t, i.e.

$$DD_{jt} \geqslant a_{jt} + b_{jt} - \text{Min}[Y_{j,t-1}(a_{jt} + b_{j,t-1}) + Y_{j,t+1}(a_{j,t+1} + b_{jt})] \tag{11}$$

then, facility j must be open in period t in the optimal solution to CMDLP.

3.1.2. Dynamic omega (DO) rule

Similar to the development of the DD rule, the maximum savings in opening facility j in time t, DO_{jt}, given that j is in K_{ft}, can be expressed as

$$DO_{jt} = Z(K_{ot}) - Z(K_{ot} + j), \tag{12}$$

whereas the lower bound to the dynamic cost in opening facility j in period t is equal to

$$\begin{aligned}\text{Min}[a_{jt}(1 - Y_{j,t-1}) + b_{jt}(1 - Y_{j,t+1})] - \text{Max}[b_{j,t-1} Y_{j,t-1} + a_{j,t+1} Y_{j,t+1}] \\ = a_{jt} + b_{jt} - \text{Max}[Y_{j,t-1}(a_{jt} + b_{j,t-1}) + Y_{j,t+1}(a_{j,t+1} + b_{jt})],\end{aligned} \tag{13}$$

where again the maximization and minimization are defined over $Y_{j,t-1}$ in $K_{f,t-1}$, and $Y_{j,t+1}$ in $K_{f,t+1}$. Therefore, if the maximum savings is not greater than the dynamic cost to open facility j in t, i.e.

$$DO_{jt} \leqslant a_{jt} + b_{jt} - \text{Max}[Y_{j,t-1}(a_{jt} + b_{j,t-1}) + Y_{j,t+1}(a_{j,t+1} + b_{jt})], \tag{14}$$

then, facility j must be closed in period t in the optimal solution to CMDLP.

The implementation of each of these two rules requires the solution of $|J| + 1$ transshipment problems. These solutions are saved for future use in the branch and bound procedures in phase II as they may appear in the evaluation of the static simplification rules introduced in the following paragraphs. Going from period 1 to period $|T|$, the DD rule is applied to all facilities in $K_{\mathrm{f}t}$. If no facilities are opened during this process, phase I terminates. Otherwise, the algorithm cycles through the time periods and facilities again, applying the DO rule in an attempt to close some facilities. If successful, the algorithm cycles back to applying the DD rule again. Otherwise, phase I terminates.

If phase I is able to fix all facilities into $K_{\mathrm{o}t}$ or $K_{\mathrm{c}t}$, then the algorithm terminates, and the solution thus obtained is the optimal solution to CMDLP. Otherwise, phase II is initiated.

3.2. Phase II — the static cycle

Phase II begins with the $K_{\mathrm{o}t}$, $K_{\mathrm{c}t}$ and $K_{\mathrm{f}t}$ sets resulting from phase I. The central idea is to generate a list of static candidate facility configurations which may be part of the optimal solution, as developed by Sweeney and Tatham [11]. A branch and bound approach based on Akinc and Khumawala [30] is adopted here, whereby the $Z(K_t)$'s already obtained in phase I may be re-used here to reduce the computational effort. Similar to the first phase, phase II also employs two simplification rules, viz., the static D&O rules. A static lower bound and a static upper bound are also needed in the implementation of the branch and bound procedure. These concepts are developed next, followed by a description of phase II of the algorithm.

The static D&O rules are conceptually similar to the dynamic D&O rules. Since these static rules are intended only to identify candidate static facility configurations of the optimal solution (in contrast to those that must be in the optimal solution, as in phase I), a penalty method similar to that employed by Khumawala [16] is adopted to reduce the computational effort in evaluating the rules. The fundamental idea is to underestimate the minimum saving computed from (11) and overestimate the maximum saving computed from (14).

3.2.1. Static delta (SD) rule

In any time period t, for any facility j in $K_{\mathrm{f}t}$, if

$$Z(K_{\mathrm{o}t} + K_{\mathrm{f}t} - j) - Z(K_{\mathrm{o}t} + K_{\mathrm{f}t}) \geqslant a_{jt} + b_{jt} - \mathrm{Min}[Y_{j,t-1}(a_{jt} + b_{j,t-1}) + Y_{j,t+1}(a_{j,t+1} + b_{jt})], \tag{15}$$

then, facility j must be open in period t in the optimal solution to CMDLP.

