JOURNAL OF ALGORITHMS 8, 305–323 (1987)

Fast Approximation Algorithms for a Nonconvex Covering Problem

DORIT S. HOCHBAUM*

University of California, Berkeley, Berkeley, California 94720

AND

WOLFGANG MAASS[†, ‡]

University of Illinois at Chicago, Chicago, Illinois 60680

Received July 1983; revised March 1986 and August 1986

We study approximation schemes for computing minimal coverings by nonconvex objects in one dimension. This problem arises, for instance, in the context of motion planning for robots. In this paper we describe a polynomial approximation scheme for this strongly *NP*-complete problem. For this purpose, we develop a general method—the shifting strategy—nested applications of which yield such a scheme (polynomial schemes for strongly *NP*-complete problems are quite rare). With some additional effort, the shifting strategy leads to algorithms that are of practical interest in that their running time is bounded by low-degree polynomials.

1. INTRODUCTION

In this paper we study approximation algorithms for some covering problems that arise in motion planning for robots. These covering problems are of particular mathematical interest since they typically require covering points by a minimal number of objects that are nonconvex. The corresponding problems of covering with convex objects are usually easier to analyze. For instance, we were able to derive polynomial approximation

* This research was supported in part by the National Science Foundation under Grant ECS-8501988.

† During the preparation of this paper, this author was supported by the Heisenberg Programm der Deutschen Forschungsgemeinschaft, West Germany.

‡ Permanent address: Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago.

0196-6774/87 $3.00

schemes for numerous convex covering and packing problems [3]. When we employ the same technique in the nonconvex case, the scheme derived is exponential in a nonconvexity parameter (later to be referred to as "sparseness").

The nonconvex objects studied in this paper are *rings*, i.e., the region enclosed between two concentric spheres of radii r_1, r_2 with $r_1 < r_2$. In this case the nonconvexity parameter is the ratio between the inner radius r_1 and the width $r_2 - r_1$. A motivation for studying this problem arises in the planning of motion for mobile robots so that all objects are reachable. The construction of a robot is characterized by a pivot support for an extendible arm. The planar region accessible by a robot's arm will typically resemble a ring around the support pivot. In this case, any object is placed between the minimum and maximum range of the arm can be reached by the robot. Since accelerating or stopping a mobile robot is a relatively slow process, we will be interested in how to identify the minimum number of placement positions, such that all objects are accessible (see [13] for additional details).

Increased flexibility of the robot is translated, in terms of the geometry of the ring-like accessibility range, to decreased inner radius of the ring compared to its width. Curiously, this smaller ratio between inner radius and width leads also to decreased complexity of the placement problem.

In this paper, we focus on the one-dimensional case of the considered covering problem. This ring cover problem proved to be hard even when restricted to one dimension [11], while covering by convex objects, i.e. intervals, is trivial in one dimension. Therefore, it provides an excellent study ground where we can develop approximation tools for the nonconvex case. In one dimension the rings are pairs of identical intervals at a fixed distance apart. The problem is thus to cover objects on a line with a minimum number of such interval pairs.

We shall refer to the objects to be covered as "points," to the pair of closed intervals of length w each and $2r$ apart as a "*ring*" of size $\langle r, w \rangle$, and to the quantity r/w as "*sparseness*" (see Fig. 1). The set of points is denoted by N, and its size by n. We denote by D the *diameter* of the ring, $2r + 2w$, by I the interval in which the points to be covered are contained and by $\lceil x \rceil$ the integer ceiling of a real number x (i.e., the smallest integer larger than or equal to x). The problem of finding the minimum number of rings covering points in I is referred to as the *ring cover problem* throughout the paper.

The ring cover problem has interesting complexity aspects. It was proved strongly NP-complete by Maass [11] even when all rings are identical—of size $\langle r, w \rangle$—but polynomial for each fixed value of the sparseness r/w. The fact that the problem is *strongly* NP-complete means that if the problem parameters, such as r, w and r/w, were given in *unary* encoding as part of the input there would still be no algorithm polynomial in the input length that solves the problem, unless $NP = P$. (For a comprehensive

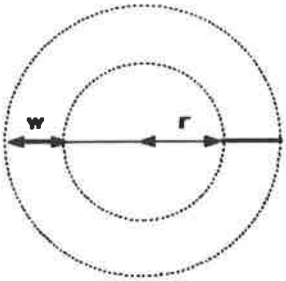

FIGURE 1

review of the theory of NP-completeness the reader is referred to Garey and Johnson [2].)

Once a problem has been identified as NP-complete, it is unlikely that one can find a fast (i.e., polynomial) algorithm that delivers the optimal solution. It may still be possible, however, to find polynomial algorithms that deliver approximate solutions. The quality of such solutions is measured in terms of the relative error. Formally, let IP be a given problem instance, $|\mathrm{OPT}(IP)|$ the value of the optimal solution, and $|\mathrm{H}(IP)|$ the value of the solution for a heuristic algorithm H. Then,

$$\sup_{IP \in \left\{\substack{\text{problem} \\ \text{instances}}\right\}} \frac{|\mathrm{H}(IP)| - |\mathrm{OPT}(IP)|}{|\mathrm{OPT}(IP)|}$$

is called the *worst case error*. We say that a heuristic is a *δ-approximation algorithm* for some $\delta > 0$, if the worst case error is bounded by δ. $(1 + \delta)$ is also called the *performance ratio* of the algorithm (for a minimization problem).

