Turán Triangles, Cell Covers, Road Placement and Train Scheduling

Luís Fernando Schultz Xavier da Silveira

Thesis submitted to the University of Ottawa
in partial Fullfilment of the requirements for
the degree of Doctor of Philosophy in
Computer Science.

School of Electrical Engineering and Computer Science

Faculty of Science

University of Ottawa

© Luís Fernando Schultz Xavier da Silveira, Ottawa, Canada, 2020

Acknowledgements (Agradecimentos)

Agradeço enormemente minha família, sem o apoio da qual jamais teria chegado até aqui. Obrigado pai, mãe, irmão e avós. O agradecimento é especial à minha avó Neuza, sempre presente, senão em pessoa, então em minhas melhores lembranças.

I thank my supervisors and colleagues at the Computational Geometry Lab at Carleton University. Thank you Vida, Pat, Jit, Anil, Michiel, Jean-Lou, Paz, Darryl and Saeed. Your guidance and collaboration will always be appreciated, and so will the contacts you have introduced me to, many through your workshops. Also thanks to you, Lúcia, for introducing me to such great people.

I would also like to thank you, Daniel, Jáuber and Ivan, for your teachings, which still influence me greatly today. A special thanks to Daniel who was great company back in Florianópolis, but also here in Ottawa, where he spent a year.

Last, but not least, I would like to thank some of the great friends I made in Ottawa. Thank you Saeed, Rob, Pascal, Alex, Serena, Bradley and Karim. Playing with you always cheered me up.

Contributions

Chapter 2 is joint work with Boris Aronov, Vida Dujmović, Pat Morin and Aurélien Ooms, Chapter 3 with Paz Carmi, Anil Maheshwari and Saeed Mehrabi, Chapter 4 with Paz Carmi, Jean-Lou De Carufel, Stephane Durocher, Darryl Hill and Thomas Shermer and Chapter 5 with Jean-Lou De Carufel, Darryl Hill, Anil Maheshwari and Sasanka Roy. As is typical in theoretical computer science, the authors are listed in alphabetical order by family name. Though it is difficult to pinpoint the exact source of any idea that appears in a multi-author paper, I made major contributions leading to the results in each of these papers.

Abstract

In this doctoral thesis, four questions related to computational geometry are considered. The first is an extremal combinatorics question regarding triangles with vertices taken from a set of n points in convex position. More precisely, two such triangles can exhibit eight distinct configurations (,,,,,,,) and, for each subset of these configurations, we are interested in the asymptotics of how many triangles one can have while avoiding configurations in the subset (as a function of n). For most of these subsets, we answer this question optimally up to a logarithmic factor in the form of several Turán-type theorems [4]. The answers for the remaining few were in turn tied to that of a long-standing open problem which appeared in the literature in the contexts of monotone matrices [52], tripod packing [54] and 2-comparable sets [37, 26].

The second problem, called Line Segment Covering (LSC), is about covering the cells of an arrangement of line segments with these line segments, where a segment covers the cells it is incident to. Recently, a PTAS, an APX-hardness proof and a FPT algorithm for variants of this problem have been shown [14]. This paper and a new slight generalization of one of its results is included as a chapter.

The third problem has been posed in the Sixth Annual Workshop on Geometry and Graphs and concerns the design of road networks to minimize the maximum travel time between two point sets in the plane. Traveling outside the roads costs more time per unit of distance than traveling on the roads and the total length of the roads can not exceed a budget. When the point sets are the opposing sides of a unit square and the budget is at most $\sqrt{2}$, we were able to come up with a few network designs that cover all possible cases and are provably optimal. Furthermore, when both point sets are the boundary of a unit circle, we managed to disprove the natural conjecture that a concentric circle is an optimal design.

Finally, we consider collision-avoiding schedules of unit-velocity axis-aligned trains departing and arriving from points in the integer lattice. We prove a few surprising results on the existence of constant upper bounds on the maximum delay that are independent of the train network. In particular, these upper bounds are shown to always exist in two dimensions and to exist in three dimensions for unit-length trains. We also showed computationally that, for several scenarios, these upper bounds are tight.

Contents

Chapter 1

Introduction

This doctoral thesis consists of two published articles which I have coauthored [4, 14] as Chapters 2 and 3 and of two other chapters based on articles yet to be published.

Two triangles whose vertices are in convex position can have eight distinct configurations (). For each of the 256 possible subsets X of these configurations and every natural number n, one can consider how many triangles can one have whose vertices are in a set of n points in convex position without any two triangles forming a configuration in X. To this maximum number of triangles, we give the name $\mathrm{ex}(n, X)$. Chapter 2 is a close to complete catalog of the asymptotics of $\mathrm{ex}(n, X)$ for each fixed X. We have determined, up to a factor of $\log n$, the asymptotic behavior of $\mathrm{ex}(n, X)$ for 248 choices of X and tied that behavior for the remaining 8 to the tripod packing problem, an open combinatorial problem that appeared in the literature under many names but still holds quite a gap between its best known lower and upper bounds. This work has been published in the Electronic Journal of Combinatorics.

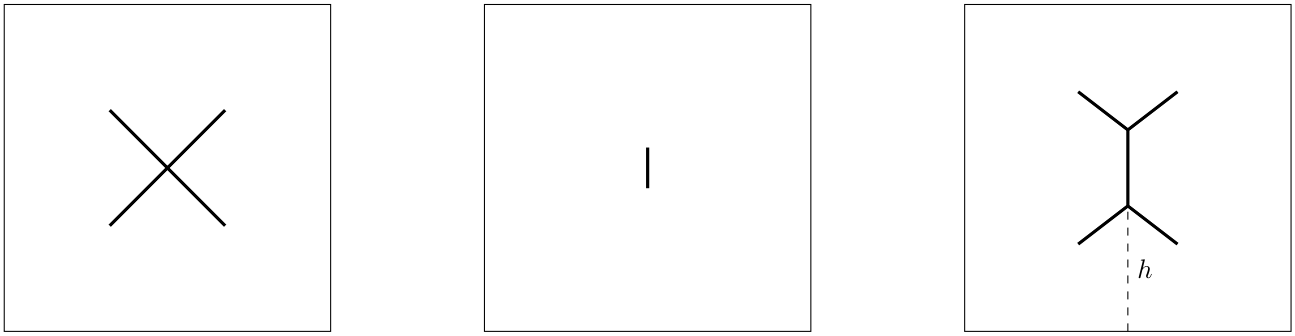
One staple of combinatorial optimization is the Set Cover problem, which is known to be both NP-hard and hard to approximate. Its difficulty is the main reason why we study geometric set cover problems, in which both the set to be covered and the covering sets are of geometric nature. One such problem, studied in Chapter 3, is the Line Segment Cover (LSC) problem, which consists of covering the faces of an arrangement of line segments by these line segments, where a line segment covers the faces it is incident to.

We recall some concepts regarding approximation algorithms from the perspective of minimization problems. A polynomial time constant-factor approximation algorithm is an algorithm that, taking as input an instance of the problem, produces a solution in polynomial time with cost bounded by a constant factor of an optimal solution's cost. A stronger notion is that of a Polynomial Time Approximation Scheme (PTAS), which is an algorithm that, besides an instance of the problem, takes as input a positive rational number ε and produces a solution whose cost is at most $1+\varepsilon$ times the cost of an optimal solution. A PTAS is required to run in polynomial time with regard to the input size for any fixed ε but has no run time restrictions with regard to ε. The class of problems that have a polynomial time constant-factor approximation algorithm is named APX, while the class of problems that have a PTAS is named PTAS.

If $\mathsf{P} \neq \mathsf{NP}$, then $\mathsf{APX} \neq \mathsf{PTAS}$ [30, page 20]. Therefore, analogously to how we define the class of NP-hard problems, we would like to define the class of APX-hard problems, which, intuitively, is the class of problems for which a PTAS would provide a PTAS for each problem in APX. The way the APX-hard class is defined is through the use of PTAS-reductions. A PTAS-reduction from a problem A to a problem B consists of three polynomial-time computable functions f, g and α as follows:

- From an instance of A, the function f produces an instance of B;
- The function $\alpha : \mathbb{Q}_+ \to \mathbb{Q}_+$ converts between the error parameters between the two problems; and
- From an instance x of A, an error parameter ε and a solution to $f(x)$ of cost at most $1 + \alpha(\varepsilon)$ times that of an optimal solution (to $f(x)$), the function g produces a solution to the instance x of cost at most $1 + \varepsilon$ times the cost of an optimal solution (to x).

The class APX-hard is then defined as the class of problems to which every problem in APX has a PTAS-reduction. Because PTAS-reductions can be composed into PTAS-reductions, to show that a problem is APX-hard, it is



(a) (b) (c)

Figure 1.1: Optimal road network layouts when (a) $\alpha \geq \sqrt{2}-1$, when (b) $\alpha < \sqrt{2}-1$ and $\beta \leq 1-2h$ and when (c) $\alpha < \sqrt{2}-1$ and $\beta > 1-2h$.

enough to reduce a problem already known to be APX-hard to the candidate problem. We do this in Chapter 3, but we use a simpler type of PTAS reduction known as an L-reduction, which we cover in more detail in Section 3.2.

Approximation algorithms are one way to handle our inability to solve NP-hard problems in polynomial time. Fixed Parameter Tractable (FPT) algorithms are another approach. In this setting, problem instances have an associated parameter, which can be thought of as a natural number representing the difficulty of the instance. A fixed parameter tractable algorithm will solve any instance of parameter at most k in polynomial time for any value of k. For more details, see Section 3.2.

For the LSC problem, we achieve three results:

- We use the local search technique of Mustafa and Ray [41] to provide a PTAS for the variant of this problem where we can only select segments from a given orientation;
- We complement the work of Korman et al. [33], using a similar kernelization technique to show that the general LSC problem is fixed-parameter tractable;
- Finally, we show that a slight variation of the LSC problem, where we need to cover only the rectangular faces, is APX-hard.

This work has been published at the 2018 Conference on Combinatorial Optimization and Applications and is included as Chapter 3.

Chapter 4 is based on ongoing work about planar road network design. Suppose travelers would like to move from points in a source set to points in a destination set and we have been allotted a budget $\beta > 0$ to build a road network to improve these trips. More precisely, traveling outside the roads takes one unit of time per unit of distance, while going along a road is faster, taking only α units of time per unit of distance ($0 \leq \alpha < 1$); the total length of the roads should be at most the budget; and we would like to minimize the longest trip, i.e., to make sure that the maximum time it takes to travel from any source point to any destination point (along a fastest path) is as short as possible.

Our main focus is when the source and destination sets are opposing sides of a unit square. When the budget is at most $\sqrt{2}$, we were able to show that the layouts in Figure 1.1, which is a copy of Figure 4.1, are optimal, where, when $\alpha < \sqrt{2}-1$,

$$h = \frac{1+\alpha}{2\sqrt{4-(1+\alpha)^2}} \in \left[\frac{1}{2\sqrt{3}}, \frac{1}{2}\right)$$

is a value depending on α, but not on β, that regulates the shape of the layout.

We also consider the variation where the source and destination sets are both the boundary of a unit circle, for which we disprove the natural conjecture that a concentric circle is an optimal network layout.

Finally, Chapter 5 is inspired by a scheduling problem posed by Joseph S. B. Mitchell and Esther M. Arkin. However, unlike in most scheduling problems, we are not concerned with optimizing a particular instance but with demonstrating the existence of universal upper bounds on the maximum delay of best schedules across all instances. So, essentially, we have a set of train lines consisting of a train track, which is an axis-aligned line segment in $\mathbb{R}^d$, and a direction of movement (decreasing or increasing coordinate along the axis). We would like to schedule the departure of trains of length ℓ moving at constant unit speed on each line while avoiding collisions. The main results in this chapter regard the existence of these universal upper bounds for $d = 2$ and for $d = 3$ and $\ell = 1$. We also show that several of the obtained upper bounds are tight through clique searches.

Chapter 2

More Turán-Type Theorems for Triangles in Convex Point Sets

Boris Aronov[1]

Vida Dujmović[2]

Pat Morin[3]

Aurélien Ooms[4]

Luís Fernando Schultz Xavier da Silveira[5]

Abstract

We study the following family of problems: Given a set of n points in convex position, what is the maximum number triangles one can create having these points as vertices while avoiding certain sets of *forbidden configurations*. As forbidden configurations we consider all 8 ways in which a pair of triangles in such a point set can interact. This leads to 256 extremal Turán-type questions. We give nearly tight (within a $\log n$ factor) bounds for 248 of these questions and show that the remaining 8 questions are all asymptotically equivalent to Stein's longstanding tripod packing problem.

2.1 Introduction

Let t_1 and t_2 be a pair of distinct triangles whose (4 to 6) vertices are in convex position. There are 8 combinatorially distinct ways that these triangles can interact: 2 ways in which the triangles can share an edge (and), 3 ways in which the triangles can share a single vertex (, , and), and 3 ways in which the triangles can have no vertices in common (, , and). Because it is difficult to keep track of nameless entities, we assign a mnemonic to each configuration (though the reader is encouraged to choose their own):

taco	mariposa	bat	nested	crossing	ears	swords	david

[1] Partially supported by NSF Grants CCF-11-17336, CCF-12-18791, and CCF-15-40656, and by BSF grant 2014/170. Department of Computer Science and Engineering, Tandon School of Engineering, New York University, USA.

[2] Partially supported by NSERC and the Ontario Ministry of Research and Innovation. Department of Computer Science and Electrical Engineering, University of Ottawa, Canada.

[3] Partially supported by NSERC. School of Computer Science, Carleton University, Canada.

[4] Supported by the Fund for Research Training in Industry and Agriculture (FRIA). Département d'Informatique, Université libre de Bruxelles (ULB), Belgium.

[5] School of Computer Science, University Ottawa, Canada

We consider the following class of problems: Given a set, X, of combinatorial configurations of pairs of triangles, what is the size of a largest set, S, of triangles one can create whose vertices are n points in convex position, and such that no pair of triangles in S forms a configuration in X? We call the size of such a set $\mathrm{ex}(n, X)$. For example,

$$\mathrm{ex}(n, \{\ ,\ ,\ ,\ ,\ \}) = n - 2\,. \tag{2.1}$$

This is because the set $X = \{\ ,\ ,\ ,\ ,\ \}$ forbids any form of crossings between the edges of triangles. Thus, the maximum number of triangles we can have while avoiding X is the number of triangles in a triangulation of a convex n-gon, i.e., $n-2$.

2.1.1 Previous Work

Since there are eight possible forbidden configurations, there are $2^8 = 256$ sets X for which we can study $\mathrm{ex}(n, X)$. Some of these sets have been previously studied. Braß, Rote, and Swanepoel showed that $\mathrm{ex}(n, \{\ ,\ ,\ ,\ \}) \le n$ in order to solve an Erdős problem on the maximum number of maximum area/perimeter triangles determined by a point set. Braß [10] later began a systematic study in which he gave asymptotically tight bounds on $\mathrm{ex}(n, X)$ for all singleton X and all pairs X of configurations in which two triangles share a single vertex.

Of course, upper and lower bounds are inherited through the subset relationship: $\mathrm{ex}(n, X) \le \mathrm{ex}(n, Y)$ for any $X \supseteq Y$. Table 2.1 shows the complete set of results we obtain when we apply this exhaustively to the list of previous results. Each entry in this table presents the asymptotic behaviour of $\mathrm{ex}(n, X)$ for the set X obtained as the union of the row and column label. Asymptotically tight bounds are coloured green, and gaps between lower and upper bounds are coloured red. Previous results imply 35 tight bounds for 256 of the possible choices of X. The configuration is omitted from the table since a simple argument (Lemma 2.1) shows that its inclusion in X does not change $\mathrm{ex}(n, X)$ by more than a constant factor. Some of the results in Table 2.1 are marked with an F if they are easy, or folklore. Some others are marked with H if they follow from a corresponding bound for 3-regular hypergraphs. Specifically, if X includes $\{\ ,\ ,\ \}$ then no pair of triangles can share a vertex, so $\mathrm{ex}(n, X) \le n/3$ and if X includes $\{\ ,\ \}$ then no pair of triangles can share an edge, so $\mathrm{ex}(n, X) \le \binom{n}{2}$.

