# KILLING TWO BIRDS WITH ONE STONE

Evangelos Kranakis, Danny Krizanc,
Anil Maheshwari, Jörg-Rüdiger Sack,
Jorge Urrutia

TR-239 APRIL 1994

School of Computer Science, Carleton University
Ottawa, Canada, KIS 5B6

# KILLING TWO BIRDS WITH ONE STONE
(Preliminary version)

Evangelos Kranakis*†
(kranakis@scs.carleton.ca)
Danny Krizanc*†
(krizanc@scs.carleton.ca)
Anil Maheshwari ‡†
(anil@mpi-sb.mpg.de)
Jörg-Rüdiger Sack*†
(sack@scs.carleton.ca)
Jorge Urrutia§†
(urrutia@csi.uottawa.ca)

**Abstract**

We consider geometric and graph-theoretic applications of the well-known paradigm "killing two birds with one stone". In the plane, this gives rise to a stage graph as follows: vertices are the points, and $\{u, v\}$ is an edge if and only if the (infinite, straight) line segment joining $u$ to $v$ intersects the stage. We give algorithms for "optimal" ray shootings of a collection of points in the plane from a given stage, including a characterization of stage graphs in time $O(n^2)$, and show how to recognize all dominances in $O(\log n)$ time using $O(n + k/\log n)$ processors on the EREW PRAM, where $k$ is the total number of dominances. Similar problems occur when we have a fixed number $k$ of stages on the plane. In this case, $\{u, v\}$ is an edge if and only if the (straight) line segment $uv$ intersects one of the $k$ stages. We consider the problem of constructing stage representations of arbitrary graphs and give upper and lower bounds on the number of stages thus required. We also study the 3-dimensional case. This gives rise to 3 ray shooting graphs; these are formed from a set of points in $\mathbb{R}^3$ and a "stage" which is a compact plane convex set. 3 ray shooting graphs are shown to be comparability graphs, but the converse is shown not to be true. We also provide a characterization of such graphs in terms of

*Carleton University, School of Computer Science, Ottawa, ON, Canada
†Research supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.
‡Max-Planck Institut für Informatik, im Stadtwald, D-66123 Saarbrücken, Germany
§University of Ottawa, Department of Computer Science, Ottawa, ON, Canada

geometric containment orders and show that the recognition problem for triangles is NP-complete.

**1980 Mathematics Subject Classification:** 68R10, 68U05
**CR Categories:** F.2.2
**Key Words and Phrases:** Algorithms and Data Structures, Dominance, Girth, Matching, Ray shooting, 2-dimensional Partial Orders, Stage Graphs.
**Carleton University, School of Computer Science:** SCS-TR-239
**Note:** This paper will be submitted for publication elsewhere.

# 1 Introduction

Suppose we have a flock of birds and wish to kill all birds by throwing stones. As the saying goes, we might be able to kill two birds with one stone. (This is possible, in case our location and the locations of the two birds are colinear.) Birds are stationary and our positions may be delimited by certain areas of the plane or the space, possible disconnected due to the presence of e.g., lakes. The objective is to find shooting positions that will minimize the total number of stones thrown. We wil investigate the planar case as well as the 3-dimensional case.

We study this new class of geometric problems, provide characterizations, present algorithms, and show how the new problem class relates to well studied problems. Using the above paradigm we show how can such problems be solved efficiently and simply, and present generalizations in several directions. A number of interesting open problems are also posed.

For the planar case, we first study the problem class in which all shooting positions lie on a line segment. This problem class has important applications and leads to an interesting characterization theorem. More, formally, 2-dimensional ray-shooting problem is defined as follows.

Consider a line segment $L$, called the stage, contained in the $x$-axis of the plane and a set of points $X$ with positive $y$-coordinates. We assume that all points are distinct and that no three points are colinear. The task is to shoot rays from the stage to each of the points of $X$ in such a way that the total number of rays needed to reach all points is minimized. (When a ray hits a point it continues.)

To solve this problem we restate it as a graph theoretic problem. We define a graph $G(X, L)$ with vertex set $X$ in which two vertices are adjacent if the (infinite) line connecting them intersects $L$ (see Figure 1). $G(X, L)$ is called a plane stage ray-shooting graph with one stage, or simply a stage graph.

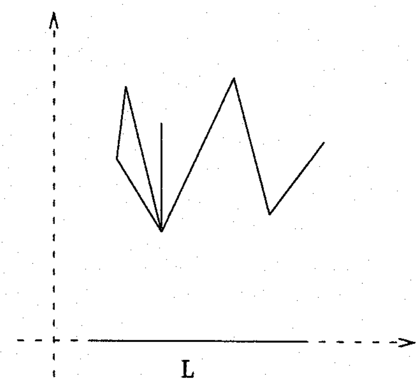


Figure 1: A graph and its stage representation.

## 1.1 Results of the paper

Minimizing the number of rays is equivalent to solving a matching problem in stage graphs and we could apply the algorithm of Micali and Vazirani [21] (see also [16, 1]) for finding a maximum cardinality matching in arbitrary graphs, which solves our problem. The approach taken here is different. We first characterize the structure of stage graphs and subsequently exploit the structural knowledge for the design of a variety of efficient algorithms.

Our characterization theorem states that a graph $G$ is a stage graph if and only if $G$ is the comparability graph of an order of dimension 2. Comparability graphs are undirected graphs $G$ which can be oriented in such a way that if we have $u \rightarrow v$ and $v \rightarrow w$ in $G$ then $u \rightarrow w$ is also an edge of $G$.

Comparablity graphs of dimension 2 have received a fair amount of attention in the literature (see [30, 23, 8]) and they also arise e.g., in scheduling. Our characteriztaion theorem implies an $O(\min\{n^2, n + m \log n\})$ algorithm to recognize stage graphs with $n$ vertices and $m$ edges; this is the first result of this paper.

While stage graphs may have a quadratic number of edges they can be compactly represented in linear space making them appealing from a practical point of view. Based on the compact representation we solve two problems which arise naturally in a variety of applications and that are well studied. These applications include: range searching, finding maximal elements and minimal layers, computing a largest area empty rectangle in a point set, determining the longest common sequence between two strings, interval/rectangle intersection problems, etc.

Let $P = \{p_1, p_2, ..., p_n\}$ be a planar point set of $n$ distinct points $p_i = (x_i, y_i)$,

$i = 1, ..., n$. A point $p_i$ is said to *dominate* a point $p_j$, if $x_i \geq x_j$ and $y_i \geq y_j$ and $i \neq j$. Preparata and Shamos [24] presented optimal sequential algorithms for counting and reporting the dominances for each point of the set $P$ and running in $O(n \log n)$ and $O(n \log n + k)$ time, respectively, where $k$ is the total number of of dominance pairs. In the reporting mode of the problem, all dominance pairs are to be enumerated. Goodrich [9] solved this problem in $O(\log n)$ time using $O(n + k/\log n)$ CREW PRAM processors, where $k$ is the total number of dominance pairs. We present a simple optimal parallel algorithm for this problem; it runs in $O(\log n)$ time using $O(n+k/\log n)$ EREW PRAM processors.

A problem related to dominances is the *rectangle query problem* for planar point sets $P$. A query consists of a pair of points $(p_i, p_j)$, where $p_i, p_j \in P$, and we need to answer whether the rectangle formed by the query points is empty or not. Such rectangle queries find application e.g., in data bases. Given $O(n^2)$ space, queries can easily be answered in $O(1)$ time. The space can be reduced to $O(n \log n)$ using range search data structures [24], unfortunately the query time increases to $O(\log n)$. We provide an $O(n \log n)$ size data-structure, where the queries can still be answered in $O(1)$ time. Furthermore, the data-structure is very-simple and can be computed in sequential $O(n \log n)$ time and in parallel $O(\log n)$ time using $O(n)$ EREW PRAM processors, respectively. Our methods for solving both problems is different from the existing methods; we reduce the problems to one dimensional problems using our characterization theorem and some new view of dominance problems via stage graphs.

## 1.2 Generalizations

We also study generalizations of plane 1-stage ray-shooting graphs to plane ray shooting graphs from several stages and to higher-dimensional ray-shooting graphs from convex domains.

In the first generalization we assume that we are given a collection of $n$ points on the plane in general position (i.e. no three co-linear), as well as $k$ fixed but arbitrary finite, closed, non-intersecting, straight line segments (also called stages). We define a graph as follows: Vertices are the given points, and $\{u, v\}$ is an edge if and only if the infinite (straight) line segment $uv$ joinning the point $u$ to $v$ intersects one of the $k$ stages. We say that the above graph is represented (via ray-shooting) by the above $k$ stages. For fixed $n$, let $\mathcal{G}_k$ denote the class of graphs (on $n$ points in general position) which can be represented by $k$ stages as above. These graphs define a hierarchy.

We prove upper and lower bounds on the number of stages needed to represented a graph and establish results on separating the classes $\mathcal{G}_k$. In particular, we prove the existence of graphs which require $\Omega(n/\log n)$ stages for their representation; furthermore, we show how to construct graphs requiring $\Omega(\sqrt{n})$ stages for their representation. We also study the determination of stage numbers for several common graphs, including lines, cycles, trees and complete bipartite.

In the second generalization we consider 3-dimensional ray shooting graphs.

Let $X = \{p_1, \ldots, p_n\}$ be a collection of points in a vector space $\mathbb{E}^3$ such that the $z$-coordinate of each element of $X$ is strictly greater than 0 and $S$ a compact plane-convex set of $\mathbb{E}^3$ contained in the hyperplane $H_0 = \{p \in \mathbb{E}^3 :$ the $z$-coordinate of $p$ is $0\}$. Given $X$ and $S$ we can now construct a graph $G(X, S)$ with vertex set $X$ such that two vertices $p_i, p_j$ of $G(X, S)$ are adjacent if the line through $p_i$ and $p_j$ intersects $S$. $G(X, S)$ will be called the 3 ray shooting graph of $X$ and $S$.

3 ray shooting graphs are shown to be comparability graphs. We prove that the set of 3 ray shooting graphs obtained when $S$ is a triangle is exactly the set of comparability graphs of orders of dimension at most 3, and thus already for 3 dimensions, the recognition problem is NP-complete. We also provide a characterization of 3 ray shooting graphs in terms of geometric containment orders, that is ordered sets arising from containment relations among the elements of families of convex sets on the plane. We prove that when $S$ is a circle, $G(S, X)$ is the comparability graph of a circle order [29]. Finally, we prove that not all comparability graphs are 3 ray shooting graphs.

# 2 Ray shooting from a stage

Consider a line segment $L$ contained in the $x$-axis of the plane and a set of points $X = \{p_1, \ldots, p_n\}$ in general position with positive $y$-coordinates. We define a graph $G(X, L)$ with vertex set $X$ in which two vertices are adjacent if the line connecting them intersects L (see Figure 1). $G(X, L)$ will be called a plane ray shooting graph. In this section we prove that a graph $G$ is a plane ray shooting graph if and only if $G$ is the comparability graph of an order of dimension 2. This yields an $O(n^2)$ algorithm to recognize planar-ray shooting graphs.

## 2.1 Terminology and Definitions

A binary relation $<$ over a set $X$ defines a partial order $P(X, <)$ on $X$ if it satisfies

1. $x < y, y < z$ implies $x < z$ (transitivity), and
2. $x < x$ (antisymmetry).

