ELSEVIER

Operations Research Letters 21 (1997) 159–164

operations research letters

Structured p-facility location problems on the line solvable in polynomial time

Vernon Ning Hsu[a,*], Timothy J. Lowe[b], Arie Tamir[c,d]

[a] *School of Business Administration, George Mason University, Fairfax, VA 22030, USA*
[b] *College of Management, University of Iowa, Iowa City, IA 52242, USA*
[c] *School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel*
[d] *Stern School of Business, New York University, New York, NY 10012, USA*

Received 1 July 1995; revised 1 May 1997

Abstract

In this paper we give an $O(pn^2)$ algorithm for solving the p-facility location problem on the line when the cost of serving any customer is a unimodal function of the location of the serving facilities. One application of our model is a generalization of the economic lot-sizing problem with backlogging allowed. © 1997 Elsevier Science B.V.

Keywords: Facility location; Dynamic programming; Computational complexity

1. Introduction

We consider the p-facility location problem on the line, where we must select no more than p sites for facilities to serve customers located at n demand points. There is a site-dependent fixed cost associated with locating facilities as well as a cost of serving each customer from a given facility site. The objective is to minimize the cost of locating facilities and serving customers. The special structure we consider involves a unimodality for the costs of serving customers. In particular, we assume that for two facilities, say j and k, if facility j lies between the location of a customer and facility k, then the cost of serving the customer from j is no larger than the cost of service from k. One application of our model is a generalization of the economic lot-sizing problem with backlogging allowed. For this specially structured problem we give an $O(pn^2)$ algorithm. We also consider two special cases of the model which lead to $O(n^2)$ solution methods. Although we cannot use the solution procedure of Hassin and Tamir [6] for our model, our solution procedure is similar to theirs in that we make use of the dynamic programming results of Aggarwal et al. [1], Wilber [8], Eppstein [3] and Galil and Park [4] to obtain our complexity bounds.

2. The model

Let $V = \{v_1, \ldots, v_n\}$, $v_1 < v_2 < \cdots < v_n$, be a set of points on the real line, representing the respective locations of n customers. We will refer to the customer located at v_i as the ith customer. The p-facility location problem is to locate at most p service facilities at a number of potential sites in V to serve the n customers

* Corresponding author.

PII S0167-6377(97)00035-7

at the various locations. We assume that the facilities are uncapacitated, and can serve any number of customers. The objective is to minimize the fixed costs of opening facilities plus the variable costs of serving the customers by the facilities. We assume that each customer will be served by one facility. Suppose that the fixed cost of opening a facility at the potential site v_i, $i=1,\dots,n$, is f_i. Since the facilities are uncapacitated and provide the same services, we may assume that at most one facility is established at any site. For each $i=1,\dots,n$, the variable cost of serving the ith customer depends on the location of the facility assigned to serve that customer. For example, we would like to account for the (likely) possibility that a customer cannot be served by facilities located at certain locations. Specifically, for $i=1,\dots,n$; $j=1,\dots,n$, we let c_{ij} denote the variable cost of serving the ith customer by a facility located at v_j. (If the ith customer cannot be served by such a facility $c_{ij}=\infty$.) We assume that the fixed costs of opening facilities and the variable costs of serving customers are all nonnegative.

There are many special cases of the above problem discussed and solved in the literature. For example, the case where the variable costs are monotone functions of the distance between the customer and the location of his server is solved by Hassin and Tamir [6] in $O(n^2)$ time. (If these functions are linear the complexity bound reported in [6] reduces to $O(pn)$.)

On the other hand, the above cost structure is too general to ensure polynomial solvability of the p-facility model. Consider the special case where each customer is associated with a pair of sites, and can only be served by a facility located at one of these two sites. It is easy to see that if the cost of opening a facility is constant and independent of the site, and if the service cost is zero, the problem is equivalent to the well known Node Cover problem on general graphs. The latter is an NP-complete problem [5].