2.1.2 New Results

In the current paper, we determine, up to a logarithmic factor, the asymptotics of $\mathrm{ex}(n, X)$ for 248 sets X. These results are shown in Table 2.2. For the remaining 8 sets, we have determined that the asymptotics are all the same and are equivalent to a problem that appears in various contexts and under different names, including monotone matrices, tripod packing, and 2-comparable triples. We discuss this problem and its rich history in Section 2.4.3.

The rest of this paper is organized as follows. In Section 2.2 we present a few easy results that we need for completeness. In Section 2.3 we discuss different ways of thinking about the problem. In particular, we present a series of puzzles whose solutions determine the asymptotic growth of $\mathrm{ex}(n, X)$. In Section 2.4, which represents the technical meat of the paper, we use these puzzles to derive new upper and lower bounds.

2.2 Easy Results

In this section we present an easy result that cuts our work in half by reducing the number of problems from 256 to 128. We then describe some easy lower bound constructions that are required for completeness.

2.2.1 Mariposas are Irrelevant

The following lemma shows that including the configuration in the set X of forbidden configurations has no effect on the asymptotics of $\mathrm{ex}(n, X)$.

Lemma 2.1. *For any X,* $\mathrm{ex}(n, X \cup \{\ \}) \ge \mathrm{ex}(n, X)/8$.

Proof. Let S be a set of triangles that achieves $\mathrm{ex}(n, X)$. For each pair of vertices u and w independently and uniformly choose a direction $\overrightarrow{uw}$ or $\overleftarrow{uw}$. We then obtain a set $S' \subseteq S$ by removing any triangle that has a directed edge for which the triangle is to the left of the edge. Observe that the set S' does not contain a configuration.

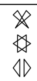
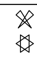
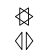
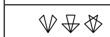
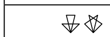
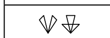
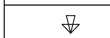
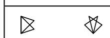
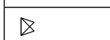
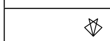
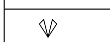
								
	1 F:F	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H
	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]
	n T2.3:[11]	n T2.3:[11]	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H
	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H
	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n H:H
	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	n^2 [10]:[10]
	n T2.3:[11]	n [11]:[11]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ [11]:[10]	$n : n^2$ T2.3:[10]	n^2 [10]:[10]
	$n : n^2$ T2.3:[10]	$n : n^2$ [11]:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ [11]:[10]	$n : n^2$ T2.3:[10]	n^2 [10]:[10]
	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]
	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]
	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H
	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	$n : n^2$ T2.3:H	n^2 H:H
	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	n^2 [10]:[10]
	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	n^2 [10]:[10]
	$n : n^2$ T2.3:[10]	$n : n^2$ [11]:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^2$ T2.3:[10]	$n : n^3$ [11]:F	$n : n^2$ T2.3:[10]	n^3 [10]:[10]
	n^2 H:H	n^2 H:[10]	n^2 H:[10]	n^2 [10]:[10]	n^2 H:[10]	n^3 [10]:[10]	n^2 [10]:[10]	n^3 F:F

Table 2.1: Previous lower and upper bounds for $\mathrm{ex}(n, X)$. T2.3 denotes a (easy) lower bound of $\Omega(n)$ that appears in Theorem 2.3. F denotes an obvious, or folklore result. H denotes a bound that follows from the corresponding bound on 3-regular hypergraphs.

	1 F:F	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H
	n^* T2.3:T2.5	n^* T2.3:T2.5	n^* T2.3:T2.5	n^* T2.3:T2.5	n^* T2.3:T2.5	n^* T2.3:T2.5	n^* T2.3:T2.5	n^* T2.3:**T2.5**
	n T2.3:[11]	n T2.3:[11]	n^* T2.3:T2.11	n^* T2.3:T2.13	n^* T2.3:T2.11	tripods	n^* T2.3:T2.11	tripods
	n^* T2.3:T2.8	n^* T2.3:T2.13	n^* T2.3:T2.8	n^* T2.3:T2.13	n^* T2.3:T2.8	tripods	n^* T2.3:**T2.8**	tripods
	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n T2.3:H	n H:H
	n^* T2.3:T2.6	n^* T2.3:T2.6	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.6	n^* T2.3:**T2.6**	n^2 **T2.20**:[10]	n^2 [10]:[10]
	n T2.3:[11]	n [11]:[11]	n^* T2.3:T2.11	n^* T2.3:T2.14	n^* T2.3:T2.11	n^2 **T2.21**:[10]	n^* T2.3:**T2.11**	n^2 [10]:[10]
	n^* T2.3:T2.10	n^* [11]:T2.14	n^* T2.3:T2.14	n^* T2.3:**T2.14**	n^* T2.3:**T2.10**	n^2 T2.21:[10]	n^2 T2.20:[10]	n^2 [10]:[10]
	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^2 **T2.18**:[10]	n^2 T2.18:[10]	n^2 T2.18:[10]	n^2 T2.18:[10]
	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^2 T2.18:[10]	n^2 T2.18:[10]	n^2 T2.18:[10]	n^2 T2.18:[10]
	n^* T2.3:T2.13	n^* T2.3:T2.13	n^* T2.3:T2.13	n^* T2.3:T2.13	n^2 T2.18:H	n^2 T2.18:H	n^2 T2.18:H	n^2 T2.18:H
	n^* T2.3:T2.13	n^* T2.3:T2.13	n^* T2.3:T2.13	n^* T2.3:**T2.13**	n^2 T2.18:H	n^2 T2.18:H	n^2 T2.18:H	n^2 H:H
	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^2 T2.18:[10]	n^2 T2.18:[10]	n^2 T2.18:[10]	n^2 [10]:[10]
	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:T2.9	n^* T2.3:**T2.9**	n^2 T2.18:[10]	n^2 T2.18:[10]	n^2 T2.20:[10]	n^2 [10]:[10]
	n^2 **T2.19**:[10]	n^2 T2.19:[10]	n^2 T2.19:[10]	n^2 T2.19:[10]	n^2 T2.18:[10]	n^3 **T2.2**:F	n^2 T2.18:[10]	n^3 [10]:[10]
	n^2 H:H	n^2 H:[10]	n^2 H:[10]	n^2 [10]:[10]	n^2 H:[10]	n^3 [10]:[10]	n^2 [10]:[10]	n^3 F:F

$n^* = n : n \log n$ tripods $= n^{1.546} : n^2/e^{\Omega(\log^* n)}$

Table 2.2: New and previous bounds for $\mathrm{ex}(n, X)$, up to a factor of $\log n$. New near-optimal results are in dark(er) green. TX denotes Theorem X in this paper and [X] denotes reference X in this paper. For example T16:[10] denotes a lower bound that appears in Theorem 16 and an upper bound due to Braß [10].

For any particular triangle $t \in S$, the probability that $t \in S'$ is exactly $1/8$ since each of t's three edges must be directed clockwise and edge directions are chosen independently. By linearity of expectation, $\mathrm{E}[|S'|] = |S|/8 = \mathrm{ex}(n, X)/8$. We conclude therefore that there exists some subset $S'' \subseteq S$ of size least $\mathrm{ex}(n, X)/8$ that does not contain a ⟠ configuration. The set S'' proves that $\mathrm{ex}(n, X \cup \{⟠\}) \geq \mathrm{ex}(n, X)/8$. ☐

2.2.2 Cubic-Sized Sets of Pairwise Crossing Triangles

Theorem 2.2. $\mathrm{ex}(n, \{◁▷, ⩔, ⟠\}) \in \Omega(n^3)$.

Proof. Partition the vertices of the convex n-gon into three contiguous sets, A, B, and C, each of size $\lfloor n/3 \rfloor$ or $\lceil n/3 \rceil$, as appropriate. Consider the set, S, of all triangles having one vertex in each of A, B, and C. It is easy to check that any two triangles in S have a pair of edges that cross, thus they do not form any of ◁▷, ⩔, or ⟠. Furthermore, $|S| \geq \lfloor n/3 \rfloor^3 \in \Omega(n^3)$, so $\mathrm{ex}(n, \{◁▷, ⩔, ⟠\}) \in \Omega(n^3)$. ☐

2.2.3 Linear-Sized Sets Using Only a Single Configuration

Since it is not explicitly stated in previous work, and we need it to complete our table, we now observe that for any configuration $x \in \{$⧐, ⩔, ⩡, ⟡, ◁▷, ⨯, ✡$\}$, one can create a linear-sized set of triangles that avoids all configurations except x.

Theorem 2.3. *For any* $X \subsetneq \{$⧐, ⩔, ⩡, ⟡, ◁▷, ⨯, ✡$\}$, $\mathrm{ex}(n, X) \in \Omega(n)$.

Proof. Let $x \in \{$⧐, ⩔, ⩡, ⟡, ◁▷, ⨯, ✡$\}$ be a configuration not in X. Label the vertices of our convex n-gon $1, \ldots, n$ in counterclockwise order. Depending on the value of x, we use one of the following constructions (see Figure 2.1):

1. For $x =$ ⧐, we use the set of triangles $\{(1, 2, i) : i \in \{3, \ldots, n\}\}$.
2. For $x =$ ⩔, we use the set of triangles $\{(1, 2i, 2i + 1) : i \in \{1, \ldots, \lfloor n/2 \rfloor - 1\}\}$.
3. For $x =$ ⩡, we use the set of triangles $\{(1, i, n + 2 - i) : i \in \{2, \ldots, \lfloor n/2 \rfloor\}\}$.
4. For $x =$ ⟡, we use the set of triangles $\{(1, i, \lfloor n/2 \rfloor + i) : i \in \{2, \ldots, \lfloor n/2 \rfloor\}\}$.
5. For $x =$ ◁▷, we use the set of triangles $\{(3i - 2, 3i - 1, 3i) : i \in \{1, \ldots, \lfloor n/3 \rfloor\}\}$.
6. For $x =$ ⨯, we use the set of triangles $\{(i, \lfloor n/3 \rfloor + 2i - 1, \lfloor n/3 \rfloor + 2i) : i \in \{1, \ldots, \lfloor n/3 \rfloor\}\}$.
7. For $x =$ ✡, we use the set of triangles $\{(i, \lfloor n/3 \rfloor + i, \lfloor 2n/3 \rfloor + i) : i \in \{1, \ldots, \lfloor n/3 \rfloor\}\}$.

In each case, the size of the set is $\Omega(n)$ and it is straightforward to verify that each pair of triangles in the set forms the configuration x and therefore avoids all configurations in X. ☐

2.3 Points of View

In this section we describe different, but equivalent (up to a logarithmic factor), views of the problem. One of these (the dot puzzle view) will be our main line of attack for the most difficult cases.

2.3.1 The Top/Bottom View

It will be helpful to gain a sense of orientation by considering a top/bottom variant of $\mathrm{ex}(n, X)$ that is defined as follows (see Figure 2.2). Partition the vertices of a convex n-gon using a horizontal line into a *top half* of size $\lceil n/2 \rceil$ and a *bottom half* of size $\lfloor n/2 \rfloor$. We define $\mathrm{ex}'(n, X)$ analogously to $\mathrm{ex}(n, X)$ except that we only count triangles having one vertex in the bottom half and two vertices in the top half. When studying ex', each triangle we count has a naturally defined *bottom vertex* in the bottom half and a *left vertex* and *right vertex*, each in the top half.

Clearly $\mathrm{ex}(n, X) \geq \mathrm{ex}'(n, X)$. The following lemma shows that, without losing much precision, we can also upper bound $\mathrm{ex}(n, X)$ by $\mathrm{ex}'(n, X)$.

Lemma 2.4. *If* $\mathrm{ex}'(n, X) \in O(n^c)$, *then*

$$\mathrm{ex}(n, X) \in \begin{cases} O(n^c) & \textit{if } c > 1, \\ O(n \log n) & \textit{if } c = 1. \end{cases}$$

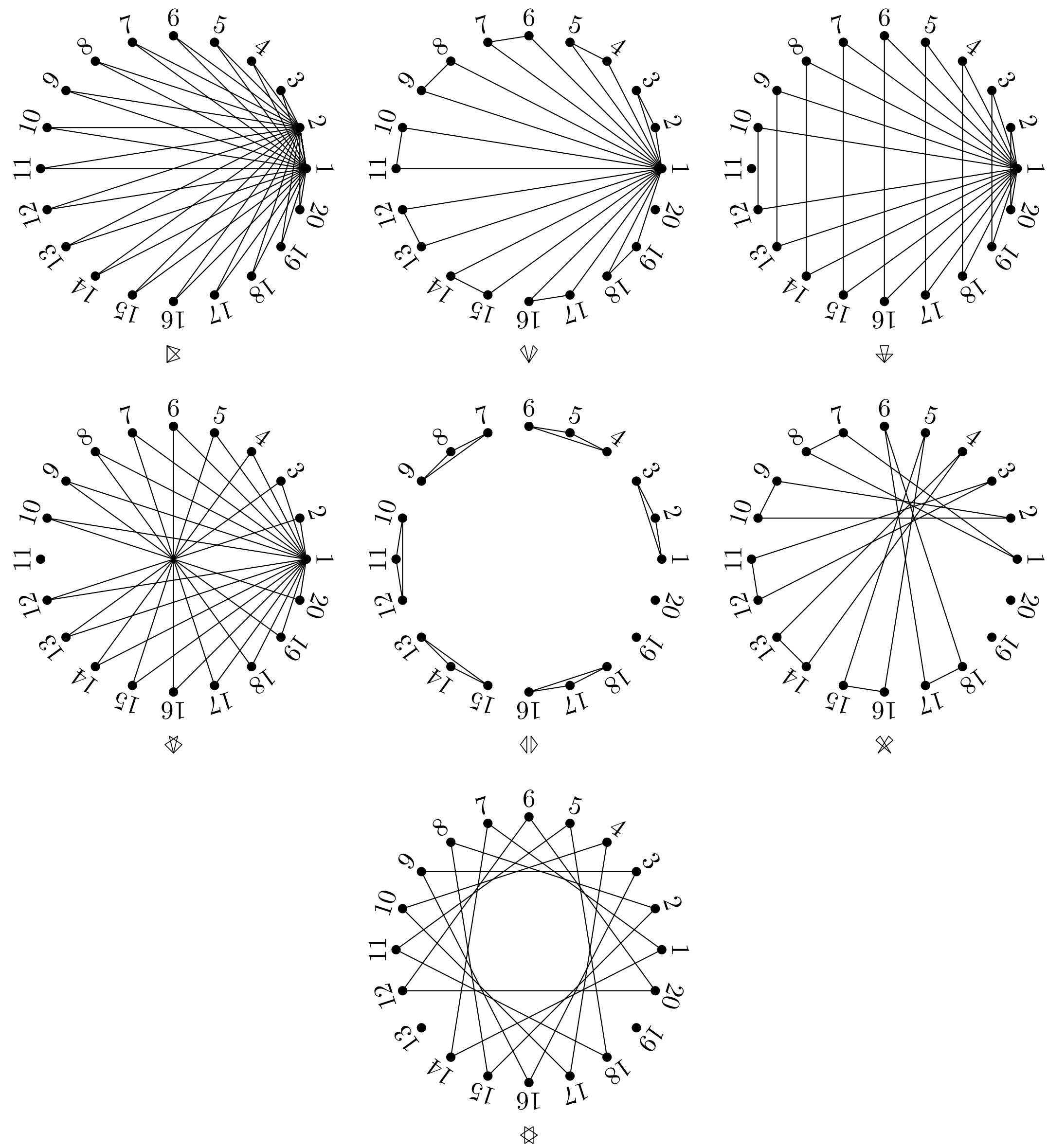


Figure 2.1: Constructions used in the proof of Theorem 2.3.

