ON THE *P*-COVERAGE PROBLEM ON THE REAL LINE

Stan Van Hoesel
and
Albert Wagelmans

OR 255-91 June 1991

ON THE P-COVERAGE PROBLEM ON THE REAL LINE

Stan Van Hoesel[1]
Albert Wagelmans[2]

June 1991

Abstract: *In this paper we consider the $p-$coverage problem on the real line. We first give a detailed description of an algorithm to solve the coverage problem without the upper bound p on the number of open facilities. Then we analyze how the structure of the optimal solution changes if the setup costs of the facilities are all decreased by the same amount. This result is used to develop a parametric approach to the $p-$coverage problem which runs in $O(pn\log n)$ time, n being the number of clients.*

OR/MS subject classification: Analysis of algorithms, computational complexity: parametric application of dynamic programming; Dynamic programming/optimal control, applications: parametric approach to $p-$coverage problem on the real line; Facilities/equipment planning, location, discrete: $p-$coverage problem on the real line

---

[1] Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

[2] Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands; currently on leave at the Operations Research Center, Massachusetts Institute of Technology, Cambridge MA; financial support of the Netherlands Organization for Scientific Research (NWO) is gratefully acknowledged.

0. Introduction

In Hassin and Tamir (1990) recent results in dynamic programming are used to improve the complexity bounds of several median and coverage location models on the real line. A general model that unifies some of the classical problems in location theory is shown to be solvable in $O(n^2)$ time, where n is the number of clients. One of the special cases of this general model is the $p-$coverage problem, where - as usual - p refers to an upper bound on the number of open facilities. Hassin and Tamir show that if this upper bound is ignored (or redundant), the resulting problem is solvable in $O(n\log n)$ time. Because the $p-$coverage problem can be formulated as a 0/1 linear program with a totally unimodular restriction matrix, an optimal solution to the Lagrangean dual problem that results from relaxing the upper bound constraint yields an optimal solution to the $p-$coverage problem. To find that optimal solution a parametric method due to Megiddo (1979) is used, resulting in an $O(n^2\log^2 n)$ algorithm.

In this paper we present an algorithm that solves the $p-$coverage problem on the real line in $O(pn\log n)$ time. After describing the problem in Section 1, we give a detailed description of an algorithm to solve the coverage problem without the upper bound on the number of open facilities (Section 2). This algorithm takes $O(n\log n)$ time. We consider the actual $p-$coverage problem in Section 3. First we analyze how the structure of the optimal solution changes if the setup costs of the facilities are all decreased by the same amount. Then this result is used to develop a parametric approach to the $p-$coverage problem which runs in $O(pn\log n)$ time. In Section 4 we describe how the algorithm should be modified to obtain an $O(pn)$ algorithm for two special cases of the $p-$coverage problem. Section 5 contains some concluding remarks.

1. Problem description

We consider the $p-$coverage problem in which n distinct points, v_1 to v_n, are located on a line. These points represent both the set of clients and the set of potential facility sites. To facilitate the exposition, we assume that the points are numbered from left to right, i.e., $j<m$ if and only if v_j is located to the left of v_m. Let $d(v_i,v_j)$ denote the distance between the points v_i and v_j. With the client at v_i we associate the radius r_i, which has the interpretation that this client can only be served by facilities at vertices v_j for which $d(v_i,v_j)\le r_i$. We will say that a client is *covered* by a subset S of facilities, if S contains at least one facility that can serve that client.

The cost structure of the problem is as follows. If a facility is opened at point v_j, a setup cost $c_j > 0$ is incurred. If the client at v_i is not covered by the set of open facilities, a penalty of $b_i > 0$ units has to be paid. The objective is to open facilities such that total costs are minimized.

We will not explicitly deal with variants of the problem. For instance one may think of the problem in which the set of potential facility sites does not coincide with the set of points where the clients are located or the problem in which several clients with different radii are located at the same point. In most cases it is easily seen that those problems can be dealt with in a similar fashion as the one described above.

It is easy to see that the $n \times n$ matrix A defined by

$$a_{ij} = \begin{cases} 1 \text{ if } d(v_i, v_j) \leq r_i \\ 0 \text{ otherwise} \end{cases}$$

has the row consecutive 1's property and that the following 0/1 linear programming formulation describes the problem.

$$\min \ \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{n} b_i z_i$$

s.t.

$$\sum_{j=1}^{n} a_{ij} y_j + z_i \geq 1 \quad \text{for all } i = 1, \ldots, n \tag{1}$$

$$\sum_{j=1}^{n} y_j \leq p \tag{2}$$

$$y_j \in \{0,1\} \quad \text{for all } j = 1, \ldots, n \tag{3}$$

$$z_i \in \{0,1\} \quad \text{for all } i = 1, \ldots, n \tag{4}$$

The restriction matrix of the above program is totally unimodular and therefore we can replace (3) and (4) by non-negativity constraints. (Because of restriction (1) and the fact that the objective function coefficients are non-negative, it is not necessary to introduce upper bounds on the variables.) An optimal solution to the resulting linear programming problem can be found by solving the Lagrangean dual with respect to (2):

$\max_{\mu \geq 0}\{\min \sum_{j=1}^{n}(c_j+\mu)y_j+\sum_{i=1}^{n}b_i z_i-\mu p\}$

s.t.

$$\sum_{j=1}^{n}a_{ij}y_j+z_i \geq 1 \quad \text{for all } i=1,\ldots,n$$

$$y_j \geq 0 \quad \text{for all } j=1,\ldots,n$$

$$z_i \geq 0 \quad \text{for all } i=1,\ldots,n$$

It follows that an optimal solution of the above Lagrangean dual provides an optimal solution to the $p-$coverage problem. The approaches followed by Hassin and Tamir (1990) and in this paper are based on this fact. For fixed μ the resulting problem can be viewed as a coverage problem without an upper bound on the number of open facilities. We will refer to such problems as ***relaxed covering problems***. As already pointed out by Hassin and Tamir, the special structure of the matrix A allows these problems to be solved very efficiently. In the next section we will discuss in great detail an algorithm that solves the relaxed coverage problem in $O(n\log n)$ time.

2. Solving the relaxed coverage problem

We will present a dynamic programming algorithm to solve the relaxed problem. In a somewhat disguised form this algorithm already appeared in Hassin and Tamir [1990]. The explicit presentation as a dynamic programming algorithm will enable us to make observations about the specific problem structure that are useful in developing our algorithm to solve the $p-$coverage problem.

The dynamic programming algorithm has n stages. We start with an empty client set and in every stage one client is added to the current set. Then we consider the coverage problem that results if only this set of clients is present (but we allow facilities to be opened in any of the n points). The order in which the clients are added to the set is determined as follows. Let $f(i)$ and $l(i)$ denote the first respectively last column that has a 1 in row i of matrix A. Note that we may assume that $f(i)>0$, because $a_{ii}=1$. First permute the rows such that they appear in order of non-decreasing $l(i)$. This results in at most n blocks of rows all having the same $l(i)$. Subsequently, permute within each block the rows such that they appear in order of non-decreasing $f(i)$. This last step is only carried out for convenience of presentation, but not really necessary. The matrix that results after permuting the rows of A in this way will be denoted by D.