In the particular application that motivated our study the variable costs depended on both the distance between the customer and his server, and the direction of travel. (E.g., traveling downhill might be less expensive than going uphill.) Thus, we assume the following unimodal cost structure which generalizes the cost structure assumed in [6]: For each $i=1,\dots,n$,

$$c_{ij} \leqslant c_{i,j+1} \quad \text{for } j=i,\dots,n-1,$$
$$c_{ij} \leqslant c_{i,j-1} \quad \text{for } j=2,\dots,i. \tag{1}$$

The unimodal structure ensures that a customer will be served by either the closest opened facility to his right or by the closest opened facility to his left, but not necessarily by the closest of the two.

Another application of the above unimodal structure is a variant of the economic lot-sizing (ELS) problem with backlogging allowed where backlogging cost is period-dependent. The classical ELS models utilize in each period (see Aggarwal and Park [2], for example) a backlogging cost function which is a function of only the total backordered quantity in the period. Such a backlogging cost is period-independent in the sense that the backorders in a certain period are treated the same regardless of which period they are backordered for and how long they have been backordered. We note that for such a period-independent backlogging cost structure, there exists an $O(n \log n)$ DP algorithm [2] to solve the ELS problem provided that the production, inventory, and backlogging costs are linear functions. It is not hard to extend this implementation to obtain an $O(pn \log n)$ algorithm to solve the ELS problem with the additional constraint that no more than p setup periods may be planned.

In a variant of the ELS model where our unimodal cost structure is applicable, we assume that the backlogging cost is period-dependent, i.e., it depends on the period from which the demand is backordered and the length of the backorder. For example, we may assume that the per unit cost of backlogging period i demand to period j is $P_{ij}=r+\beta_i(j-i)^{\alpha_i}$; where $\alpha_i>0$, $\beta_i>0$, and where r is the unit production cost (we assume production cost is the same in all periods). The term $\beta_i(j-i)^{\alpha_i}$ reflects a penalty cost for backlogging demand in period i, which is increasing in the length of backlogging. (In particular, if backlogging is not allowed at some period i, we can model that by setting $\beta_i=\infty$.) Letting D_i denote the demand in period i, the total cost of satisfying demand in period i by production in period j is $c_{ij}=P_{ij}D_i$. This cost structure clearly satisfies (1).

We note that the solution procedure suggested by Hassin and Tamir [6] is not always applicable to the problem with unimodal cost structure. This is due to

the fact that in the latter case it is not always true that a customer is served by the closest open service facility. However, we will show that the unimodal model can also be solved efficiently by using an appropriate dynamic programming recursion. Specifically, we will present an $O(pn^2)$ algorithm for the unimodal p-facility location problem. To ensure a finite solution value to the problem we assume that for each customer i, there exists a site v_j such that $c_{ij}+f_j$ is finite.

3. The algorithm

Having assumed that $v_1 < v_2 < \cdots < v_n$, we consider the following sequence of subproblems. For each j, $j=1,\ldots,n$, and q, $q=1,\ldots,p$, let $P(j,q)$ denote the problem of locating q facilities in $\{v_j, v_{j+1},\ldots,v_n\}$ in order to minimize the total cost of opening the q facilities plus the total cost of serving the customers $j, j+1,\ldots,n$ by these facilities. Let $P'(j,q)$ be the restricted version of problem $P(j,q)$, where one of the q facilities has to be established at v_j. Let $V(j,q)$ and $V'(j,q)$ denote the optimal objective value of problems $P(j,q)$ and $P'(j,q)$, respectively. In particular, the solution value to the p-facility location model is given by $V(1,p)$.

To simplify the notation we assume without loss of generality that $c_{ii}=0$, for $i=1,\ldots,n$. We now have the following recursion for $V'(j,q)$.

$$V'(j,1)=f_j+\sum_{i=j}^{n} c_{ij},$$

and for $q\geqslant 2$,

$$V'(j,q)=f_j+\min_{j<k\leqslant n}\left\{\left(\sum_{i=j}^{n}\min\{c_{ij},c_{ik}\}\right)+V'(k,q-1)\right\}.$$