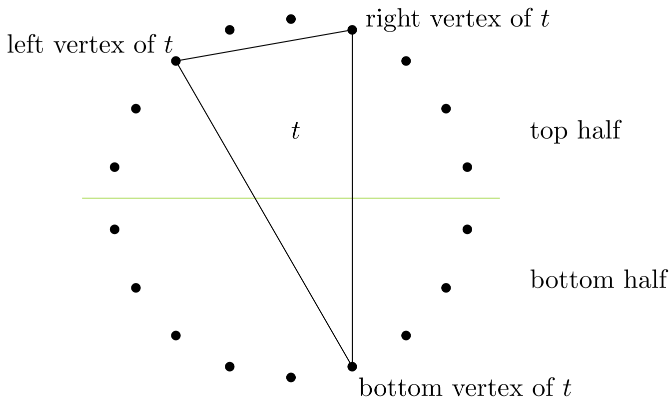


Figure 2.2: ex$'$ only counts triangles with two vertices in the top half and one vertex in the bottom half.

Proof. Let S be a set of triangles that avoids X. Every triangle in S is of one of the following types:

1. It has one vertex in the top half and two in the bottom half; there are $O(n^c)$ such triangles.
2. It has two vertices in the top half and one in the bottom half; there are $O(n^c)$ such triangles.
3. It has all three vertices in the top half; there are at most $\mathrm{ex}(\lceil n/2 \rceil, X)$ such triangles.
4. It has all three vertices in the bottom half; there are at most $\mathrm{ex}(\lfloor n/2 \rfloor, X)$ such triangles.

Thus, we obtain the recurrence inequality

$$\mathrm{ex}(n, X) \le O(n^c) + \mathrm{ex}(\lceil n/2 \rceil, X) + \mathrm{ex}(\lfloor n/2 \rfloor, X)\,,$$

which solves to $O(n^c)$ for $c > 1$ and $O(n \log n)$ for $c = 1$ [19, Section 4.3]. $\square$

2.3.2 The Dot Puzzle View

The top-bottom version of the problem gives us a sense of orientation, but it is still difficult to visualize the sets of triangles obtained this way. Next, we show that there is a corresponding puzzle that is easy to visualize. Refer to Figure 2.3.

In this puzzle, we are given $\binom{n}{2}$ points,

$$Q = \{(x, y) : y \in \{1, \dots, n-1\}, x \in \{y+1, \dots, n\}\}\,.$$

These points model the top/bottom view on a convex $2n$-gon, where the point (x, y) represents a triangle whose vertices are some point on the bottom and the xth and yth points on the top, where the top vertices are labelled $1, \dots, n$ from left to right.

The dot puzzle proceeds in n rounds and during the ith round, the player selects a set $Q_i \subseteq Q$ subject to certain constraints that depend on the points selected in rounds $1, \dots, i-1$. In the top/bottom view, the ith round determines which pairs of top vertices form a triangle with the ith bottom vertex, where the bottom vertices are labelled $1, \dots, n$ from right to left.

Of course, the constraints on which points can be selected during round i depend on the set of forbidden configurations and the set $\bigcup_{j=1}^{i-1} Q_j$ of points played during previous rounds. By proving bounds on $\sum_{i=1}^{n} |Q_i|$ we obtain bounds on the maximum number of triangles obtained in the top-bottom view on a set of $2n$ points, i.e., bounds on $\mathrm{ex}'(2n, X)$.

Figure 2.4(a) shows restrictions on the locations of points placed during a single round. It is interpreted as follows: If the central point, $p = (x, y)$, is placed during round i, and we wish to avoid some particular configuration, c, then we should not place any points in the parts of the figure that have label c. For instance, if we wish to avoid the $c = ▷$ configuration, then we should not place any points in the same row or column as p; such a point creates a ▷ configuration in which the shared edge joins a bottom vertex to a left (same row) or right (same column) vertex.

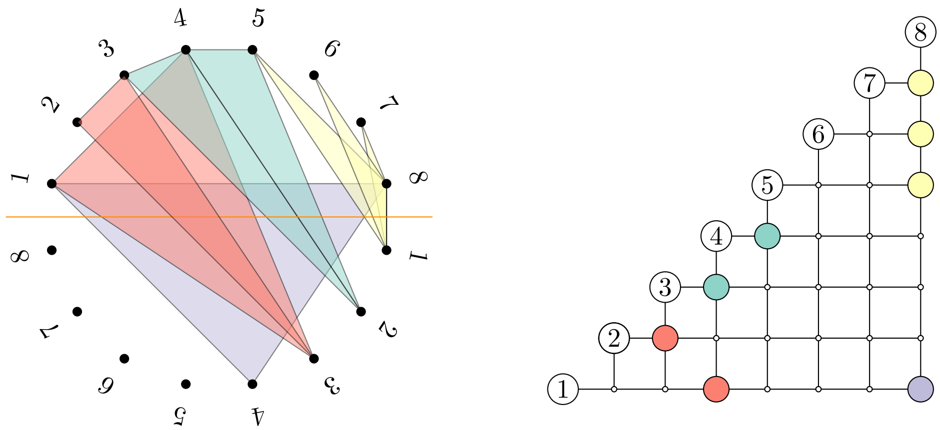


Figure 2.3: The dot puzzle view of the top/bottom view. In this example, four rounds of the Dot Puzzle have been played.

Figure 2.4(b) shows the restrictions on the locations of points placed in subsequent rounds. Its interpretation is similar Figure 2.4(a). For example, if we wish to avoid a ⩡ configuration and we place the central point, p, during round i, then, in every round $j > i$, we should not place any point directly to the left or directly below p. Any such point creates a ⩡ configuration in which the shared vertex is the left vertex (to the left of p) or the right vertex (below p) of both triangles.

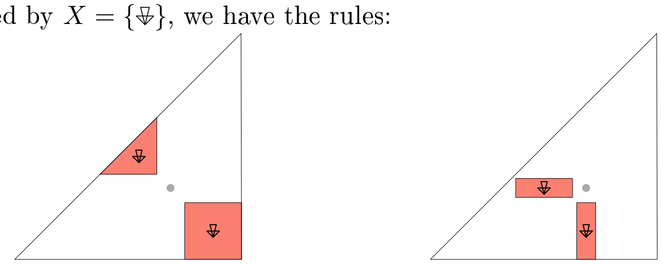
For any $X \subseteq \{$⧐, ⟠, ⩔, ⩡, ⟱, ⧏⧐, ⨳, ⟡$\}$ and any $S \subseteq Q$, we define $\kappa(X, S)$ as the subset of Q that can no longer be played in the dot puzzle game (for configurations in X) if the points in S have been played in previous rounds (these are the points of Q κilled by S). We use the complementary notation $\overline{\kappa}(X, S) = Q \setminus \kappa(X, S)$ to be the subset of points in Q that can still be played in the dot puzzle game if the points is S have been played in previous rounds.

2.3.3 Some Warm-Up Exercises

For the remainder of the paper, we will study ex$'$ using the dot puzzle view. Thus, all of our results are bounds on solutions to these dot puzzles.

We say that a point set is *non-decreasing* (respectively, non-increasing) if, when sorted lexicographically, the y-coordinates of the points form a non-decreasing (respectively, non-increasing) sequence. A point set is *increasing* (respectively, decreasing) if it is non-decreasing (respectively, non-increasing) and no two of its points points have the same x-coordinate or the same y-coordinate.

From Figure 2.4, some previous upper bounds naturally fall out. Consider Braß's results [10] that ex$(n, \{$⩡$\}) \in O(n^2)$. For the game defined by $X = \{$⩡$\}$, we have the rules:



In particular, these rules imply that points selected during a single round of the dot puzzle must be non-decreasing, and thus at most $2n-3$ points can be selected take part in Q_i. Thus $\sum_{j=1}^{n} |Q_i| \leq 2n^2 - 3n$, so ex$'(n, \{$⩡$\}) \in O(n^2)$ and the bound ex$(n, \{$⩡$\}) \in O(n^2)$ immediately follows from Lemma 2.4.

Similarly, we can almost recover the result ex$(n, \{$⧏⧐, ⨳, ⩔, ⩡$\}) \leq n$ of Braß, Rote and Swanepoel [11]. Here, the rules are:

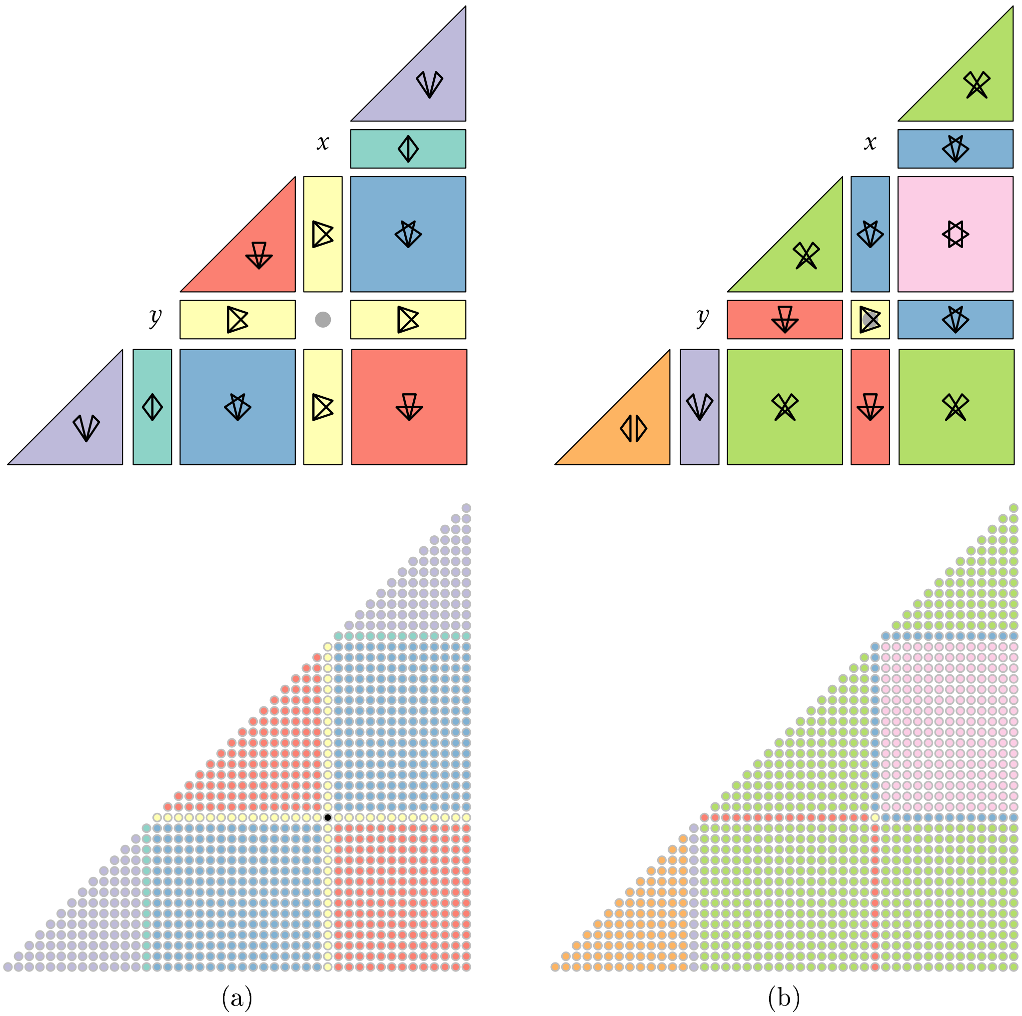
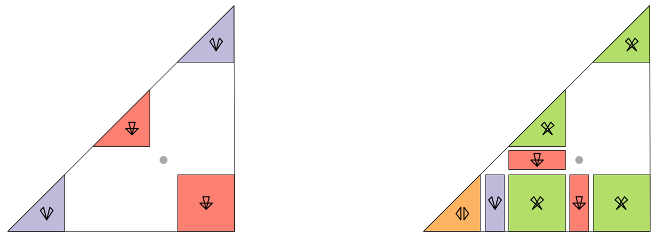


Figure 2.4: The regions killed by forbidden configurations during (a) the current round and (b) subsequent rounds.



The rule for ⟱ ensures that the set of points taken during a single round form a non-decreasing point set. The rules for points allowed in subsequent rounds ensure that, after round i any points chosen are not below or to the left of the topmost-rightmost point in Q_i. Taken together, these rules imply that

$$\mathrm{ex}'(n, \{⧓, ⨳, ⩒, ⟱\}) \leq \sum_{i=1}^{n} |Q_i| \leq 3n - 4\,,$$

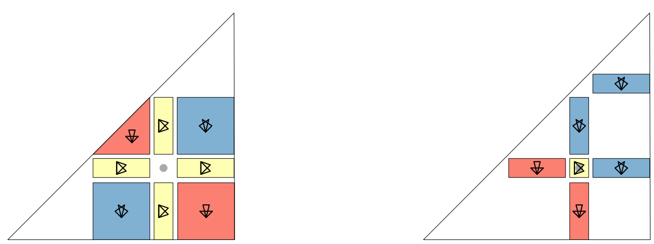
since the union of Q_i is a non-decreasing point set (whose size is therefore at most $2n - 3$), and each Q_i shares at most one point with Q_{i+1}. The bound $\mathrm{ex}(n, \{⧓, ⨳, ⩒, ⟱\}) \in O(n \log n)$ then follows from Lemma 2.4.

2.4 Results Based on Dot Puzzles

After this warm-up, and armed with the dot puzzle view, we are ready to prove some new results. We begin by proving several linear upper bounds. From this point on, each proof of a theorem will begin with a picture, similar to Figure 2.4, that shows the rules of the dot puzzle considered by the theorem. For a point $q \in Q$, we use the notations $\mathrm{x}(q)$ and $\mathrm{y}(q)$ to denote the x- and y-coordinates of q.

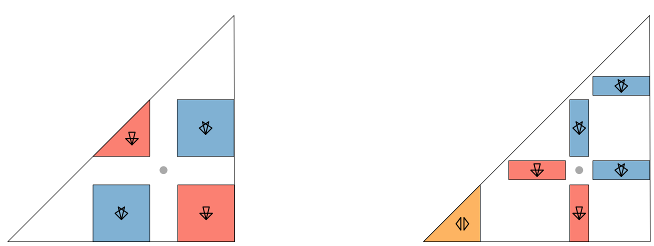
2.4.1 Linear Upper Bounds

Theorem 2.5. $\mathrm{ex}'(n, \{⧐, ⟱, ⟠\}) \in O(n)$.



Proof. Taking the union of the rules for ⧐, ⟱, and ⟠, we obtain the rule which ensures that during subsequent rounds we can not take a point from any column or row used in a previous round. The rules for ⧐ ensure that the set of points taken in each Q_i includes at most one point in each row (or column). Therefore each new point played can be charged to a unique row, so the total number of points played is at most n. □

Theorem 2.6. $\mathrm{ex}'(n, \{⟱, ⟠, ⧓\}) \in O(n)$.