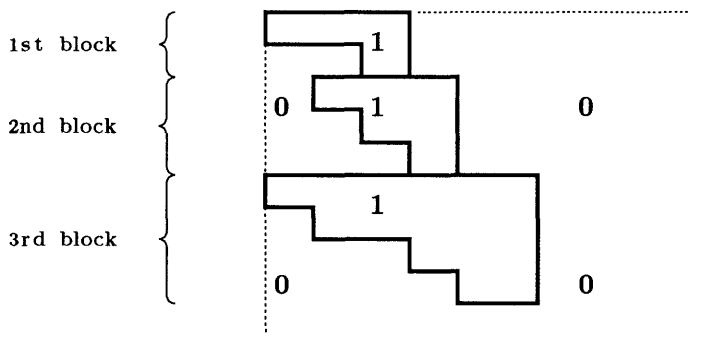

Figure 1: *Structure of matrix D*

It is easily checked that matrix D is in *standard greedy form* (cf. Kolen and Tamir, 1990), which implies that the relaxed coverage problem can be solved using a greedy algorithm. Our dynamic programming algorithm is essentially this greedy algorithm applied to the special case that the restriction matrix has the row consecutive 1's property.

The row order of D defines the order in which we will consider the clients in the dynamic programming algorithm. From now on we let u_i denote the client that corresponds to the $i-$th row of matrix D. We accordingly re-index the cost coefficients b_i, i.e., b_i corresponds to u_i. Furthermore redefine $f(i)$ and $l(i)$ to be the first respectively last column that has a 1 in row i of matrix D (instead of A). We also define $l(0)\equiv 0$, $l(n+1)\equiv n+1$ and for $j\in\{1,\ldots,n\}$ we let i_j be such that $l(i_j-1)<r\leq l(i_j)$, i.e., row i_j of D is the first row with a 1 in a column greater than or equal to j. Note that we have not altered the column order. Therefore we will use our original notation v_j, $j=1,\ldots,n$, to refer to the potential facilities.

We are now able to describe the dynamic programming algorithm in more detail. Let $Z(i,j)$, $1\leq i,j\leq n$, denote the optimal solution value of the coverage problem in which the client set is $\{u_1,\ldots,u_i\}$ and the largest indexed open facility is restricted to be v_j. Furthermore we define $Z(0,0)\equiv 0$. Now consider client u_i and suppose that $Z(i-1,j)$ is known for all $j\in\{0,\ldots,l(i-1)\}$. First suppose that $l(i)=l(i-1)$. It is obvious that if a facility that covers u_i is already open, then this client can be added at no extra cost, i.e.,

$$Z(i,j)=Z(i-1,j) \quad \text{if} \quad f(i)\leq j\leq l(i) \tag{5}$$

Because u_i is not covered by facilities v_j with $j<f(i)$, we will incur the cost b_i. The best thing we can do is to cover the other clients optimally. Hence,

$$Z(i,j) = b_i + Z(i-1,j) \quad \text{if} \quad 0 \leq j < f(i) \tag{6}$$

Using (5) or (6) we are able to compute $Z(i,j)$ from the already known value $Z(i-1,j)$, $j=1,\ldots,l(i)$, in a straightforward way. Now suppose that $l(i-1)<l(i)$, i.e., i is the first row of a block of rows k having all the same $l(k)$ value. In this case we first of all determine $Z(i-1,j)$ for $j=l(i-1)+1,\ldots,l(i)$. If facilities v_j with $j>l(i-1)$ are opened we will incur the cost c_j, but none of these facilities covers any of the first $i-1$ clients. Therefore it is not difficult to see that in an optimal policy one incurs additional costs equal to $\min_{0\leq t\leq l(i-1)}\{Z(i-1,t)\}$. Hence, it follows that

$$Z(i-1,j) = c_j + \min_{0\leq t\leq l(i-1)}\{Z(i-1,t)\} \quad \text{if} \quad j > l(i-1) \tag{7}$$

It is now obvious how the values $Z(i,j)$ can be computed recursively for all $i\in\{1,\ldots,n\}$ and $j\in\{0,\ldots,n\}$. The optimal value of the coverage problem is equal to $\min_{0\leq j\leq n}\{Z(n,j)\}$. To show how an optimal set of open facilities can be determined, we first prove the following.

Lemma 1 Let $r\in\{2,\ldots,n\}$ and consider any subset S of $\{v_r,\ldots,v_n\}$ that contains v_r. Suppose the facilities in S are opened and let $T\subseteq\{v_1,\ldots,v_{r-1}\}$ be a choice of additional open facilities. Then T is an optimal choice if and only if T represents an optimal solution for the coverage problem in which the set of potential facilities is $\{v_1,\ldots,v_{r-1}\}$ and the client set is $\{u_1,\ldots,u_{i_r-1}\}$.

Proof It follows from the structure of matrix D that the clients that are covered by $\{v_r,\ldots,v_n\}$ correspond exactly to the rows with an index greater than or equal to i_r. A subset $S\subseteq\{v_r,\ldots,v_n\}$ with $v_r\in S$, may only cover a subset of these clients. However, we will show that any client with an index larger than i_r which is not covered by S is also not covered by $\{v_1,\ldots,v_{r-1}\}$. Hence, these clients can be ignored when determining an optimal set T, i.e., choosing T optimally is equivalent to making an optimal choice for the coverage problem with potential facilities v_1 to v_{r-1} and clients u_1 to u_{i_r-1}.

Consider a client u_m, $m>i_r$, that is not covered by S. Because of the structure of D, it holds that $l(m)\geq l(i_r)\geq r$. Furthermore, by definition $d_{m,l(m)}=1$. Now suppose that u_m is covered by v_p, $p\leq r-1$, then $d_{mp}=1$. Because D has the row consecutive 1's property it follows that also $d_{mr}=1$. This is a contradiction with the assumption that u_m is not covered by S, because $v_r\in S$. Hence, any

client u_m, $m > i_r$, not covered by S is also not covered by $\{v_1,\ldots,v_{r-1}\}$. This completes the proof. □

Note that an optimal choice of T in Lemma 1 does only depend on the lowest indexed facility in S. We will now use this fact to construct an optimal solution of the coverage problem. Let j_0 be such that $Z(n,j_0) = \min_{0\le j\le n}\{Z(n,j)\}$, then we know that v_{j_0} is the largest indexed open facility in some optimal solution. We can determine the other open facilities in order of decreasing index as follows. Let S denote the current set of facilities that have already been chosen to be opened in the optimal solution. If $r{:=}\min\{j|v_j\in S\}$, then Lemma 1 states that we should add to S the largest indexed open facility in an optimal solution of the coverage problem in which one has to choose facilities from $\{v_1,\ldots,v_{r-1}\}$ to serve $\{u_1,\ldots,u_{i_r-1}\}$. It is not difficult to see that the optimal value to the latter problem is $\min_{0\le k\le l(i_r-1)}\{Z(i_r-1,k)\}$ and that the facility that should be added to S is one for which this minimum is attained. This facility - which need not be unique - is an optimal choice for the first open facility to the left of v_r given that v_r is open. We will refer to it as an *optimal predecessor* of v_r. If it is not optimal to open a facility to the left of a facility, we define its optimal predecessor to be v_0. For all facilities j with $l(i_r-1) < j \le l(i_r)$, an optimal predecessor is found while determining the minimum in (7). By simply storing its index at that time, an optimal solution of the coverage problem can be constructed later on in the way indicated above.