Here, $K'_{\mathrm{o}t}$ and $K'_{\mathrm{f}t}$ are sets without any capacity constraints except on facility j, and the minimization is over $Y_{j,t-1}$ and $Y_{j,t+1}$ in $K_{\mathrm{f},t-1}$ and $K_{\mathrm{f},t+1}$ respectively. Note that discarding the capacity constraints of the facilities in $K_{\mathrm{o}t}$ and $K_{\mathrm{f}t}$ sets has the effect of adding more facilities to these sets. The savings from opening an additional facility j, in a larger set is no greater than the savings from adding the same facility to a smaller set, i.e., the left-hand side of (15) is a lower bound on the minimum savings from opening facility j. Evaluation of this lower bound requires the solution of the following continuous knapsack problem (KP):

$$\mathrm{Max} \quad \sum_{i\in I}\sum_{m\in M} \delta_{i\theta m} x_{i\theta m} + \sum_{n\in N}\sum_{m\in M} \delta_{\theta nm} x_{\theta nm} \tag{16}$$

s.t.

$$\sum_{m\in M} r_m\left(\sum_{i\in I} x_{i\theta m}\right) \leqslant h_\theta \tag{17}$$

$$\sum_{i\in I} x_{i\theta m} = \sum_{n\in M} x_{\theta nm} \quad \forall m\in M, \tag{18}$$

$$x_{i\theta m} \geqslant 0 \quad \forall i\in I,\ m\in M, \tag{19}$$

$$x_{\theta nm} \geqslant 0 \quad \forall n\in N,\ m\in M, \tag{20}$$

where

$$\delta_{i\theta m} = \operatorname*{Min}_{j\in(K_{ot}+K_{ft}-\theta)} \{\mathrm{Max}(C_{ijm} - C_{i\theta m};\ 0)\}, \tag{21}$$

$$\delta_{\theta nm} = \operatorname*{Min}_{j\in(K_{ot}+K_{ft}-\theta)} \{\mathrm{Max}(C_{jnm} - C_{\theta nm};\ 0)\}, \tag{22}$$

and Θ is any facility in the K_{ft} set to be tested for opening.

It can be readily seen that the optimal solution to KP can be obtained by selecting the largest pair of $\delta_{i\Theta m}$ and $\delta_{\Theta nm}$ which yields a positive sum, and allocates the maximum allowable flow to both $x_{i\Theta m}$ (from factory i to facility Θ) and $x_{\Theta nm}$ (from facility Θ to customer n). Unless all the remaining pairs of $\delta_{i\Theta m}$ and $\delta_{\Theta nm}$ are non-positive, or the capacity of facility Θ is exhausted, the next largest pair of $\delta_{i\Theta m}$ and $\delta_{\Theta nm}$ is selected, and the same operations are performed. The allocated flows constitute the optimal solution to KP.

3.2.2. *Static omega (SO) rule*

For any facility j in K_{ft}, if:

$$Z(K_{ot}) + L(K_{ot}+\Theta) \leqslant a_{jt} + b_{jt} - \mathrm{Max}[Y_{t,t-1}(a_{jt} + b_{j,t-1}) + Y_{j,t+1}(a_{t,t+1} + b_{jt})], \tag{23}$$

then, facility j must be closed in period t in the optimal solution to CMDLP.

If we denote $P(K_{ot})$ as the primal multi-commodity transshipment problem for period t in CMDLP (with K_{ot} as the set of open facilities), then $L(K_{ot}+\Theta)$ is the Lagrangean relaxation of $P(K_{ot}+\Theta)$, with the entire set of constraints corresponding to $P(K_{ot})$ relaxed. $Z(K_{ot}) - L(K_{ot}+\Theta)$ is then given by

$$\mathrm{Max} \quad \sum_{i\in I}\sum_{m\in M}(-C_{i\theta m} - P_{im})x_{i\theta m} + \sum_{n\in N}\sum_{m\in M}(-C_{\theta nm} + q_{nm})x_{\theta nm} \tag{24}$$

s.t.

$$\sum_{m\in M} r_m\left(\sum_{i\in I} x_{i\theta m}\right) \leqslant h_\theta, \tag{25}$$

$$\sum_{i\in I} x_{i\theta m} = \sum_{n\in N} x_{\theta nm} \quad \forall m\in M, \tag{26}$$

$$u_{i\theta m} \geqslant x_{i\theta m} \geqslant 0 \quad \forall i\in I,\ m\in M, \tag{27}$$

$$u_{\theta nm} \geqslant x_{\theta nm} \geqslant 0 \quad \forall n\in N,\ m\in M, \tag{28}$$

where p_{im} and q_{nm} are the dual variables corresponding to the factory capacity and customer demand constraints in $P(K_{\mathrm{ot}})$, respectively. Hence, the left-hand side of (23) is given by another continuous knapsack problem, which can be solved in a fashion similar to KP.