Proof. Observe that if Q_i contains $k > 1$ points, $p_1, \ldots, p_k$ in a single column (or row), then each of the k rows (or columns) containing one of these points is completely covered by $\kappa(\{⟱, ⟠\}, \{p_1, \ldots, p_k\})$, i.e., $Q_{i+1}, \ldots, Q_n$ can not contain any points in these rows (or columns) (see Figure 2.5). Therefore, when summing $\sum_{i=1}^{n} |Q_i|$, the contribution of points that are not alone in their row or column is at most $2n$. We therefore assume that each Q_i contains at most one point from each row and column. Since the rules for $\triangledown$ imply that Q_i is non-decreasing, this implies that each Q_i is an increasing set of points.

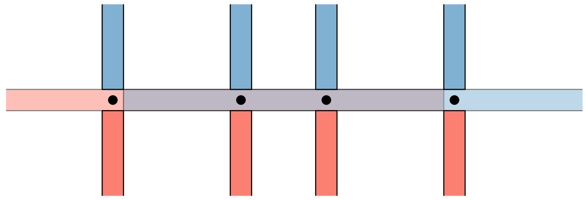


Figure 2.5: A step in the proof of Theorem 2.6.

Let $S = \bigcup_{i=1}^{n} Q_i$ and notice that S contains at most one point from each row: each Q_i contains at most one point in each row and the first time a point p appears in some row, $\kappa(\{\triangledown, \diamondsuit\}, \{p\})$ eliminates every other point from that row. Therefore $|S| \leq n$. All that remains is to account for multiplicity; a single point in S can appear in more than one Q_i.

Now, because of the rules for $\diamondsuit$, the condition that Q_i is increasing is quite restrictive. In particular, if we consider the last (top rightmost) point, p, of Q_i, then it must be placed so that $\kappa(\{\triangleleft\!\triangleright\}, \{p\})$ contains all of Q_i except p and the second-to-last point in Q_i (see Figure 2.6). That is, Q_i contains $|Q_i| - 2$ points that cannot appear in $Q_{i+1}, \ldots, Q_n$. We can think of Q_i as eliminating $|Q_i| - 2$ points from S, so $\sum_{i=1}^{n}(|Q_i| - 2) \leq |S| \leq n$, so $\sum_{i=1}^{n} |Q_i| \leq 3n$. $\square$

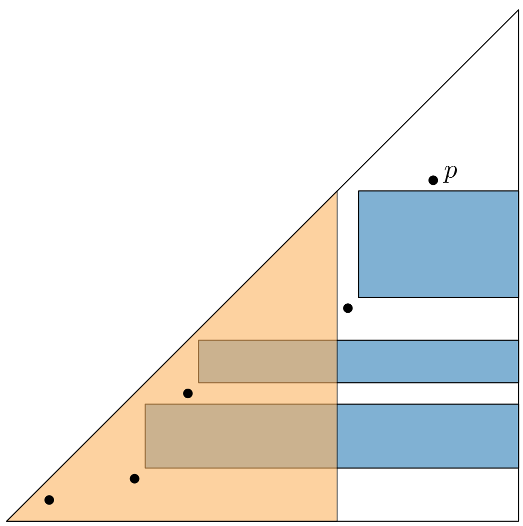


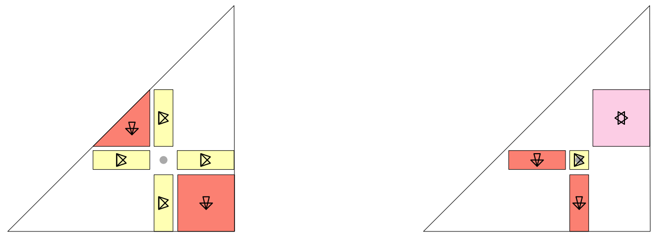
Figure 2.6: Another step in the proof of Theorem 2.6.

Our next four upper bounds depend on a simple lemma about forbidden configurations of points. We say that three points $a = (x_0, y_0)$, $b = (x_0, y_1)$, and $c = (x_1, y_1)$ form a Γ-*configuration* if $y_0 < y_1$ and $x_0 < x_1$.

Lemma 2.7. *Let S be a subset of $\{1, \ldots, n\}^2$ with no three points a, b, and c that form a Γ-configuration. Then $|S| \leq 2n$.*

Proof. If we remove the rightmost point from each row of S, then each column in what remains of S contains at most one point. Otherwise, we could take a to be the lowest point in a column, b to be the highest point in the same column, and c to be the removed rightmost point in b's row. $\square$

Theorem 2.8. $\mathrm{ex}'(n, \{\triangleright, \triangledown, \hexstar\}) \in O(n)$.



Proof. Let $S = \bigcup_{i=1}^{n} Q_i$ be the set of points played in a solution to the resulting dot puzzle. Note that, by the inclusion of ⊳, each element of S appears in exactly one Q_i, so $|S| = \sum_{i=1}^{n} |Q_i|$ is the quantity we are interested in bounding.

Next, we claim that S does not contain any three points a, b, and c forming a Γ-configuration. Refer to Figure 2.7. Suppose, for the sake of contradiction, that this were not the case and that $a \in Q_i$, $b \in Q_j$, and $c \in Q_k$. We have $a \in \kappa(\{⊽\}, \{b\})$ so $i \leq j$ and the rules for ⊳ implies $i \neq j$, so $i < j$. We also have $b \in \kappa(\{⊽\}, c)$ so $j \leq k$. Finally, we have $c \in \kappa(\{✧\}, a)$, so $k \leq i$. Taken together, this gives the contradiction $i < j \leq k \leq i$. Therefore, S contains no Γ-configuration and applying Lemma 2.7 then implies that $|S| \leq 2n$. $\square$

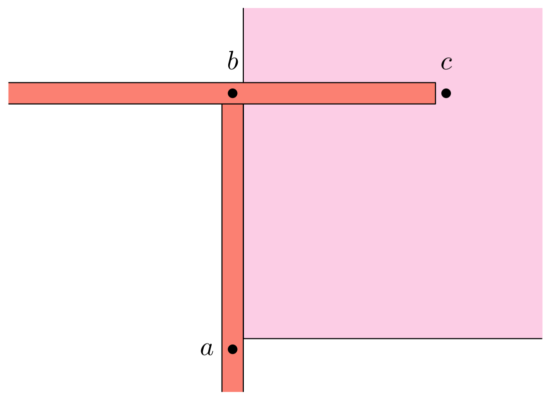
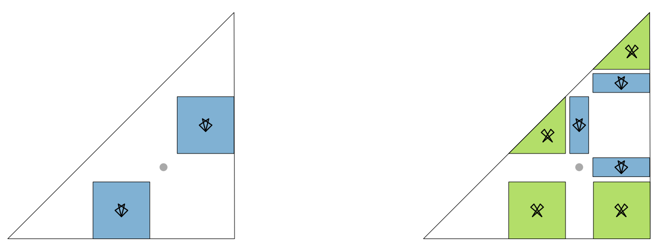


Figure 2.7: The proof of Theorem 2.8.

Theorem 2.9. $\mathrm{ex}'(n, \{◊, ✕\}) \in O(n)$.



Proof. Let $S = \bigcup_{i=1}^{n} Q_i$. We claim that S has no Γ-configuration so, by Lemma 2.7, $|S| \leq 2n$. To see this, assume S contains a Γ-configuration a, b, and c. Then the rules for ◊ imply that no set Q_i contains both a and c. However, the rules for ✕ and ◊ imply that, if $a \in Q_i$, $b \in Q_j$ and $c \in Q_k$ then

$$i \geq j \geq k \geq i\,.$$

(See Figure 2.8). But this is a contradiction, since it implies that $i = j = k$.

Now, for some Q_i, consider a point $p \in Q_i$ with minimum x-coordinate and, in case more than one such point exists, take the the one that minimizes $\mathrm{y}(p)$. Observe that $\kappa(\{◊, ✕\}, \{p\}) \supseteq Q_i \setminus \{p\}$. Indeed, every point directly above and every point to the right of p is killed by p or cannot be included in Q_i because of the rule for ◊. Therefore, Q_i eliminates at least $|Q_i| - 1$ points of S. It follows that $\sum_{i=1}^{n} |Q_i| \leq |S| + n \leq 3n$. $\square$

Theorem 2.10. $\mathrm{ex}'(n, \{⊽, ◁▷, ✧\}) \in O(n)$.

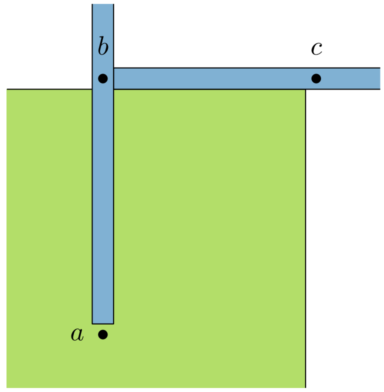
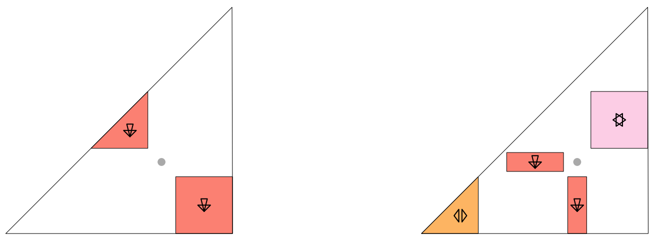


Figure 2.8: The proof of Theorem 2.9.



Proof. Let $S = \bigcup_{i=1}^{n} Q_i$. We will first show that, for each $i \in \{1,\dots,n\}$, there are at most two points of Q_i that appear in $Q_{i+1},\dots,Q_n$. Refer to Figure 2.9. Define

$$Q_i^* = \{(x,y) \in Q_i : x \geq \max\{y(p) : p \in Q_i\}\},$$

and let p and r be the top rightmost and bottom leftmost points in Q_i^*, respectively. If there is more than one point in Q_i^* with x-coordinate equal to $\mathrm{x}(r)$ (as in Figure 2.9(a)) then we define q to be the highest such point (note that this includes the case where $p = r$). Otherwise (as in Figure 2.9(b)), we define q to be the rightmost point with y-coordinate $\mathrm{y}(r)$ (note that this includes the case where $q = r$). Now, observe that $\kappa(\{⊽, ⧓, ✡\}, \{p,q,r\}) \supseteq Q_i \setminus \{p,q\}$, so p and q are the only points of Q_i that can appear again in $Q_{i+1},\dots,Q_n$. (Note that Figure 2.9 only illustrates the case in which $y(p) = x(r)$; if $y(p) < x(r)$, then even p is contained in $\kappa(\{⊽, ⧓, ✡\}, \{p,q,r\}$.)

We can therefore think of Q_i as eliminating $|Q_i| - 2$ points from S, so $\sum_{i=1}^{n}(|Q_i| - 2) \leq |S|$, which implies that $\sum_{i=1}^{n} |Q_i| \leq |S| + 2n$. All that remains now is to bound $|S|$.

For each $i \in \{1,\dots,n\}$, let Q'_i be obtained from Q_i by removing the leftmost point in each row. Let $S' = \bigcup_{i=1}^{n} Q'_i$. We claim that S' contains no Γ-configuration so, by Lemma 2.7, $|S'| \leq 2n$. To see why this is so, suppose that S' contains a Γ-configuration a, b, and c. Then, as argued in the proof of Theorem 2.8, it must be that $a, b, c \in Q'_i$ for some i. However, this contradicts the fact (due to ⊽) that Q_i is non-decreasing since the leftmost point, b', of Q_i in the same row as b is to the left of a (see Figure 2.10).

Therefore, $|S'| \leq 2n$. Now let $S'' = S \setminus S'$. We claim that S'' also satisfies the conditions of Lemma 2.7. Indeed, by the same reasoning as above, if there were $a, b, c \in S''$ forming a Γ-configuration, then it must be that $a, b, c \in Q_i \setminus Q'_i$ for some i. But this is a contradiction since b and c are in the same row, and $Q_i \setminus Q'_i$ contains at most one point per row.

Wrapping up, we have $|S| = |S'| + |S''| \leq 4n$ so $\sum_{i=1}^{n} |Q_i| \leq 6n$. □

Theorem 2.11. $\mathrm{ex}'(n, \{⊽, ⩔, ✡\}) \in O(n)$.

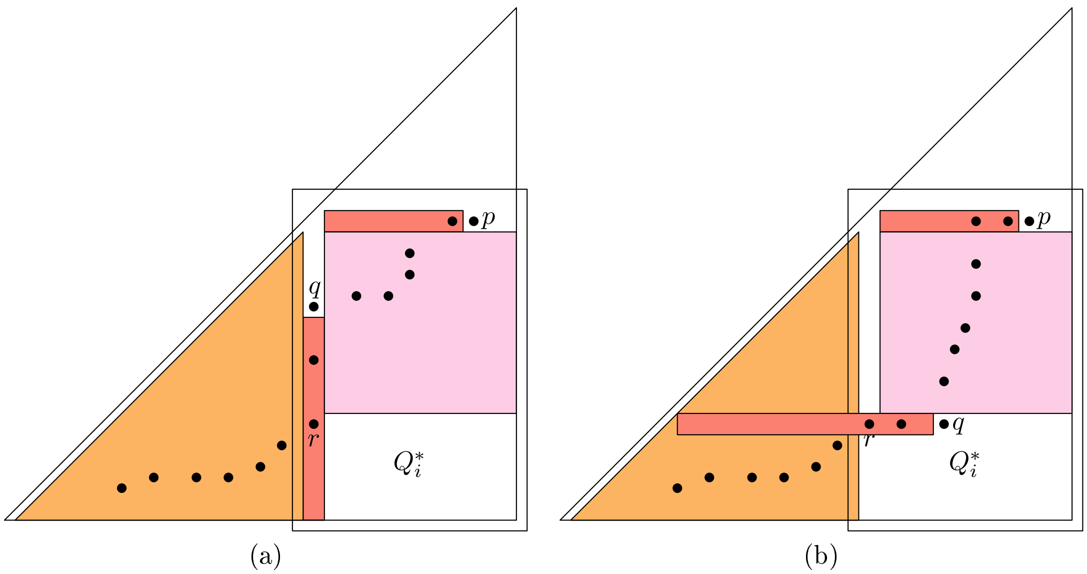


Figure 2.9: A step in the proof of Theorem 2.10.

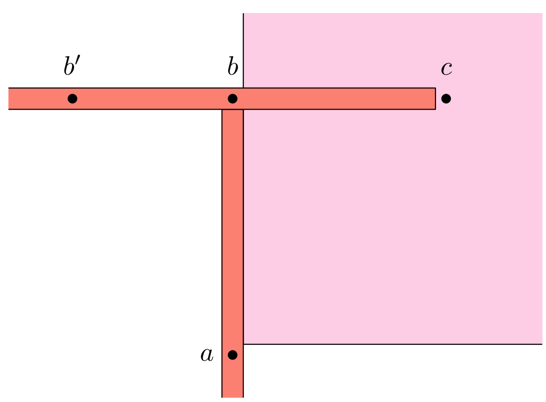


Figure 2.10: Another step in the proof of Theorem 2.10.