Let us define an optimal predecessor of v_{n+1} to be a facility that has the largest index among the open facilities in some optimal solution. As already indicated in the preceding paragraph, a facility may have more than one optimal predecessor. In particular we have the following result.

Lemma 2 Let $0 \le h < j < k < m \le n+1$ be such that v_h is an optimal predecessor of v_m and v_j is an optimal predecessor of v_k, then v_h and v_j are both optimal predecessors of both v_k and v_m.

Proof We know that

- $i_k \le i_m$,
- $j \le l(i_k-1)$ and $Z(i_k-1,j) = \min_{0\le t\le l(i_k-1)}\{Z(i_k-1,t)\}$, and
- $h \le l(i_m-1)$ and $Z(i_m-1,h) = \min_{0\le t\le l(i_m-1)}\{Z(i_m-1,t)\}$.

Because $j \le l(i_k - 1) \le l(i_m - 1)$ it follows that

$$Z(i_m - 1, h) \le Z(i_m - 1, j) \tag{8}$$

and $h < j \le l(i_k - 1)$ implies

$$Z(i_k - 1, j) \le Z(i_k - 1, h). \tag{9}$$

For all $t \in \{i_k, \ldots, i_m - 1\}$ we have $l(t) \ge j$. Using the consecutive 1's property this implies that client u_t is covered by $\{v_1, \ldots, v_j\}$ if and only if it is covered by v_j. Therefore,

$$Z(i_m - 1, j) = Z(i_k - 1, j) + \sum_{t \in J} b_t \tag{10}$$

where $J \equiv \{t \mid i_k \le t < i_m \text{ and } d_{tj} = 0\}$. Analogously one can prove

$$Z(i_m - 1, h) = Z(i_k - 1, h) + \sum_{t \in H} b_t \tag{11}$$

where $H \equiv \{t \mid i_k \le t < i_m \text{ and } d_{th} = 0\}$. Again from the consecutive 1's property it follows that $d_{tj} = 0$ implies $d_{th} = 0$ for $t \ge i_k$, i.e., $J \subseteq H$. Therefore, using (8), (10) and (11),

$$Z(i_k - 1, j) = Z(i_m - 1, j) - \sum_{t \in J} b_t \ge Z(i_m - 1, h) - \sum_{t \in H} b_t = Z(i_k - 1, h)$$

which combined with (9) yields

$$Z(i_k - 1, h) = Z(i_k - 1, j) = \min_{0 \le t \le l(i_k - 1)} \{Z(i_k - 1, t)\}$$

Hence, v_h is an optimal predecessor of v_k. The fact that v_j is an optimal predecessor of v_m follows from similar arguments. □

Lemma 2 will be used in the next section to develop our algorithm for the p-coverage problem. In the remainder of this section we will present an efficient implementation of the dynamic programming algorithm for the relaxed coverage problem. This implementation is based on the following result.

Lemma 3 Let $i \in \{2, \ldots n\}$ and suppose that $j < k \le l(i - 1)$ and $Z(i - 1, j) \ge Z(i - 1, k)$, then $Z(h, j) \ge Z(h, k)$ for all $h = i, \ldots, n$.

Proof Consider a fixed $h \in \{i,\ldots,n\}$. By the same arguments as in the proof of Lemma 2 one can show

- $Z(h,j) = Z(i-1,j) + \sum_{t\in J} b_t$ where $J \equiv \{t \mid i \le t \le h \text{ and } d_{tj} = 0\}$,
- $Z(h,k) = Z(i-1,k) + \sum_{t\in K} b_t$ where $K \equiv \{t \mid i \le t \le h \text{ and } d_{tk} = 0\}$, and
- $K \subseteq J$.

The statement now follows easily. □

The importance of Lemma 3 is that it implies that if $Z(i-1,j) \ge Z(i-1,k)$ for $j < k \le l(i-1)$, v_j may be ignored as a potential facility from stage i onwards. In that case we will refer to v_j as a *dominated facility.*

We are now able to present the algorithm in full detail. First note that the smallest/largest indexed facility that is able to serve a given client can be found by binary search among the facilities. Hence, it takes $O(n\log n)$ time to determine a compact representation of matrix A. Obtaining matrix D requires $O(n)$ time if a bucket sort procedure is used twice.

At any point in time we let Q be the index set of relevant facilities, i.e, initially $Q = \{0\}$ and at the end of stage $i-1$ it contains all non-dominated $j \in \{0,\ldots,l(i-1)\}$. We store the elements of Q in a balanced tree (cf. Aho, Hopcroft and Ullman, 1974). This enables us to perform the following operations in $O(\log n)$ time: add an element to Q, delete an element from Q and find the smallest element of Q which is greater than a given value. To keep track of the relevant $Z(i,j)$-values we introduce variables Δ_j, $j = 0,\ldots,n$, which are initialized to 0 and which at the end of stage $i-1$ satisfy $Z(i-1,j) = \sum_{t\le j, t\in Q} \Delta_t$ for all $j \in Q$. Note that the fact that Q contains the indices of the non-dominated facilities implies $\Delta_j > 0$ for all $j \in Q$. Moreover, let j_{min} be the smallest element of Q, then $\min_{0\le j\le l(i-1)}\{Z(i-1,j)\} = Z(i-1,j_{min}) = \Delta_{j_{min}}$. Furthermore, we explicitly store the value $Z(i-1,l(i-1))$ in the variable ZL.

We will now show how to update Q, ZL and the Δ_j-variables such that they possess similar properties at the end of stage i. First we check whether $l(i) = l(i-1)$. If this is not the case then we add $l(i-1)+1$ to $l(i)$ to Q, set $\Delta_{l(i-1)+1} := c_{l(i-1)+1} + \Delta_{j_{min}} - ZL$ and $\Delta_j := c_j - c_{j-1}$ for all $j = l(i-1)+2,\ldots,l(i)$. Using (7) it is easy to see that $Z(i-1,j) = \sum_{t\le j, t\in Q} \Delta_t$ for all elements j of the current set Q. Furthermore, we update ZL in this case by setting it equal to

$Z(i,l(i)) = c_{l(i)} + \Delta_{j_{min}}$.