3.2.3. Static lower bound

Static lower bounds on all non-terminal nodes in the branch and bound procedure can be obtained via a Lagrangean or continuous relaxation (see Geoffrion and McBride [31]). The Lagrangean relaxation tends to provide tighter bounds over nodes closer to the top of the branch and bound tree, while continuous relaxation is computationally more effective around the lower hierarchy nodes. Lagrangean relaxation is obtained by incorporating the factory capacity and customer demand constraints of the static subproblems of CMDLP generated in the branch and bound procedure during phase II into the objective function, together with the corresponding Lagrangean multipliers. Optimal dual multipliers corresponding to the continuous relaxation of the static subproblem can be used, or alternatively a subgradient optimization procedure can be employed.

3.2.4. Static upper bound

This static upper bound was originally developed and effectively used by Sweeney and Tatham [11]. Let v (CMDLP) denote the value of any feasible solution to CMDLP, and $\underline{v}$ $(\mathrm{CMDLP}_{\bar{t}})$ denote the sum of all optimal static solutions for all time period except for period t, then the static upper bound UB_t, is given by

$$\mathrm{UB}_t = v(\mathrm{CMDLP}) - \underline{v}(\mathrm{CMDLP}_{\bar{t}}). \tag{29}$$

It has been shown [30] that any feasible static facility configuration in period t, K_{ot}, which has a total operation cost greater than UB_t, cannot be part of the optimal solution.

3.3. The branch and bound algorithm

Phase II is partitioned into two passes, both are based on the branch and bound approach. The objective of each pass is as follows:

Pass I — Obtain the optimal static solution of each time period;

Pass II — Generate a list of candidate static facility configurations for each period.

The optimal static solution is obtained using the branch and bound algorithm by Akinc and Khumawala [30]. For each time period, and beginning with the $K_{\mathrm{ot}'}$ $K_{\mathrm{ct}'}$ and $K_{\mathrm{f}t}$ sets from phase I, the SD and SO rules are applied iteratively to facilities in $K_{\mathrm{f}t}$ as in phase I in an attempt to fix them open or closed. However, the right-hand side (RHS) of these two rules are set to zero in pass I since there are no applicable dynamic costs in the computation of the static optimal solutions. If $K_{\mathrm{f}t}$ becomes empty during this procedure, then the optimal static solution is obtained for the period under consideration, and the algorithm proceeds to the next period.

When the SD or SO rules fail to fix any facility open or closed while $K_{\mathrm{f}t}$ is non-empty, a facility in the $K_{\mathrm{f}t}$ set is selected to be branched. A branch is created with the chosen facility opened (the open branch), and another branch is created with the chosen facility closed (the closed branch). Then, starting with the SO rule (23) at the open branch, and the SD rule (15) at the closed branch, the two rules are applied iteratively as in phase I again (using zeros for the RHS). Whenever one of the two static rules fails, another facility in the K_{ft} set is selected to be branched until the optimal static solution is reached. Static lower bounds are computed for all non-terminal nodes, and those exceeding the current incumbent static solution are fathomed.

It should be noted that since the static D&O values are estimates of their dynamic counterparts, dynamic D&O from phase I can be used to substitute for their corresponding static D&O values whenever they are available. The solutions to all transshipment problems solved in pass I are also saved for possible use in pass II. The algorithm terminates in pass I if all the optimal static facility configurations are identical in all periods. In such a case, the sequence of optimal static facility configurations constitute the optimal solution.

Pass II begins with computing a static upper bound for each time period, using the optimal static solutions obtained in pass I. Since the original RHS of (15) and (23) are used here, pass II then proceeds with the selection of a free facility to branch on, starting from period 1 again. The same branch and bound procedure as in pass I is performed, except that non-terminal nodes are fathomed if the corresponding static lower bounds are greater than the static upper bound value (instead of the current incumbent solution as in pass I). Pass II terminates when all the static facility configurations with total operation costs not exceeding the static upper bound of their own period are identified. These configurations constitute the list of "alternatives" in each period to be used in the final solution phase.