The partially ordered set $P(X, <)$ is a linear order if it also satisfies $x < y$ or $y < x$, for all distinct $x, y \in X$.

Let $P(X, <)$ be a poset. A realizer of $P$ of size $k + 1$ is a collection of linear orders $\{L_0(X, <_0), L_1(X, <_1), \ldots, L_k(X, <_k)\}$ such that

$$L_0(X, <_0) \cap L_1(X, <_1) \cap \cdots \cap L_k(X, <_k) = P(X, <),$$

where the intersection is defined by $x < y \Leftrightarrow x <_i y$, for all $i$. It can be proved easily that every poset can be obtained as the intersection of a number of linear

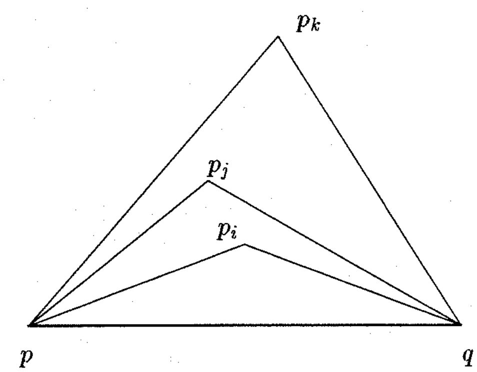


Figure 2: A comparability graph

orders. Dushnik and Miller [6] define the dimension of $P$, denoted $\dim P$, to be the smallest possible size of a realizer of $P$. Such a realizer is called a minimum realizer of $P$.

## 2.2 Ray shooting on the plane

Our aim in this section is to prove the following theorem.

**Theorem 2.1** *A graph $G$ is a plane ray shooting graph if and only if $G$ is the comparability graph of an ordered set of dimension 2.*

PROOF Consider a set $X = \{p_1, \ldots, p_n\}$ of $n$ points on the plane with $y$-coordinates greater than 0 and a line segment $L$ contained in the $x$-axis, with end points $p$ and $q$. Let $G(X, L)$ be the ray shooting graph of $X$ and $L$. We start first by proving that $G(X, L)$ is a comparability graph, i.e. we show that it is possible to orient the edges of $G(X, L)$ such that if $p_i \rightarrow p_j$ and $p_j \rightarrow p_k$ then $p_i \rightarrow p_k$. To this end, let us assume that two vertices $p_i$ and $p_j$ of $G(X, L)$ are adjacent, i.e. the line through $p_i$ and $p_j$ intersects $L$. We orient the edge $\{p_i, p_j\}$ of $G(X, L)$, $p_i \rightarrow p_j$ if the $y$-coordinate of $p_i$ is smaller than that of $p_j$, otherwise we orient $p_j \rightarrow p_i$. We now prove that the orientation thus obtained in $G(X, L)$ is transitive. Observe that $p_j \rightarrow p_i$ if and only if the triangle $\Delta(p_i, p, q)$ defined by $p_i$ and the end points $p$ and $q$ of $L$ is contained in the triangle $\Delta(p_j, p, q)$ defined by $p_j$ and $p$ and $q$. Thus if $p_i \rightarrow p_j$ and $p_j \rightarrow p_k$ then $\Delta(p_k, p, q) \supset \Delta(p_j, p, q) \supset \Delta(p_i, p, q)$ and thus $p_i \rightarrow p_k$ (see Figure 2). This orientation of $G(X, L)$ defines a partial order $P(X, <)$ on $X$ in which $p_i < p_j$ if $p_i \rightarrow p_j$.

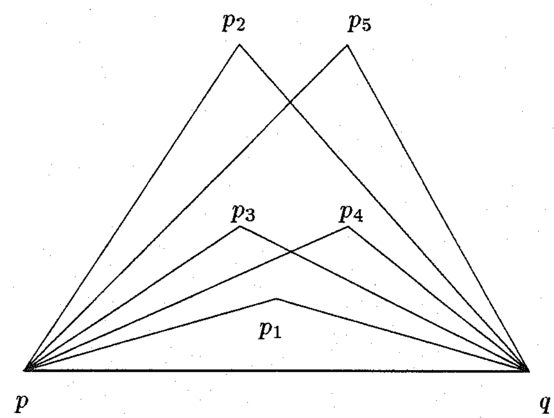


Figure 3: The orders $L_1(X, <_1)$ and $L_2(X, <_2)$

We now show that $P(X, <)$ has dimension 2. To prove this we will produce two linear extensions $L_1(X, <_1)$ and $L_2(X, <_2)$ of $P(X, <)$ such that $L_1(X, <_1) \cap L_2(X, <_2) = P(X, <)$. To produce $L_1(X, <_1)$ sort the points of $X$ in the counterclockwise direction with respect to $p$, i.e. $p_i <_1 p_j$ if the slope of the line joining $p_i$ to $p$ is smaller than the slope of the line joining $p_j$ to $p$. In $L_2(X, <_2)$ we now define $p_i <_2 p_j$ if the slope of the line joining $p_i$ to $p$ is greater than the slope of the line joining $p_j$ to $q$ (see Figure 3, where $L_1(X, <_1) = \{p_1 <_1 p_4 <_1 p_3 <_1 p_5 <_1 p_2\}$ and $L_2(X, <_2) = \{p_1 <_2 p_3 <_2 p_4 <_2 p_2 <_2 p_5\}$.) It now follows that $P(X, <) = L_1(X, <_1) \cap L_2(X, <_2)$.

Conversely, let $P(X, <)$ be an ordered set of dimension 2 and $L_1(X, <_1), L_2(X, <_2)$ be two total orders on $X$ such that $P(X, <) = L_1(X, <_1) \cap L_2(X, <_2)$. Choose two points $p, q$ on the $x$-axis as depicted in Figure 4. Let $p_i$ be an element of $X$. Let $r(i)$ and $s(i)$ be the ranks of $p_i$ in $L_1(X, <_1)$ and $L_2(X, <_2)$, respectively. Consider a set $\{\lambda_1, \ldots, \lambda_n\}$ of $n$ lines through $p$ sorted in increasing order according to their slopes and a set $\{\beta_1, \ldots, \beta_n\}$ of $n$ lines through $q$ sorted in decreasing order according to their slopes such that each $\lambda_i$ intersects each $\beta_j$ at a point with positive $y$-coordinate, $1 \leq i, j \leq n$. Let us label with $p_i$ the point at which $\lambda_{r(i)}$ and $\beta_{s(i)}$ intersect and identify the points of $X$ with $p_1, \ldots, p_n$ (see Figure 3, where $L_1(X, < 1) = \{p_2 <_1 p_4 <_1 p_3 <_1 p_1 <_1 p_5\}$ and $L_2(X, <) = \{p_3 <_2 p_1 <_2 p_5 <_2 p_2 <_2 p_4\}$.) It is now easy to see that the set $X$ of points on the plane labeled $p_1, \ldots, p_n$ and the line segment $L$ are such that $G(X, L)$ is the ray shooting graph of $P(X, <)$. ∎

**Corollary 2.1** *Recognizing 2-ray shooting graphs can be done in $O(n^2)$ time.*

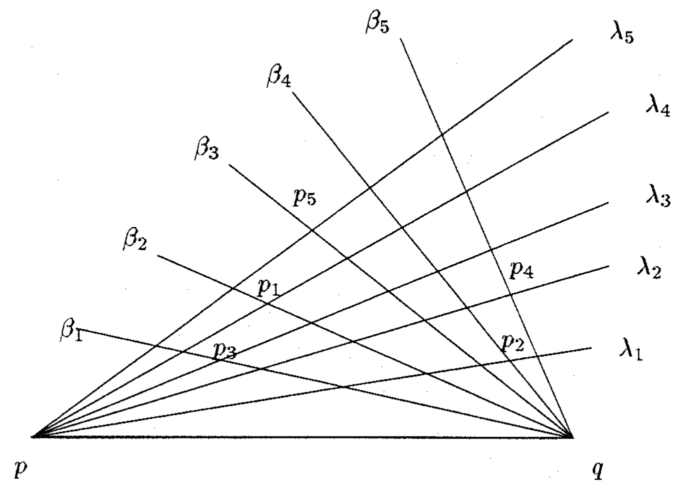


Figure 4: The orders $L_1, L_2$

PROOF Recognition of orders of dimension two can be done in $O(n^2)$ time [30]. ■

In fact, our characteriztaion theorem implies an $O(\min\{n^2, n + m \log n\})$ algorithm to recognize stage graphs with $n$ vertices and $m$ edges.

Now it is possible to solve the original stage shooting problem using the Micali and Vazirani algorithm [21]. However, a more efficient algorithm which is more appropriate to the case of comparability graphs is currently under investigation.

# 3 Dominance Problems

In this section we present algorithms for dominance problems on a planar point set. These problems arise naturally in a variety of applications and they are directly related to well studied geometric and non-geometric problems. These problems include: range searching, finding maximal elements and minimal layers, computing a largest area empty rectangle in a point set, determining the longest common sequence between two strings, and interval/rectangle intersection problems, etc.

## 3.1 Related results

Direct dominance problems have been studied for the CREW-PRAM (details for this model can be found in [12, 13]). The two-set dominance counting problem was solved by Atallah et al. [2] in optimal $O(\log n)$ time using $O(n)$ processors, where $n$ is the total number of points in the given sets. In this problem, given two point sets $A$ and $B$, all pairs $(a, b)$ are to be counted where $a \in A$ dominates $b \in B$. In the reporting mode of the problem, all dominance pairs are to be enumerated. Goodrich [9] solved this problem in $O(\log n)$ time using $O(n/\log n + k)$ CREW PRAM processors, where $k$ is the total number of dominance pairs.

## 3.2 Complexity of the dominance problem

Let $P = \{p_1, p_2, ..., p_n\}$ be a planar point set of $n$ distinct points $p_i = (x_i, y_i)$, $i = 1, ..., n$. A point $p_i$ is said to *dominate* a point $p_j$, if $x_i \geq x_j$ and $y_i \geq y_j$ and $i \neq j$. The *dominance problem* is to enumerate all dominances of a given point set. Preparata and Shamos [24] presented an optimal sequential algorithm for this problem and it runs in $O(n \log n + k)$ time, where $k$ is the total number of dominance pairs. Goodrich [9] presented a parallel algorithm for this problem; it runs in $O(\log n)$ time using $O(n + k/\log n)$ CREW PRAM processors. In this section we present a simple optimal parallel algorithm for this problem; it runs in $O(\log n)$ time using $O(n + k/\log n)$ EREW PRAM processors. For details on the parallel model of computation, see [9, 12].

The other problem we study in this section is a *rectangle query problem* for a planar point set $P$. A query consists of a pair of points $(p_i, p_j)$, where $p_i, p_j \in P$, and we need to answer whether the rectangle formed by the query points is empty or not. Given $O(n^2)$ space, we can answer such queries in $O(1)$ time, since we can precompute information about each pair of points in $P$. The space can be reduced to $O(n \log n)$ using range search data structures [24], but unfortunately the query time increases to $O(\log n)$. We provide an $O(n \log n)$ size data-structure, where the queries can still be answered in $O(1)$ time. Furthermore, the data-structure is very-simple and can be computed in sequential $O(n \log n)$ time and in parallel $O(\log n)$ time using $O(n)$ EREW PRAM processors, respectively.

Our method for solving both the problems is different from the existing methods; we reduce the problems to one dimensional problems. This is achieved by ordering the points with respect to $x$-coordinate and then redefining these problems with respect to the corresponding permutation on $y$-axis.