For convenience, for each pair of indices (j,k), $j\leqslant k$, define

$$C(j,k)=\sum_{i=j}^{k}\min\{c_{ij},c_{ik}\}.$$

Then, the above recursive equations can be rewritten as

$$V'(j,1)=f_j+\sum_{i=j}^{n} c_{ij}$$

and for $q\geqslant 2$,

$$V'(j,q)=f_j+\min_{j<k\leqslant n}\{C(j,k)+V'(k,q-1)\}. \quad (2)$$

The optimal solution value to the p-facility location problem is given by

$$V(1,p)=\min\left\{V'(1,p),\ \min_{1<j\leqslant n}\left\{\sum_{i=1}^{j} c_{ij}+V'(j,p)\right\}\right\}.$$

To evaluate the complexity of the algorithm we first note that for each pair (j,k), it takes $O(n)$ time to compute $C(j,k)$. Therefore, the marginal effort needed to compute $V'(j,q)$ for a pair (j,q) is $O(n^2)$. The total computational effort to solve the problem amounts to $O(pn^3)$.

We now show how a lower-order solution procedure is possible by taking advantage of the unimodal structure. The approach is similar to that of [6], making use of results in dynamic programming as reported in [1, 3, 4, 8]. The improvement will follow directly from the above results after we prove that the matrix $\{C(j,k)\}$ satisfies the concavity (supermodularity) property stated in the lemma.

Lemma 1. *Let j,k,l,m satisfy $1\leqslant j\leqslant k\leqslant l\leqslant m\leqslant n$. Then,*

$$C(j,m)-C(j,l)\geqslant C(k,m)-C(k,l). \quad (3)$$

Proof.

$$C(j,m)-C(j,l)=\sum_{i=l+1}^{m}\min\{c_{im},c_{ij}\}+\sum_{i=j}^{l}\{\min\{c_{im},c_{ij}\}-\min\{c_{il},c_{ij}\}\}. \quad (4)$$

$$C(k,m)-C(k,l)=\sum_{i=l+1}^{m}\min\{c_{im},c_{ik}\}+\sum_{i=k}^{l}\{\min\{c_{im},c_{ik}\}-\min\{c_{il},c_{ik}\}\}. \quad (5)$$

For each i, $i=l+1,\ldots,m$, the unimodality property implies that $c_{ij}\geqslant c_{ik}$, since $j\leqslant k\leqslant l\leqslant i$. Therefore, the

first sum in (4) is greater than or equal to the first sum in (5). Again, due to the unimodality property, for each i, $i=j,\ldots,l$, $c_{im} \geqslant c_{il}$. Therefore,

$$\min\{c_{im}, c_{ij}\} - \min\{c_{il}, c_{ij}\} \geqslant 0. \tag{6}$$

Thus, using (6), it follows that the second sum in (4) is greater than or equal to S, where

$$S = \sum_{i=k}^{l} (\min\{c_{im}, c_{ij}\} - \min\{c_{il}, c_{ij}\}).$$

Summarizing, to prove the lemma, it will suffice to prove that S is greater than or equal to the second sum in (5). In fact we will now prove that for each i, $i=k,\ldots,l$,

$$\min\{c_{im}, c_{ij}\} - \min\{c_{il}, c_{ij}\} \geqslant \min\{c_{im}, c_{ik}\} - \min\{c_{il}, c_{ik}\}. \tag{7}$$

Consider an index i, $i=k,\ldots,l$.

Case I: $c_{im} \leqslant c_{ij}$. Combining this condition with the unimodality property we have $c_{il} \leqslant c_{im} \leqslant c_{ij}$. Thus, (7) is equivalent to

$$c_{im} + \min\{c_{il}, c_{ik}\} \geqslant \min\{c_{im}, c_{ik}\} + c_{il}. \tag{8}$$

If $c_{il} \geqslant c_{ik}$, then the unimodality implies $c_{im} \geqslant c_{il} \geqslant c_{ik}$. In this case the left-hand side of (8) is equal to $c_{im} + c_{ik}$, while its right-hand side is equal to $c_{ik} + c_{il}$. Thus, (8) is clearly satisfied. If $c_{il} \leqslant c_{ik}$, then the left-hand side of (8) is equal to $c_{im} + c_{il}$, while its right-hand side is equal to $\min\{c_{im}, c_{ik}\} + c_{il}$. Again, (8) is satisfied.