One can easily devise a 1-approximation algorithm for the ring cover problem by placing one ring at a time with its left interval's left border placed on the leftmost point yet uncovered. In order to see why the worst case error of this algorithm is bounded by 1, we consider the related problem of covering with the minimum number of identical (single) intervals, or one-dimensional disks. This is easily solvable by placing the left end of the disk on the leftmost point yet uncovered, and repeating until all points are covered. Since the number of rings needed to cover all points does not exceed the number of disks needed, and the number of disks is at most twice the minimum number of rings, the result follows easily. More elaborate agruments are used in section 5 to concoct a $\frac{1}{2}$-approximation algorithm that takes $O(\min(n^5, n^4 \cdot r/w))$ steps. By now, a natural question is how much can the performance improve; that is, whether there is an

ε-approximation algorithm for $\varepsilon < \frac{1}{2}$ and whether there is some inherent lower bound on ε. This question is related to the existence of a polynomial-time approximation scheme, that is, a family of algorithms such that for every $\varepsilon > 0$ there is an algorithm in the family delivering an ε-approximate solution in polynomial time. If the running time of such ε-approximation algorithms depends also polynomially of $1/\varepsilon$, then the scheme is called fully polynomial. It has, however, been proved by Garey and Johnson [2] that no such fully polynomial approximation scheme exists for strongly NP-complete problems, unless $P = NP$. (This proof requires that the optimal value be bounded by a polynomial in the problem length. This is certainly the case for the ring cover problem—the optimum never exceeds n.) For strongly NP-complete problems it may be possible to find a polynomial ε-approximation scheme that depends exponentially on $1/\varepsilon$, but is polynomial in all other problem parameters. Such schemes, however, are quite rare. One such scheme is reported by Ibarra and Kim [7] for a generalization of the knapsack problem. Another independent work by Baker [1], describes polynomial schemes using a technique similar to ours for problems on planar graphs. Other schemes proposed [9], [10] are what we call "asymptotic" polynomial approximation schemes in that the procedure described works only for problem instances with input size $> n_0$, where $n_0 = n_0(1/\varepsilon)$ is growing with $1/\varepsilon$. Instances of size $\leq n_0$ are handled using different methods such as enumeration that ensure ε-approximation while are still polynomial for fixed ε. Recently approximation schemes were devised for two strongly NP-complete scheduling problems [5], [6]. The technique used is called dual approximation since it exploits a dual relationship between those problems and some bin packing problems.

The main result of this paper (Sect. 4) is the existence of a polynomial time approximation scheme for the considered strongly NP-complete ring cover problem. We also give particularly efficient approximation algorithms for special classes of problem instances.

The plan of the paper is as follows: in Section 2 we describe the shifting strategy that is the fundamental building block of all our approximation algorithms. When r/w is relatively small, we propose in Section 3 an approximation scheme that has running time exponential in r/w, but runs faster than both the optimal algorithm and the polynomial time approximation scheme for small r/w. Section 4 describes the above-mentioned polynomial approximation scheme. In the subsequent two sections we describe algorithms of practical interest in that their running time is bounded by low degree polynomials. The algorithm in Section 5 works for arbitrary r/w and guarantees a bound on the absolute error divided by the optimum, called relative error, of at most $\frac{1}{2}$. For $r/w \leq 1/2$, we propose in Section 6 an approximation scheme that is much faster than the general scheme. Section 7 is a summary.

2. The "Shifting" Strategy

All the algorithms presented in this paper make use of the shifting strategy. The basic idea of this strategy is the conversion of solutions derived by a "local" algorithm on certain bounded intervals to an approximate solution to the "global" problem (that covers all points). The shifting strategy allows us to bound the error of this simple approach by repetitive applications of it, followed by the selection of the single most favorable resulting solution. Such a strategy adds only a multiplicative factor to the algorithm's running time.

Let the set N of the n given points be enclosed in an interval I and let l be a positive integer. In the first phase the interval I is subdivided into intervals of length D each, with the possible exception of the leftmost and rightmost subintervals that can be shorter. Each such subinterval will be considered left closed and right open. These subintervals are then considered in groups of l consecutive intervals resulting in intervals of length $l \cdot D$ each (again with the possible exception of the leftmost and rightmost such intervals). For any fixed subdivision of I into intervals of length D, (see Fig. 2 for an illustration) there are l different ways of grouping the intervals of the partition into periods of length $l \cdot D$. These partitions can be ordered such that each can be derived from the previous one by shifting the cuts to the right over distance D. Repeating the shift l times we end up with the same partition we started from. We denote such l distinguished partitions by $S_1, S_2, \dots, S_l$. We shall refer henceforth to all intervals created by the partitions as $l \cdot D$-intervals with the interpretation that their length is $l \cdot D$ or less.

Let A be some algorithm that delivers a solution—i.e., a set of rings that covers all given points—in any $l \cdot D$-interval (or shorter). For a given partition S_i, let $A(S_i)$ be the algorithm that applies algorithm A to each interval in the partition S_i, and outputs the union of all rings used to cover these intervals. Such a set of rings is clearly a feasible solution to the global problem defined on I. This process of finding a global solution is repeated for each partition S_i, $i = 1, 2, \dots, l$. The shift algorithm S_A, defined for a given local algorithm A, delivers the set of rings of minimum cardinality among the l sets delivered by $A(S_1), \dots, A(S_l)$.

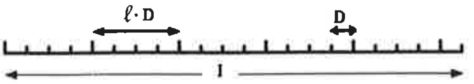

Fig. 2. Illustration of one partition to intervals of length $l \cdot D$, where $l = 4$.

Let the performance ratio of algorithm S_A be denoted by r_{S_A}, i.e.,

$$r_{S_A} = \sup_{IP \in \left\{ \substack{\text{problem} \\ \text{instances}} \right\}} \frac{|S_A(IP)|}{|\mathrm{OPT}(IP)|}.$$

We now prove the shifting lemma.

LEMMA 2.1 (the shifting lemma).

$$r_{S_A} \le r_A\left(1 + \frac{1}{l}\right), \tag{2.1}$$

where r_A is the performance ratio of the local algorithm A.

Proof. The proof uses the idea that if for one partition S_i the resulting solution delivered by the algorithm $A(S_i)$ has large error associated with it (typically because most of the points of N lie close to the cuts of the partition), then there is another partition S_j such that the solution delivered by $A(S_j)$ has a small error associated with it. Technically, this is proved by producing an upper bound on the sum of errors of the solution delivered by all algorithms $A(S_i)$ for $i = 1, 2, \ldots, l$.

By the definition of r_A we have

$$|A(S_i)| \le r_A \cdot \sum_{J \in S_i} |\mathrm{OPT}_J|, \tag{2.2}$$

where $|\mathrm{OPT}_J|$ is the number of rings in an optimal cover of the points in interval J and "$J \in S_i$" indicates that interval J is an $l \cdot D$-interval in partition S_i.