## 3.3 Reporting Dominances

In this subsection, we provide algorithms for reporting dominances of a planar point set $P$. Without loss of generality assume that the points of the set

$P = \{p_1, p_2, ..., p_n\}$, where $p_i = (x_i, y_i)$, $i = 1, ..., n$, are sorted with respect to increasing $x$-coordinate. Therefore, we relabel each point $p_i$ by its index $i$. From now on, we refer to a point $p_i$ by its index $i$. Let $Y$ be the array consisting of labels of points in $P$, sorted with respect to increasing $y$-coordinate; i.e. $Y$ is a permutation of $\{1, ..., n\}$. Let $i$ appear at the position $pos_i$, where $1 \leq pos_i \leq n$, in $Y$. From the above definitions it follows that a point $i$ dominates a point $j \in P$ if and only if $i > j$ and $pos_i > pos_j$. Hence the points dominated by $i$ are the elements of the subarray $Y[1, ..., pos_i]$ which are less than $i$. So the dominance problem reduces to that of reporting all elements of subarray $Y[1, ..., i-1]$ which are less than $Y[i]$, for all $i \in \{2, ..., n\}$.

We provide first a sequential algorithm for the above problem and then show that it can be easily parallelized. The sequential algorithm is based on the merge sort algorithm; it runs in $O(n \log n + k)$ time using linear space, where $k$ is the total number of dominance relations in the given point set $P$.

The sequential algorithm has $O(\log n)$ merge stages. In order to simplify notation, we present the last merge stage. Assume that we know all dominances for each point within subarrays $Y[1, ..., n/2]$ and $Y[n/2+1, ..., n]$. We wish to compute dominances for each point in $Y[1, ..., n]$. Observe that we need only compute the points dominated by $Y[n/2+1, ..., n]$ in $Y[1, ..., n/2]$, since no point in $Y[1, ..., n/2]$ dominates any point in $Y[n/2+1, ..., n]$. The dominances are computed as follows. First note that the arrays $Y[1, ..., n/2]$ and $Y[n/2+1, ..., n]$ have already been sorted in increasing order during the recursion. Now rank each element of $Y[n/2+1, ..., n]$ in $Y[1, ..., n/2]$. Suppose an element $Y[i]$, where $n/2+1 \leq i \leq n$, is ranked at the position $j$ $(1 \leq j \leq n/2)$ in $Y[1, ..., n/2]$, the points dominated by $Y[i]$ in $Y[1, ..., n/2]$ are $Y[1]$, $Y[2]$, ..., $Y[j]$. After cross ranking, we can report dominances in time proportional to the number of dominance pairs. We summarize the result in the following theorem.

**Theorem 3.1** *All dominances of an $n$-point planar set can be computed in $O(n \log n + k)$ time using $O(n)$ space, where $k$ is the total number of dominance pairs.*

PROOF The correctness of the algorithm is straightforward. Now we analyze its complexity. The merge-sort algorithm takes $O(n \log n)$ time using $O(n)$ space. During each stage in merging, the ranking of the sub-arrays can be achieved in linear time with respect to their sizes. Since in each stage we report a set of new dominance pairs, overall time complexity of the algorithm follows. In order to perform the $i+1$st stage of merge-sort, we need only the computation of the $i$th stage; thus the algorithm requires only linear space. ■

Now we parallelize the above algorithm by using the results of [4, 15]. The parallel-merge sort algorithm of [4] cross-ranks elements of each subarray during each stage of merging. As observed above, after cross ranking, the problem reduces to that of reporting subarrays $Y[1, ..., j]$ for an appropriate $j$, where $1 \leq j \leq n/2$, for each $Y[i]$, where $n/2+1 \leq i \leq n$. Subarrays can be optimally reported on EREW PRAM by the algorithm of [15]. We summarize the result in the following theorem.

**Theorem 3.2** *All dominances of an $n$-point planar set can be computed in $O(\log n)$ time using $O(n + k/\log n)$ processors on the EREW PRAM, where $k$ is the total number of dominances.*

Proof The correctness of the algorithm is straightforward. We analyze the complexity of the algorithm. Parallel merge sort requires $O(\log n)$ time using $O(n)$ processors on the EREW PRAM [4]. Further, it also cross ranks subarrays in each step. Using this information, the value of $k$ can be computed in $O(\log n)$ time using $O(n)$ processors. Allocate $O(n + k/\log n)$ processors to report all dominances. We also need to store the sorted sub-arrays at each intermediate stage in merge-sort. Using the algorithm of [15], the required subarrays can be reported in $O(\log n)$ time using using $O(n + k/\log n)$ processors [15]. Hence, it follows, that all dominances can be reported in $O(\log n)$ time using $O(n + k/\log n)$ EREW PRAM processors. ■

## 3.4 Rectangle Query Problem

In this subsection we address the rectangle query problem. Given an $n$-point planar set $P$, the queries are of the form $(p_i, p_j)$, where $p_i, p_j \in P$, and we need to output, whether or not the rectangle formed by $p_i$ and $p_j$ contains a point of $P$ in its interior. We provide sequential and parallel algorithms to compute an $O(n \log n)$ size data-structure, such that the queries can be answered in $O(1)$ time.

As in the previous subsection, we assume that the points of the set $P = \{p_1, p_2, ..., p_n\}$, where $p_i = (x_i, y_i)$, $i = 1, ..., n$, are sorted with respect to increasing $x$-coordinate. Therefore, we relabel each point $p_i$ by its index $i$. We refer to a point $p_i$ by its index $i$. Notice that our queries are of type $(i, j)$, where $1 \leq i, j \leq n$. Furthermore, we can assume that $i < j$, otherwise we can interchange $i$ and $j$.

We compute two data-structures, the first one answers the queries where $y_i \leq y_j$, and the other one answers the queries where $y_i > y_j$. Since the procedure for computing both data-structures and answering the corresponding queries is analogous, we only discuss the computation of data-structure which handles the queries where $y_i \leq y_j$.

Let $Y$ be the array corresponding to the labels of points in $P$ sorted with respect to *increasing* $y$-coordinate. Let $(i, j)$ be a query pair, where $i < j$ and $y_i \leq y_j$. Let $i$ appear at the position $pos_i$ in $Y$, where $1 \leq pos_i \leq n$. The following lemma enables us to reduce our problem of detecting whether a rectangle is empty or not to a one dimensional problem on $Y$.

**Lemma 3.1** *The rectangle formed by $(p_i, p_j)$ is empty if and only if there does not exist any element $Y[k]$, such that $i \leq Y[k] \leq j$, where $pos_i < k < pos_j$.*

PROOF Follows from the definition of the array $Y$. ■

In the following we first state a sequential algorithm to compute a data-structure, which can answer the existence of $Y[k]$ between $(pos_i, pos_j)$ as stated in the above lemma, and then show how the queries can be answered. Further, we show that the algorithm for computing the data-structure can be easily parallelized.

Before stating our algorithm, we simplify notation by restating the problem. Our aim is to preprocess the array $Y$ (assume $n = 2^l$) such that, given any two indices $a$ and $b$, where $1 \leq a < b \leq n$, we can determine whether there exist an element in the subarray $\{Y[a+1], ..., Y[b-1]\}$, which is between $Y[a]$ and $Y[b]$ in $O(1)$ sequential time. Intuitively, it seems that we need to precompute this information for some subarrays, and then given a query array, the relevant information should be deduced from a constant number of pre-computed sub-arrays. We achieve our goal by constructing a complete binary tree $T$ on the elements of $Y$ such that each internal node $u$ of $T$ keeps some information about the array determined by the leaves in the subtree rooted at $u$. In the following, we precisely state the information maintained at each internal node $u$ of $T$.

Let $LCA(a, b)$ denote the lowest common ancestor node of the leaves of $T$ holding $Y[a]$ and $Y[b]$. Given two indices $a$ and $b$, we can determine $LCA(a, b)$, say the node $u$, of $T$ in $O(1)$ sequential time since $T$ is a complete binary tree. If the leaves of the subtree rooted at $u$ correspond exactly to the subarray $\{Y[a], ..., Y[b]\}$, then it is sufficient to store an information at $u$, about the presence or absence of an element between $Y[a]$ and $Y[b]$ in the subarray $\{Y[a+1], ..., Y[b-1]\}$. However, the subarray associated with $u$, denoted by $Y_u$, is typically of the form of $\{Y[l], ..., Y[a], ..., Y[b], ..., Y[r]\}$, where $l \leq a < b \leq r$. Hence the information stored at the node $u$ is not sufficient to answer our query, and some additional information is needed, as described next.

Let $v$ and $w$ be the left and right child of $u$, respectively. Let the subarrays associated with $v$ and $w$, respectively are $Y_v = \{Y[l], ..., Y[a], ..., Y[p]\}$ and $Y_w = \{Y[p+1], ..., Y[b], ..., Y[r]\}$ for some $a \leq p < b$. Notice that the subarrays $Y_v$ and $Y_w$ partition $Y_u$. Let us define two quantities, called *suffix minima* and *prefix maxima*, respectively over the elements of arrays $Y_v$ and $Y_w$, which we need for answering our queries.

For any $\alpha$, where $l \leq \alpha \leq p$, the suffix minima for $\alpha$ in $Y_v$ is defined as follows. Among the elements of the subarray $\{Y[\alpha+1], ..., Y[p]\}$ consider only the set of elements which are larger than $Y[\alpha]$, and call this set as $Suff_\alpha$. If $Suff_\alpha \neq \emptyset$, then the suffix minima for $\alpha$ is the element with the minimum value in $Suff_\alpha$, otherwise suffix minima does not exist for $\alpha$. Similarly we define prefix maxima. For any $\beta$, where $p+1 \leq \beta \leq r$, the prefix maxima for $\beta$ in $Y_w$ is defined as follows. Among the elements of the subarray $\{Y[p+1], ..., Y[\beta-1]\}$, consider only the set of elements which are smaller than $Y[\beta]$, and call this set $Pref_\beta$. If $Pref_\beta \neq \emptyset$, then the prefix maxima for $\beta$ is the element with the maximum value in $Pref_\beta$, otherwise it does not exist for $\beta$.

Let us first analyze the complexity of constructing the whole data structure.

The algorithm constructs a complete binary tree whose leaves are the elements of $Y$ such that each internal node $u$ has associated with it two arrays, suffix minima and prefix maxima arrays. It can be seen that the data-structure needs $O(n \log n)$ space. Now we show that the data-structure can be computed in $O(n \log n)$ time.

We make two copies of array $Y$, and on one copy we perform a merge-sort algorithm. The merge-sort algorithm, computes a complete binary tree $T'$, over $Y$, and at each internal node $u$ of $T'$ it computes a sorted list of elements in the subtree rooted at $u$. Furthermore, if $v$ and $w$ are the left and right child of $u$ in $T'$, then we also cross rank the elements of $v$ and $w$. Also store the sorted list, and the cross ranking information, at each internal node of $T'$. It is easy to see that this can be accomplished in $O(n \log n)$ time and space. Now we work on the other copy of $Y$ to compute the suffix minima and prefix maxima arrays. Consider a node $u$ of $T$, and let $v$ and $w$ be its left and right child, respectively, as mentioned above. Assume that we know the suffix minima and prefix maxima arrays for $v$ and $w$ and we wish to compute these arrays for the node $u$. Notice that the suffix minima and prefix maxima for each element in $u$ can be computed by using the cross ranking information among the elements of $v$ and $w$ in the merge-sort tree $T'$.