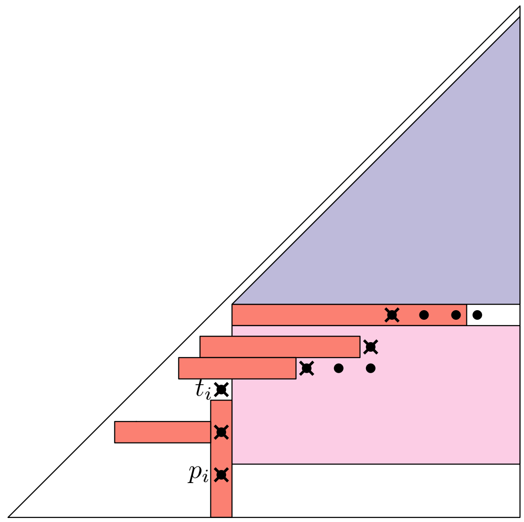
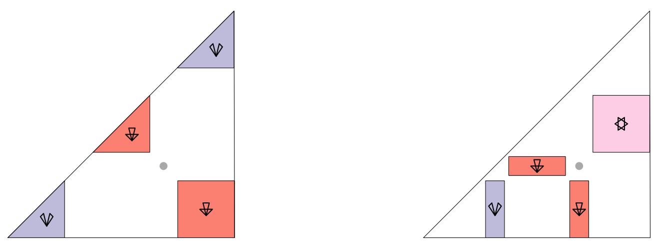


Figure 2.11: The proof of Theorem 2.11.



Proof. Consider the set Q_i played during some round i. The rules for ⇓ imply that Q_i is non-decreasing. Consider the following subsets of Q_i:

1. the set L_i of points in the leftmost column of Q_i;
2. the set T_i of points in the topmost row of Q_i;
3. the set B_i of points in the bottommost row of Q_i; and
4. the set $N_i = Q_i \setminus (L_i \cup T_i \cup B_i)$.

Let p_i denote the lowest leftmost point of Q_i (the unique point in $B_i \cap L_i$). The rule for ⩛ implies that every point of N_i is contained in $\kappa(\{✡\}, \{p_i\})$.

Refer to Figure 2.11. Observe that, for any point $p \in N_i$, the entire row containing p is killed in the sense that it is contained in $\kappa(\{✡\}, \{p_i\}) \cup \kappa(\{⇓\}, \{p\})$. Next, let t_i be the topmost point in L_i and observe that, for any point $p \in L_i \setminus \{t_i, p_i\}$, the entire row containing p is killed by $\kappa(\{✡\}, \{p_i\}) \cup \kappa(\{⇓\}, \{t_i, p\})$.

Consider the operation of removing the leftmost point from each row of Q_i, for each $i \in \{1, \dots, n\}$. We claim that this removes a total of at most $4n$ points from $Q_1, \dots, Q_n$. Indeed, by the preceding discussion, if we remove a point $p \in N_i$ or $p \in L_i \setminus \{t_i, p_i\}$ then this point can be charged to a row that is never used again in $Q_{i+1}, \dots, Q_n$. For each round $i \in \{1, \dots, n\}$, there are only three other choices for p: $p = p_i$, $p = t_i$, or p is the leftmost point in T_i. Thus, we can charge each row for removing at most one point and each round for removing at most 3 points.

For each $i \in \{1, \dots, n\}$, let Q'_i be the subset of Q_i obtained by removing the leftmost point in each row and observe that Q'_i is an increasing set of points. By the preceding discussions $\sum_{i=1}^{n} |Q_i| \le 4n + \sum_{i=1}^{n} |Q'_i|$. Let $S = \bigcup_{i=1}^{n} Q'_i$. We claim that S contains no Γ-configuration. To see why this is so, observe that, since each Q'_i is an increasing set, if two points $a \in Q'_i$ and $b \in Q'_j$ of S are in the same column then $i \ne j$. Assume b is above a, then $a \in \kappa(\{⇓\}, \{b\})$ so $i < j$. Now, if $c \in Q'_k$ is to the right of b, then $b \in \kappa(\{⇓\}, \{c\})$, so $j \le k$. Therefore, $i < j \le k$, but this is not possible since $c \in \kappa(\{✡\}, \{a\})$, so $k \le i$. Therefore, by Lemma 2.7, $|S| \le 2n$.

All that remains is to account for points in S that are played multiple times. In each Q_i there are at most two points that can be played in subsequent rounds: the rightmost point in B_i and the rightmost point in T_i.

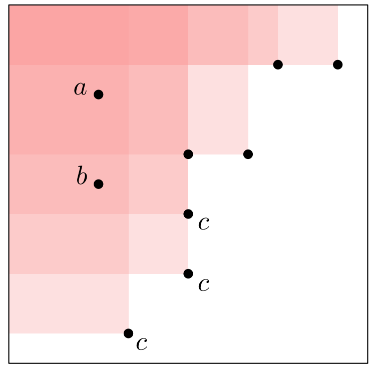


Figure 2.12: The proof of Lemma 2.12.

We charge each occurrence of such repeated points to the rounds in which they are these extreme points. In this way, each round is charged for at most two such points and the total contribution of these points to $\sum_{i=1}^{n} |Q_i|$ is at most $2n$.

In summary,

$$\sum_{i=1}^{n} |Q_i| \leq 4n + \sum_{i=1}^{n} |Q_i'| \leq 4n + 2n + |S| \leq 8n \,. \qquad \square$$

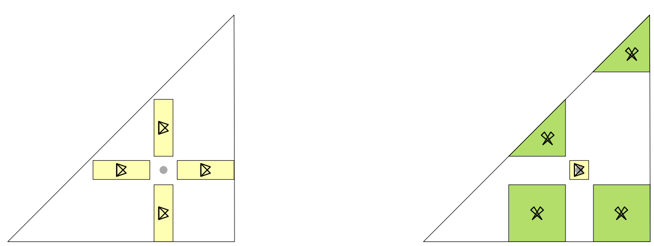
2.4.2 Forbidding Swords

Next we focus on the configuration ⚔ and give linear upper bounds bounds on $\mathrm{ex}'(n, \{$⚔,⊳$\})$ and $\mathrm{ex}'(n, \{$⚔,⟟$\})$. We begin with another lemma about forbidden configurations of points that is similar to Lemma 2.7. We say that a point (x_i, y_i) *SE-dominates* a point (x_j, y_j) if $x_i > x_j$ and $y_i < y_j$. We say that three points $a = (x_0, y_0)$, $b = (x_0, y_1)$, and c form an *obtuse-L-configuration* if $y_1 < y_0$ and c SE-dominates b.

Lemma 2.12. *Let S be a subset of $\{1, \ldots, n\}^2$ with no three points a, b, and c forming an obtuse-L-configuration. Then $|S| \leq 3n$.*

Proof. Refer to Figure 2.12. Consider the *Pareto boundary* $P \subseteq S$ containing each point of S that is not SE-dominated by any other point in S. The set P is non-decreasing, so it has size at most $2n$. We claim that the set $S \setminus P$ has at most one point in each column, so $|S| \leq 3n$. To see why this claim is true, observe that if some column of $P \setminus S$ contains two points a and b with a above b, then at least one point c in P SE-dominates b, so that a, b, and c would form the obtuse-L-configuration. $\square$

Theorem 2.13. $\mathrm{ex}'(n, \{$⊳,⚔$\}) \in O(n)$.



Proof. Let $S = \bigcup_{i=1}^{n} Q_i$. Then the rules for ⊳ imply that $|S| = \sum_{i=1}^{n} |Q_i|$, so it suffices to bound $|S|$. We claim that S contains no obtuse-L-configuration so, by Lemma 2.12, $|S| \leq 3n$.

Suppose there were $a \in Q_i$, $b \in Q_j$, and $c \in Q_k$ forming an obtuse-L-configuration. Now, $a \in \kappa(\{$⚔$\}, \{c\})$ and $c \in \kappa(\{$⚔$\}, \{a\})$, so it must be that $i = k$. The same argument, applied to b and c, implies that $j = k$, so $i = j = k$. But this is a contradiction since the rules for ⊳ imply that $i \neq j$. $\square$

Theorem 2.14. $\mathrm{ex}'(n, \{$⟟,⚔$\}) \in O(n)$.

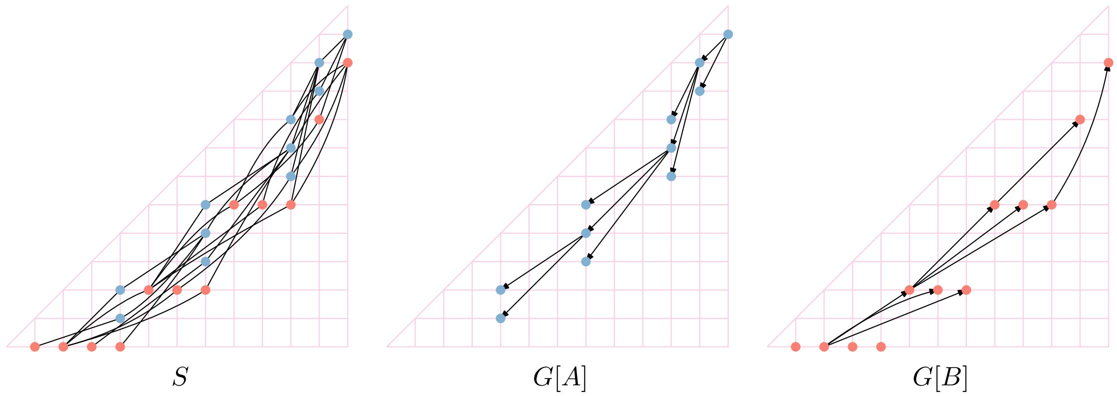
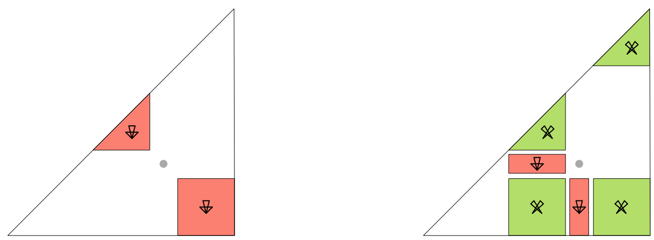


Figure 2.13: Four-colouring the graph G in the proof of Theorem 2.14.



Proof. Let $S = \bigcup_{i=1}^{n} Q_i$. We claim that S is non-decreasing. Indeed, the rule for ⍒ prevents two decreasing points from being played in the same round, while the rule for ⚔ prevents two decreasing points from being played in different rounds. This implies that $|S| \le 2n$. What remains is to account for points of S that are played in multiple rounds.

Consider the graph G with vertex set S that contains an edge uw if and only if the x-coordinate of u is equal to the y-coordinate of w. We claim that G is 4-colourable. To prove this, we partition S into two sets A and B and show that each of the graphs $G[A]$ and $G[B]$ induced by A and B is 2-colourable (in fact, $G[A]$ and $G[B]$ are each forests). Thus, if we colour A with colours $\{1, 2\}$ and B with colours $\{3, 4\}$, then we obtain a 4-colouring of G.

Refer to Figure 2.13. Remove the bottom-most point of S from each column and what remains is the set A. Observe that, since S is non-decreasing, A contains at most one point per row. Imagine directing the edges of $G[A]$ from right to left (top to bottom). This directed graph is obviously acyclic and, since each row contains at most one point of A, has maximum in-degree 1. Therefore $G[A]$ is a forest and can be 2-coloured using the colours $\{1, 2\}$. By a similar argument, using the fact that each column contains at most one point of B, the graph $G[B] = G[S \setminus A]$ can be 2-coloured using the colours $\{3, 4\}$.

The resulting 4-colouring of G partitions S into 4 colour classes $S_1, \ldots, S_4$. We now argue that each of these colour classes contributes $O(n)$ to $\sum_{i=1}^{n} |Q_i|$. Consider a (new) directed graph $H_j = (S_j, E_j)$ that contains the edge $\overrightarrow{uw}$ if and only if u kills w, i.e., $w \in \kappa(\{⍒, ⚔\}, \{u\})$. We claim that this graph is *complete*, i.e., for any $u, w \in S_j$ at least one of $\overrightarrow{uw}$ or $\overrightarrow{wu}$ is in E_j. To see why this is so, consider any two distinct points $u, w \in S_j$ with $u = (x_0, y_0)$ and $w = (x_1, y_1)$. Since S (and hence S_j) is non-decreasing, we may assume without loss of generality that $x_0 \le x_1$ and $y_0 \le y_1$. There are five cases to consider:

1. $y_0 = y_1$, so $x_0 < x_1$. In this case, $u \in \kappa(\{⍒\}, w)$, so $\overrightarrow{wu} \in E$.
2. $x_0 = x_1$, so $y_0 < y_1$. In this case, $u \in \kappa(\{⍒\}, w)$, so $\overrightarrow{wu} \in E$.
3. $x_0 < x_1$, $y_0 < y_1$, and $y_1 > x_0$. In this case, $w \in \kappa(\{⚔\}, \{u\})$, so $\overrightarrow{uw} \in E$.
4. $x_0 < x_1$, $y_0 < y_1$, and $y_1 < x_0$. In this case, $u \in \kappa(\{⚔\}, \{w\})$, so $\overrightarrow{wu} \in E$.
5. $x_0 < x_1$, $y_0 < y_1$, and $y_1 = x_0$. This case cannot occur since, in this case, the graph G contains the edge uw, so u and w are assigned different colours and at most one of them is j.

Suppose now that H_j contains a directed cycle $C = u_0, \ldots, u_{\ell-1}$. Since the points in this cycle all kill each other, i.e., $u_{k+1 \bmod \ell} \in \kappa(\{⍒, ⚔\}, \{u_k\})$, it must be the case that all the vertices of C are played in the same round $i' \in \{1, \dots, n\}$ and never played again, i.e., $V(C) \subseteq Q_{i'}$ and $V(C) \cap Q_{j'} = \emptyset$ for all $j' \in \{1, \dots, n\} \setminus \{i'\}$.

Now, if we repeatedly find a cycle in H_j and remove its vertices, we will eventually be left with an acyclic subgraph H'_j with vertex set $S'_j \subseteq S_j$. From the preceding discussion, we know that each cycle vertex we remove contributes only 1 to $\sum_{i=1}^n |Q_i|$:

$$\sum_{i=1}^{n} |(S_j \setminus S'_j) \cap Q_i| = |S_j \setminus S'_j| \,.$$

Finally, we are left with the complete acyclic subgraph H'_j with vertex set S'_j and whose topological sort order we denote by $\prec$. Now, if Q_i contains vertices $v_1 \prec \cdots \prec v_k$ of H'_j, then v_1 kills all of $v_2, \dots, v_k$ so that these vertices can not appear in any $Q_{i'}$ with $i' > i$. This implies that

$$\sum_{i=1}^{n} (|S'_j \cap Q_i| - 1) \le |S'_j| \,,$$

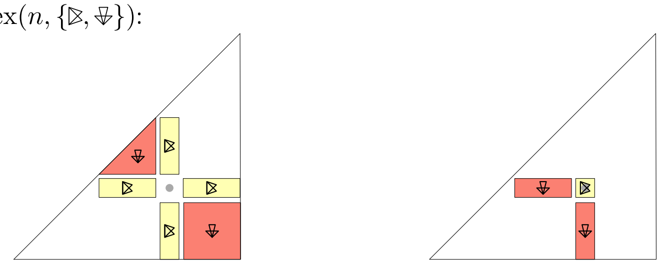
so $\sum_{i=1}^n |S'_j \cap Q_i| \le |S'_j| + n$. Putting everything together, we have

$$\begin{aligned}
\sum_{i=1}^{n} |Q_i| &= \sum_{j=1}^{4} \left(\sum_{i=1}^{n} |Q_i \cap S_j| \right) \\
&= \sum_{j=1}^{4} \left(\sum_{i=1}^{n} (|Q_i \cap (S_j \setminus S'_j)| + |Q_i \cap S'_j| \right) \\
&\le \sum_{j=1}^{4} \big(|S_j \setminus S'_j| + |S'_j| + n \big) \\
&= |S| + 4n \le 6n \,.
\end{aligned}$$

□

2.4.3 Monotone Matrices, Tripod Packing, and 2-Comparable Sets

In this section, we discuss $\mathrm{ex}(n, \{\rhd, \Downarrow\})$:



Determining the asymptotics of $\mathrm{ex}(n, \{\rhd, \Downarrow\})$ was given explicitly as an open problem in the conclusion of Braß's paper. We spent more than a year working on this problem and this work included computer searches for a variant of the problem played on the square grid $\{1, \dots, n\}^2$.[1] Using the results of these computer searches in the Online Encyclopedia of Integer Sequences [49], we discovered that this problem, when played on the square grid, is equivalent to several other known problems. See Figure 2.14.