Let OPT be the set of rings in an optimal solution and $\mathrm{OPT}^{(1)}, \ldots, \mathrm{OPT}^{(l)}$ the set of rings, in OPT, containing points in two $l \cdot D$-intervals (that must be adjacent) in the $S_1, S_2, \ldots, S_l$ partitions, respectively. It can easily be seen that (Fig. 3)

$$\sum_{J \text{ in } S_i} |\mathrm{OPT}_J| \le |\mathrm{OPT}| + |\mathrm{OPT}^{(i)}|. \tag{2.3}$$

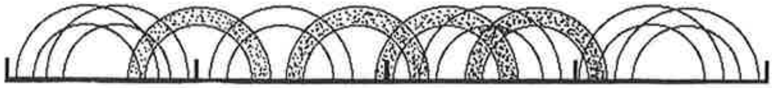

FIG. 3. For each given partition S_i there is a different set of rings that cover points in two adjacent intervals. The shaded rings are the ones in the set OPT^i.

Due to the geometric design of the partitions, each with cut points at a distance of at least D from the others, there can be no ring in the set OPT that contains points in two adjacent intervals in more than one partition. Therefore, the sets $\mathrm{OPT}^{(1)}, \ldots, \mathrm{OPT}^{(l)}$ of rings in the optimal cover are disjoint. It follows that

$$\sum_{i=1}^{l} \left(|\mathrm{OPT}| + |\mathrm{OPT}^{(i)}| \right) \leq (l+1) \cdot |\mathrm{OPT}|. \tag{2.4}$$

(2.3) and (2.4) imply

$$\min_{i=1,\ldots,l} \sum_{J \text{ in } S_i} |\mathrm{OPT}_J| \leq \frac{1}{l} \cdot \sum_{i=1}^{l} \left(\sum_{J \text{ in } S_i} |\mathrm{OPT}_J| \right) \leq \left(1 + \frac{1}{l}\right) |\mathrm{OPT}|. \tag{2.5}$$

Joining the inequality (2.5) with (2.2) we derive that

$$|S_A| = \min_{i=1,\ldots,l} |\mathrm{A}(S_i)| \leq r_{\mathrm{A}} \cdot \left(1 + \frac{1}{l}\right) \cdot |\mathrm{OPT}|, \tag{2.6}$$

which establishes (2.1). Q.E.D.

3. An ε-Approximation Scheme for r/w Bounded

Practical applications of the 1-dimensional ring cover problem would often use rings with bounded sparseness. The consideration of the case where r/w is bounded, is of interest for the sake of algorithmic analysis as well. For each given distribution of n points there is a bound on r/w beyond which the ring problem becomes very easy. Once the ring width w is smaller than the smallest gap between any pair of points, any ring could cover at most two points. This version of the problem can be solved in polynomial time: All rings in this case can cover either one or two points. We delete the points which cannot be covered by a ring covering any other point; the remaining problem of choosing a set of rings covering two points each can be represented as an edge cover problem (choose a subset of mimimum size of the edges of a graph, such that all vertices of the graph are adjacent to at least one edge in this subset). The edge cover problem can be solved in time $0(\sqrt{|V|}|E|)$, using the maximum matching algorithm (see [12] for the maximum matching algorithm), where $|V|$ is the number of vertices and $|E|$ the number of edges. Note that in the specific graph described here, the maximum degree of a vertex is 4. As a result, $|E| \leq 4|V| = 4n$. It follows that one can solve this edge cover problem in $O(n^{1.5})$.

Once r/w is treated as fixes, there is a polynomial algorithm available (see Maass [11]). Nevertheless, the running time of that algorithm,

$O(n^{16\lceil r/w\rceil+18} \cdot \lceil r/w\rceil^2 \cdot \log n)$, makes it impractical even for moderately small values of r/w. For instance, for $r/w = 1$, the optimal algorithm runs in time exceeding $O(n^{34})$. Therefore, there is clearly a need for an algorithm that could deliver a solution, if only approximate, in reasonable running time.

The algorithm used involves the shifting strategy described in Section 2. The interval I, in which the set N of points is placed, is subdivided into intervals of length $l \cdot D$, where $l = \lceil 1/\varepsilon \rceil$, and $\varepsilon > 0$ is the prescribed bound on the worst case error. In each interval of length $l \cdot D$ we employ an optimal algorithm denoted by B as the "local" algorithm. Each $l \cdot D$-interval can be completely covered by a collection of rings tightly packed next to each other with minimal overlap, using no more than $\lceil l/2 \rceil \cdot (\lceil D/w \rceil + 1)$ rings. We call such collection a "compact" packing. It is enough to enumerate all sets of ring positions of size no larger than that quantity and select the smallest such subset that is a feasible cover; this essentially describes Algorithm B. This algorithm is obviously exponential in both r/w and l. (This fact will be established more precisely in the proof of Lemma 3.2.) Its exponential dependence on r/w is, however, far less serious than it is for the optimal dynamic programming algorithm in Maass [11]. The following lemma will be used in the proof of Lemma 3.2.

LEMMA 3.1. We can verify a guess that at most K rings are required to cover the n points in an interval by examining only $K \cdot \binom{2n}{K}$ covers each at K steps at most. If the guess is correct the procedure will result in identifying a minimum cover in $K^2\binom{2n}{K} \leq 2 \cdot (2n)^K$ steps at most.