It is easy to see that the above data-structure can be computed in $O(n \log n)$ sequential time and in parallel in $O(\log n)$ time using $O(n)$ EREW PRAM processors by using the parallel merge-sort algorithm of [4]. Now we show that the queries can be answered in $O(1)$ time. The following lemma is crucial for the correctness and the complexity.

**Lemma 3.2** *Let $u$ be the lowest common ancestor node corresponding to $a$ and $b$ in $T$, where $a < b$. Let $v$ and $w$ be the left and right child of $u$, respectively. Let the subarrays associated with $v$ and $w$ be*

$$Y_v = \{Y[l], ..., Y[a], ..., Y[p]\} \text{ and } Y_w = \{Y[p+1], ..., Y[b], ..., Y[r]\},$$

*where $a \leq p < b$, respectively. There exists an element between $Y[a]$ and $Y[b]$ in the subarray $\{Y[a+1], ..., Y[b-1]\}$ if and only if either the suffix minima of $Y[a]$ in $Y_v$ is smaller than $Y[b]$, if it exists, or the prefix maxima of $Y[b]$ in $Y_w$ is larger than $Y[a]$, if it exists.*

Proof Follows from the definition of suffix minima and prefix maxima. ■

Let us recall our problem. We are given a set $P$ of points, sorted with respect to $x$-coordinate and labeled accordingly. Our queries are of the form $(p_i, p_j)$, where $p_i, p_j \in P$. We want to report whether the rectangle formed by $p_i$ and $p_j$ is empty or not. We first test whether $i < j$, if not, we interchange $i, j$. We compute two data-structures, one to handle the queries where $y_i < y_j$ and the other one to handle the queries where $y_i > y_j$. Let us concentrate on the queries of the first type. We defined the array $Y$, which was the order of the indices of points of $P$ with respect to increasing $y$-coordinate. We compute a data-structure over $Y$, i.e., a complete binary tree $T$, where nodes of $T$ also contain appropriate suffix minima and prefix maxima arrays. Given a rectangle query $(p_i, p_j)$, where $i < j$ and $y_i < y_j$, we find the position $a = pos_i$ and $b = pos_j$ in $Y$ of $i$ and $j$, respectively. Now determine the lowest common ancestor node of $a$ and $b$, say $u$ in $T$. Locate the position of $Y[a]$ and $Y[b]$ among the children of $u$ in $T$ and then using the suffix minima and prefix maxima informations computed in $T$, answer the query. Since finding the lowest common ancestor in a complete binary tree and locating the appropriate $Y[a]$ and $Y[b]$ requires constant time, the queries can be answered in $O(1)$ time. We summarize the results in the following theorem.

**Theorem 3.3** *A data structure of size $O(n \log n)$ can be computed in $O(n \log n)$ sequential time and in $O(\log n)$ parallel time using $O(n)$ EREW PRAM processors, rectangle queries can be answered in $O(1)$ sequential time.* ■

# 4 Graphs Represented with Many Stages

Suppose we have a collection of $n$ points on the plane in general position (i.e. no three co-linear), as well as $k$ fixed but arbitrary finite, closed, non-intersecting, straight line segments (also called stages). We define a graph as follows: Vertices are the given points, and $\{u, v\}$ is an edge if and only if the infinite (straight) line segment $uv$ joinning the point $u$ to $v$ intersects one of the $k$ stages. We say that the above graph is represented (via ray-shooting) by the above $k$ stages.* For fixed $n$, let $\mathcal{G}_k$ denote the class of graphs (on $n$ points in general position) which can be represented by $k$ stages as above.

It is easy to show that a simple graph with $m$ edges is in the class $\mathcal{G}_m$. To see this, represent the graph by $n$ points in the plane which are in general position. For each edge $\{u, v\}$ of the graph position a stage in such a way that it intersects the line segment $uv$, but such that for no other edge of the graph does the corresponding line segment intersect this stage. In particular, every graph on $n$ vertices can be represented with at most $n(n-1)/2$ stages. It is clear that we have the following containments for graphs on $n$ vertices:

$$\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_{n(n-1)/2}. \tag{1}$$

For any graph $G$ let $\mathrm{st}(G)$ be the miniimum number of stages needed in order to represent the graph $G$ as above.

Any graph $G \in \mathcal{G}_k$ can be writen as the "union" of $k$ graphs $G_i$, $i = 1, \ldots, k = \mathrm{st}(G)$, in the following way. Consider a ray-shooting representation of $G$ with stages $S_1, \ldots, S_k$. $G_i$ has the same vertices as $G$. Moreover, $\{u, v\}$ is an edge of $G_i$ if and only if the infinite line segment $uv$ intersects $S_i$.

*Such graphs are also called plane ray shooting graphs in section 2.

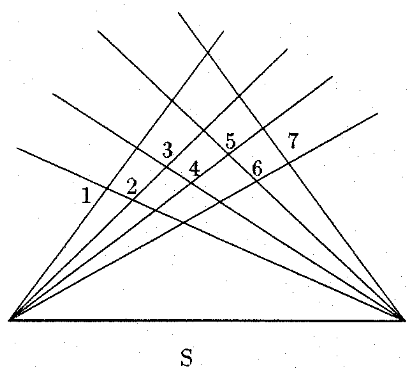


Figure 5: Representing the 7 node line graph with the stage $S$

This gives a "natural" factoring [14] of the given graph $G$ in $\mathrm{st}(G)$ subgraphs each represented with one stage via ray-shooting.

In the sequel we consider the size $\mathrm{st}(G)$ as a function of the number of vertices of the graph. We consider upper and lower bounds on the number of stages needed to represented a graph and establish results on separating the above classes of graphs.

## 4.1 Stage number of certain graphs

In this section we consider the stage number of several simple graphs, including lines, cycles, trees and complete bipartite.

**Theorem 4.1**

1. *The line graph $L_n$ on $n$ vertices can be represented with a single stage. Hence* $\mathrm{st}(L_n) = 1$.
2. *The cycle $C_n$ on $n$ vertices can be represented with a single stage if and only if $n \leq 4$. Moreover,* $\mathrm{st}(C_n) = 2$, *for $n \geq 5$.*
3. *A graph with girth $\geq 5$ requires at least two stages for its representation.*

PROOF (OUTLINE)
(1) The representation of the line graph $L_n$ is depicted in Figure 5. We draw two pencils of lines from the endpoints of the stage $S$. It is easy to check that the $n$ points can be represented with the numbers $1, 2, 3, \ldots, n$ as appropriate intersections of these lines (see Figure 5).

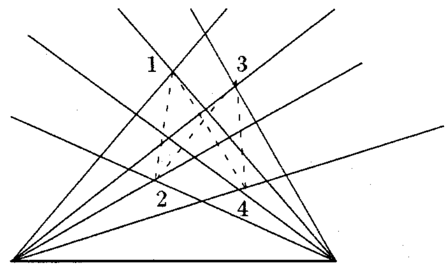


Figure 6: Representing the 4-element cycle

(2) It is easy to represent with one stage the 2- or 3-element cycles. The representation of the 4-element cycle is given in Figure 6. The edges

$$\{1,2\}, \{2,3\}, \{3,4\}, \{4,1\}$$

of the corresponding cycle are represented by the dashed lines.

To represent an $n$-cycle with two 2 stages, we first represent the line graph $L_n$ with one stage and then add an extra stage in order to represent the edge joinning its two endpoints.

The fact that an $n$-cycle with $n \geq 5$ requires at least two stages will be derived from a property about partially ordered sets. Let $(P, <)$ be a strict partially ordered set (finite or infinite). Define a graph on $P$ by joinning $x, y \in P$ if and only if either $x < y$ or $y < x$ [18][9.32]. First we show that no cycle of length $\geq 5$ in the graph $G$ can have an odd number of vertices. To see this we argue as follows. Let a cycle in this graph consist of the points $1, 2, \ldots, n$, for some $n \geq 5$, where $\{i, j\}$ is an edge if and only if either $j = i+1$ or $i = 1, j = n$. Without loss of generality let us assume that $1 < 2$ (a similar argument works if $2 < 1$). Since $1, 3$ are $<$-incomparable we must have that $3 < 2$. Since $2, 4$ are incomparable we must also have that $3 < 4$, and so on. Hence we have the following inequalities

$$\begin{array}{l} 1, 3 < 2 \\ 3, 5 < 4 \\ 5, 7 < 6 \\ \ldots \\ n-2, n < n-1 \end{array} \tag{2}$$

Now, if $1 < n$ then also $1 < n-1$, since $n < n-1$, a contradiction; if $n < 1$ then also $n < 2$, since $1 < 2$, a contradiction.

If we apply this result to the partially ordered set induced via ray-shooting by a single stage on the given points of the plane we derive that the only possible

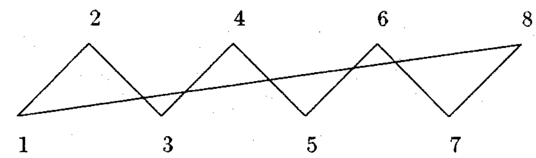


Figure 7: The lattice corresponding to an 8-cycle

cycles must be of length $n$, for some even $n \geq 6$. Let us assume without loss of generality that $1 < 2$ (the case $2 < 1$ is entirely analogous). As in (2) we can show that

$$\begin{array}{l} 1,3 < 2 \\ 3,5 < 4 \\ 5,7 < 6 \\ \ldots \\ n-3, n-1 < n-2 \\ n-1, 1 < n. \end{array} \qquad (3)$$

(See Figure 7 where we depict the ordering relations corresponding to (2) for an 8-node cycle.) But it is straightforward to check that this is an impossible configuration in the ray-shooting ordering.

(3) This follows immediately from part (2) of the Theorem. ■

**Theorem 4.2** *The complete bipartite graph $K_{m,n}$ can be represented with* 1 *stage.*

PROOF The idea for the representation is depicted in Figure 8. To represent $K_{m,n}$ we choose two sets $A, B$ of sizes $m, n$, respectively. Say, $A = \{1, 2, \ldots\}$ and $B = \{a, b, \ldots\}$. Place the sets of points parallel to each other and to the stage as in Figure 8. ■

**Theorem 4.3** *Every tree can be represented with at most two stages. In addition, caterpillars are precisely the trees representable with one stage.*

PROOF (OUTLINE) First of all we show that caterpillars are representable with one stage. The representation is depicted in Figure 9. The stage is $S$. We arrange the points of the body of the caterpillar on the two dashed lines. On the top dashed line we place the points $1, 3, 5, 7, 9, \ldots$ and on the bottom dashed line the points $2, 4, 6, 8, \ldots$. Each of the these points has its "legs" located on the dotted line. The odd (respectively, even) points have their legs placed in the region below (respectively, above) them and delimited by the two dashed lines. It is clear that in this way we can represent all caterpillars.