From (5) and (6) we see that $Z(i-1,j) = \sum_{t \le j, t \in Q} \Delta_t$ must be increased by b_i for all $j \in Q$ with $j < f(i)$, and should remain the same for all $j \in Q$ with $f(i) \le j \le l(i)$. If g denotes the smallest element of Q greater than or equal to $f(i)$, then this can be effectuated by setting $\Delta_{j_{min}} := \Delta_{j_{min}} + b_i$ and $\Delta_g := \Delta_g - b_i$. At this point $Z(i,j) = \sum_{t \le j, t \in Q} \Delta_t$ for all $j \in Q \subseteq \{0,\ldots,l(i)\}$. However, Q may contain indices of dominated facilities. Note that $k \in Q$ is dominated if the smallest $j \in Q$ with $j > k$ has $\Delta_j \le 0$. It is easy to see that this is only possible for $j \in O \equiv \{g, l(i-1)+1, \ldots, l(i)\}$. To update Q we consider the elements of O in decreasing order (although, as we will see, some elements may be skipped). Let r be the current element under consideration and let k be the largest element in Q with $k < r$. If $\Delta_r \le 0$, then k is deleted from Q and we set $\Delta_r := \Delta_r + \Delta_k$. We repeat this until $\Delta_r > 0$. Next we consider the largest element of $O \cap Q$ that is smaller than r. After this procedure the indices of all dominated facilities have been removed from Q and stage i of the algorithm has been completed.

To derive the complexity of the algorithm we first note that in every stage the total amount of work can be split into three parts:

(a) a number of operations that depends on the number of elements added to Q at the start of the stage,

(b) a number of operations that depends on the number of elements deleted from Q during the stage, and

(c) a number of operations associated with finding g and the corresponding update of the Δ_j-variables.

Clearly, the number of operations in (c) is $O(\log n)$ per stage and the operations in (a) and (b) can be performed in $O(\log n)$ time per element added to Q, respectively deleted from Q. There are n stages and each of the n indices is added exactly once to Q and deleted at most once. Therefore the total complexity of the algorithm is $O(n \log n)$.

3. Solving the p-coverage problem

As already mentioned in Section 1, the approach to solve the p-coverage problem proposed by Hassin and Tamir (1990) is based on the observation that it suffices to find an optimal solution to the Lagrangean dual problem that results when the restriction on the number of open facilities is dualized. This fact is also used in the approach to be presented here. Consider the

parametric relaxed coverage problem where the cost of opening facility v_j is equal to $c_j+\sum_{i=1}^{n}b_i-\lambda$ for all $j=1,\ldots,n$, and λ ranges from 0 to $\sum_{i=1}^{n}b_i$. It is easy to see that for $\lambda=0$ it is optimal to keep all facilities closed. The optimal value of the parametric problem is a non-increasing piece wise linear concave function of λ. Let λ^* be the largest value in $[0,\sum_{i=1}^{n}b_i]$ for which there exists an optimal solution with at most p open facilities, then this solution is optimal for the $p-$coverage problem. The latter follows from the fact that $\sum_{i=1}^{n}b_i-\lambda^*$ is the value of the optimal Lagrange multiplier. Hence, solving the $p-$coverage problem boils down to finding λ^*. This can be done in several ways. Hassin and Tamir indicate that an approach due to Megiddo (1979) can be used. This approach has a computational complexity equal to the square of the running time of the algorithm to solve the relaxed coverage problem, i.e., it takes $O(n^2\log^2 n)$ time. However, there exists other methods with lower complexities. For instance, it is easily seen that the optimal value function has at most n breakpoints on $[0,\sum_{i=1}^{n}b_i]$. Using a well-known method often attributed to Eisner and Severance (1976), this entire function can be determined in $O(n^2\log n)$ time. Then λ^* can be found as the value for which the absolute value of the slope of this function changes from a value less than or equal to p to a value greater than p. We also note that Hassin and Tamir provide a general method to solve location problems on the real line. This method - which is not based on the Lagrangean relaxation - solves the $p-$coverage problem in $O(n^2)$ time.

We propose a parametric approach to the $p-$coverage problem that differs from the parametric approaches mentioned above in the fact that it explicitly exploits the problem structure. This will enable us to determine the optimal value function of the parametric problem for increasing λ in an on-line fashion. Given the optimal solution for $\lambda=0$ in which all facilities are closed, we determine largest value of λ, say λ_1, for which this solution is optimal. Because λ_1 is a breakpoint of the optimal value function, there exists for that value an alternative optimal solution with at least one open facility. Actually, it turns out that we may assume that the alternative solution has exactly one open facility and we will find such a solution as a byproduct of determining λ_1. Subsequently we determine the largest λ, say λ_2, for which the just found solution is optimal. Again it turns out that there must exist an alternative optimal solution for λ_2 with exactly two open facilities. We continue in this way until we find a solution that has p open facilities. It is easy to see that this must be the optimal solution of the p-coverage problem. So actually we are solving a complete family of coverage

problems in which the bound on the number of open facilities ranges from 0 to p. The value λ^*, although of secondary importance, can be determined as the largest value of λ for which this solution is optimal. Of course, as soon as $\lambda_t \geq \sum_{i=1}^{n} b_i$ for some $t \leq p$ we conclude that there are not more than p open facilities in an optimal solution of the relaxed problem and we terminate the algorithm.

By now it will be clear that the most important part of our algorithm is a procedure that calculates for a given optimal solution of an relaxed coverage problem with cost coefficients b_i and $\bar{c}_j$, $i,j=1,\ldots,n$, the maximal amount that can be subtracted from all $\bar{c}_j$'s simultaneously such that the solution remains optimal. This resembles the problem in which one wants to determine the maximal amount by which the setup costs in the well-known Wagner-Whitin economic lot-sizing model can be reduced such that a given production plan remains optimal. In Van Hoesel and Wagelmans (1991) it is shown how that problem can be solved in linear time. It turns out that a similar approach can be used for the current problem, yielding an $O(n\log n)$ algorithm. The latter implies an $O(pn\log n)$ algorithm for the $p-$coverage problem.

Our approach is as follows. Let $v_{k(1)} < \ldots < v_{k(q)}$ be the open facilities in an optimal solution of the relaxed coverage problem with cost coefficients b_i and $\bar{c}_j$, $i,j=1,\ldots,n$. Furthermore, define $k(q+1) \equiv n+1$. For $r \in \{1,\ldots,q+1\}$ we consider the coverage problem in which the set of open facilities with an index greater than or equal to $v_{k(r)}$ is restricted to be exactly $\{v_{k(r)},\ldots,v_{k(q)}\}$. Define λ_r as the smallest non-negative value of λ with the property that if the setup costs are decreased to $\bar{c}_j - \lambda$ for all $j=1,\ldots,n$, the restricted coverage problem above has an optimal solution with at least $q+1$ open facilities (this is equivalent to having at least r open facilities to the left of $v_{k(r)}$). Note that for $r=q+1$ there is no restriction on the choice of open facilities. This means that λ_{q+1} is the smallest non-negative value such that when all setup costs are decreased by it, there exists an optimal solution of the coverage problem with at least $q+1$ open facilities. Hence, λ_{q+1} is the value we want to determine. When r increases the corresponding coverage problems become less restricted and this implies that the λ_r's are non-increasing in r. Our algorithm uses this fact to determine the λ_r's in order of increasing index.