Case II: $c_{ij} \leqslant c_{im}$. The proof follows from symmetric arguments. Combining the condition with the unimodality property we have $c_{ik} \leqslant c_{ij} \leqslant c_{im}$. Thus (7) is equivalent to

$$c_{ij} + \min\{c_{il}, c_{ik}\} \geqslant c_{ik} + \min\{c_{il}, c_{ij}\}. \tag{9}$$

If $c_{il} \geqslant c_{ik}$, the left-hand side of (9) is equal to $c_{ij} + c_{ik}$, while its right-hand side is equal to $\min\{c_{il}, c_{ij}\} + c_{ik}$. Thus, (9) is satisfied. If $c_{il} \leqslant c_{ik}$, then the unimodality implies $c_{il} \leqslant c_{ik} \leqslant c_{ij}$. In this case the left-hand side of (9) is equal to $c_{ij} + c_{il}$, while its right-hand side is equal to $c_{ik} + c_{il}$. Again, (9) is satisfied. This completes the proof of the lemma. □

We can now apply the results in [1,3,4,8] to the recursion (2) defined above. Suppose that the parameter q in (2) is fixed. For each pair (j,k) let

$$C'(j,k) = C(j,k) + f_j + V'(k, q-1).$$

With this notation (2) can be rewritten as

$$V'(j,q) = \min_{j<k\leqslant n} \{C'(j,k)\}.$$

It is easy to see that the concavity property of the matrix $\{C(j,k)\}$ implies the concavity of the matrix $\{C'(j,k)\}$. We can now apply Theorem 4.3 in [1], the algorithm in [8], (which also applies to our model), or the newer and simpler algorithms in [3, 4]. We conclude that for each fixed value of q, $1 < q \leqslant p$, the total effort to compute $V'(j,q)$ for all $j=1,\ldots,n$, is $O(nT)$, where T is the effort needed to compute the term $C(j,k)$ for a given pair (j,k), (provided that the sequence $V'(j,q-1)$, $j=1,\ldots,n$, has already been computed). We have noted above that $T = O(n)$. Thus, the p-facility location problem with a unimodal cost structure can be solved in $O(pnT) = O(pn^2)$ time.

Remarks

1. It has been mentioned above that the Node Cover problem on a general graph is a special case of the general p-facility location problem. The cost structure of the p-facility location problem associated with the Node Cover problem can be viewed as a very simple bimodal function. The costs associated with the two sites that can serve a customer are zero, while the costs of the other sites are infinity. From this point of view, the unimodal case, can now be regarded as a "maximal" case which is known to be solvable in polynomial time.
2. We note in passing that the above p-facility location problem with a unimodal cost structure can be transformed into a simple p-covering problem, where the 0/1 constraint matrix defining that problem has the row consecutive 1's property. It is easy to check that the general transformation suggested by Tamir (see [7]), for converting p-facility location problems into covering problems, will result in the consecutive 1's property for unimodal structures. The resulting covering problem has $O(n^2)$ constraints. Therefore, the best known algorithms to solve such covering problems yield complexity bounds that are significantly inferior to the $O(pn^2)$ bound reported above.

3. When the bound on the number of facilities is ineffective, e.g., $n \leqslant p$, the recursive equation can be modified to be independent of the parameter q. Accordingly, the running time of the algorithm reduces to $O(n^2)$.

4. Special cases

In this section we briefly discuss two special cases of the unimodal model to which the above algorithm can be implemented more efficiently. The first case is the *one-sided* model. Each customer can be served only from one direction, and there is a penalty for no service. Specifically, the customer set $N = \{1,\dots,n\}$, is partitioned into two groups, say N_1 and N_2. If the ith customer is in N_1 (N_2), then he can be served only by a facility located to his right (left). There is a nonnegative penalty b_i, if the ith customer is not served. Using the above notation we have the following cost structure:

If i is in N_1, then $c_{ij} = b_i$ for $j = 1,\dots,i-1$.
If i is in N_2, then $c_{ij} = b_i$ for $j = i+1,\dots,n$.

We also assume without loss of generality that $c_{ij} \leqslant b_i$ for $j = 1,\dots,n$. We will show that the one-sided model can be solved in $O(n^2)$ time. We have already noted that the running time of the above algorithm is $O(pnT)$, where T is the time needed to compute the coefficient $C(j,k)$ for a pair of indices (j,k). Thus, to obtain the $O(n^2)$ bound for the one-sided model, we will show how to preprocess the data in $O(n^2)$ time, so that any coefficient $C(j,k)$ can be computed in $T = O(1)$.