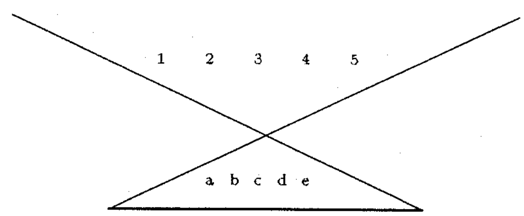


Figure 8: Representing $K_{5,5}$

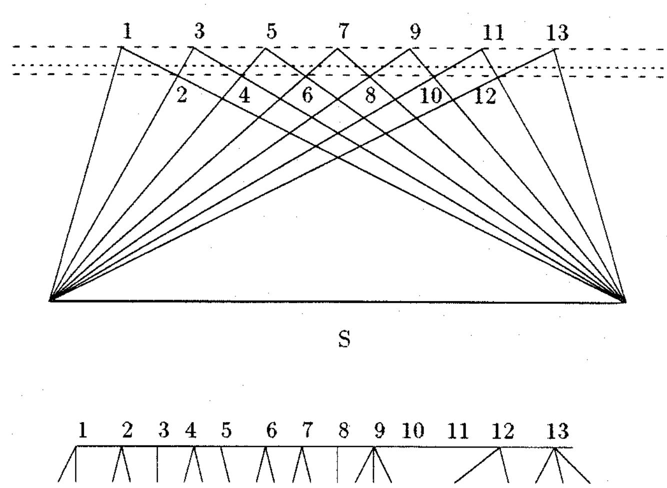


Figure 9: A caterpillar and its representation

The representation of an arbitrary tree with two stages is depicted in Figure 10 and is achieved as follows. Let the stages be $L$ and $R$. Assume the tree to be represented has $h \leq 2k-1$ levels. Consider $k$ different slopes

$$\frac{\pi}{2} > l_1 > l_2 > \cdots > l_k > 0$$

equally distributed in the interval $(0, \pi/2]$. For each slope $l_i$ draw two parallel lines from the endpoints of the stage $L$ with inclination $l_i$. Thus each slope corresponds to a "trunk" with base $L$ and delimited by these two rays. Now consider the stage $R$. Choose a sequence of slopes

$$\pi > r_1 > r_2 > \cdots > r_k > \frac{\pi}{2}$$

and form as before a sequence of trunks each with base $R$ and corresponding slopes $r_i$, $i = 1, \ldots, k$. We can choose the slopes $l_i, r_j$ in such a way that the quadrangles formed by the intersection of the $l_i$-trunk with the $r_j$-trunk, for $i, j = 1, \ldots, k$, are pairwise disjoint. Now we select a sequence of $h$ of these quadrangles, say $Q_1, Q_2, \ldots, Q_h$, in such a way that for any points $u \in Q_i$ and $v \in Q_j$ the straight line $uv$ intersects either $L$ or $R$ only if $|i-j| = 1$. The idea here is that if in each quadrangle $Q_i$ we place points then two points will be adjacent in the ray-shooting ordering only if the corresponding quadrangles they belong to are also "adjacent". This makes possible the representation of the levels of the tree. A sequence of five such quadrangles $Q_1, \ldots, Q_5$ delimited by the regions $1, 2, 3, 4, 5$, respectively, is depicted in Figure 10.

Now we place points within these quadrangles in order to represent the nodes of the tree. Each of the regions labeled $1, 2, 3, 4, 5$ contains a certain set of nodes of the tree. Let the tree be of height $h$. Region 1 contains the leaves; region 2 the nodes at height $h-1$, and in general region $i$ the nodes at height $h-i+1$. The previous condition on the regions guarantees that if $u \in Q_i, v \in Q_j$ and $|i-j| > 2$ then $u, v$ are not adjacent in the tree.

Therefore it remains to show how to place the points in each region. Indeed, nodes are placed in such a way that for each $i$, if $u, v \in Q_i$ then $uv$ does not intersect neither stage $L$ nor stage $R$. In addition, if $i$ is odd (respectively, $i$ is even) and $u \in Q_i, v \in Q_{i+1}$ then $uv$ intersects the stage $R$ (respectively, $L$). This completes the proof of the representation of trees by two stages.

If a tree is not a caterpillar then it must contain the tree depicted in the left-hand side of Figure 11 as a subtree. However it can be shown that this tree can not be represented with one stage. The idea is the following. If we represented the subgraph consisting of the nodes $1, 2, 3, 4, 5$ using a single stage, say $S$, as in the right-hand side of Figure 11, then it can be seen that the nodes $6, 7$ can only be placed inside the region marked with $R$. But then it is clear that they would both have to be adjacent to 3, which is a contradiction. Hence this tree is not representable with one stage. ■

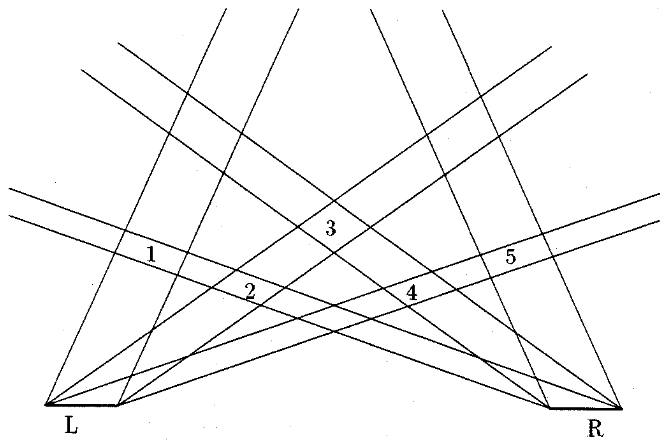


Figure 10: Representing a tree with two stages

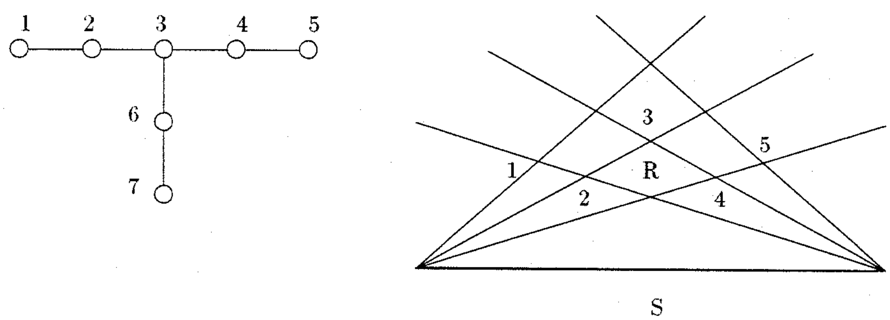


Figure 11: A tree requiring two stages

## 4.2 Upper Bounds

For arbitrary graphs we can establish the following upper bound on $\mathrm{st}(G)$.

**Theorem 4.4** *Every $n$ vertex graph can be represented with at most $\lfloor n(n-1)/4 \rfloor$ stages.*

PROOF Let $G$ be a graph with $n$ vertices and $m$ edges. Represent the graph on a set of $n$ points in the plane which are in general position such that for all points $u, v$ in the given set the line segment $uv$ is not parallel to the $x$-axis. Further assume all the points lie above the $x$-axis.

For each edge $e$ of the graph locate a "small" stage $S_e$ (i.e. a closed interval) on the $x$-axis in such a way that the infinite line segment determined by $e$ intersects $S_e$, but for no other edge of the graph does the corresponding infinite line segment intersect $S_e$. Without loss of generality we may assume the line segments $S_e$, for $e$ an edge of the graph, are pairwise nointersecting.

Color every such stage $S_e$ "blue". In addition, add a "red" stage for each non-edge of the graph in such a way that the red and blue stages are pairwise non-intersecting. There are $m$ many blue stages. Let $r$ be the number of red stages. Notice that

$$m + r = \frac{n(n-1)}{2}.$$

Now traverse the $x$-axis from $-\infty$ to $+\infty$ and join adjacent stages of the same color into single stages of that same color. It is clear that the number $s$ of the resulting blue stages is $\leq \min\{m, r\}$. It follows that

$$2s \leq 2\min\{m, r\} \leq m + r = \frac{n(n-1)}{2},$$

which proves that $s \leq \lfloor n(n-1)/4 \rfloor$, as desired. ■

## 4.3 Lower Bounds

In the sequel we establish lower bounds on the number of stages needed to represent a graph. The first result proves the existence of graphs which require $n/\log n$ stages without giving any indication on how to construct them.

**Theorem 4.5** *For any class $\mathcal{G}$ of graphs on $n$ vertices there exist graphs $G \in \mathcal{G}$ requiring at least $\lceil \log |\mathcal{G}| / n \log n \rceil$ stages for their representation.*

PROOF (OUTLINE) As was shown in section 2, a 1-stage graph is a comparibility graph of dimension 2 and therefore it may be represented using a permutation $\pi$ on $\{1, 2, \ldots, n\}$ as follows:

1. $V = \{(i, \pi(i)) : i = 1, \ldots, n\}$,
2. $G = \{\{(i, \pi(i)), (j, \pi(j))\} : i < j, \pi(i) < \pi(j)\}$.

Thus every such graph can be encoded with a single permutation. Hence a graph which is representable with $k$ stages can be encoded with $2k-1$ permutations. In turn, each of these $2k-1$ permutations can be encoded with $n \log n$ bits, for a total of $(2k-1)n \log n$ bits.

Now assume that $k$ is such that every graph in the class $\mathcal{G}$ is representable with $k$ stages. There are at least $|\mathcal{G}|$ possible graphs and they can all be encoded with $(2k-1)n \log n$ bits. It follows that $(2k-1)n \log n \geq \log |\mathcal{G}|$. This proves the required lower bound. ∎

By using standard results on the number of graphs of specific type (e.g. regular, bipartite etc) it is possible to determine lower bounds for such classes of graphs [11][Chapter 15], [10]. For example, since there are $2^{\Omega(n^2)}$ graphs on $n$ vertices we obtain the following result as a corollary.

**Theorem 4.6** *There exist graphs which require $\Omega(n/\log n)$ stages for their representation.* ∎

Nevertheless, Theorems 4.5 and 4.6 still give no indication on how to construct graphs requiring a large number of stages. To give such a construction we use the previous observation that every cycle with 5 or more nodes requires at least two stages for its representation. This means that graphs which are representable with a single stage must have girth $\leq 4$. We take advantage of this fact in order to prove the following rersult.

**Theorem 4.7** *Every graph $G$ with minimal degree $d$ and girth $\geq 5$ requires at least $\lfloor d/2 \rfloor$ stages for its representation via ray-shooting.*

Proof Assume on the contrary that $G$ can be represented with less than $\lfloor d/2 \rfloor$ stages, say $s$. Let $G_i$ be the subgraph of $G$ corresponding to the $i$th stage. Let $e, e_i$ be the number of edges of the graphs $G, G_i$, respectively. Observe that

$$\begin{aligned} 2\sum_{i=1}^{s} e_i &\geq 2e \\ &= \sum_{u \in V} \deg_G(u) \\ &\geq nd. \end{aligned}$$

It follows that for some $i \leq s$ we must have that $e_i \geq nd/2s \geq n$. This implies that the graph $G_i$ must have a cycle. However since the girth of the graph $G$ is $\geq 5$ so is the girth of the graph $G_i$. This means that the graph $G_i$ cannot be representable with one stage, which contradicts its very definition. ∎

What is the best lower bound that can be achieved via the construction implied by Theorem 4.7? In other words, for a given $d$ what is the smallest possible number of nodes $n$ of a regular graph of degree $d$? A well-known theorem of Erdös and Tutte [25] (see also [18]10.11]) gives an indication on the number of stages required by $n$-node graphs with girth $\geq 5$.