For convenience, we define $k(0) \equiv 0$ and $\lambda_0 \equiv \infty$. The following theorem is basically a characterization of how the structure of the optimal solution changes when the setup costs are decreased sufficiently.

Theorem 1 Let $r \in \{1,\ldots,q+1\}$ and suppose $\lambda_r < \lambda_{r-1}$, then there exists an optimal solution for the restricted coverage problem corresponding to λ_r with the following properties:

- there are exactly r open facilities $v_{\gamma(1)} < \ldots < v_{\gamma(r)}$ with an index less than $k(r)$, and
- there exists an m, $0 \leq m < r$, such that

$$\gamma(t) = k(t) \qquad \text{for all } t = 1,\ldots,m, \text{ and}$$
$$k(t-1) < \gamma(t) < k(t) \text{ for all } t = m+1,\ldots,r.$$

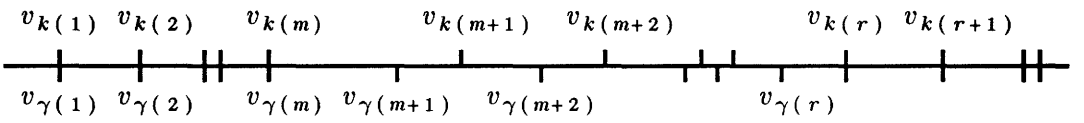

Figure 2: Structure of optimal solution in Theorem 1

Proof We may assume $q < n$, because otherwise $\lambda_r = \infty$ for all $r = 1,\ldots,q+1$. Let $r \in \{1,\ldots,n+1\}$. Because of Lemma 1 the coverage problem corresponding to λ_r boils down to choosing facilities from $\{v_1, v_2, \ldots, v_{k(r)-1}\}$ to serve the client set $\{u_1,\ldots,u_{i_r-1}\}$. From now on we will therefore focus on the latter problem. Note that $\{v_{k(1)},\ldots,v_{k(r-1)}\}$ is an optimal solution to this problem for $\lambda = \lambda_r$. Consider any solution with at least r open facilities that is optimal for $\lambda = \lambda_r$ and denote the indices of its open facilities by $h(1) < h(2) < \ldots < h(s)$, where $s \geq r$. Let $v_{k(m_1)}$ be the largest indexed facility in the intersection of $\{v_{k(1)},\ldots,v_{k(r-1)}\}$ and $\{v_{h(1)},\ldots,v_{h(s)}\}$ (if this intersection is empty, take $m_1 = 0$). Furthermore, let m_2 be such that $k(m_1) = h(m_2)$, then it follows from Lemma 1 that we can also take $\{v_{k(1)},\ldots,v_{k(m_1-1)}\} \cup \{v_{h(m_2)},\ldots,v_{h(s)}\}$ as an optimal solution. Because $\lambda_r < \lambda_{m_1}$ all optimal solution in which $v_{h(m_2)} = v_{k(m_1)}$ is open have at most $m_1 - 1$ open facilities to the left of $v_{k(m_1)}$. Therefore it holds that $|\{v_{h(1)},\ldots,v_{h(m_2-1)}\}| \leq m_1 - 1 = |\{v_{k(1)},\ldots,v_{k(m_1-1)}\}|$, which implies that also the just constructed optimal solution has at least r open facilities.

If $v_{h(m_2+1)}$ to $v_{h(s)}$ are such that $k(m_1+t-1) < h(m_2+t) < k(m_1+t)$ for all $t = 1,\ldots,s-m_2$, then we have obtained a solution of the desired form:

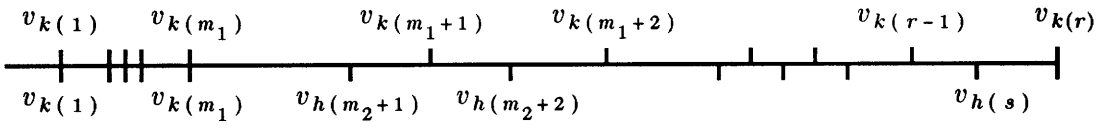

Figure 3: *Solution of the desired form*

Otherwise, there exists at least one $t \in \{m_1+1,\ldots,r\}$ such that the (possibly empty) set $\{v_{k(t-1)+1},\ldots,v_{k(t)-1}\}$ does not contain exactly one element of $\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$. Consider the largest t with that property, say t_1.

Suppose first that $\{v_{k(t_1-1)+1},\ldots,v_{k(t_1)-1}\}$ contains more than one element of $\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$ and let $v_{h(t_2)}$ and $v_{h(t_2+1)}$ be the two largest indexed of those:

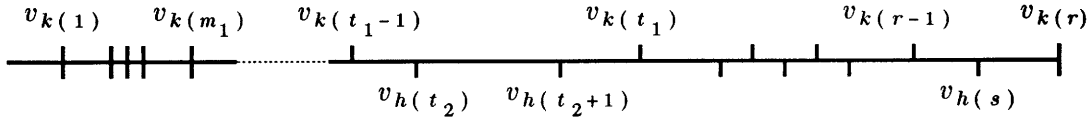

Figure 4: $\{v_{k(t_1-1)+1},\ldots,v_{k(t_1)-1}\}$ *contains more than one element of* $\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$

Because $v_{k(t_1-1)}$ is an optimal predecessor of $v_{k(t_1)}$ and $v_{h(t_2)}$ is an optimal predecessor of $v_{h(t_2+1)}$, it follows from Lemma 2 that $v_{k(t_1-1)}$ is an optimal predecessor of $v_{h(t_2+1)}$. Hence, $\{v_{k(1)},\ldots,v_{k(t_1-1)}\} \cup \{v_{h(t_2+1)},\ldots,v_{h(s)}\}$ is also an optimal solution. Moreover, this solution has the form stated in the theorem:

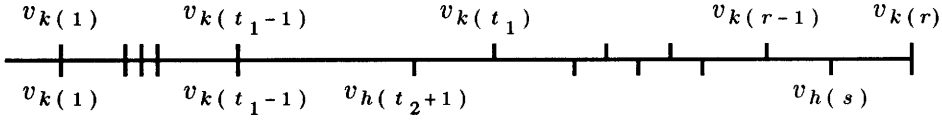

Figure 5: *Solution of the desired form*

For the case that $\{v_{k(t_1-1)+1},\ldots,v_{k(t_1)-1}\}$ does not contain any element of $\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$ we will deduce a contradiction. From the fact that $|\{v_{k(1)},\ldots,v_{k(m_1)}\} \cup \{v_{h(m_2+1)},\ldots,v_{h(s)}\}| \geq r$ it follows immediately that $|\{v_{h(m_2+1)},\ldots,v_{h(s)}\}| \geq r-m_1$, and therefore there must be at least one $t \in \{m_1+1,\ldots,t_1-1\}$ such that $\{v_{k(t-1)+1},\ldots,v_{k(t)-1}\}$ contains more than one

element of the set $\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$. Let t_3 be the largest index with this property and let $v_{h(t_4)}$ and $v_{h(t_4+1)}$ be the two largest indexed elements in $\{v_{k(t_3-1)+1},\ldots,v_{k(t_3)-1}\}\cap\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$:

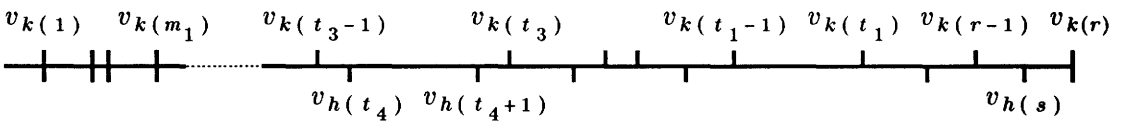

Figure 6: $\{v_{k(t_1-1)+1},\ldots,v_{k(t_1)-1}\}$ *does not contain any element of* $\{v_{h(m_2+1)},\ldots,v_{h(s)}\}$

Note that $|\{v_{k(t-1)+1},\ldots,v_{k(t)-1}\}\cap\{v_{h(t_4+2)},\ldots,v_{h(s)}\}|\leq 1$ for all $t\in\{t_3+1,\ldots,r\}$, and strict inequality holds for $t=t_1$. Hence the set $\{v_{h(t_4+1)},\ldots,v_{h(s)}\}$ has at most $r-t_3$ elements, which implies that $|\{v_{k(1)},\ldots,v_{k(m_1-1)}\}\cup\{v_{h(m_2)},\ldots,v_{h(t_4)}\}|\geq t_3$. Using Lemma 2 we deduce that $v_{h(t_4)}$ is an optimal predecessor of $v_{k(t_3)}$. Therefore $\{v_{k(1)},\ldots,v_{k(m_1-1)}\}\cup\{v_{h(m_2)},\ldots,v_{h(t_4)}\}$ is an optimal choice of open facilities to the left of $v_{k(t_3)}$ in any solution in which $v_{k(t_3)}$ is open. However this leads to a contradiction, because by definition of λ_{t_3} there does not exist such an optimal solution with at least t_3 open facilities for $\lambda=\lambda_r<\lambda_{t_3}$. This completes the proof. □

Our algorithm to determine λ_{q+1} consists of $q+1$ stages, where in the $r-$th stage λ_r is calculated. To this end we will determine for every $j\in\{k(r-1)+1,\ldots,k(r)\}$ the value $W(j)$, which is defined as follows:

$W(j)$ = the optimal value of the problem in which clients u_1 to $u_{i_{k(r)}-1}$ have to be served at minimum cost by r facilities from $\{v_1,\ldots,v_{k(r)}\}$, under the condition that exactly one facility is chosen from the set $\{v_{k(t-1)+1},\ldots,v_{k(t)}\}$ for every $t=1,\ldots,r-1$, and v_j is the facility chosen from $\{v_{k(r-1)+1},\ldots,v_{k(r)}\}$

Note that $W(k(r))=\bar{c}_{k(r)}+Z\left(i_{k(r)}-1,k(r-1)\right)$. The reason why these values are important is the following. Assume that $\lambda_r<\lambda_{r-1}$ and consider the problem in which clients u_1 to $u_{i_{k(r)}-1}$ have to be served by facilities from $\{v_1,\ldots,v_{k(r)-1}\}$. Theorem 1 states that when all setup costs are reduced by λ_r, then there exists an optimal solution in which for every $t=1,\ldots,r-1$ the set $\{v_{k(t-1)+1},\ldots,v_{k(t)}\}$ contains exactly one open facility, and furthermore

exactly one facility from $\{v_{k(r-1)+1},\ldots,v_{k(r)-1}\}$ is opened. It is not difficult to see that this optimal solution has value $\min_{k(r-1)<j<k(r)}\{W(j)\}-r\lambda_r$. Because λ_r is the smallest non-negative value of λ for which this solution is optimal and $Z\left(i_{k(r)}-1,k(r-1)\right)-(r-1)\lambda$ is the value of the optimal solution for all $\lambda\in[0,\lambda_r]$, it holds that

$$\min_{k(r-1)<j<k(r)}\{W(j)\}-r\lambda_r=Z\left(i_{k(r)}-1,k(r-1)\right)-(r-1)\lambda_r$$

or equivalently

$$\lambda_r=\min_{k(r-1)<j<k(r)}\{W(j)\}-Z\left(i_{k(r)}-1,k(r-1)\right). \tag{12}$$

Equality (12) only holds if $\lambda_r<\lambda_{r-1}$. Because $\lambda_r\leq\lambda_{r-1}$, it follows that λ_r can be calculated as the minimum of the already known value λ_{r-1} and $\min_{k(r-1)<j<k(r)}\{W(j)\}-Z\left(i_{k(r)}-1,k(r-1)\right)$.

Let us define for a fixed $r\in\{1,\ldots,q-1\}$ the *efficient facilities* as those v_j in $\{v_{k(r-1)+1},\ldots,v_{k(r)}\}$ for which $W(j)<W(t)$ for all $t=j+1,\ldots,k(r)$. We will now discuss how the $W(j)$-values can be determined efficiently. Consider the first stage. For every $j\in\{1,\ldots,k(1)-1\}$ the value $W(j)$ is equal to $\bar{c}_j$ plus the sum of the b_i's of those clients $u_i\in\{u_1,\ldots,u_{i_{k(1)}-1}\}$ that can not be served by v_j. These values are implicitly calculated and stored using Δ_j-variables as in the algorithm described in Section 2. This is simply done by considering the clients u_1 to $u_{i_{k(1)}-1}$ in any order. If client u_i is considered, then b_i is added to Δ_1 and $\Delta_{l(i)+1}$ and the same quantity is subtracted from $\Delta_{f(i)}$. Actually, we are only interested in the $W(j)$-values of facilities that are efficient and these can subsequently easily be determined. If $v_{j_{min}}$ is the smallest indexed efficient facility, then $\min_{0<j<k(1)}\{W(j)\}=\Delta_{j_{min}}$; hence, $\lambda_1=\Delta_{j_{min}}$.

At the beginning of stage r, $1<r\leq q+1$, we have already calculated λ_{r-1} and $W(h)$ for every efficient $h\in\{k(r-2)+1,\ldots,k(r-1)\}$. Note that these values are defined with respect to the client set $\{u_1,\ldots,u_{i_{k(r-1)}-1}\}$. Calculating the $W(j)$-values for $j\in\{k(r-1)+1,\ldots,k(r)\}$ is done in two steps. In the first step we determine for all $j\in\{k(r-1)+1,\ldots,k(r)\}$ an optimal predecessor in $\{k(r-2)+1,\ldots,k(r-1)\}$ as follows. Consider for a fixed $j\in\{k(r-1)+1,\ldots,k(r)\}$ and all $h\in\{k(r-2)+1,\ldots,k(r-1)\}$ the quantities $Y(h,j)$, defined as follows:

$Y(h,j) = W(h)$ plus the sum of b_i's of those clients in $\{u_{i_{k(r-1)}},\ldots,u_{i_{j-1}}\}$ that can not be served by v_h