Proof. We may assume, without loss of generality, that for each ring in a cover at least one of the two left borders of its two w-intervals coincides with one of the n given points it contains, i.e., each ring is "right justified." Therefore, the n given points correspond to at most $2n$ ring positions that have to be considered (Fig. 4). By preprocessing the data we can obtain a list of the $2n$ ring positions and specify for each ring the leftmost and rightmost point in each of the pairs of intervals. This can be done in time $O(n \cdot \log n)$. Using enumeration we consider all possible subsets of rings where the number of rings in each subset does not exceed K. The number of such subsets in the interval is bounded by $\sum_{i=1}^{K}\binom{2n}{i} \leq K\binom{2n}{K}$. (The

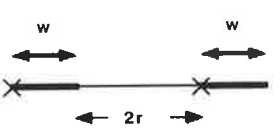

FIG. 4. Possible locations of border points.

inequality is immediate since we can assume $K \leq n$.) The feasibility check for a given set of rings can be executed in time linear in the number of rings in the set: Recall that each ring is represented by the two pairs of leftmost and rightmost points in each of its two w-intervals (e.g., (4, 7; 10, 11) indicates that the ring covers the points indexed by 4, 5, 6, 7, 10, 11). For a set of rings that are given in this representation (sorted by the leftmost point covered by each ring) one can check in linear time whether they cover all points in a considered interval. This may be done as follows. First one computes in linear time the union of the left intervals of the rings (again in terms of intervals of indices of given points that are covered), where two intervals are merged into one larger interval if they leave no point uncovered in between. Then one inserts analogously in linear time the right intervals of the rings. The considered rings form a feasible cover of the points in the respective interval J, if the constructed union of the intervals consists of *one* interval of indices that starts with the index of the leftmost point in J and ends with the index of the rightmost point in J. The number of steps for the examination of each cover is hence at most K and the number of covers examined at most $K\binom{2n}{K}$. The total number of steps is bounded by $K^2\binom{2n}{K}$ as claimed. Now

$$K^2\binom{2n}{K} \leq \frac{K}{(K-1)!}(2n)^K.$$

But, $K/(K-1)! \leq 2$, so the statement of the lemma is proved. Q.E.D.

LEMMA 3.2. The running time of the shift algorithm S_B is bounded by

$$O\left(\frac{1}{\varepsilon} \cdot (2n)^{\lceil 1/2\varepsilon \rceil \cdot (\lceil 2r/w \rceil + 3)}\right).$$

Proof. We consider a "compact" covering in the ith $l \cdot D$-subinterval containing n_i points. Such a subinterval can hence be covered by at most $L_i = \min\{\lceil l/2 \rceil \cdot (\lceil D/w \rceil + 1), n_i\}$ rings. (Note that $D/w + 1 = 2r/w + 3$.) By Lemma 3.1 the positions of these rings can be selected from a set of at most $2n_i$ ring positions. Furthermore, we can find the minimum cover of the points in no more than $L_i^2 \cdot \binom{2n_i}{L_i}$ or $O((2n_i)^{L_i})$. We now define the quantity $L = \min\{\lceil l/2 \rceil(\lceil 2r/w \rceil + 3), n\}$, and note that $L \geq L_i$ for all i. It follows that the total running time summed up for all subintervals, $O(\sum_i (2n_i)^{L_i})$, does not exceed $O((2n)^L)$. Finally, we substitute $\lceil 1/\varepsilon \rceil$ for l, so the running time for each of the l partitions of I is at most

$$O\left((2n)^{\lceil 1/2\varepsilon \rceil(\lceil 2r/w \rceil + 3)}\right). \tag{3.1}$$

Q.E.D.

Though the running time of the shift algorithm is still exponential in the sparseness, it is faster than the dynamic programming optimization algorithm for ε-approximate solutions with $\varepsilon \geq 0.07$. For $r/w \leq 2$, the running time of this shift algorithm is considerably better than that of the general polynomial approximation scheme described in the next section.

4. A Polynomial Approximation Scheme

The scheme presented in the previous section is not a polynomial approximation scheme since the running time depends exponentially on the sparseness r/w. This is due to the local algorithm used in the shift strategy that solves the problem optimally in each interval of length $l \cdot D$. In this section we circumvent this difficulty by using an approximation algorithm from another approximation scheme as the local algorithm. This is done by partitioning further each interval of length $l \cdot D$ to sets of subintervals with the property that a compact covering of each of these sets of intervals does not take more than $O(l^2)$ rings. Therefore one can find an optimal cover for these sets of intervals by enumerating all possible sets of rings with no more than $O(l^2)$ rings in a set. This procedure removes the exponential dependence on r/w in the running time of the optimal local algorithm, and replaces it with the exponential dependence on $O(l^2)$. We apply this "local" polynomial approximation scheme for $r/w > 1$, otherwise the scheme described in the previous section is used.

In this local approximation scheme the parameter l of the shift algorithm is set equal to $\lceil 3/\varepsilon \rceil$. The local algorithm parametrized by l, call it $C(l)$, delivers an approximate solution in intervals of length $l \cdot D$ or less. Each such interval is subdivided to K intervals of length $2r + w$ each (where the last one might be shorter), and each of those is further partitioned to subintervals of length $l \cdot w$ each. All these intervals are considered left-closed right-open. Recalling that $D = 2r + 2w$, we note that $K = \lceil lD/(2r + w) \rceil < \lceil \frac{4}{3} l \rceil$ since $r > w$.

We denote the collection of all K, leftmost $l \cdot w$-intervals in the $(2r + w)$-intervals by $U_{1,1}$ (Fig. 5). The next set of $l \cdot w$-intervals, adjacent to the previous one to its right, is denoted by $U_{2,1}$; the next by $U_{3,1}$, etc., up to $U_{q,1}$

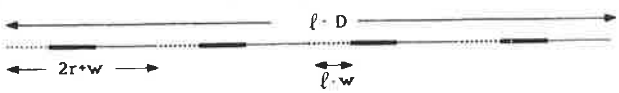

Fig. 5. The subdivision of the $l \cdot D$-interval into intervals of length $l \cdot w$ with $K = 4$. The union of the (disjoint) intervals marked by the wavy line is $U_{1,1}$ nd the union of the underlined intervals is $U_{2,1}$.

with $q = \lceil (2r + w)/(l \cdot w) \rceil$ (see Fig. 5). The sets of intervals $U_{i,1}$, $i = 1, \ldots, Q$ have the important property that any ring with one of its w-intervals intersecting in an $l \cdot w$-interval of $U_{j,1}$ must have its other w-interval intersecting in $U_{j,1}$ or in an adjacent $l \cdot w$-interval of $U_{j-1,1}$ or of $U_{j+1,1}$ (at a distance of $2r + w$ away). We now consider all possible $l - 1$ shifts of the $l \cdot w$-intervals by distance w. Given $i \in \{1, \ldots, q\}$ and $s \in \{2, \ldots, l\}$ we define $U_{i,s}$ to be the set of intervals that results from shifting all intervals in $U_{i,1}$ by distance $(s - 1) \cdot w$ to the right. An alternative definition of $U_{i,s}$ views it as the union of all K intervals of length $l \cdot w$ (or less for the rightmost one) whose endpoints are at distance $(s - 1) \cdot w + j(2r + w)$ for $j = 1, \ldots, K$, from the left end of the $l \cdot D$-interval. By including the set, call it $U_{0,s}$ of those K $(s - 1) \cdot w$-intervals that lie leftmost in the $(2r + w)$-intervals, we make sure that for every $s = 1, \ldots, l$ the sets $U_{0,s}, U_{1,s}, \ldots, U_{q,s}$ form a partition of the considered interval of length $l \cdot D$.