**Theorem 4.8** *Every graph $G$ with minimal degree $d$ and girth $\geq 5$ must have more than $d^2$ vertices.*

PROOF Let $u \in V$ be an arbitrary but fixed vertex of the graph. Let $V_i$ be the set of vertices at distance exactly $i$ from $u$, where $i = 0, 1, 2$. Notice that since the girth of the graph is $\geq 5$ every vertex $v \in V_i$ has exactly one edge to a vertex of $V_{i-1}$. This means that

$$|V_0| = 1, |V_1| \geq d, |V_2| \geq (d-1)|V_1|.$$

It follows that $n \geq |V_0| + |V_1| + |V_2| \geq d^2 + 1$, as desired ∎

There are constructions in the literature of $d$-regular graphs with girth 5. For example, see [25, 26, 27, 22] as well as [3] and the inductive construction in [32], [18][10.12]. An interesting construction of a regular bipartite graph of degree $p+1$, $p^2+p+1$ nodes and girth 6, $p$ prime, is the projective plane over the Galois Field on $p$ elements, with $p+1$ lines each line containing exactly $p+1$ points [18][10.15]. It is clear from Theorem 4.7 that this last graph requires at least $(p+1)/2$ stages for its representation. This gives a graph $G$ on $n$ vertices and $\Theta(n^{3/2})$ edges such that $\mathrm{st}(G) = \Omega(\sqrt{n})$. It is also known [17][Theorem 4.2] that a graph with $n > 2$ vertices, girth $\geq 5$ can have at most $\frac{1}{2}n\sqrt{n-1}$ edges. Hence, $\Theta(\sqrt{n})$ is the highest possible stage number for a graph obtained by Theorem 4.7.

# 5 Ray Shooting From Convex Ranges

In this section we study ray shooting graphs in the three-dimensional vector space. Let $X = \{p_1, \ldots, p_n\}$ be a collection of points in a vector space $\mathbb{E}^3$ such that the $z$-coordinate of each element of $X$ is strictly greater than 0 and $S$ a compact plane-convex set of $\mathbb{E}^3$ contained in the hyperplane $H_0 = \{p \in \mathbb{E}^3 :$ the $z$-coordinate of $p$ is $0\}$. Given $X$ and $S$ we can now construct a graph $G(X, S)$ with vertex set $X$ such that two vertices $p_i, p_j$ of $G(X, S)$ are adjacent if the line through $p_i$ and $p_j$ intersects $S$. $G(X, S)$ will be called the ray-shooting graph of $X$ and $S$. $G(X, S)$ is called a 3-ray-shooting graph (see Figure 12).

We prove that 3-ray shooting graphs are comparability graphs. These are graphs $G$ that can be oriented in such a way that if we have $u \rightarrow v$ and $v \rightarrow w$ in $G$ then $u \rightarrow w$ is also an edge of $G$. We prove that the set of 3-ray-shooting graphs obtained when $S$ is a triangle is exactly the set of comparability graphs of orders of dimension at most 3, and thus in this case, the recognition problem is NP-complete [33]. We also provide a characterization of 3-ray shooting graphs in terms of geometric containment orders, that is ordered sets arising from containment relations among the elements of families of convex sets on the plane. For instance, we prove that when $S$ is a circle, $G(S, X)$ is the comparability graph of a circle order [29]. Finally, we prove that not all comparability graphs are 3-ray shooting graphs. In order to prove these results we will need to review some concepts of geometric containment and ordered sets.

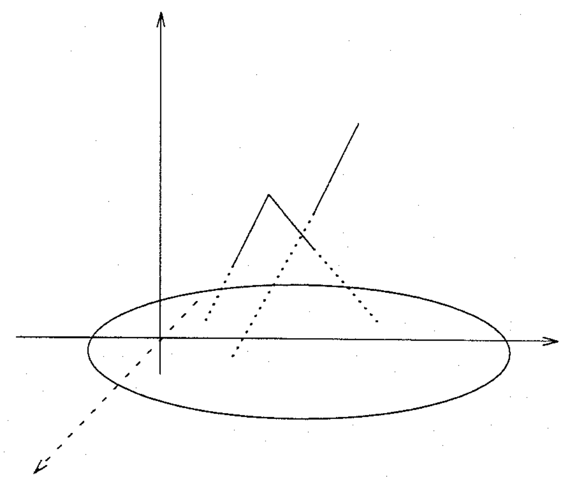

Figure 12: A circle representation of a graph.

## 5.1 Function diagrams and crossing numbers

Let $\xi = \{f_1, \ldots, f_m\}$ be a family of continuous functions $f_i : [0,1] \rightarrow \mathbb{R}$, $i = 1, \ldots, m$. The family $\xi$ is called regular if the following conditions are satisfied.

1. For any pair of elements $f_i, f_j \in \xi, i \neq j$, the set of values $S(i,j) = \{x \in [0,1] : f_i(x) = f_j(x)\}$ is finite.
2. $f_i(0) \neq f_j(0), f_i(1) \neq f_j(1); i \neq j$.
3. Each time the graphs of two different functions intersect, they cross each other; that is if $f_i(x_0) = f_j(x_0)$ there exists an $\epsilon > 0$ such that $x_0 - \epsilon < x < x_0 < y < x_0 + \epsilon$ implies that $f_i(x) < f_j(x)$ and $f_i(y) > f_j(y)$ or $f_i(x) > f_j(x)$ and $f_i(y) < f_j(y)$.

Informally speaking, a set of functions $\xi$ as above is regular if the graphs of any two elements $\xi$ intersect a finite number of times and each time they intersect, they cross each other.

Let $X = \{x_1, \ldots, x_m\}$ be a set, and $P(X, <)$ a partial order on $X$. $P(X, <)$ is called a function order ($f$-order for short) if there exists a regular set of functions $\xi = \{f_1, \ldots, f_m\}$ such that $x_i < x_j$ if $f_i(x) < f_j(x)$, for all $x \in [0,1]$. The set $\xi$ will be called an $f$-diagram for $P(X, <)$. We will also say that $P(X, <)$ represents $\xi$. It is easy to prove that every poset is an $f$-order [31].

Given an $f$-diagram $\xi = \{f_1, \ldots, f_m\}$, the crossing number $\chi(\xi)$ is defined as the maximum over the set $\{|S(i,j)| : f_i, f_j \in \xi, i \neq j\}$; that is the maximum

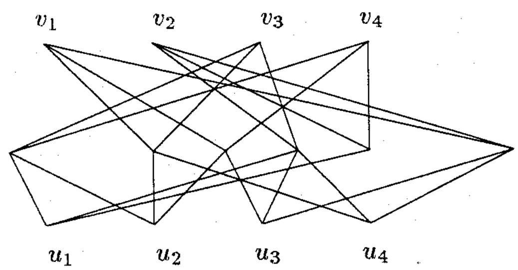


Figure 13: The order set $\Psi_n$.

number of times two elements of $\xi$ intersect. The crossing number $\chi(P(X,<))$ of a poset $P(X,<)$ is now defined as $\min\{\chi(\xi) : \xi$ is an $f$-diagram for $P(X,<)\}$.

Informally speaking, every partial order can be represented in many ways using a regular set $\xi = \{f_1, \ldots, f_m\}$ of continuous real functions with domain $[0,1]$. In each such representation, the graphs of some elements of $\xi$ intersect a number of times. The crossing number of a poset $P(X,<)$ is $k$ if in any $f$-diagram $\xi = \{f_1, \ldots, f_m\}$ representing $P(X,<)$ there are at least two elements of $\xi$ that intersect at least $k$ times. Notice that if $\chi(P(X,<)) = 0$, then $P(X,<)$ has an $f$-diagram $\xi$ in which no pair of functions of $\xi$ intersects, thus $P(X,<)$ is a linear order. It is also easy to prove that if $\chi(P(X,<)) = 1$, then $\dim P(X,<) = 0$ and that in general $\chi(P(X,<)) \leq \dim P(X,<) - 1$ [31].

Let $H_n(X,<)$ be the ordered set with elements

$$X = \{u_1, \ldots, u_n, v_1, \ldots, v_n\}$$

such that $u_i < v_j, i \neq j$, and all other pairs of elements in $H_n(X,<)$ are not comparable.

Let $\Psi_n$ be the ordered set obtained from $H_n(X,<)$ as follows. For each subset $S_k$ of $\{1, \ldots, n\}$ with either $\lfloor n/2 \rfloor$ or $\lfloor (n+1)/2 \rfloor$ elements, insert in $H_n(X,<)$ a new element $s_k$ such that $s_k > u_j, j \in S_k, s_k < v_i, i \notin S_k; s_i < s_j$ if $S_i \subset S_j, i \neq j$ (see Figure 13). The next result was proved in [29]:

**Theorem 5.1** *The ordered set $\Psi_n$ has crossing number $n-1$ and dimension $n$.* ∎

## 5.2 Ray shooting in $\mathbb{E}^3$

In the rest of this section, unless otherwise specified, $S$ will always refer to a two dimensional convex set of $\mathbb{R}^3$ contained in the plane $z = 0$ such that $S$ contains the origin. $F = \{S_1, \ldots, S_n\}$ will be a family of convex sets homothetic to $S$

and contained in the plane $z = 0$. $X$ will always refer to a collection of points in $\mathbb{R}^3$ such that the $z$- coordinate of all the elements of $X$ is strictly greater that 0. As in the two dimensional case, we define a graph $G(S, X)$ with vertex set $X$, and in which two vertices are adjacent if and only if the line joinning them intersects $S$. We now say that a graph $G$ is a 3-ray shooting graph if there is a two-dimensional convex set $S$ (contained in the plane $z = 0$ and containing the origin) and a collection of points $X$ (with $z$- coordinate greater than 0) such that $G$ is isomorphic to $G(S, P)$.

**Lemma 5.1** *Let $S$ be a plane convex set of $\mathbb{R}^3$ and $X$ a collection of points in $\mathbb{R}^3$ with $z$ coordinates greater than 0, then $G(S, X)$ is a comparability graph.*

PROOF For every point $p_i \in X$ let $C(p_i, S)$ be the truncated cone formed by the set of all line segments joining $p_i$ to points in $S$. It is easy to see that if the $z$ coordinate of $p_i$ is smaller than the $z$ coordinate of $p_j$ then $p_i$ and $p_j$ are adjacent in $G(S, X)$ if and only if $C(p_i, S)$ is contained in $C(p_j, S)$. In this case, orient the edge $\{p_i, p_j\}$ of $G(S, X)$ as $p_i \rightarrow p_j$. To verify the transitivity of this orientation, we simply observe that if $p_i \rightarrow p_j$ and $p_j \rightarrow p_k$ then $C(p_i, S)$ is contained in $C(p_j, S)$ which in turn is contained in $C(p_k, S)$. It follows that $C(p_i, S)$ is contained in $C(p_k, S)$ and thus $p_i \rightarrow p_k$. ■

The orientation induced on $G(S, X)$ induces a partial order $P(X, <)$ on $X$. $P(X, <)$ will be called a ray shooting order. If in addition we want to specify that a ray shooting order $P(X, <)$ arises from a specific convex set $S$, we will call $P(X, <)$ an $S$-shooting order. A natural question arises.

> Is it true that for every ordered set $P(Y, <)$ is a ray-shooting order? That is, is it true that for every ordered set $P(Y, <)$ there is a convex set $S$ and a point set $X$ such that $P(Y, <)$ is isomorphic to the ray shooting order generated by $X$ and $S$? Can we characterize ray-shooting orders?