3.4. Phase III — the solution phase

Phase II results in a list of candidate static facility configurations for each period. The objective of phase III is to obtain the optimal solution from these lists by evaluating the alternate combinations of these configurations. This problem structure strongly suggests the use of dynamic programming, which was employed by Sweeney and Tatham [11]. The periods form the stages while the static configurations form the states within each stage. The application of dynamic programming to phase III is straight forward. An illustration of the algorithm is given in the following section.

4. An illustrated application

The purpose of providing the illustrated example problem in this section is primarily to show the richness and accuracy of data required in the application of the algorithm developed in this study. As shown in this small example problem, the size of the problem will increase the computations required. Companies that are facing facility location decisions will most likely consider more than two commodities, four potential facility sites, five customers and three periods. In this case, it becomes very significant that companies have accurate data for all of the costs, and capacities for their facilities, and demand from all of the customers for each of the commodities.

An example problem consisting of two equally weighed commodities (M), two factories (I), four potential facility sites (J), five customers (N), and three time periods (T) is chosen to illustrate the algorithm, with the data given in Appendix A. Closing penalties are taken to be the same as the set-up costs of the corresponding facilities, while reopening penalties are twice of the closing penalties. As in most cases in the literature, the reopening penalties for the initial time period and the closing penalties for the last time period are taken to be zero.

4.1. Phase I (dynamic cycle)

Phase I begins with $K_{ot} = K_{ct} = \phi$, the empty set, and $K_{ft} = \{1, 2, 3, 4\}$, for all time periods. The DD rule is first applied to determine if any facility should be fixed open permanently. This step requires the solution to $|K_{ft} + 1|$ transshipment problems for each period, which are given in Table 1.

Beginning with period 1, the DD rule as given in (7) is applied to the free facilities. For example, in T_1, $\mathrm{DD}_3 = Z(1, 2, 4) - Z(1, 2, 3, 4) = 200$; while the RHS of (7) is equal to $a_{31} + b_{31} = 110$. The minimization in (7) would yield a value of zero since the Y values are free and that the dynamic costs are non-negative. Since the DD rule is satisfied, J_3 should be opened in T_1. Note that when a facility is opened T_1, its corresponding RHS of (7) in T_2 becomes $b_{j2} - b_{j1}$, since $Y_{j1} = 1$. This applies in the computation of the RHS values for J_1 and J_3 in both T_2 and T_3. The resulting values are given in Table 2. Since the DD rule is satisfied by J_1 and J_3 throughout the three periods, they are removed from K_{ft} to K_{ot}.

Since the DD rule was successful in opening some facilities, the DO rule as given in (8) is now applied to the facilities remaining in K_{ft}, beginning from T_1 again. As in the DD rule, there are also $|K_{ft} + 1|$ transshipment problems to be solved in each period. However, the K_{ft} sets have been reduced by the DD rule, and most of these solutions have already been computed. The only

Table 1
Solutions to transshipment problems

	T_1	T_2	T_3
$Z(1, 2, 3, 4)$	5,075	4,965	5,155
$Z(2, 3, 4)$	5,865	5,895	6,410
$Z(1, 3, 4)$	4,940	4,830	5,075
$Z(1, 2, 4)$	5,275	5,080	5,215
$Z(1, 2, 3)$	5,020	5,030	5,310

Table 2
Dynamic delta values

		J_1	J_2	J_3	J_4
T_1	DD_{jt}	790	− 135	200	− 55
	RHS of (7)	250	160	110	140
T_2	DD_{jt}	930	− 135	115	65
	RHS of (7)	0	480	0	420
T_3	DD_{jt}	1255	− 80	60	155
	RHS of (7)	− 250	320	− 110	280

Table 3
Dynamic omega values

		J_2	J_4
T_1	DO_{jt}	– 120	40
	RHS of (8)	– 320	– 280
T_2	DO_{jt}	– 25	175
	RHS of (8)	– 480	– 420
T_3	DO_{jt}	345	580
	RHS of (8)	– 160	– 140

additional solutions needed are $Z(1,\ 3)$, which are equal to 4900, 5005, and 5655, respectively, for periods 1, 2, and 3.

Table 3 gives the resulting values in evaluating the DO rule. for example, in T_2, $\mathrm{DO}_4 = Z(1,\ 3) - Z(1,\ 3,\ 4) = 175$; while the RHS of (8) is equal to $-b_{41} - a_{43} = -420$. Since the DO rule is not satisfied in all cases, no facility can be closed and phase I terminates.