The local algorithm $C(l)$ is an approximation algorithm which belongs to a polynomial approximation scheme for bounded intervals. It consists of solving optimally for each set of intervals $U_{i,s}$ for $i = 0, 1, \ldots, q$, $s = 1, \ldots, l$. By construction, each set $U_{i,s}$ consists of at most K $l \cdot w$-intervals. Therefore it can be completely covered by $\lceil K/2 \rceil \cdot l$ rings which is no more than l^2 rings. Let an optimal set of rings covering $U_{i,s}$ be denoted by $\text{OPT}_{i,s}$. For each shift s, $\bigcup_{i=0}^{q} \text{OPT}_{i,s}$ is a feasible ring cover for the $l \cdot D$ interval. The local algorithm $C(l)$ delivers as a solution the set of such rings of minimum cardinality compared to all possible shifts. More precisely, let $\bar{s}$ be the shift such that $\sum_{i=0}^{q} |\text{OPT}_{i,\bar{s}}| = \min_{s=1,\ldots,l} \sum_{i=0}^{q} |\text{OPT}_{i,s}|$, then the solution set delivered by the local algorithm is $\bigcup_{i=0}^{q} \text{OPT}_{i,\bar{s}}$. (Note that a solution that is at least as good could be derived by taking s for which $\min_{s=1,\ldots,l} |\bigcup_{i=0}^{q} \text{OPT}_{i,s}|$ is attained).

THEOREM 4.1. (a) *for $r > w$, choosing $l = \lceil 3/\varepsilon \rceil$, the relative error of the algorithm $S_{C(l)}$ is bounded by ε:*

$$\frac{|S_{C(l)}|}{|\text{OPT}|} \leq \left(1 + \frac{1}{l}\right)^2 \leq 1 + \varepsilon.$$

(b) *The running time of $S_{C(l)}$ is at most $O(l^2 \cdot (2n)^{l^2})$ (i.e. $O((1/\varepsilon^2)(2n)^{\lceil 3/\varepsilon \rceil^2})$).*

Proof. (a) It suffices to show that the performance ratio of the local algorithm, $r_{C(l)}$, is bounded by $(1 + 1/l)$, since then it follows from Lemma 2.1 that

$$\frac{|S_{C(l)}|}{|\text{OPT}|} \leq r_{S_{C(l)}} \leq r_{C(l)}\left(1 + \frac{1}{l}\right) \leq \left(1 + \frac{1}{l}\right)^2 \leq 1 + \varepsilon. \tag{4.1}$$

Let $\overline{\mathrm{OPT}}$ be an optimal set of rings (of minimum cardinality) covering the points in an interval of length $l \cdot D$ (or less). We denote by $\overline{\mathrm{OPT}}^{(s)}$ the sets of those rings in $\overline{\mathrm{OPT}}$ that contain points in two adjacent $U_{i,s}$, $i = 1, \ldots, q$, for shifts $s = 1, 2, \ldots, l$. Note that the sets $\overline{\mathrm{OPT}}^{(1)}, \overline{\mathrm{OPT}}^{(2)}, \ldots, \overline{\mathrm{OPT}}^{(l)}$ are disjoint, since the "ring phase," $2r + w$, is equal to the length of each of the K subintervals in the considered $(l \cdot D)$-interval. Let the optimal set of rings covering the points in $U_{0,s}, \ldots, U_{q,s}$ for shift s be denoted by $\mathrm{OPT}_{0,s}, \ldots, \mathrm{OPT}_{q,s}$, respectively. Now,

$$\sum_{i=0}^{q} |\mathrm{OPT}_{i,s}| \leq |\overline{\mathrm{OPT}}| + \left|\overline{\mathrm{OPT}}^{(s)}\right|, \qquad s = 1, \ldots, l, \tag{4.2}$$

since the optimal solution, $\overline{\mathrm{OPT}}$, on the $l \cdot D$-interval can be split into individual feasible covers for each set $U_{i,s}$. This holds due to the following argument. Let $Q_{i,s}$ be the set of rings in $\overline{\mathrm{OPT}}$ intersecting $U_{i,s}$. Then $|Q_{i,s}| \geq |\mathrm{OPT}_{i,s}|$, and adding all these cardinalities yields $|\overline{\mathrm{OPT}}| + |\overline{\mathrm{OPT}}^{(s)}|$, since only the rings in $\overline{\mathrm{OPT}}^{(s)}$ are counted twice: hence the inequality. Repeating arguments similar to the ones in Lemma 2.1 we find a shift $\bar{s}$ for which the following holds:

$$l \cdot \sum_{i=0}^{q} |\mathrm{OPT}_{i,\bar{s}}| \leq \sum_{s=1}^{l} \sum_{i=0}^{q} |\mathrm{OPT}_{i,s}|$$

$$\leq l \cdot |\overline{\mathrm{OPT}}| + \sum_{s=1}^{l} \left|\overline{\mathrm{OPT}}^{(s)}\right| \leq (l + 1) \cdot |\overline{\mathrm{OPT}}|. \tag{4.3}$$

Consequently, the approximate solution value in each $l \cdot D$-interval does not exceed $(1 + 1/l) \cdot |\overline{\mathrm{OPT}}|$, and $r_{C(l)} \leq 1 + 1/l$. Substituting this inequality in (4.1) we get the desired result.