It is easily seen that a facility v_h for which $Y(h,j)$ is minimal is an optimal predecessor of v_j. To determine this minimum it suffices to consider only those facilities v_h that are efficient at the start of stage r. The latter follows from arguments similar to those in the proof of Lemma 3 and the fact that every client $u_i \in \{u_{i_{k(r-1)}},\ldots,u_{i_{j-1}}\}$ has $l(i) \geq k(r-1) \geq h$ for all $h \in \{k(r-2)+1,\ldots,k(r-1)\}$. Moreover, suppose $j \in \{k(r-1)+1,\ldots,k(r)\}$ and $h \in \{k(r-2)+1,\ldots,k(r-1)\}$ are such that $Y(h,j) \geq Y(t,j)$ for some t, $h < t \leq k(r-1)$. Using again the same arguments, it follows that for any $m \in \{j+1,\ldots,k(r)\}$ it is not necessary to consider $Y(h,m)$ in determining $\min_{t:k(r-2)<t\leq k(r-1)}\{Y(t,m)\}$. In that case we will refer to v_h as being a *non-efficient predecessor*. To calculate $\min_{t:k(r-2)<t\leq k(r-1)}\{Y(t,j)\}$ for all $j \in \{k(r-1)+1,\ldots,k(r)\}$ we proceed as follows. Add the clients $u_{i_{k(r-1)}}$ to $u_{i_{k(r)^{-1}}}$ in order of increasing index to the current client set. For each client this boils down to adjusting at most two Δ_h-values corresponding to facilities v_h in $\{v_{k(r-2)+1},\ldots,v_{k(r-1)}\}$ which are currently efficient predecessors. After a client has been added, this set of efficient predecessors is updated (if necessary). Suppose the client just added is u_i and $j \in \{k(r-1)+1,\ldots,k(r)\}$ is such that $i = i_j - 1$, then $\min_{t:k(r-2)<t\leq k(r-1)}\{Y(t,j)\}$ equals the Δ_h-value of the currently smallest indexed efficient predecessor. This minimum value is stored at this point in order to be used in the next step.

In the second step of calculating $W(j)$ for all $j \in \{k(r-1)+1,\ldots,k(r)\}$, we take into consideration the clients that were ignored in the first step. For given j those clients are u_{i_j} to $u_{i_{k(r)-1}}$. Note that $l(i) \geq j$ for all $i \in \{i_j,\ldots,i_{k(r)-1}\}$. Therefore, it is easy to verify that $W(j)$ is equal to $\bar{c}_j + \min_{t:k(r-2)<t\leq k(r-1)}\{Y(t,j)\}$ plus the b_i's of those clients $u_i \in \{u_{i_{k(r-1)}},\ldots,u_{i_{k(r)-1}}\}$ for which $f(i) > j$. Hence, for given $i \in \{i_{k(r-1)},\ldots,i_{k(r)-1}\}$ we should include b_i in $W(j)$ for all j with $i_{k(r-1)}+1 \leq j < f(i)$. This justifies the following procedure. Initially, we take the Δ_j-values such that $\sum_{t=k(r-1)+1}^{j}\Delta_t = \bar{c}_j + \min_{t:k(r-2)<t\leq k(r-1)}\{Y(t,j)\}$. Then we consider every $i \in \{i_{k(r-1)},\ldots,i_{k(r)-1}\}$ with $f(i) > i_{k(r-1)}+1$ and set $\Delta_{k(r-1)+1} := \Delta_{k(r-1)+1} + b_i$ and $\Delta_{f(i)} := \Delta_{f(i)} - b_i$. At the end of this procedure the Δ_j-values represent the values to be calculated and the value λ_r is easily obtained. The r-th stage ends with determining the efficient facilities in $\{v_{k(r-1)+1},\ldots,v_{k(r)}\}$.

After the $q+1-$st stage we have computed the desired value λ_{q+1}. If we have stored the smallest r for which $\lambda_r = \lambda_{q+1}$ and the optimal predecessor of every facility, it is easy to construct a solution with $q+1$ open facilities that is optimal for $\lambda = \lambda_{q+1}$.

The analysis of the complexity of the above algorithm is similar to the complexity analysis in Section 2. Most of the work done in stage r is linearly bounded by the cardinalities of the sets $\{k(r-2)+1,\ldots,k(r-1)\}$, $\{k(r-1)+1,\ldots,k(r)\}$ and $\{i_{k(r-1)},\ldots,i_{k(r)-1}\}$. Summing up over all stages yields an $O(n)$ bound on the number of operations involved. The only exception on this bound is the amount of work needed in the first step of the stages to determine which Δ_h-values should be adjusted when a client is added to the client set. As in the algorithm in Section 2, this takes $O(\log n)$ time per client. Hence, the algorithm runs in $O(n\log n)$ time. The $p-$coverage problem can be solved by running the algorithm p times, i.e., in $O(pn\log n)$ time. In the next section we consider two special cases that allow a lower running time.

There are several facts worth mentioning that follow immediately from the algorithm in presented in this section. In the first place we have found that the $p-$coverage problem on the line has always an optimal solution with exactly p open facilities, unless the optimal solution to the relaxed problem is feasible. Actually, this is a consequence of the fact that if the relaxed coverage problem has alternative optimal solutions with q_1 and q_2 open facilities, where $0 \leq q_1 < q_2 \leq n$, then there exists an optimal solution with q_3 open facilities for every q_3 with $q_1 < q_3 < q_2$. The following property is also related to this fact. Consider the coverage problem in which all setup costs are equal to 0 and let $B(q)$ denote the optimal value of the problem in which the number of open facilities is exactly q, $q \in \{0,\ldots,n\}$. Hence $B(q)$ gives the minimal total penalty costs as a function of q, the number of open facilities. The following property holds.

Theorem 2 $B(q)$ is a convex function of q.

Proof Consider the parametric problem coverage problem in which all setup costs are equal to $\sum_{i=1}^{n} b_i - \lambda$, where λ ranges from 0 to $\sum_{i=1}^{n} b_i$. Clearly, for $\lambda = 0$ it is optimal to keep all facilities closed and for $\lambda = \sum_{i=1}^{n} b_i$ an optimal solution is to open all facilities. Moreover, we have seen that there exist values $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq \sum_{i=1}^{n} b_i$, such that there exists an optimal solution with q, $0 < q < n$, open facilities if and only if $\lambda \in [\lambda_q, \lambda_{q+1}]$. For a fixed $q \in \{1,\ldots,n\}$

it holds that

$$B(q)+q\left(\sum_{i=1}^{n}b_i-\lambda_q\right)=B(q-1)+(q-1)\left(\sum_{i=1}^{n}b_i-\lambda_q\right)$$

or equivalently

$$\lambda_q=B(q)-B(q-1)+\sum_{i=1}^{n}b_i \tag{13}$$

Analogously, it holds that

$$\lambda_{q+1}=B(q+1)-B(q)+\sum_{i=1}^{n}b_i \tag{14}$$

Combining (13), (14) and the fact that $\lambda_q \leq \lambda_{q+1}$, yields

$$B(q)-B(q-1)\leq B(q+1)-B(q)$$

Because this inequality holds for every $q\in\{1,\ldots,n\}$, this proves the statement. □

4. Two special cases

Hassin and Tamir (1990) show how two special cases of the relaxed coverage problem can be solved in $O(n)$ time, while it takes $O(pn)$ time to solve the corresponding $p-$coverage problems. In this section we will briefly indicate how the same bounds can be obtained after a slight modification of the algorithms presented in Sections 2 and 3. We will only discuss the relaxed coverage problems, because the modifications for the $p-$coverage problems are similar.