Our main objective is now to show that there are ordered sets that are not ray-shooting orders regardless of the choice of $S$. We will also give a partial characterization of ray-shooting orders in terms of "geometric containment" orders. To answer these questions, we need to recall some results on geometric containment and ordered sets.

Let $H = \{T_1, \ldots, T_n\}$ be family of plane convex sets. A containment order $P(H, <)$ on $H$ can be defined in which $S_i < S_j$ if and only if $S_i$ is a subset of $S_j$. In the case that all the elements of $H$ are circles, i.e. circles together with their interiors, $P(H, <)$ is called a circle order [29]. When the elements of $H$ are polygons with $n$ vertices, $P(H, <)$ is called an $n$-gon order. When in addition the elements of $H$ are regular $n$-gons with the same orientation, $P(H, <)$ is called a regular $n$-gon order [28].

Given a convex set $S$ and a family $F = \{S_1, \ldots, S_n\}$ of convex sets homothetic to $S$, the containment order $P(F, <)$ arising from $F$ will be called an $S$-order, e.g. if $S$ is a circle, $P(F, <)$ is a circle-order.

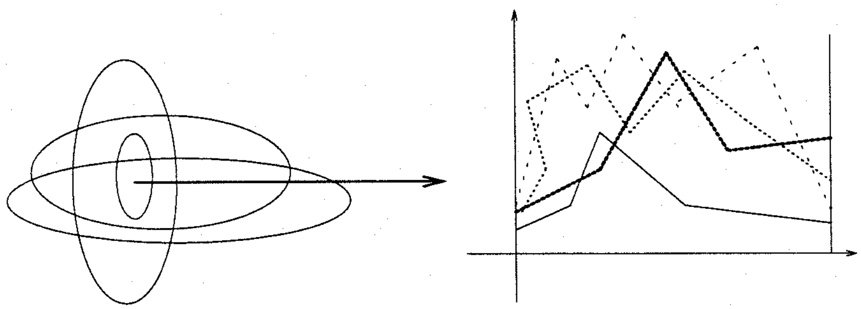

Figure 14:

A family $H = \{T_1, \ldots, T_n\}$ of convex sets is called normal if the intersection of all the elements of $H$ is non-empty.

**Lemma 5.2** *Let $P(H, <)$ be a containment order of a normal family $H = \{T_1, \ldots, T_n\}$ of convex sets such that the boundaries of every two elements $T_i$ and $T_j$ intersect at most $k$ times, then the crossing number of $P(H, <)$ is at most $k$.*

PROOF To prove our result, all we need to do is to produce a function diagram for $P(H, <)$ in which every pair of functions intersects at most $k$ times. Since $H$ is normal, there is a point $p$ in the interior of all $T_i, i = 1, \ldots, n$. Using what in topology is known as surgery, cut the plane along a ray emanating from $P$ (that does not go through any point in the intersection of the boundaries of any two $S_i, S_j$) and map the two sides of the cut line to the lines $x = 0$ and 1 of the plane, the upper cut to $x = 1$ and the lower to $x = 0$. It is easy to see (e.g. using polar coordinates) that this can be done in such a way that the boundary of each $S_i$ is mapped to a continuous function from $[0, 1]$ to the reals and that intersection points of boundaries are mapped to intersection points of corresponding functions. The result now follows (see Figure 14). ■

We now prove the following result:

**Theorem 5.2** *Let $S$ be a convex set and $P(X, <)$ an $S$-order. Then the crossing number of $P(X, <)$ is at most 2.*

Some terminology and basic geometric results will be needed before we prove our result. Let $S$ be a plane convex set of $\mathbb{R}^3$ and $F = \{S_1, \ldots, S_n\}$ be a family of convex sets homothetic to $S$. Let $C(S)$ be the cone containing all the rays joining the point $(0, 0, 2)$ to points in $S$. The point $(0, 0, 2)$ will be called the apex of $C(S)$. It is easy to see that every $S_i \in F$ is the intersection of the plane

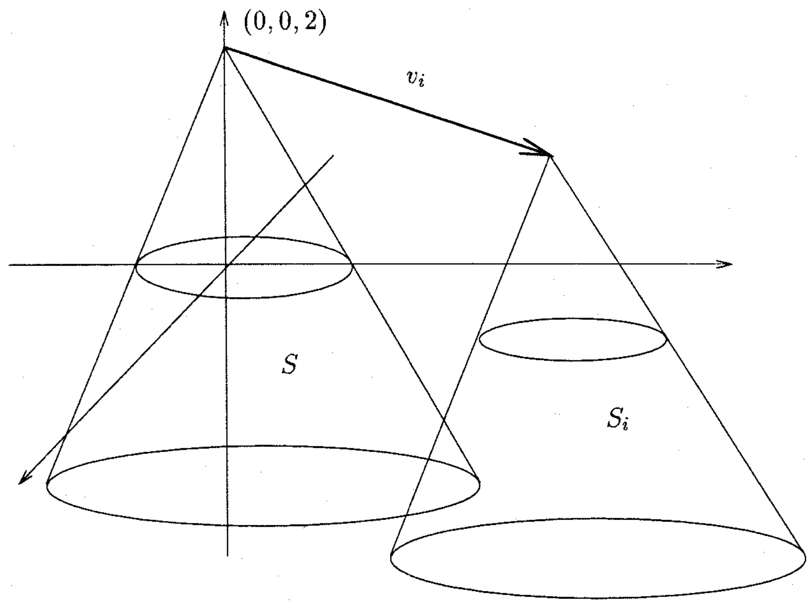


Figure 15: $S, S_i$ are homothetic and contained in the plane $z = 0$,

$z = 0$ with a translate $C_i(S) = \{q = p+v_i; p \in C(S)\}$ by a vector $v_i, i = 1, \ldots, n$. The point $a_i = (0, 0, 2) + v_i$ will be called the apex of $C_i(S), i = 1, ..., n$.

Let $H_m$ be the plane $z = -m$ of $\mathbb{R}^3$, $m$ a positive integer. Let us now define $F_m = \{S_{i,m} = C_i(S) \cap H_m, i = 1, ..., n\}$. The following lemma is straightforward.

**Lemma 5.3** *The containment order induced by $F_m$ is the same as that of $F$. Moreover if $m$ is large enough, then $F_m$ is a normal family of sets, i.e. the intersection of all sets of $F_m$ is non empty.* ■

PROOF of Theorem 5.2. Let $F = \{S_1, \ldots, S_n\}$ be a family of convex sets on the plane homothetic to a convex set $S$ and let $P(F, <)$ be the containment order set generated by $F$. By Lemma 5.3, there is an $m$ such that $F_m = \{S_{1,m}, \ldots, S_{n,m}\}$ is a normal family of convex sets homothetic to $S$ such that $F_m$ has the same containment order as $F$.

But now by Lemma 5.2, and the fact that the boundaries of any two of the homothetic convex sets intersect at most twice, the crossing number of $P(F, <)$ is at most 2. This completes the proof of Theorem 5.2. ■

**Theorem 5.3** *Not every comparability graph is a 3-ray-shooting graph.*

PROOF It is proved in [29] that for every $n$ there are orders of dimension $n$ and crossing number $n-1$. Since the crossing number of ray shooting orders is at most two, our result now follows. ■

We finish this section by proving the following result.

**Theorem 5.4** *An ordered set $P(Y,<)$ is an $S$-shooting order if and only if $P(Y,<)$ is an $S$-ordered set.*

Some extra results will be needed to prove Theorem 5.4. Given $C(S)$ let $D(S)$ be the set of directions of the rays contained in $C(S)$ emanating from the apex $(0,0,2)$ of $C(S)$, that is the set of unit vectors $u_i$ of $\mathbb{R}^3$ such that $(0,0,2)+u_i$ belongs to $C(S)$. $D(S)$ is a compact subset of the unit sphere in $\mathbb{R}^3$.

Consider a convex set $S'$ contained in the plane $z=0$ homothetic to $S$ and containing all the elements of $F$. Let $C'(S)$ be a translate of $C(S)$ such that $C'(S)\cap H_0 = S'$. Since $S'$ contains all the elements of $F, S'_m = H_m \cap C'(S)$ contains all the elements of $F_m$. The next lemma is easy to prove.

**Lemma 5.4** *For every $\epsilon > 0$, there is an $m_0$ such that if $m > m_0$ and $q \in S'_m$ then the unit vector $u$ determined by the ray from the apex $a_i$ of $C_i(S)$ to $q$ belongs to the set of directions $D(S)$ of $C(S)$ or is at distance at most $\epsilon$ from $D(S), i = 1,\ldots,n$.* ■

PROOF of Theorem 5.4. We prove first that $S$-shooting orders are $S$-containment orders. Let $X$ be a collection of points with $z$ coordinate greater than 0 and let $P(X,<)$ be the $S$-shooting order generated by $S$ and $X$. Assume without loss of generality that all the elements of $X$ have $z$-coordinate greater than 2. Each point $p_i$ of $X$ together with $S$ defines a truncated cone $C(p_i,S)$ containing all line segments joining $p_i$ to points in $S$. Consider the plane $z=1$. Each truncated cone $C(p_i,S)$ intersects the plane $z=1$ in a convex set $Q_i$ homothetic to $S$. Moreover if an element $p_i$ is smaller than $p_j$ in $P(X,<)$ then $C(p_i,S)$ is contained in $C(p_j,S)$ and then $Q_i$ is contained in $Q_j$. It now follows that $P(X,<)$ is an $S$-order.

Conversely, let $P(F,<)$ be an $S$-containment order and $F=\{S_1,\ldots,S_n\}$ a family of $n$ convex sets homothetic to $S$ that generates $P(F,<)$. For every $S_i$ of $F$ let $C_i(S)$ be the cone defined in the proof of Theorem 5.3. As in Theorem 5.3 let $S_{i,m} = C_i(S)\cap H_m$ and let $S'_m = H_m \cap C'(S)$.

Notice that if two sets $S_i$ and $S_j$ of $F$ are such that $S_i$ is contained in $S_j$ then the line joining the apex $a_i$ of $C_i(S)$ to the apex $a_j$ of $C_j(S)$ intersects $S'_m$. Suppose then that $S_i$ is not contained in $S_j$ and $S_j$ is not contained in $S_i$. Assume without loss of generality that the $z$-coordinate of $a_i$ is greater than that of $a_j$. We proceed now to prove that if $m$ is large enough, then the line joining $a_i$ to $a_j$ does not intersect $S_{i,m}$. To prove this, it is equivalent to prove that the ray emanating at $a_i$ through $a_j$ does not intersect $S_{i,m}$. To prove this we notice that since $S_i$ is not contained in $S_j$ and $S_j$ is not contained in $S_i$ the

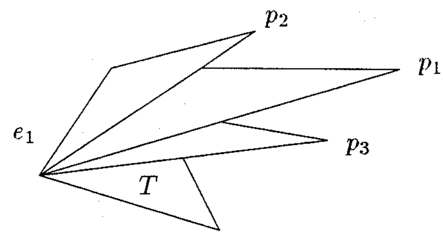


Figure 16: $L_1$ is the order $p_3 < p_1 < p_2$

direction $d_{i,j}$ determined by the ray from $a_i$ to $a_j$ does not belong to $D(S)$. Let $e_{i,j} > 0$ be the distance from $d_{i,j}$ to $D(S)$ and let

$$\epsilon = \min\{\epsilon_{i,j} : S_i \not\subseteq S_j \ \& \ S_j \not\subseteq S_i\}.$$

Then by Lemma 5.4 there is an $m_0$ such that if $m > m_0$ then the direction joining any $a_i$ to any point of $S_{i,m}$ is at distance at most $\epsilon$ from $D(S)$. Then if $S_i$ is not contained in $S_j$ and $S_j$ is not contained in $S_i$ the line joining $a_i$ to $a_j$ cannot intersect $S_{i,m}$, and the direction of its distance to $D(S)$ is greater than or equal to $\epsilon$ which is a contradiction. ■

We conclude by proving that recognizing triangle-shooting graphs is equivalent to recognizing orders of dimension 3, which is NP-hard [33].