1. *Monotone matrix problem*: How many values from $\{1, \dots, n\}$ can one write in an $n \times n$ matrix, so that each row is increasing from left-to-right, each column is increasing from bottom-to-top, and for each $i \in \{1, \dots, n\}$, the positions of i in the matrix form an increasing sequence?
2. *Tripod packing problem*: A *tripod* with top $p = (x, y, z) \in \mathbb{R}^3$ is the union of three closed rays originating at p and directed in the positive x-, y-, and z-directions, i.e.,

$$\mathrm{tripod}(x, y, z) = \bigcup_{0 \le t < \infty} \{(x+t, y, z), (x, y+t, z), (x, y, z+t)\} \,.$$

   How many disjoint tripods can be packed with tops in $\{1, \dots, n\}^3$?

[1] Playing on the square grid does not change the asymptotics of the problem. Any solution for $\{1, \dots, \lfloor n/2 \rfloor - 1\}^2$ can be used as a solution for the triangular grid Q and any upper bound for $\{1, \dots, n\}^2$ is also an upper bound for the triangular grid Q.

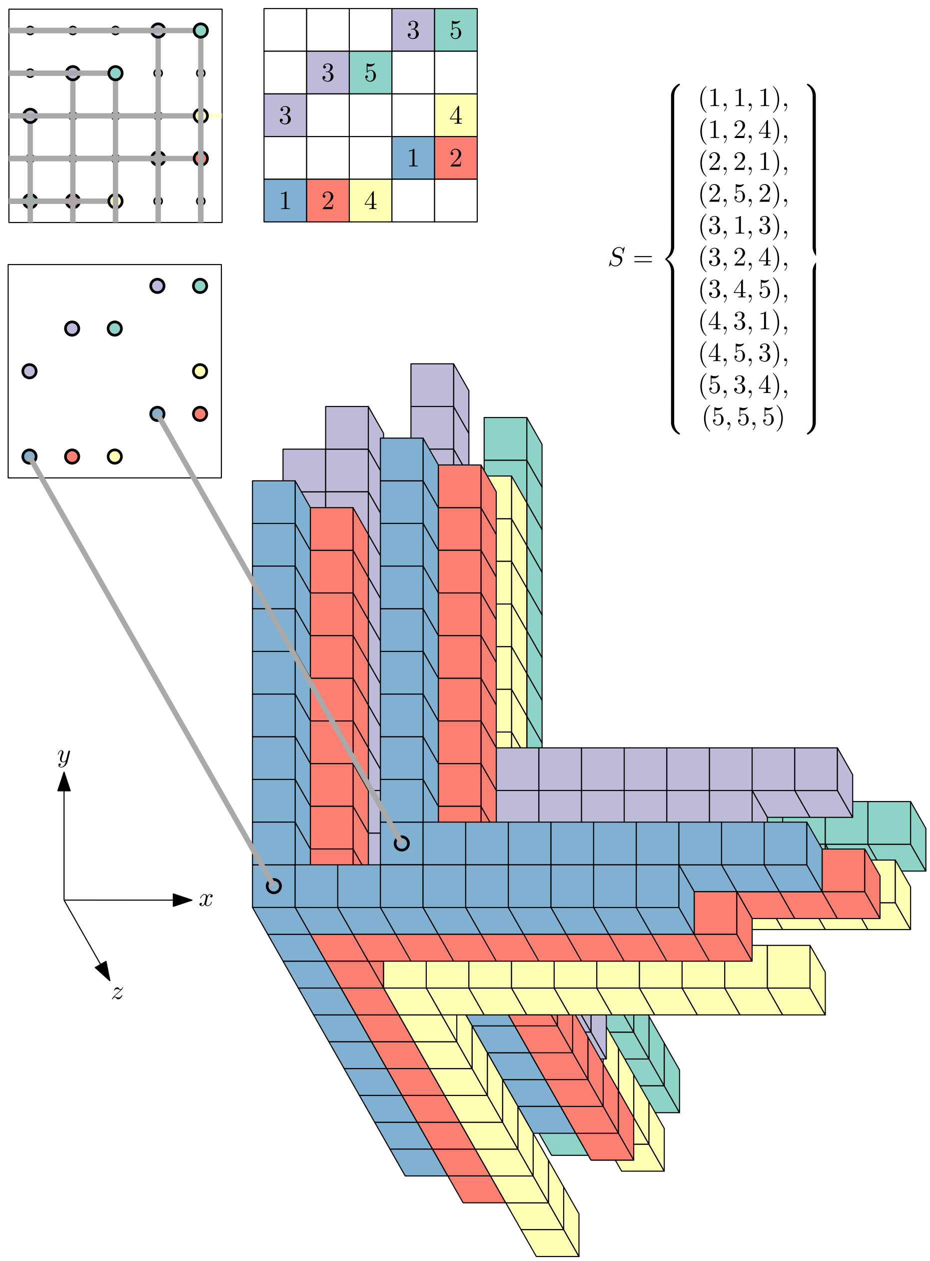


Figure 2.14: The dot puzzle induced by excluding the taco and nested configurations has already been studied under several equivalent formulations.

3. *2-comparable sets of triples problem.* Two triples of integers (a_1, a_2, a_3) and (b_1, b_2, b_3) are *2-comparable* if $a_i < b_i$ for at least two values of $i \in \{1, 2, 3\}$ or $a_i > b_i$ for at least two values of $i \in \{1, 2, 3\}$. What is the largest set, S, of pairwise 2-comparable triples one can make whose entries come from $\{1, \ldots, n\}$?

Several simple and natural constructions give lower bounds of $\Omega(n^{3/2})$ for these problems. However, this bound is not tight. A sequence of recursive constructions has steadily raised this lower bound [26, 50, 51, ?, 54]. The current record is held by Gowers and Long [26], who describe a construction of size $\Omega(n^{1.546})$.

Theorem 2.15 (Gowers and Long [26])**.** $\mathrm{ex}'(n, \{\triangleright, \triangledown\}) \in \Omega(n^{1.546})$.

The only known upper bound for this problem comes from the fact that a solution to this problem gives a solution to the Ruzsa-Szemerédi induced matching problem [47].

4. *Induced-matching problem*: What is the maximum number of edges in a bipartite graph $G = (A, B, E)$ with $|A| = |B| = n$ such that E can be partitioned into n *induced matchings* $M_1, \ldots, M_n$? That is, each M_i is a matching, and for any two edges $e, f \in M_i$ there is no edge in E that joins an endpoint of e to an endpoint of f.

It is simple to verify that if one takes a 2-comparable set of triples $S = \{(a_i, b_i, c_i) : i \in \{1, \ldots, m\}\}$ then the bipartite graph $G = (A, B, E)$ with $A = B = \{1, \ldots, n\}$ and

$$E = \{(b_j, c_j) : j \in \{1, \ldots, m\}\}$$

satisfies the conditions of the induced matching problem, with the partition into matchings given by

$$M_i = \{(b, c) : (i, b, c) \in S\} .$$

Thus, any upper bound for the induced-matching problem is also an upper bound on the size of a 2-comparable set of triples.

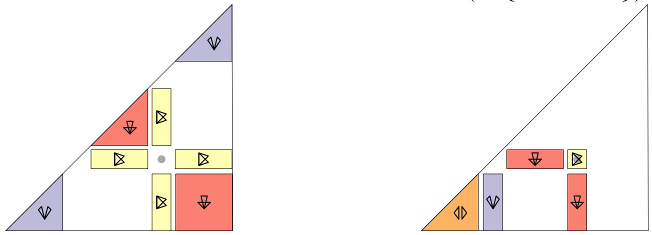
Known upper bounds for the induced matching problem are barely subquadratic, with the current record being held by Fox's improved version of the triangle removal lemma [24], which gives an upper bound of $n^2/e^{\Omega(\log^* n)}$. See the discussion, for example, in Gowers and Long [26]. Lower bounds for the induced matching problem are surprisingly high; a result of Behrend [8] can be used to construct n vertex graphs with $n^2/e^{O(\sqrt{\log n})}$ edges that can be decomposed into induced matchings.

Theorem 2.16 (Fox [24])**.** $\mathrm{ex}'(n, \{\triangleright, \triangledown\}) \in n^2/e^{\Omega(\log^* n)}$.

While discovering these results, we noticed that the relationships between some of these problems have gone unnoticed. Here we make a few bibliographic notes.

- Braß [10] seems to have been unaware that the question he posed was equivalent to tripod packing and monotone matrices (or, like us, had never heard of these problems).
- Tiskin [54], apparently unaware of the relation between tripod packing and induced matchings, proved an upper bound of $o(n^2)$ for tripod packing. His proof does not depend on any properties of tripod packing that are not also true for induced matchings, and uses the same tools (namely Szemeredi's Regularity Lemma) as the original upper bounds for the induced matching problem.
- Gowers and Long [26] seem to be unaware that the problem on 2-comparable sets of triples was studied under other names.
- Gowers and Long [26] arrived at 2-comparable sets as a relaxation of a problem (the size of the largest 2-increasing sequence of triples) proposed by Loh [37]. In his discussion of this problem, Loh formulates a restricted version of the induced matching problem, in which the matching must satisfy a certain Σ-free property and expresses hope [37, remark on page 9] that this restricted version has an $O(n^{3/2})$ upper bound. However, solutions for tripod packing correspond to Σ-free induced matchings, so Σ-free induced matchings of size $\omega(n^{3/2})$ are already known.

In our context, the only new observation we have pertains to $\mathrm{ex}'(n, \{\rhd, \bigtriangledown, \mathbb{V}, \lhd\!\rhd\})$:



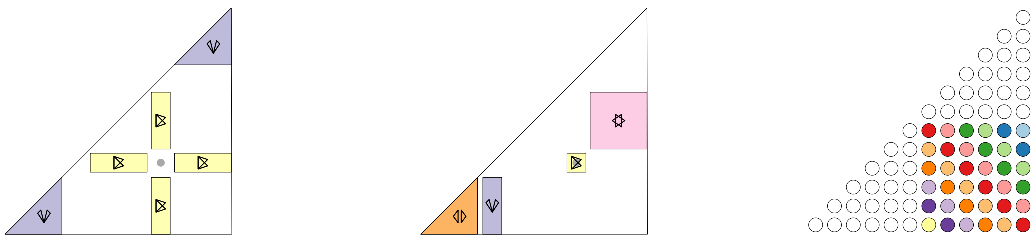
We observe that $\mathrm{ex}'(n, \{\rhd, \bigtriangledown\}) \in O(\mathrm{ex}'(n, \{\rhd, \bigtriangledown, \mathbb{V}, \lhd\!\rhd\})$ so these two functions therefore have the same asymptotic growth. This comes from the fact that a solution for the dot puzzle of size n resulting from $X = \{\rhd, \bigtriangledown\}$ can be used as a solution for the dot puzzle of size $2(n+1)$ resulting from $X = \{\rhd, \bigtriangledown, \mathbb{V}, \lhd\!\rhd\}$ by only playing in the lower-right quadrant. When played this way, the extra restrictions caused by $\mathbb{V}$ and $\lhd\!\rhd$ do not affect the lower-right quadrant.

Theorem 2.17. $\mathrm{ex}'(n, \{\rhd, \bigtriangledown\}) \in \Theta(\mathrm{ex}'(n, \{\rhd, \bigtriangledown, \mathbb{V}, \lhd\!\rhd\})$.

2.4.4 Lower Bounds

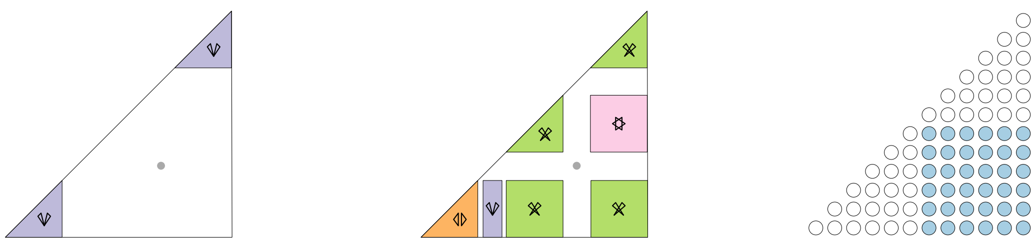
Finally, we finish up with some $\Omega(n^2)$ lower bound constructions. In each case, a matching upper bound follows from one of the results in Braß [10]. The following are essentially "proofs by figure" in which a brief description of the solution is illustrated alongside the rules of each dot puzzle. In order to avoid floors and ceilings, we assume n is even.

Theorem 2.18. $\mathrm{ex}(n, \{\rhd, \text{✡}, \nabla, \mathbb{V}, \lhd\!\rhd\}) \in \Theta(n^2)$.



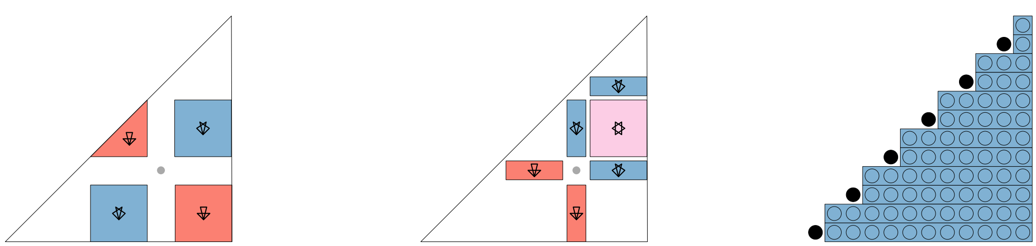
Proof. For each $i \in \{1, \ldots, n/2\}$, we take Q_i to be all the points of $Q' = \{n/2, \ldots, n\}^2$ on the line $\{(x, y) : y = 3n/2 - x - i + 1\}$. $\square$

Theorem 2.19. $\mathrm{ex}(n, \{\times, \mathbb{V}, \lhd\!\rhd, \text{✡}\}) \in \Theta(n^2)$.



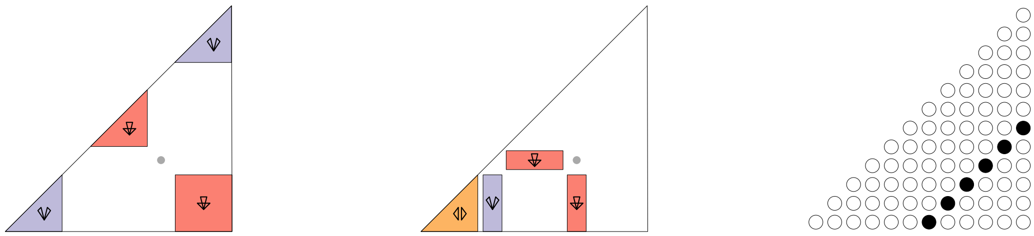
Proof. We take $Q_1 = \{(x, y) \in Q : x > n/2, y < n/2\}$ and set $Q_2 = Q_3 = \cdots = Q_n = \emptyset$. $\square$

Theorem 2.20. $\mathrm{ex}(n, \{\text{✡}, \bigtriangledown, \nabla\}) \in \Theta(n^2)$.