(b) The points in each set of intervals $U_{i,s}$ can be covered by at most $\lceil \frac{1}{2} \cdot K \rceil \cdot l$ rings, using "compact" covering, where as previously indicated $K = \lceil lD/(2r + w) \rceil < \lceil \frac{4}{3} l \rceil$. Each ring can be chosen such that at least one of the points in $U_{i,s}$ is on one of its two left borders. Invoking Lemma 3.1 with number of rings $\overline{K} = \min\{\lceil \frac{1}{2} K \rceil \cdot l,\ n_{i,s}\}$, where $n_{i,s}$ denotes the number of points in $U_{i,s}$, we need at most $O((2n_{i,s})^{\lceil (1/2) K \rceil \cdot l})$ steps to find the minimum cover of $U_{i,s}$. This is repeated for all sets $U_{0,s}, U_{l,s}, \ldots, U_{q,s}$ and for all l shifts. Hence, the total running time of algorithm C(l) does not exceed $O(l \cdot (2\bar{n})^{\lceil (1/2) K \rceil \cdot l})$ for $\bar{n}$ the number of points in an interval of length $l \cdot D$. Since $K < \lceil \frac{4}{3} l \rceil$, this expression is bounded by $O(l(2\bar{n})^{\lceil (1/2) \lceil (4/3) l \rceil \rceil l})$ for each $l \cdot D$-interval and by $O(l \cdot (2n)^{\lceil (1/2) \lceil (4/3) l \rceil \rceil l})$ for all intervals. This is repeated for all l partitions. Substituting $\lceil 3/\varepsilon \rceil$ for l, we derive the stated result. Q.E.D.

5. More Efficient Local Algorithms Yield Faster Approximation Algorithms

In this section we sketch a method that allows us to improve the time bound for one of our previously described approximation algorithms. The preceding algorithms used the shifting strategy in order to reduce a global covering problem to a local covering problem. In the case where the local covering problem was solved optimally it was done via an enumeration of all possible solutions. In many cases there exist substantially faster optimal local algorithms that one can use in order to get a more efficient approximation algorithm for the global covering problem.

We sketch this approach here for the case of $\frac{1}{2}$-approximation algorithm for the ring cover problem. The $\frac{1}{2}$-approximation algorithm from the polynomial approximation scheme in Section 4 yields time bound of $O(n^{36})$. We show here that with the help of a more efficient local algorithm one gets a $\frac{1}{2}$-approximation algorithm of time complexity $O(\min(n^5, n^4 \cdot r/w))$.

The new $\frac{1}{2}$-approximation algorithm uses again the shifting strategy to reduce the global covering problem to a local covering problem for intervals of length $2 \cdot D$ (where $D = 2r + 2w$). For the local covering problem we use instead of the time consuming enumeration algorithm a more efficient algorithm that constructs an optimal covering of $\bar{n}$ given points in an interval of length $2 \cdot D$ by rings of size $\langle r, w \rangle$ in $O(\min(\bar{n}^5, \bar{n}^4 \cdot r/w))$ steps. This local algorithm makes use of some simple facts about the geometrical structure of optimal local coverings in intervals of length $2 \cdot D$.

The more efficient local covering algorithm runs repeatedly through two phases. In the first phase it places up to four rings G_1, G_2, $\tilde{G}_1$, $\tilde{G}_2$ in an arbitrary fashion (but with one of the given points on the left end of one of the two intervals of each of them; there are $O(\bar{n}^4)$ possibilities of doing this). If the only points that are uncovered lie either between the two intervals of G_1 or between the two intervals of G_2 then the algorithm covers in the second phase these remaining points via algorithm F of Lemma 5.1 in $O(\min(\bar{n}, r/w))$ steps. Otherwise, the first phase will be repeated for another collection of four rings. In the end the local algorithm outputs the covering with the fewest rings that it can generate in this way.

It is obvious from this description that the new local algorithm is of the desired time complexity. Thus we only have to show that it generates an optimal local covering. In view of Lemma 5.1 it is sufficient to show that every optimal local covering OPT of $\bar{n}$ given points in a $2 \cdot D$ interval J contains four rings G_1, G_2, $\tilde{G}_1$, $\tilde{G}_2$ such that those of the $\bar{n}$ points that are not covered by these four rings lie between the two intervals of G_1 and between the two intervals of G_2. We first note that without loss of generality OPT contains at most one ring whose right end is left of the center m of the considered $2 \cdot D$ interval J (if there are several such rings "flip over" all except the rightmost one; it is easy to see that all $\bar{n}$ points remain covered). Analogously we can assume that there is in OPT at most one ring whose left endpoint lies to the right of the center m.

Let G_1 be the leftmost ring in OPT whose right end is $\geq m$ and let G_2 be the rightmost ring in OPT whose left end is $\leq m$. Let $\tilde{G}_1$ ($\tilde{G}_2$) be the unique ring left of G_1 (ring of G_2), if it exists. Assume that γ is one of the $\bar{n}$ given points in interval J that is not covered by G_1, G_2, $\tilde{G}_1$, $\tilde{G}_2$. If γ lies left of G_1, then γ is covered in OPT by a ring whose left endpoint lies left of that of G_1. By assumption this ring can only be $\tilde{G}_1$. The case where γ lies to the right of G_2 is handled analogously. Finally, since by construction there is no"gap" between the right end of G_1 and the left end of G_2, γ can only lie between the two intervals of G_1 or between the two intervals of G_2. The following lemma describes in detail phase 2 of the preceding algorithm. We write S for the smallest interval that contains all points which lie between the two intervals of G_1 and to the left of G_2, T is the smallest interval that contains all points which lie between the two intervals of G_2.

LEMMA 5.1. *Let the ring size be fixed at $\langle r, w \rangle$. Let S and T be two disjoint intervals on the line, both of length $\leq 2r$, with S lying left of T and the distance between S and T bigger than w. Let ST be a set of $\bar{n}$ points in $S \cup T$. Then the following Algorithm* F *computes a minimal cover of ST by rings of size $\langle r, w \rangle$ in $O(\min r/w, \bar{n})$ steps.*

ALGORITHM F. Always place the next ring in the leftmost of the following two positions:

(i) where the left end of its left interval coincides with the leftmost uncovered point of ST in S;

(ii) where the left end of its right interval coincides with the leftmost uncovered point of ST in T.