For each pair of indices (j,k), $j \leqslant k$, define

$$C_1(j,k) = \sum_{i\in N_1, i=j}^{k} c_{ik}, \qquad C_2(j,k) = \sum_{i\in N_2, i=j}^{k} c_{ij}.$$

It is easy to see that for each k, the total effort to compute $C_1(j,k)$ for all $j = 1,\dots,k$, is only $O(n)$. Similarly, for each j, the total effort to compute $C_2(j,k)$ for all $k = j,\dots,n$, is also $O(n)$. Therefore the total time to compute $C_1(j,k)$ and $C_2(j,k)$ for all pairs (j,k) is $O(n^2)$. To see that any coefficient $C(j,k)$ in the recursion can now be obtained in constant time, we note that

$$C(j,k) = C_1(j+1,k) + C_2(j,k-1).$$

To summarize, we have demonstrated that the total time to solve the p-facility location problem when the cost structure is one-sided is $O(n^2)$.

The second case that we consider is the case where a service cost is a monotone function of the distance between the customer and the serving facility. As mentioned in Section 2, this case has already been solved in $O(n^2)$ time by Hassin and Tamir [6], using a different recursion. We will show that our algorithm will yield the same bound. In view of the above, it is sufficient to demonstrate that any $C(j,k)$ coefficient can be computed in constant time, after some $O(n^2)$ preprocessing. For each pair (j,k), $j<k$, let $x_{jk} = (v_j + v_k)/2$. Suppose that v_j and v_k are two consecutive sites, where facilities are located. Then for each i, $i = j,\dots,k$, the ith customer is served at v_j if $v_i \leqslant x_{jk}$ and otherwise it is served at v_k. Therefore, in this case

$$C(j,k) = \sum_{i:v_i \leqslant x_{jk}} c_{ij} + \sum_{i:v_i > x_{jk}} c_{ik}.$$

In the preprocessing phase we compute the following expressions: For each pair (j,k), $j \leqslant k$, let $C_1(j,k) = \sum_{i=j}^{k} c_{ik}$, and let $C_2(j,k) = \sum_{i=j}^{k} c_{ij}$. The total effort to compute $C_1(j,k)$ and $C_2(j,k)$ for all pairs (j,k) is clearly $O(n^2)$. Also, for each pair (j,k), $j<k$, we compute the largest index i, $j \leqslant i \leqslant k$, such that $v_i \leqslant x_{jk}$. Denote this index by $i(j,k)$. The total effort to compute $i(j,t)$ for all pairs (j,t) is clearly $O(n^2)$. With the above notation we have

$$C(j,k) = C_2(j, i(j,k)) + C_1(i(j,k)+1, k).$$

Therefore, the time to compute $C(j,k)$ for any pair (j,k), $j<k$, is constant. To summarize, the total time to solve the p-facility location problem for the case where the cost of serving a customer is a monotone function of its distance to the server, is also $O(n^2)$.

References

[1] A. Aggarwal, M. Klawe, S. Moran, P. Shor, R. Wilber, Geometric applications of a matrix searching algorithm, Algorithmica 2 (1987) 195–208.
[2] A. Aggarwal, J. Park, Improved algorithms for economic lot size problems, Oper. Res. 41 (1993) 549–571.
[3] D. Eppstein, Sequence comparison with mixed convex and concave costs, J. Algorithms 11 (1990) 85–101.
[4] Z. Galil, K. Park, A linear-time algorithm for concave one-dimensional dynamic programming, Inform. Process. Lett. 33 (1989/90) 309–311.

[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979.
[6] R. Hassin, A. Tamir, Improved complexity bounds for location problems on the real line, Oper. Res. Lett. 10 (1991) 395–402.
[7] A. Kolen, Solving covering problems and the uncapacitated plant location problem on trees, European J. Oper. Res. 12 (1983) 266–278.
[8] R. Wilber, The concave least-weight subsequence problem revisited, J. Algorithms 9 (1988) 418–425.