Proof. We repeatedly take every second point on the diagonal $y = x - 1$, i.e., for each $i \in \{1, \ldots, n\}$, $Q_i = \{(2j, 2j-1) : j \in \{1, \ldots, n/2\}\}$. $\square$

Theorem 2.21. $\mathrm{ex}(n, \{\mathbb{V}, \triangledown, \bowtie\}) \in \Theta(n^2)$.



Proof. We repeatedly take points on the diagonal of the lower-right quadrant, i.e., for each $i \in \{1, \ldots, n\}$, we take $Q_i = \{(n/2 + i, i) : i \in \{1, \ldots, n/2\}$. $\square$

2.5 Conclusions

We have been able to document the asymptotic behavior of $\mathrm{ex}(n, X)$ up to a $\log n$ factor for 248 of the 256 possible choices of $X \subseteq \{\rhd, \Diamond, \mathbb{V}, \triangledown, \diamondsuit, \bowtie, \times, \ast\}$. We have also tied, again up to a $\log n$ factor, the asymptotics of $\mathrm{ex}(n, X)$ for the remaining 8 choices of $X \subseteq \{\rhd, \triangledown, \mathbb{V}, \bowtie, \Diamond\}$ containing $\{\rhd, \triangledown\}$ to that of the well-known tripod packing problem. In doing so we made major progress towards a full catalog of these asymptotics, but solving the tripod packing problem and devising a methodology to close the $\log n$ gap are two open problems.

Acknowledgement

Some of this work was carried out at the Fourth Annual Workshop on Geometry and Graphs, held at the Bellairs Research Institute in Barbados, March 6–11, 2016. The authors are grateful to the organizers and to the other participants of this workshop for providing a stimulating working environment.

Chapter 3

Approximability of Covering Cells with Line Segments

Paz Carmi[1] Anil Maheshwari[2] Saeed Mehrabi[2]
Luís Fernando Schultz Xavier da Silveira[3]

Abstract

Korman et al. (Line segment covering of cells of arrangements, IPL, 129:25–30, 2018) studied the following geometric covering problem: given a set S of n line segments in the plane, find a minimum number of line segments such that every cell in the arrangement of the line segments is covered. Here, a line segment s *covers* a cell f if s is incident to f. The problem was shown to be NP-hard, even if the line segments in S are axis-parallel, and it remains NP-hard when the goal is to cover the "rectangular" cells (i.e., cells that are defined by exactly four axis-parallel line segments).

In this paper, we consider the approximability of the problem. We first give a PTAS for the problem when the line segments in S are in any orientation, but we can only select the covering line segments from one orientation. Then, we show that when the goal is to cover the rectangular cells using line segments from both horizontal and vertical line segments, then the problem is APX-hard. We also consider the parameterized complexity of the problem and prove that the problem is FPT when parameterized by the size of an optimal solution. Our FPT algorithm works when the line segments in S have two orientations and the goal is to cover *all* cells, complementing that of Korman et al. [33] in which the goal is to cover the "rectangular" cells.

3.1 Introduction

Set Cover is a well-studied problem in computer science. The input to the problem is a ground set $\mathcal{G}$ of n elements and a set $\mathcal{S}$ of m subsets of $\mathcal{G}$; that is, $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ such that $S_i \subseteq \mathcal{G}$ for all $1 \leq i \leq m$. The objective is to find a minimum-cardinality subset of $\mathcal{S}$ whose union is $\mathcal{G}$. Set Cover is known to be NP-hard [25] and even hard to approximate [23]. Because of these inapproximability results, there has been a rich body of research to study the set cover problem for geometric objects: the *geometric set cover* problem. For example, consider a set of n points

[1]Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel
[2]School of Computer Science, Carleton University, Ottawa, Canada
[3]School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada

in the plane as the ground set $\mathcal{G}$ and let each set $S_i \subseteq \mathcal{S}$ be a disk in the plane (i.e., each disk defines a subset of points). Then, the goal is to select a minimum number of disks to cover all the points. While the problem is still NP-hard, it admits a PTAS [41]. Other variants of the geometric set cover includes covering line segments with points [35, 12], multicoloured set cover [17], covering line segments with squares [1]. See [27, 18, 13, 55, 15, 46] and the references therein for more results on the geometric set cover problem.

In this paper, we consider a geometric variant of the set cover problem that was first studied by Korman et al. [33]. A set of line segments in the plane is said to be *non-overlapping* if any two line segments from the set intersect in at most one point. Given a set S of n non-overlapping line segments in the plane, a *cell* in the arrangement of S is a maximally connected region that is not intersected by any line segment in S [33]. In other words, cells are "empty" (i.e., not intersected by any line segment) regions in the arrangement defined by S. Then, the objective of the *Line Segment Covering* (LSC) problem is to select a minimum number of line segments such that every cell in the arrangement of the line segments is covered. Here, a cell is *covered* by a line segment if it is incident to the line segment (i.e., the line segment is in the set of line segments defining the boundary of the cell). We assume that at most two line segments may share a fixed point in the plane.

Related work Korman et al. [33] proved that when the line segments are only horizontal and vertical, the LSC problem is NP-hard and it remains NP-hard when the goal is to cover the "rectangular" cells. By a closer look at their hardness proof, one can see that the problem is NP-hard even if we are only allowed to select the line segments from one orientation (they only select vertical line segments when constructing a solution from a given truth assignment for the corresponding 3SAT problem). Moreover, the authors gave an $O(k^{O(k)} \cdot n \log n + C)$-time FPT algorithm for covering the rectangular cells when parameterized by k (the size of an optimal solution) by computing a kernel of size $O(k^3)$ for the problem. However, the algorithm does not work when the goal is to cover *all* cells of the arrangement. The authors left open studying the approximability of the problem. Recently, Basu Roy et al. [46] studied the geometric set cover problem for non-piercing objects in the plane. Two connected regions A, B in the plane are *non-piercing* if both $A \setminus B$ and $B \setminus A$ are connected. Given a set of non-piercing objects and a set of points in the plane, the goal is to find a minimum-cardinality subset of the objects so as to cover all the points. The authors gave a PTAS for this problem. However, the ground set in LSC problem consists of line segments, which are not non-piercing. So, it is not clear if their algorithm could be applied to the LSC problem.

The LSC problem is closely related to a guarding problem studied by Bose et al. [9]. Given a set of lines in the plane, they studied the problems of guarding cells of the arrangement by selecting a minimum number of lines, or guarding the lines by selecting a minimum number of cells. Here, "guarding" has the same meaning as "covering" in the LSC problem. However, their results do not extend to the LSC problem, because (as also noted by Korman et al. [33]) they use some properties of lines that are not true for the case of line segments.

Our results In this paper, we prove the following results.

- We first consider the variant of the LSC problem in which the line segments in S can have arbitrary orientations, but we are allowed to select the covering line segments from only one orientation. We give a PTAS for this variant, which along with the NP-hardness of Korman et al. [33], settles the complexity of this variant of the problem.
- Then, we consider the next immediate variant: when the input line segments are axis-parallel and we allow selecting the covering line segments from more than one direction. We show that the LSC problem is APX-hard for this variant.
- We give an FPT algorithm for the LSC problem when the line segments in S have only two orientations and the goal is to cover *all* cells of the arrangement. This complements the FPT algorithm of Korman et al. [33] as we do not restrict the covering only to rectangular cells.

Organization In Section 3.2, we give some definitions and revisit some necessary background. We show our PTAS in Section 3.3 and the APX-hardness result in Section 3.4. Finally, the FPT algorithm is given in Section 3.5 and we conclude the paper in Section 3.6.

3.2 Preliminaries

In the following, we revisit some techniques and background that are used throughout this paper.

Local search Our PTAS for the LSC problem is based on the local search technique, which was introduced to computational geometry independently by Mustafa and Ray [41], and Chan and Har-Peled [16]. Consider an optimization problem in which the objective is to compute a feasible subset S' of a ground set S whose cardinality is minimum over all such feasible subsets of S. Moreover, it is assumed that computing some initial feasible solution and determining whether a subset $S' \subseteq S$ is a feasible solution can be done in polynomial time. The local search algorithm for a minimization problem is shown in Algorithm 1.

Algorithm 1 LocalSearch(S)

1: fix some parameter k;
2: let A be some initial feasible solution;
3: **while** $\exists A' \subseteq A$, $M \subseteq S \setminus A$: $|A'| \leq k$, $|M| < |A'|$ and $(A \setminus A') \cup M$ is a feasible solution **do**
4: set $A = (A \setminus A') \cup M$;
5: **return** A;

Clearly, the local search algorithm runs in polynomial time. Let $\mathcal{B}$ and $\mathcal{R}$ be the solutions returned by the algorithm and an optimal solution, respectively. The following result establishes the connection between local search technique and obtaining a PTAS.

Theorem 1 ([16, 41]). *Consider the solutions $\mathcal{B}$ and $\mathcal{R}$ for a minimization problem, and suppose that there exists a planar bipartite graph $H = (\mathcal{B} \cup \mathcal{R}, E)$ that satisfies the local exchange property: for any subset $\mathcal{B}' \subseteq \mathcal{B}$, $(\mathcal{B} \setminus \mathcal{B}') \cup N_H(\mathcal{B}')$ is a feasible solution, where $N_H(\mathcal{B}')$ denotes the set of neighbours of $\mathcal{B}'$ in H. Then, the local search algorithm yields a* PTAS *for the problem.*

The local search was used by Mustafa and Ray [41] to obtain a PTAS for the geometric hitting set problem and by Chan and Har-Peled [16] to obtain a PTAS for the independent set problem on pseudodisks. Since then, the technique has been used to get a PTAS for several other geometric problems, such as covering and packing [5], geometric dominating set [7] and unique covering [6].

Fixed-parameter tractability The theory of parameterized complexity was developed by Downey and Fellows [21]. Let Σ be a finite alphabet. Then, a *parameterized* problem is a language $L \subseteq \Sigma^* \times \Sigma^*$ in which the second component is called the *parameter* of the problem. A parameterized problem L is said to be *fixed-parameter tractable* or FPT, if the question "$(x_1, x_2) \in L$?" can be decided in time $f(|x_2|) \cdot |x_1|^{O(1)}$, where f is an arbitrary function. We call an algorithm with such running time $f(|x_2|) \cdot |x_1|^{O(1)}$, an FPT algorithm.

For the rest of this paper, we denote a set of n line segments in the plane by S (i.e., $|S| = n$) and the set of cells (induced by the arrangement) by $\mathcal{A}(S)$.

3.3 PTAS

In this section, we show that the LSC problem admits a PTAS when the line segments in S have any orientations, but we can select line segments from only one orientation to cover the cells. In fact, we give a PTAS for a more general problem: given $M \subseteq S$ and a subset D of cells in $\mathcal{A}(S)$, the goal is to cover the cells in D with the minimum number of line segments in M that are from the "covering orientation". Let us call this variant as the LSC problem on (M, D). In the following, we assume that each cell in D is incident to at least one line segment from the "covering orientation"; otherwise, we let D to denote the set of those cells that are incident to at least one line segment from the covering orientation.

We run the local search algorithm with parameter $k = c/\epsilon^2$ for some $\epsilon > 0$, where c is a constant. Let $\mathcal{B}$ be the solution returned by the algorithm and let $\mathcal{R}$ be an optimal solution. We can assume that $\mathcal{B} \cap \mathcal{R} = \emptyset$; if $\mathcal{B} \cap \mathcal{R} \neq \emptyset$, then let $I = \mathcal{B} \cap \mathcal{R}$, $\mathcal{B}' = \mathcal{B} \setminus I$, $\mathcal{R}' = \mathcal{R} \setminus I$, $M' = M \setminus I$ and let D' be the set of cells that are

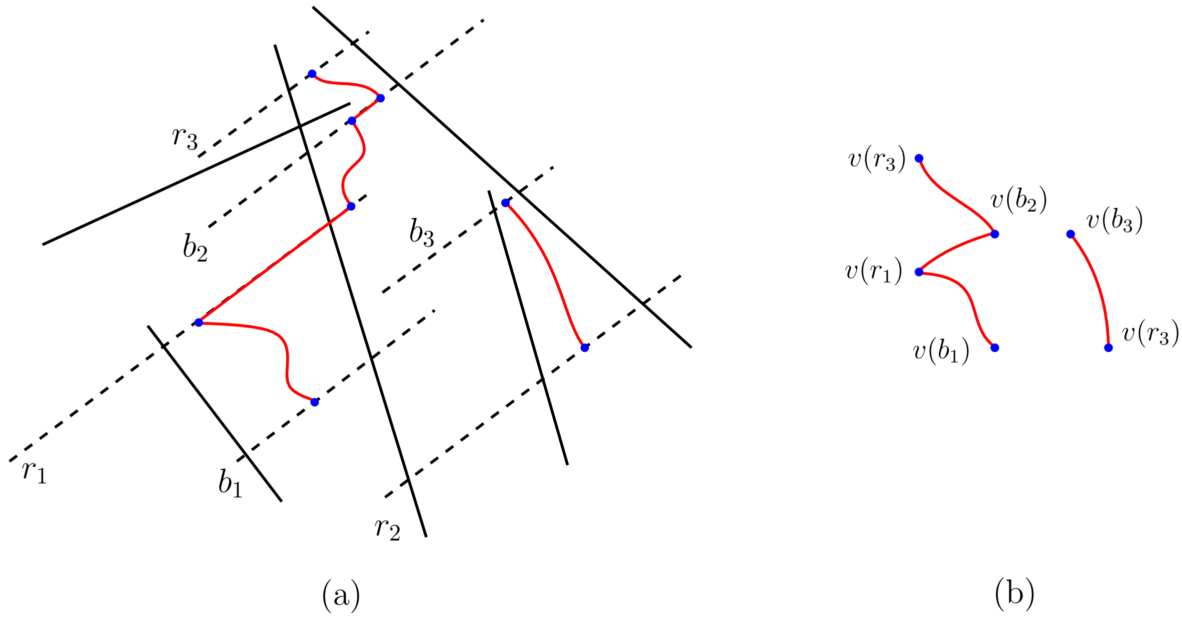


Figure 3.1: (a) An example of graph H', where $b_1, b_2, b_3 \in \mathcal{B}$ and $r_1, r_2, r_3 \in \mathcal{R}$. (b) Graph H after contracting the edges of H' that correspond to the same line segment.

not covered by line segments in I. Now, $\mathcal{B}'$ and $\mathcal{R}'$ are disjoint. Moreover, $\mathcal{R}'$ is an optimal solution for the LSC problem on (M', D'). Therefore, if we show that $|\mathcal{B}'| \leq (1+\epsilon)|\mathcal{R}'|$, then we can conclude that $|\mathcal{B}| \leq (1+\epsilon)|\mathcal{R}|$.

We now construct a planar bipartite graph $H = (\mathcal{B} \cup \mathcal{R}, E)$ that satisfies the local exchange property, hence proving that the problem admits a PTAS by Theorem 1. To this end, we first construct an auxiliary planar graph H' and then show how to obtain H from H' by edge contraction. For each cell $f \in \mathcal{A}(S)$, let $b \in \mathcal{B}$ and $r \in \mathcal{R}$ be two line segments that cover f; we select a point $p \in b \cap f$ and $q \in r \cap f$ and connect them by a curve c that lies in the interior of f (except its endpoints p and q). Notice that since both $\mathcal{B}$ and $\mathcal{R}$ are feasible solutions, we know that $\mathcal{B}$ contains at least one line segment that covers f and $\mathcal{R}$ also contains at least one line segment that covers f, for all $f \in \mathcal{A}(S)$. We add p and q to $V(H')$ and c to $E(H')$. We complete the definition of H' by connecting every pair of consecutive points in $s \cap V(H')$, for all $s \in S$, by an edge that is exactly the portion of s that lies between the pair of points. See Figure 3.1(a) for an example. Clearly, H' is planar because the first set of edges are drawn in the interior of cells and each cell contains at most one edge. Moreover, the second set of edges are aligned with the line segments in $\mathcal{B} \cup \mathcal{R}$. Since the line segments in $\mathcal{B} \cup \mathcal{R}$ are non-overlapping and all have the same orientation, the second set of edges are also non-crossing. To obtain the graph H, for each segment $s \in \mathcal{B} \cup \mathcal{R}$, we contract the edges of H' that are contained in s such that we get a single point $v(s)$ corresponding to s; see Figure 3.1(b). So, $V(H) = \{v(s) | s \in \mathcal{B} \cup \mathcal{R}\}$. Graph H is planar since H' remains planar after this edge contraction. Moreover, H is a bipartite graph as the edges of H' with both endpoints belonging to a line segment in $\mathcal{B}$ or both endpoints belonging to a line segment in $\mathcal{R}$ are collapsed into a single point (i.e., $v(s)$).