Proof. We show by induction on the cardinality $\bar{n}$ of ST that Algorithm F computes a minimal cover. Let OPT be an optimal covering of ST by rings of size $\langle r, w \rangle$. Our assumption on S and T implies that if a ring in OPT covers some point in T (S) with its left (right) interval, this interval does not cover in addition a point in S (T). Further, its other interval covers no point of ST. Thus we can "flip over" this other interval of the ring to the opposite side without changing the cardinality of the covering OPT. Therefore we can assume w.l.o.g. that the covering OPT is "normal," i.e., every ring in OPT covers with its left interval only points of S and with its right interval only points of T.

Let R_0 be the leftmost ring in the covering OPT and let R be the first ring that is placed by algorithm F. We want to show that R covers every point of ST that is covered by R_0. If R_0 were positioned strictly to the right of R, then OPT would not cover ST (using the assumption that OPT is normal).

Thus we only have to consider the case where R_0 is positioned left of R and R_0 covers some point $p \in N$ that is not covered by R. If $p \in S$ then R_0 covers p with its left interval (since OPT is normal). Therefore p lies left of the left interval of R, which contradicts the definition of R. If $p \in T$ we arrive at an analogous contradiction. Therefore in any case the set $S\tilde{T}$ of points in ST that are not covered by R is covered by the $|\text{OPT}| - 1$ rings of $\text{OPT} - \{R_0\}$.

Together with the induction hypothesis this implies that algorithm F covers $S\tilde{T}$ with $\leq |\text{OPT}| - 1$ rings. Therefore algorithm F covers ST with $|\text{OPT}|$ rings.

For the time analysis we assume that the given $\bar{n}$ points have been preprocessed as in Lemma 3.1 so that the placement of each ring requires only a constant number of steps (the preprocessing itself requires only $O(\bar{n} \log \bar{n})$ steps, thus it does not affect the time bound of the constructed $\frac{1}{2}$-approximation algorithm). Q.E.D.

Remark 5.2. We refer to Hochbaum and Maass [4] for a more efficient version of the previously described $\frac{1}{2}$-approximation algorithm that runs in time $O(\min(n^3, n^2 \cdot r/w))$. This saving of a factor of n^2 can be achieved by noting that in phase 1 of the described algorithm for each considered positioning of the rings G_1, G_2 there are only constantly many possibilities (instead of $\bar{n}^2$ many) that have to be considered for the positioning of the rings $\tilde{G}_1, \tilde{G}_2$. However, the analysis of these constantly many cases is quite cumbersome because the optimal positioning of $\tilde{G}_1, \tilde{G}_2$ interacts with the optimal positioning of the remaining rings that are neither to the left of G_1 nor to the right of G_2.

Remark 5.3. The preceding efficient $\frac{1}{2}$-approximation algorithm makes use of the special structure of optimal coverings in a $2D$-interval. It is also likely that for natural numbers $l > 2$ optimal coverings in $l \cdot D$ intervals have similar properties that can be exploited for designing efficient optimal algorithms for covering $\bar{n}$ points in a $l \cdot D$ interval. This would provide for $l > 2$ efficient $(1/l)$-approximation algorithms for the global ring cover problem. Also, one could design fast approximation algorithms for the local covering problem in order to get efficient approximation algorithms for the (global) ring cover problem. We suggest these approaches as topics for further research.

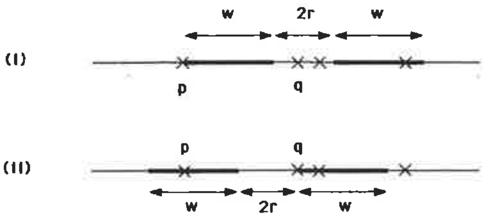

FIG. 6. Two possible positions of a ring.

6. An Efficient ε-Approximation Scheme for $r/w \leq 1/2$

For rings with $w \geq 2r$, and only for these rings, there is an optimal covering OPT of the given points which has the "right justified" property. That is, each ring is placed in a rightmost position that still covers the leftmost point, p, yet uncovered by the rings on its left or it is placed in the rightmost position that covers point p while leaving no points uncovered in the $2r$ inner disk of the ring. In the latter case the leftmost uncovered point following the positioning of this ring is right to its rightmost border. More formally, a covering OPT is said to have *property* $(*)$ if for every ring R in OPT:

(i) *either* R is positioned with its left end at the leftmost point p that is not covered by another ring of OPT that is left of R;

(ii) *or* the left endpoint of the right interval of R covers a point q to the right of p and all points left of q are already covered by R or rings left of R (see Maass [11, Corollary 4.2]).

The positioning of a ring according to case (i) (Fig. 6) will generally leave some uncovered points in the inner disk of length $2r$. Precisely one more ring will have to be placed in order to cover those points (since $2r \leq w$). Case (ii) will offer an advantage only if following the placement of a ring R_1 according to case (ii), the gap of length $2r$ between the two intervals—the inner disk of R_1—does not contain any uncovered point. If otherwise is the case then an additional ring R_2 needs to be placed to cover the uncovered points in the inner disk of length $2r$ which together with ring R_1 cover an uninterrupted interval of points up to its rightmost border. But then we could have been better off using case (i) for R as this will only cover additional points with the next ring R_2 compared to the positioning of those two rings with R_1 according to case (ii).

The following quick ε-approximation scheme for $r/w \leq 1/2$ makes use of property $(*)$.

THEOREM 6.1. For $r/w \leq 1/2$ there is an ε-approximation scheme that covers n given points on the line in $O(\lceil 1/\varepsilon \rceil^2 2^{2\lceil 1/\varepsilon \rceil} \cdot n \cdot \log n)$ steps with error ε, for any given $\varepsilon > 0$.

Proof. Set $l := \lceil 1/\varepsilon \rceil$. By the shifting lemma (Lemma 2.1) it is sufficient to give an algorithm that computes for m given points that lie in an interval of length $l \cdot D$ in $O(l \cdot 2^{2 \cdot l} \cdot \log m)$ steps an optimal covering. (Recall that in computing a cover for a given shift S there are no more than n such non-empty $l \cdot D$-intervals that require covering. This implies at most n applications of the $O(l \cdot 2^{2l} \cdot \log m)$ local algorithm).