Subcase I: $r_i=r$ for all $i=1,\ldots,n$. It is easy to see that in this case the matrix D can be taken equal to the matrix A, because $f(i)$ and $l(i)$ are both monotonically non-decreasing functions of i in the original indexation. Moreover, the latter also implies that determining $f(i)$ and $l(i)$ for all $i=1,\ldots,n$ can be done in $O(n)$ time. Instead of storing the elements of Q - the indices of the non-dominated facilities - in a balanced tree, we now use a double linked list. Furthermore, we use a pointer to indicate g, which is in stage i the smallest element of Q greater than or equal to $f(i)$. Again using the monotonicity of $f(i)$, it is not difficult to see that finding the correct value of g in every stage can be done in a computational effort that is

overall $O(n)$.

Subcase II: $b_i = \infty$ for all $i = 1, \ldots, n$. Hence, in this problem every client has to be served. We will show that instead of matrix A, we may use matrix A' which is defined as follows. Let $fc(j)$, $j = 1, \ldots, n$, denote the smallest row such that $a_{ij} = 1$ for all $i = fc(j), \ldots, j$ (note that $a_{jj} = 1$). Similarly, let $lc(j)$ denote the largest row such that $a_{ij} = 1$ for all $i = j, \ldots, lc(j)$. Column j of the (0,1)-matrix A' is defined by $a'_{ij} = 1$ if and only if $fc(j) \leq i \leq lc(j)$. Hence, A' has the column consecutive 1's property. This, combined with the fact that A has the row consecutive 1's property, implies that A' has also the row consecutive 1's property. Define $f'(i)$ and $l'(i)$ to be the first respectively last column in which row i of matrix A' contains a 1. We have the following results with respect to the structure of A'.

Lemma 4 $fc(j)$ and $lc(j)$ are monotonically non-decreasing in j.

Proof Suppose there exists an $j < n$ and an i such that $fc(j) > i = fc(j+1)$. Because $a'_{ij} = 0$ and $j \geq fc(j) > i$, it follows that j must be greater than $l'(i)$. Hence, also $j+1$ is greater than $l'(i)$ and therefore $a'_{i,j+1} = 0$. This contradicts $fc(j+1) = i$. One can prove analogously that $lc(j)$ is monotonically non-decreasing in j. □

Lemma 5 $f'(i)$ and $l'(i)$ are monotonically non-decreasing in i.

Proof Analogously to the proof of Lemma 4. □

From Lemma 4 it follows that a compact representation of matrix A' can be obtained in $O(n)$ time, while Lemma 5 implies that the same bound holds for solving the coverage problem w.r.t. matrix A' (cf. Subcase I). Hence, to show that the linear time bound applies to Subcase II, it now suffices to prove that replacing A by A' does not really alter the problem.

Lemma 6 Let S be a feasible choice of open facilities, i.e., every client is being covered by S.. Then S is still feasible if all facilities $v_j \in S$ are restricted to serve only clients u_i with $fc(j) \leq i \leq lc(j)$.

Proof It suffices to show that if we let every client be served by an open facility that is nearest to it, then every $v_j \in S$ serves only clients v_i with $fc(j) \leq i \leq lc(j)$. Suppose this is not true, then we may assume w.l.o.g. that there exists a client v_i that is being served by facility v_j, while $i > lc(j)$. Because $a_{ij} = 1$, it follows that $a_{hj} = 0$ for some h with $j < h < i$. Client v_h is located between v_j and v_i, and therefore it holds that

$$d(v_j, v_i) = d(v_j, v_h) + d(v_h, v_i) > r_h + d(v_h, v_i) \tag{15}$$

Let v_l be the facility that serves v_h. Clearly, $l \leq j$ is impossible because if v_j can not serve v_i, then this would also hold for v_l. If $j < l \leq i$ then we would obtain a contradiction with the fact that client v_i is being served by an open facility that is closest. Hence, the only possibility left is $l > i$, in which case

$$d(v_i, v_l) = d(v_h, v_l) - d(v_h, v_i) \leq r_h - d(v_h, v_i) \tag{16}$$

Combining (15) and (16) yields $d(v_i, v_l) < d(v_i, v_j)$ and this contradicts the fact that v_j is a facility closest to v_i. □

5. Concluding remarks

We have shown how the $p-$coverage problem on the real line can be solved in $O(pn\log n)$ time using a parametric approach that exploits the problem structure. Our approach differs significantly from the one proposed by Hassin and Tamir (1990) and has a lower complexity for problem instances in which the upper bound on the number of open facilities is small compared to the number of potential facilities.

In Van Hoesel and Wagelmans (1991) a similar approach as in this paper is used to design algorithms for several economic lot-sizing problems in which the setup costs can be viewed as linear functions of a single parameter. Hassin and Tamir already showed that location problems on the real line and the economic lot sizing problem allow similar solution techniques. In particular we would like to point out here that Lemma 2 of this paper states a property which resembles Wagner and Whitin's Planning Horizon Theorem, while Lemma 1 can be viewed as an analog of Theorem 4 in Wagner and Whitin (1958). It seems worthwhile to identify other dynamic programming problems for which such structural properties hold.

Acknowledgement

This research was carried out while the second author was visiting the Operations Research Center at the Massachusetts Institute of Technology with financial support of the Netherlands Organization for Scientific Research (NWO). He would like to thank the students, staff and faculty affiliated with the ORC for their kind hospitality.

References

Aho, A.V., J.E. Hopcroft and J.D. Ullman (1974), *The Design and Analysis of Computer Algorithms*, Addison-Wesley, London

Eisner, M.J. and D.G. Severance (1976), "Mathematical Techniques for Efficient Record Segmentation in Large Shared Databases", *Journal of the Association for Computing Machinery* 23, 619 – 635

Hassin, R. and A. Tamir (1990), "Improved Complexity Bounds for Location Problems on the Real Line", Working Paper, Department of Statistics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel

Kolen, A. and A. Tamir (1990), "Covering Problems", in *Discrete Location Theory*, eds. P.B. Mirchandani and R.L. Francis , John Wiley & Sons, Inc., New York

Megiddo, N. (1979), "Combinatorial Optimization with Rational Objective Functions", *Mathematics of Operations Research* 4, 414 – 424

Van Hoesel, C.P.M. and A.P.M. Wagelmans (1991), "On Setup Cost Reduction in the Wagner-Whitin Model", Working Paper, Operations Research Center, Massachusetts Institute of Technology, Cambridge MA (forthcoming)

Wagner, H.M. and T.M. Whitin (1958), "Dynamic Version of the Economic Lot Size Model", *Management Science* 5, 89 – 96