4.2. Phase II (static cycle)

With J_1 and J_3 opened throughout the periods, while J_2 and J_4 remains free, optimal solutions to the static problems in each period are computed in pass I. Note that DD and DO values computed in phase I can be used in place of their static counter parts whenever available, and that the RHS values are replaced by zeros. Table 2 indicates that J_4 should be opened in T_2 and T_3 since their corresponding DD values are greater than zero; while J_2 should be closed in T_1 and T_2, and J_4 should be closed in T_1 from Table 3. Thus, the static optimal solutions for T_1 and T_2 are determined. In T_3, since J_4 has just been opened, the SO rule is applied to J_2, the only remaining free facility. Again, the corresponding DD value is readily available using values in Table 1: $\mathrm{DD}_2 = Z(1,\ 3,\ 4) - Z(1,\ 2,\ 3,\ 4) = -80 \leqslant 0$, and J_2 is closed in T_3. Note that the D&O rules were effective enough bounds to provide the static optimal solutions without branching, and that over/under estimates of the D&O values were not required since the exact values were available from phase I. Thus, pass I terminates with the static optimal solutions given in Table 4.

At the beginning of pass II, the static upper bound is computed based on a feasible solution to CMDLP and the static optimal solutions. We will use the sequence of optimal static solutions with applicable dynamic costs as a feasible solution to CMDLP, with $v(\mathrm{CMDLP}) = 15{,}085$. Then, the static upper bounds can be computed as

$$\mathrm{UB}_1 = 15{,}085 - 4830 - 5075 = 5180,$$

$$\mathrm{UB}_2 = 15{,}085 - 4900 - 5075 = 5110,$$

$$\mathrm{UB}_3 = 15{,}085 - 4900 - 4830 = 5355.$$

Period 3 will now be used to illustrate the generation of candidate static facility configurations in pass II. Beginning with $K_{\mathrm{o}t} = \{1,\ 3\}$ and $K_{\mathrm{f}t} = \{2,\ 4\}$ from phase I, J_2 is arbitrarily selected to be branched (see Khumawala [16] for judicious branching strategies). Now the SD rule is applied to

Table 4

Commodity 1 Distribution Cost

	N_1	N_2	N_3	N_4	N_5
I_1	12	—	17	24	—
I_2	6	25	15	—	25

	J_1	J_2	J_3	J_4
I_1	5	—	6	12
I_2	7	4	5	—
N_1	5	9	—	7
N_2	7	17	—	10
N_3	6	6	4	9
N_4	6	—	12	15
N_5	10	15	5	4

Commodity 2 Distribution Cost

	N_1	N_2	N_3	N_4	N_5
I_1	10	—	—	– 20	—
I_2	5	21	12	—	22

	J_1	J_2	J_3	J_4
I_1	4		4	—
I_2	5		3	3
N_1	4	—	3	4
N_2	3	4	—	2
N_3	2	5	4	—
N_4	—	5	7	4
N_5	5	3	—	2

Customer Demands for Commodity 1

	T_1	T_2	T_3
N_1	200	150	100
N_2	50	50	50
N_3	50	70	120
N_4	30	60	80
N_5	60	40	20

Customer Demands for Commodity 2

	T_1	T_2	T_3
N_1	100	70	50
N_2	25	25	25
N_3	30	30	50
N_4	15	35	45
N_5	25	25	20

Capacity of Factories

	I_1	I_2
M_1	220	200
M_2	120	120

Facilities

	J_1	J_2	J_3	J_4
Capacity	215	145	105	130
Set-up Cost	250	160	110	140

the "close J_2" branch for the remaining free facilities J_4. Again, its dynamic counter part is computed since it is readily available: $\mathrm{DD}_4 = Z(1, 3) - Z(1, 3, 4) = 580$ (the SD value would have been 305, an under-estimate of DD). The RHS value remains as 280, similar to that computed in phase I. Therefore, J_4 should be opened. This results in a terminal node, with a corresponding value of $Z(1, 3, 4)$ equal to 5075. This node becomes a candidate static configuration since it is not fathomed for exceeding the static lower bound value of 5355.