**Theorem 5.5** *An ordered set $P(X, <)$ is a triangle-shooting order if and only if the dimension of $P(X, <)$ is at most 3.*

**Proof** Let $T$ be a triangle contained in the plane $z = 0$ of $\mathbb{E}^3$ with edges $e_1, e_2, e_3$, and as usual, $X$ a set of $n$ points in $\mathbb{E}^3$ with $z$-coordinate greater than 0. Let $P(T, <)$ be the ray shooting order generated by $T$ and $X$. For every point $p_i$ of $X$ consider the triangles $T_{i,j}$ defined by $e_j$ and $p_i, i = 1, \ldots, n, j = 1, 2, 3$. For each $k = 1, 2, 3$ we can define a linear order $L_k$ on $X$ in which a point $p_i$ is smaller than $p_j$ if when we rotate $T$ along $e_k$ in the upward direction, we meet $p_i$ before we meet $p_j$. It is easy to see that $P(X, <) = L_1 \cap L_2 \cap L_3$ (see Figure 16).

Conversely, let $P(Y, <)$ be an ordered set of dimension at most 3 and three linear extensions $L_1, L_2, L_3$ of $P(Y, <)$ such that $P(Y, <) = L_1 \cap L_2 \cap L_3$. Consider a point $q$ in the interior of $T$ and the perpendicular $L$ to $T$ through $q$. Choose $n$ points $q_i$ in general position "around" $L$ with $z$-coordinate equal to $i, i = 1, \ldots, n$. For $k = 1, 2, 3$ let $H_{k,i}$ be the plane containing $e_k$ and $q_i, k = 1, 2, 3, i = 1, \ldots, n$. For every $y \in Y$ let $r(k, y)$ be the rank or position of $y$ in $L_k, k = 1, 2, 3$. It is now easy to see that the set of points

$X = \{H_{k,r(1,y)} \cap H_{k,r(2,y)} \cap H_{k,r(3,y)} : y \in Y\}$ is such that the ray shooting order induced by $T$ and $X$ is isomorphic to $P(Y, <)$. ■

# 6 Conclusion

In this paper we focused on geometric and graph-theoretic applications of the paradigm "killing two birds with one stone". There are several interesting open problems suggested by our investigations. More efficient matching algorithms for comparability graphs seem to be posiible and are currently under investigation. The notion of stage number, as a graph theoretic parameter, seems to be interesting in its own right. This suggests, the search for tighter (constructive or not) upper and lower bounds on the stage number of an arbitrary graph, as well as determining the complexity of the recognition problem $G \in \mathcal{G}_k$, both for fixed as well variable $k$.

# References


[1] H. Alt, N. Blum, K. Mehlhorn and M. Paul, "Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log m})$", Infor Proc. Letters, Vol. 37, 1991, pp. 237-240.

[2] M. Atallah, R. Cole and M. Goodrich. "Cascading divide-and-conquer: a technique for designing parallel algorithms." SIAM J. Computing, **18** (1989), pp. 499-532.

[3] B. Bollobás, "Extremal Graph Theory", Academic Press, 1978.

[4] R. Cole, "Parallel Merge Sort", SIAM J. on Comp., 17(4):770-785, 1988.

[5] A. Datta, A. Maheshwari, J.-R. Sack, "Optimal parallel algorithms for direct dominance problems", Proc. of the First Annual European Symposium on Algorithms, Lecture Notes in Computer Science, 1993.

[6] B. Dushnik and E. Miller, "Partially Ordered Sets", Amer. Math. Montly 55 (1948) 26-28.

[7] P.C. Fishburn and W.T. Trotter, "Angle Orders", Order 1, (1985) 81-88.

[8] M.C. Golumbic, "Comparability graph recognition and coloring", Computing 18, 1977, pp. 199-208.

[9] M. T. Goodrich, "Intersecting line segments in parallel with an output-sensitive number of processors", SIAM J. Computing, 20, 1991, pp. 737-755.

[10] I. P. Goulden and D. M. Jackson, "Combinatorial Enumeration", John Wiley & Sons, New York, 1983.

[11] F. Harary, "Graph Theory", Addison-Wesley Publishing Company, 1969.

[12] J. JáJá, "An introduction to parallel algorithms", Addison-Weseley Publishing Company, 1992.

[13] R. M. Karp and R. Vijaya Ramachandran, "Parallel Algorithms for Shared-Memory Machines", Handbook of Theoretical Computer Science, Edited by J. van Leeuwen, Volume 1, Elsevier Science Publishers B.V., 1990.

[14] D. König, "Theorie der endlichen und unendlichen Graphen", Leipzig, 1936 (reprinted by Chelsea, New York, 1950).

[15] A. Lingas and A. Maheshwari, "Simple optimal parallel algorithm for reporting paths in trees", STACS94, Lecture Notes in Computer Science, 1994.

[16] J. van Leeuwen, "Graph Algorithms", in: Handbook of Theoretical Computer Science, Edited by J. van Leeuwen, Volume 1, Elsevier Science Publishers B.V., 1990.

[17] J. H. van Lint and R. M. Wilson, "A Course in Combinatorics", Cambridge University Press, 1992.

[18] L. Lovász, "Combinatorial Problems and Exercises", North Holland Publishing Company, 1979.

[19] R.M. McConnell and J.P. Spinrad, "Linear-time modular decomposition and efficicent transitive orientation of comparability graphs", ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 536-545.

[20] K. Mehlhorn and S. Näher, "Dynamic Fractional Cascading", Algorithmica, Vol. 5, No. 2, pp. 215-241, 1990.

[21] S. Micali and V. V. Vazirani, "An $O(\sqrt{V}E)$ algorithm for finding maximum matching in general graphs", in: Proc. 21st Ann. IEEE Symp. Foundations of Computer Science, pages 17-27, 1980.

[22] V. Neumann-Lara, "$k$-Hamiltonian Graphs with Given Girth", in: Infinite and Finite Sets, Colloquia Mathematica Societatis János Bolyai, Keszthely, Hungary, 1973, pp. 1133 - 1142.

[23] A. Pnueli, S. Even and A. Lempel, "Transitive orientation of graphs and identification of permutation graphs", Canad. J. Math 23, 1971, pp. 160-175.

[24] F. P. Preparata and M. I. Shamos, "Computational Geometry: An Introduction", Springer-Verlag, New York, 1985.

[25] H. Sachs, "Einführung in die Theorie der endlichen Graphen", Teil I - II, Teubner, 1970 - 1972.

[26] H. Sachs, "Regular Graphs with Given Girth and Restricted Circuits", Journal of the London Mathematical Society, 38 (1963) 423 - 429.

[27] H. Sachs, "On Regular Graphs with Given Girth", in: Theory of Graphs and its Applications, (M. Fiedler, ed.), Academic Press, New York, 1965, pp. 91 - 97.

[28] N. Santoro and J. Urrutia, "Angle Orders, Regular $n$-gon Orders and the Crossing Number of a Partial Order", Order 4 (1987) 209-220.

[29] J.B. Sidney, S.J. Sidney and J. Urrutia, "Circle Orders, $n$-gon Orders and the Crossing Number", Order 5 (1988) 1-10.

[30] J. Spinrad, "On Comparability and Permutation Graphs", Siam J. Comp. 14, 658-670, 1985.

[31] J. Urrutia, "Partial Orders and Euclidean Geometry". In Algorithms and Order, I. Rival (ed), 387-434 (1989) Kluwer Academic Publishers.

[32] H. Walther and H.-J. Voß, "Über Kreise in Graphen", VEB Deutscher Verlag der Wissenschaften, 1974.

[33] M. Yannakakis, "The Complexity of the Partial Order Dimension Problem", SIAM J. Discr. Meth., 3, 351-358, 1982.

## School of Computer Science, Carleton University
## Recent Technical Reports

**TR-219** **Three Algorithms for Selection on the Reconfigurable Mesh**
Dipak Pravin Doctor and Danny Krizanc, February 1993

**TR-220** **On Multi-label Linear Interval Routing Schemes**
Evangelos Kranakis, Danny Krizanc, and S.S. Ravi, March 1993

**TR-221** **Note on Systems of Polynomial Equations over Finite Fields**
Vincenzo Acciaro, March 1993

**TR-222** **Time-Message Trade-Offs for the Weak Unison Problem**
Amos Israeli, Evangelos Kranakis, Danny Krizanc and Nicola Santoro, March 1993

**TR-223** **Anonymous Wireless Rings**
Krzysztof Diks, Evangelos Kranakis, Adam Malinowski, and Andrzej Pelc, April 1993

**TR-224** **A consistent model for noisy channels permitting arbitrarily distributed substitutions, insertions and deletions**
B.J. Oommen and R.L. Kashyap, June 1993

**TR-225** **Mixture Decomposition for Distributions from the Exponential Family Using a Generalized Method of Moments**
S.T. Sum and B.J. Oommen, June 1993

**TR-226** **Switching Models for Non-Stationary Random Environments**
B. John Oommen and Hassan Masum, July 1993

**TR-227** **The Probability of Generating Some Common Families of Finite Groups**
Vincenzo Acciaro, September 1993

**TR-228** **Power Roots of Polynomials over Arbitrary Fields**
Vincenzo Acciaro, September 1993

**TR-229** **Optimal Parallel Algorithms for Direct Dominance Problems**
Amitava Datta, Anil Maheshwari and Jörg-Rüdiger Sack, October 1993

**TR-230** **Uniform Generation of Forests of Restricted Height**
M.D. Atkinson and J.-R. Sack, October 1993

**TR-231** **Optimal Elections in Labeled Hypercubes**
Paola Flocchini and Bernard Mans, December 1993

**TR-232** **On the Complexity of Computing Gröbner Bases in Characteristic 2**
Vincenzo Acciaro, December 1993

**TR-233** **Broadcasting Session Keys**
Mike Just, Evangelos Kranakis, Danny Krizanc, and Paul van Oorschot, February 1994

**TR-234** **String Taxonomy Using Learning Automata**
B. John Oommen and Edward V. de St. Croix, March 1994

**TR-235** **Distributed Cyclic Reference Counting**
Frank Dehne and Rafael D. Lins, March 1994

**TR-236** **Exact and Approximate Computational Geometry Solutions of an Unrestricted Point Set Stereo Matching Problem**
Frank Dehne and Katia Guimaraes, March 1994

**TR-237** **Scalable and Architecture Independent Parallel Geometric Algorithms with High Probability Optimal Time**
Frank Dehne, Claire Kenyon and Andreas Fabri, March 1994

**TR-238** **Finding the Extrema of a Distributed Multiset**
Paola Alimonti, Paola Flocchini and Nicola Santoro, March 1994