We first note that we need at most $2l$ rings to cover compactly all the intervals of length $l \cdot D$ (actually $\lceil l \cdot \frac{3}{4} \rceil \cdot 2$ rings are sufficient since each interval of length $\frac{4}{3} \cdot D$ can be covered completely by two rings). The algorithm attempts to place $2 \cdot l$ rings in left-to-right order following this scheme, but each time, before placing a new ring to cover the leftmost uncovered point, it guesses which case, (i) or (ii) applies. There are $2^{2 \cdot l}$ ways to guess a cover according to this scheme (though not all are feasible): hence at most $2^{2 \cdot l}$ attempts will be made.

For each guess we position successively rings according to this guess. In case (i) this is trivially done. In case (ii) we recall that the only advantage offered by a positioning according to (ii) is when given the leftmost uncovered point p, there is a gap or an open interval (i, j) of length $\geq 2r$ to the right of p with its left endpoint i at a distance $\leq w$ from p (see Fig. 7). This allows p and i to be covered together by the left interval of the currently positioned ring while leaving no uncovered points in the inner disk of that ring. Among all such rings we shall choose the rightmost one. (Note that i could be equal to p.) The algorithmic implementation of this idea is straightforward: Among all gaps (i, j) of length $\geq 2r$ we choose the rightmost such gap with the property that the distance of i from p, d_{ip}, is $\leq w$ (see Fig. 8). We then inspect the distance d_{jp}; if it is $\leq w + 2r$

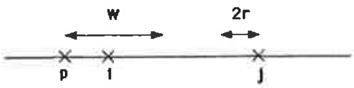

FIGURE 7

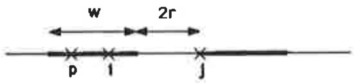

FIGURE 8

then we place the ring with the leftend border of its right interval coinciding with j.

If such gap does not exist or if the distance $d_{jp} > w + 2r$ then we place the ring with its leftmost border on point p which is consistent with case (i). In this event we may discard the current guess as inapplicable and attempt another guess.

The procedure described calls for finding the farthest (rightmost) gap from p with its left end at a distance $\leq w$ from p, and for finding the next (closest) uncovered point. This can be done using binary search (or even a refined form of such search on which we shall not elaborate). Hence we need $O(\log m)$ steps to compute the exact position of one such ring. We assume here that suitable data structures are used for the positions of the yet uncovered points and for the gaps of length $\geq 2r$ between such points. The run time for each guess is hence no more than $O((2l) \cdot \log m)$ and the desired result follows immediately. Q.E.D.

7. Summary

In this paper we derive various new results for the ring cover problem by exploiting the so-called shifting technique. The applicability of this technique can be generalized well beyond the ring cover problem. The technique applies, for instance, to a variety of geometric covering and packing problems with convex objects in any Euclidean space [2]. Such problems include covering points in a d-dimensional Euclidean space by balls or cubes or packing with such objects. In fact, this technique applies to *any* covering and packing problem with convex objects; however, the irregularity of the object—the ratio between the maximum and minimum diameter—affects the running time of the polynomial approximation scheme. Such applications of the shifting technique to strongly NP-complete problems that arise in image processing and VLSI are described in [2].

The ring cover problem has been studied in this paper as a paradigm of covering with nonconvex objects. There are numerous extensions and generalizations of this problem that are still open. In the one-dimensional case the ring problem can be viewed as covering with pairs of intervals. This problem is a special case of covering with objects that consist of $m \geq 2$ indentical intervals. Another generalization is to the case where the pair (or multiple) intervals of each nonconvex object are not of equal length.

It seems that our approach could be useful beyond the realm of geometric problems. Scheduling with intermittent schedules to cover all tasks is a nongeometric problem for which our technique may prove applicable. As all the problems mentioned so far are strongly NP-complete, the existence of a fast approximation scheme that solves such a problem is of particular interest.

Finally, there is another wide family of covering problems where the covering objects are not identical, and each has a cost associated with its use in the cover. The objective is to find a minimum cost cover. Such problems appear also in the context of the location–allocation problem, and none of them have yet been resolved in terms of polynomial schemes.

References

1. B. S. Baker, Approximation algorithms for *NP*-complete problems on planar graphs, in "Proceedings, 24th Symposium FOCS, November 1983."
2. M. R. Garey and D. S. Johnson, "*Computer and Interactability: A Guide to the Theory of NP-Completeness*," Freeman, San Francisco, 1978.
3. D. S. Hochbaum and W. Maass, "Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI," manuscript, University of California, Berkeley, 1983, *J. Assoc. Comput. Mach.*, **32**, No. 1 (1985), 130–136.
4. D. S. Hochbaum and W. Maass, "Fast Approximation Algorithms for the Robot Placement Problem," U.C. Berkeley Business School Working Papers Series MS-30, University of California, Berkeley, May 1983.
5. D. S. Hochbaum and D. B. Shmoys, Using dual approximation algorithms for scheduling problems: Theoretical and practical results, *J. Assoc. Comput. Mach.*, in press.
6. D. S. Hochbaum and D. B. Shmoys, "A Polynomial Approximation Scheme for Machine Scheduling on Uniform Processors: Using the Dual Approximation Approach," U.C. Berkeley Manuscript, May 1986.
7. O. H. Ibarra and C. E. Kim, Approximation algorithms for certain scheduling problems, *Math. Oper. Res.* **3**, No. 3 (1978), 197–204.
8. D. S. Johnson, The *NP*-completeness column: An ongoing guide, *J. Algorithms* **3** (1982), 182–195.
9. R. M. Karp and N. Karmarkar, An efficient approximation scheme for the one-dimensional bin packing problem, *in* "Proceedings, 23rd Symposium FOCS, November 1982.
10. R. J. Lipton and R. E. Tarjan, Applications of a planar separator theorem, *in* "Proceedings, 18th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Long Beach, California, 1977, pp. 162–170.
11. W. Maass, On the complexity of nonconvex covering, *SIAM J. Comput.*, in press.
12. S. Micali and V. Vazirani, An $O(\sqrt{|V|}\,|E|)$ Algorithm for finding maximum matching in general graphs, in "Proceedings, 21st Annual Symposium on the Foundations of Computer Science, IEEE, 1980," pp. 17–27.
13. Robots are coming to industry's service, *The Economist*, August 29, 1981.