The branch and bound process then backtracks to the "open J_2" branch, and the SO rule is applied to J_4. The corresponding DO value is equal to $Z(1, 2, 3) - Z(1, 2, 3, 4) = 155$ (the SO value would have been 335, an over-estimate of DO). Since the RHS value is -140, J_4 cannot be closed. Now the static lower bound is evaluated using continuous relaxation for this node, yielding

Table 5
Candidate static configurations

T_1	$Z(1, 3) = 4{,}900$	$Z(1, 3, 4) = 4{,}940$	$Z(1, 2, 3) = 5{,}020$	$Z(1, 2, 3, 4) = 5{,}075$
T_2	$Z(1, 3, 4) = 4{,}830$	$Z(1, 2, 3, 4) = 4{,}965$	$Z(1, 3) = 5{,}005$	$Z(1, 2, 3) = 5{,}030$
T_3	$Z(1, 3, 4) = 5{,}075$	$Z(1, 2, 3, 4) = 5{,}155$	$Z(1, 2, 3) = 5{,}310$	

a value of 5090 (Lagrangean relaxation yields the same value here), implying that the node cannot be fathomed. Thus, the remaining free facility, J_4 is selected to be branched. The resulting nodes are terminal nodes, with their objective function values less than the static upper bound. Thus, both nodes become candidate configurations for phase III.

In T_1 and T_2, the branch and bound process resulted in all four possible terminal nodes remaining as candidate configurations, since the SD and SO rules were not able to fix any of the free facilities, and that no nodes were fathomed by the static upper bounds. The resulting candidate configurations are given in Table 5.

4.3. Phase III (solution phase)

In this phase, the minimum cost combination with one configuration selected from each period is determined using dynamic programming. The periods act as stages while the configurations form the states within each stage. The dynamic programming setup is similar to finding the shortest path through the three stages, where distance in a path is measured by the sum of the $Z(J_{ot})$'s on the path, plus applicable dynamic costs. The optimal solution opens facilities 1, 3, and 4 throughout the three periods, with a total cost of \$14,845.

4.4. Application

The algorithm was used to solve the application problem employed by Sweeney and Tatham [11]. The problem consists of three plants, five possible facility locations, fifteen customer zones, and a five year planning horizon. There are also penalties for reopening and closing each facility site. All deliveries must be originated from the plants and shipments to customers must be made via the open facilities, i.e., no direct shipments from plants to customers are allowed (note that the proposed algorithm allows direct shipments). Hence, the total capacity of open facilities must exceed the total demand of all customers in each period.

The solution methodology used by Sweeney and Tatham is an integration of Bender's decomposition and dynamic programming. For each period, the optimal static solution is computed first using Bender's decomposition. As they reported, each static solution required an average of 15.2 s of CPU time. Then, a list of candidate static facility configurations is obtained for each period, as in the second phase of this algorithm. The entire problem required the computation of sixty static suboptimal solutions. Each of these solutions required an average of 3.9 s.

In phase I of the proposed algorithm, only one facility could be fixed open permanently in periods 4 and 5. Since no direct transshipment from factories is allowed in this application, a feasible network is therefore not attained in any period. In phase II, the process of generating the list of candidate facility configurations required only a fraction of a second for each period (including all input and output processing time). When compared to the computational results reported by Sweeney and Tatham [11], there is no reduction in the number of candidate facility configurations generated. However, the most prominent difference is the total computation time required for solving the entire problem. This improvement can only be attributed to the efficiency of phase II of the proposed algorithm.

5. Conclusions

In this paper, we have proposed an algorithm for the capacitated, multi-commodity, multi-period, multi-stage facility location problem based on the solution framework by Sweeney and Tatham [11]. The solution methodology proposed in this paper was motivated by Bender's decomposition approach developed by Sweeney and Tatham [11], and has benefitted greatly from several techniques. The algorithm can be partitioned into three phases. The first phase identifies the dominant facilities which can be readily assigned open or closed permanently. These dominant facilities are the ones which, if open, decrease (if closed, increase) the total distribution cost significantly. When phase I fails to reach termination, the second phase generates a list of static facility configurations which may become part of the optimal dynamic solution. The size of this list depends critically on the tightness of the static upper bounds. The final phase then computes the optimal dynamic solution from the configuration list generated in phase II. In this phase, any dynamic programming approach can be employed.

This research shows that the proposed algorithm is applicable in a variety of practical applications. Several observations pertaining to the algorithm are made below;

1. The algorithm will be more efficient when dominant facilities exist. This is not unexpected since the dynamic phase (phase I) is designed to detect the existence of such dominant facilities.
2. Lower reopening and closing penalties will favor the dynamic delta cycle, and higher penalties will favor the dynamic omega cycle. If the penalties are low, the optimal dynamic solution tends to switch facilities more often, and the sequence of optimal static solutions is usually a good feasible solution. Conversely, if these penalties are high, switching of facilities tends to occur less often. Also, with higher penalties, it is vital that the direct transshipment routes from the factories to the customers provide a feasible distribution network. Otherwise, the dynamic omega rule will not be applicable.
3. The static phase tends to be more efficient when the fixed warehousing costs are either high or low, relative to the distribution costs. This phase can be a computational burden if the fixed costs do not fit in either category. When the fixed costs are high, the optimal static solution tends to open fewer number of facilities, and vice versa. In many cases, the optimal static solution is attained quite early in the search process, and much of the computation time will be spent on the verification of optimality.
4. Terminal node pruning by the submodularity test tends to increase as phase I requires more iterations, which is also the case when the number of potential sites is large.
5. The Lagrangean lower bounds to be obtained at the earlier stage (higher hierarchy) of the branch and bound tree will usually be much stronger than the continuous relaxation bounds.

On the contrary, these continuous relaxation bounds computed at the lower hierarchy will usually be just as tight as the Lagrangean bounds. Hence, at the lower hierarchy, it may be more beneficial to employ continuous relaxation, and not to incur the additional computation required by computing the Lagrangean bounds.
6. The static upper bounds, UB_t's, will tend to be tighter when dominant facilities exist.
7. In phase II, the static delta and omega rules are designed to avoid solving the transshipment subproblems directly. However, in generating the list of candidate static facility configurations, without strong static upper bounds, many transshipment subproblems which are avoided in the first pass will need to be computed in the second pass. These static rules have their virtue in solving multiple commodity FLPs. But for single commodity FLPs, since most contemporary primal simplex based network algorithms are highly efficient, it may perhaps be more efficient to solve the subproblem directly using reoptimization.

The efficiency of the algorithm will increase with the efficacy of the first phase. One area which should receive immediate attention is the development of good branching strategies. Normally, rapid pruning of non-terminal nodes depends greatly on the selection of the branching facility, e.g., see Khumawala [16,17,19].

Since the multiple commodity network flow problems are not unimodal in general, fractional flows are allowed in the optimal solution. For most distribution system design problems in which supplies and demands are in large quantities, a realistic solution can still be reached by judicious rounding of the fractional flows. Otherwise, this may create a significant drawback; the decision variables to be integral. Unless there is a very efficient multiple commodity network integral flow algorithm available, this remedy makes the FLP even more intractable.

The algorithm developed in this study can be extended to solve international facility location problems where there are additional parameters and constraints to be included into the model. Further research in this area should be promising since in today's global economy there are more companies considering facility locations in other countries.

Appendix A

An example problem consisting of two equally weighed commodities (M), two factories (I), four potential facility sites (J), five customers (N), and three time periods (T) is chosen to illustrate the algorithm, with the data given in Table 4.

References

[1] Geoffrion AM, Graves GW. Multicommodity distribution system design by Bender's decomposition. Management Science 1974;20:822–44.

[2] Ballou RH. Dynamic Warehouse Location Analysis. Journal of Marketing Research 1968;5:271–6.

[3] Erlenkotter D. A comparative study of approaches to dynamic location problems. European Journal of Operational Research 1981;6:133–43.

[4] Aikens CH. Facility planning models for distribution planning. European Journal of Operational Research 1985;22:263–79.

[5] Krarup J, Pruzan PM. The simple plant location problem: survey and synthesis. European Journal of Operational Research 1983;12:36–81.
[6] Francis RL, McGinnis LF, White JA. Locational analysis. European Journal of Operational Research 1983;12:220–52.
[7] Brandeau M, Chiu S. An overview of representative problems in location research. Management Science 1989;35(6):645–74.
[8] Wesolowsky GO. Dynamic facility location. Management Science 1973;19:1241–8.
[9] Wesolowsky GO, Truscott WG. The multi-period location-allocation problem with relocation of facilities. Management Science 1975;22:57–65.
[10] Warzawski A. Multi-dimensional location problems. Operational Research Quarterly 1973;24:165–79.
[11] Sweeney DJ, Tatham RL. An improved long run model for multiple warehouse location. Management Science 1976;22(7):748–58.
[12] Hormozi AM, Khumawala BM. An improved model for multi-period facility location problems. IIE Transactions 1996;28(2):105–15.
[13] Van Roy TJ, Erlenkotter D. A dual-based procedure for dynamic facility location. Management Science 1982;28:1091–5.
[14] Shulman A. An algorithm for solving capacitated plant location problems with discrete expansion sizes. Operations Research 1991;39(3):423–36.
[15] Kelly DL, Marucheck AJ. Planning horizon results for the dynamic warehouse location problem. Journal of Operations Management 1984;4:279–94.
[16] Khumawala BM. An efficient branch and bound algorithm for the warehouse location problem. Management Science 1972;18(12):718–31.
[17] Khumawala BM. An efficient heuristic procedure for the uncapacitated warehouse location problem. Naval Research Logistics Quarterly 1973;20(1):109–21.
[18] Effroymson MA, Ray TL. A branch and bound algorithm for plant location. Operations Research 1966;14: 361–8.
[19] Khumawala BM. An efficient heuristic procedure for the capacitated warehouse location problem. Naval Research Logistics Quarterly 1974;21:609–23.
[20] Nauss RM. An improved algorithm for capacitated facility location problem. Journal of Operational Research Society 1978;29:1195–201.
[21] Neebe AW, Khumawala BM. An improved algorithm for the multi-commodity location problem. Journal of Operational Research 1981;32:143–9.
[22] Roodman GM, Schwarz LB. Optimal and heuristic facility phase-out strategies. AIIE Transactions 1975;7:177–84.
[23] Barnhart C, Sheffi Y. A network-based primal-dual heuristic for the solution of multicommodity network flow problems. Transportation Science 1993;27(2):102–17.
[24] Crainic TG. Dual-ascent procedures for multicommodity location-allocation problems with balancing requirements. Transportation Science 1993;27(2):90–101.
[25] Aggarwal AK, Oblak M, Vemuganti RR. A Heuristic solution procedure for multicommodity integer flows. Computers and Operations Research 1995;22(10):1075–88.
[26] Lee CY. An optimal algorithm for the multiproduct, capacitated facility location problem with a choice of facility type. Computers and Operations Research 1991;18(2):167–82.
[27] Lee CY. The multi-product warehouse location problem: applying a decomposition algorithm. International Journal of Physical Distribution and Logistics Management 1993;23(6):3–13.
[28] Lee CY. A cross decomposition algorithm for a multi-product, multi-type facility location problem. Computers and Operations Research 1993;20(5):527–40.
[29] Pirkul H, Jayaraman V. A multi-commodity. multi-plant, capacitated facility location problem: formulation and efficient heuristic solution. Computers and Operations Research 1998;25(10):869–78.
[30] Akinc U, Khumawala BM. An efficient branch and bound algorithm for capacitated warehouse location problem. Management Science 1977;23:585–94.
[31] Geoffrion AM, McBride R. Lagrangean relaxation applied to capacitated facility location problem. AIIE Transactions 1978;10:40–5.

Cem Canel is an Associate Professor of Operations Management at University of North Carolina at Wilmington. He received his Ph.D. from University of Houston in Operations Management. His research interests include facilities location, management and control of quality. He has published articles in *International Journal of Production Research*, *International Journal of Production Economics*, *International Journal of Operations and Production Management*, *International Journal of Physical Distribution and Logistics Management*, *Industrial Management and Data systems.*

Basheer M. Khumawala is John & Rebecca Moores Professor of Operations Management at University of Houston. He received his Ph.D. from Purdue University in Industrial Administration. His research interests include facilities location, manufacturing and distribution planning, and application of heuristics. He has published articles in *Management Science*, *Naval Research Logistics Quarterly*, *AIIE Transactions*, *Journal of Operations Management*, *Production and Inventory Management*, *International Journal of Production Research*, *International Journal of Production Economics and others.*

Dr. Japhet S. Law obtained his Ph.D. in Mechanical/Industrial Engineering from the University of Texas at Austin and is currently a professor in the Department of Decision Sciences and Managerial Economics, and Dean of the Faculty of Business Administration. He is currently the President of the Operations Research Society of Hong Kong, Regional Vice President of Asia Pacific decision Science Institute, and Associate Editor of the Asia-Pacific Journal of Operational Research. His research interest is in applied operations research in production and management assessment Dr. Law's articles have appeared in *International Journal of Production Economics*, *IEEE Transactions*, *Journal of Production & Inventory Management*, *Journal of Computers and Industrial Engineering*, *International Journal of Systems Science*, *and Journal of Industrial Management.*

Anthony Loh received his Ph.D. from University of Houston in Industrial Engineering, and served as a visiting professor at University of Houston in Industrial Engineering before joining his private business.