Lemma 2. *Graph H is planar and bipartite.*

We next show that H satisfies the exchange property.

Lemma 3. *Graph H satisfies the local exchange property.*

Proof. It is sufficient to show that for every cell $f \in \mathcal{A}(S)$, there are vertices $b \in \mathcal{B}$ and $r \in \mathcal{R}$ such that both segments corresponding to these vertices cover f and $(b, r) \in E(H)$. Take any cell $f \in \mathcal{A}(S)$ and let $M \subseteq \mathcal{B} \cup \mathcal{R}$ be the set of all line segments that cover f. Notice that $M \cap \mathcal{B} \neq \emptyset$ and $M \cap \mathcal{R} \neq \emptyset$ because $\mathcal{B}$ and $\mathcal{R}$ are each a feasible solution. Then, by definition, there must be a $b \in M \cap \mathcal{B}$ and $r \in M \cap \mathcal{R}$ for which $(b, r) \in E(H)$. This completes the proof of the lemma. □

Putting everything together, we have the main result of this section.

Theorem 4. *Let S be a set of line segments of any orientation. Moreover, let $M \subseteq S$ and D be a subset of cells in $\mathcal{A}(S)$. Then, there exists a* PTAS *for the line segment covering* (LSC) *problem on (M, D) when the line segments in S can have any orientation and we are allowed to select the covering line segments from only one orientation.*

3.4 APX-Hardness

In this section, we show that the LSC problem is APX-hard when the line segments in S have only two orientations and the goal is to cover the rectangular cells. To this end, we give an L-reduction from the Minimum Vertex Cover (MVC) problem on graphs with maximum-degree three, which is known to be APX-hard [3], to this variant of the LSC problem. Our reduction is inspired by the construction of Mehrabi [40]. As a reminder, we first give a formal definition of L-reduction [43], which is one of the gap-preserving reductions. Let Π and Π' be two optimization problems with the cost functions $c_\Pi(.)$ and $c_{\Pi'}(.)$, respectively. We say that Π L-*reduces* to Π' if there are two polynomial-time computable functions f and g such that the followings hold.

1. For any instance x of Π, $f(x)$ is an instance of Π'.
2. If y is a solution to $f(x)$, then $g(y)$ is a solution to x.
3. There exists a constant $\alpha > 0$ such that

$$OPT_{\Pi'}(f(x)) \leq \alpha OPT_\Pi(x),$$

   where $OPT_Y(x)$ denotes the cost of an optimal solution for problem Y on its instance x.
4. There exists a constant $\beta > 0$ such that for every solution y for $f(x)$,

$$|OPT_\Pi(x) - c_\Pi(g(y))| \leq \beta |OPT_{\Pi'}(f(x)) - c_\Pi(y)|,$$

   where $|x|$ denotes the absolute value of x.

Lemma 5. *The minimum vertex cover* (MVC) *problem on graphs with maximum-degree three is* L*-reducible to the* LSC *problem, where S is a set of horizontal and vertical line segments and the goal is to cover the rectangular cells of $\mathcal{A}(S)$.*

Proof. Let I be an instance of MVC on graphs of maximum-degree three; let $G = (V, E)$ be the graph corresponding to I and let k be the size of the smallest vertex cover in G. First, let $u_1, \ldots, u_n$ be an arbitrary ordering of the vertices of G, where $n = |V|$. In the following, we give a polynomial-time computable function f that takes I as input and outputs an instance $f(I)$ of the LSC problem.

We first describe the vertex gadgets. For each vertex u_i, $1 \leq i \leq n$, construct a horizontal line segment H_i and a vertical line segment V_i, and connect them as shown in Figure 3.2. We call the (blue) horizontal line segment used in the connection of H_i and V_i the *horizontal connector* C_i of i. Moreover, there are four line segments used in the connection of H_i and V_i that we call the *small connectors* of i. Notice that these five "connectors" along with H_i and V_i form exactly two rectangular cells. For each edge $(u_i, u_j) \in E$, where $i < j$, we add two small line segments, one horizontal and one vertical, at the intersection point of V_i and H_j such that they intersect each other as well as each intersects one of V_i and H_j, hence forming a rectangular cell; see the two (red) line segments at the intersection of V_1 and H_2 in Figure 3.2 for an example. We call such a pair *edge line segments* and denote them by $E_{i,j}$. Finally, for every rectangular cell whose four sides are *all* defined by the line segments corresponding to a 4-subset of $\{H_i, V_i | 1 \leq i \leq n\}$ (i.e., the cell is not covered by a horizontal connector or edge line segments), we insert a vertical line segment into the cell so as to make it non-rectangular; see the vertical (red) line segment in Figure 3.2. This ensures that every rectangular cell is incident either to a horizontal connector or to edge line segments $E_{i,j}$ for some i and j. This gives the instance $f(I)$ of the LSC problem. Notice that f is a polynomial-time computable function. In the following, we denote an optimal solution for the instance X of a problem by $s^*(X)$. We now prove that all the four conditions of L-reduction hold.

First, let M be a vertex cover of G of size k. Denote by $H[M] = \{H_i | u_i \in M\}$ the set of horizontal line segments induced by M and define $V[M]$ analogously. Moreover, let $C[M] = \{C_i | u_i \notin M\}$ be the set of horizontal connectors whose corresponding vertex is not in M. We show that $F = H[M] \cup V[M] \cup C[M]$ is a

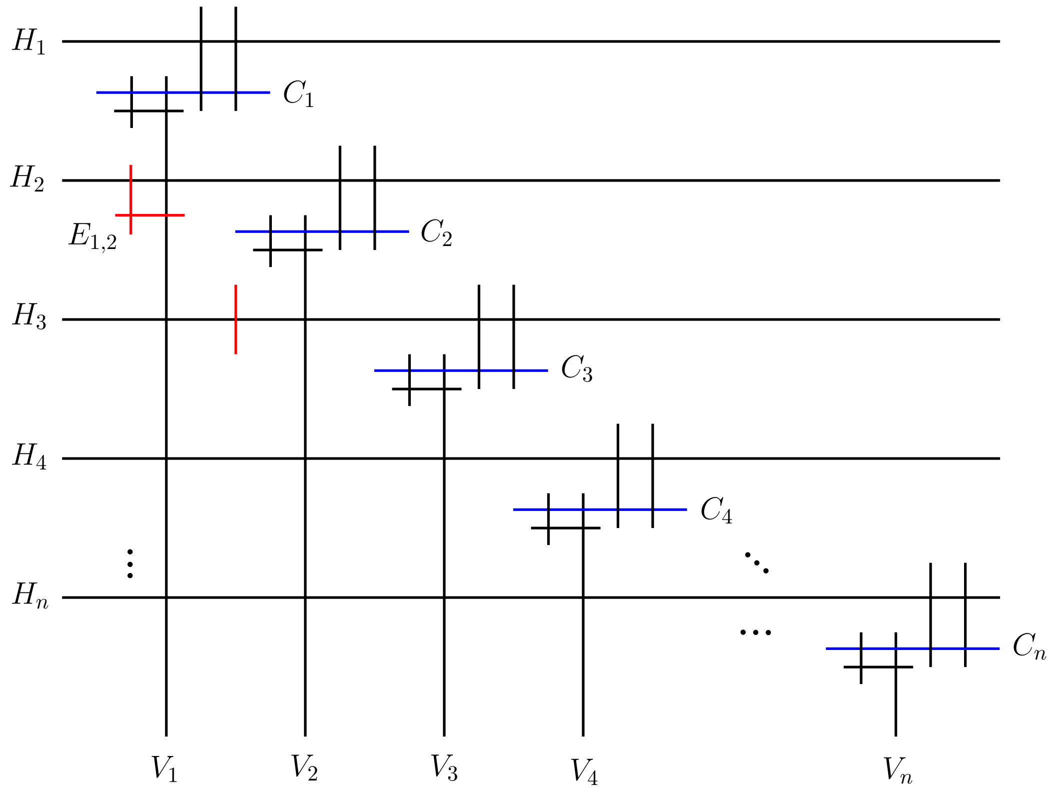


Figure 3.2: An illustration in support of the construction in the proof of Lemma 5.

feasible solution for covering all the rectangular cells of $f(I)$. Let f be a rectangular cell. Then, f must be incident either to a horizontal connector or to edge line segments $E_{i,j}$ for some i and j. First, if f is incident to a horizontal connector C_i, then either $C_i \in F$ or $H_i \in F$ and $V_i \in F$ by the construction of F and so f is covered either way. Next, if f is incident to edge line segments $E_{i,j}$ for some i and j, where w.l.o.g. $i < j$, then either $V_i \in F$ or $H_j \in F$ because we know that $u_i \in M$ or $u_j \in M$. So, f is again covered in this case. Therefore, F is a feasible solution.

Second, let F be any feasible solution for $f(I)$. Notice that we can construct a feasible solution F' for $f(I)$ such that $|F'| \leq |F|$ and F' consists of only H_i or V_i for some i, or a C_i. More precisely, if F contains a small connector, then we replace it by the corresponding horizontal connector. Moreover, if F contains exactly one of the line segments in a pair of edge line segments (resp., contains both line segments in a pair of edge line segments $E_{i,j}$), then we replace that line segment with the line segment it intersects (resp., we replace those line segments with V_i and H_j). The new solution F' is feasible because (i) any rectangular cell covered by a small connector is also covered by a horizontal connector, and (ii) any cell covered by a pair of edge line segments $E_{i,j}$ (for some i and j) is also covered by V_i and H_j. For (ii), if exactly one of the line segments in $E_{i,j}$ is in F, then we replace it with exactly one of V_i or H_j. Otherwise, if both line segments of $E_{i,j}$ are in F, then we replace both of them with V_i and H_j. So, $|F'| \leq |F|$ and F' is a feasible solution for $f(I)$. Now, let $M = \{u_i | H_i \in F' \text{ or } V_i \in F'\}$. To show that M is a vertex cover for G, consider any edge $(u_i, u_j) \in E$, where $i < j$. Then, we know that there exists a rectangular cell at the intersection of V_i and H_j that must be covered by F'. Since none of the two edge line segments of $E_{i,j}$ are in F', we conclude that at least one of V_i and H_j is in F', which means that $u_i \in M$ or $u_j \in M$. Hence, M is a vertex cover.

Third, let M be any vertex cover of G of size k. Then, observe that $|H[M]| = |V[M]| = |M| = k$ and also $|C[M]| = n - k$. Given that G has degree three, $k \geq n/4$ and so $|s^*(f(I))| \leq n - k + k + k \leq 5k \leq 5|s^*(I)|$.

We now prove the last condition of L-reduction. First, define $\mathsf{Both}[F']$ as follows: for each i, $\mathsf{Both}[F']$ contains H_i and V_i if and only if both of them are contained in F'. Also, define $\mathsf{One}[F']$ to be the remaining line segments corresponding to either H_i or V_i for some i; i.e., those of u_i, where *only* one of its line segments appears in F'. Take any vertex i. To cover the two rectangular cells incident to the horizontal connector of i, we must have $C_i \in F'$ or $H_i, V_i \in F'$; this is true for all i. Then, $|C[F']| + |\mathsf{Both}[F']|/2 \geq n$. Moreover, $|\mathsf{One}[F']| + |\mathsf{Both}[F']|/2 \geq k$ since M is a vertex cover of G. Therefore, $|F'| \geq |\mathsf{Both}[F']| + |\mathsf{One}[F']| + |C[F']| \geq |\mathsf{One}[F']| + |\mathsf{Both}[F']|/2 + n \geq k + n$. By this and our earlier inequality $|s^*(f(I))| \leq n - k + k + k$, we have $|s^*(f(I))| = n + k$. Now, suppose that $|F| = |s^*(f(I))| + c$ for some $c \geq 0$. Then,

$$\begin{aligned}
&|F| - |s^*(f(I))| = c \\
&\Rightarrow |F| - (n+k) = c \\
&\Rightarrow |F'| - (n+k) \leq c \\
&\Rightarrow |\mathsf{One}[F']| + |\mathsf{Both}[F']|/2 + n - (n+k) \leq c \\
&\Rightarrow |\mathsf{One}[F']| + |\mathsf{Both}[F']|/2 - k \leq c \\
&\Rightarrow |M| - |s^*(I)| \leq c.
\end{aligned}$$

That is, $|M| - |s^*(I)| \leq |F| - |s^*(f(I))|$. This concludes our L-reduction from MVC on graphs of maximum-degree three to LSC with $\alpha = 5$ and $\beta = 1$. $\square$

Theorem 6. *The line segment covering* (LSC) *problem is* APX*-hard when the line segments in S are either horizontal or vertical and the goal is to cover the rectangular cells of $\mathcal{A}(S)$.*

3.5 FPT

In this section, we show that the LSC problem is fixed-parameter tractable (parametrized by the size of an optimal solution) when the line segments in S are either horizontal or vertical, and the goal is to cover all the cells in $\mathcal{A}(S)$. This complements the FPT result of Korman et al. [33], where the goal is to cover the rectangular cells. Throughout this section, let k be the size of an optimal solution.

Our FPT follows the framework of Korman et al. [33]. That is, we formulate the LSC problem as a hitting set problem and argue that we only need to hit an $O(k^3)$ number of sets; hence, obtaining a kernel of size $O(k^3)$ for the problem. The FPT of Korman et al. [33] is based on the fact that any three axis-parallel line segments can cover at most two "rectangular" cells (i.e., at most two rectangular cells can be incident to all the three line segments). As an analogous result, we prove in Lemma 7 that the number of such cells can be at most six when the goal is to cover *all* cells, including non-rectangular ones. We will then apply this result to obtain the desired kernel.

Lemma 7. *Let S be a set of n axis-parallel line segments in the plane. Then, for any three line segments $s_1, s_2, s_3 \in S$, there are at most six cells in $\mathcal{A}(S)$ that can be covered by all three line segments s_1, s_2 and s_3.*

Proof. Take any three line segments s_1, s_2 and s_3 that are subsegments of line segments in S and let F be the set of all cells in $\mathcal{A}(S)$ that are covered by all three of s_1, s_2 and s_3. We first will show that, when s_1, s_2 and s_3 are pairwise disjoint, $|F| \leq 2$. To this end, we construct a planar graph H corresponding to s_1, s_2, s_3 and the cells in F and will then argue that this graph must contain a subdivision of $K_{3,3}$ if $|F| > 2$. We next give the details. Let f be a cell in F. Consider a point $p(f)$ in the interior of f as well as distinct points $p(s_i, f)$ in $s_i \cap f$ for all $i = 1, 2, 3$ (notice that $p(s_i, f)$ is on the boundary of f). These points together form the set of vertices of H; that is, $V(H) = \{p(f) : f \in F\} \cup \{p(s_i, f) : i = 1, 2, 3, f \in F\}$. Now, for each $i = 1, 2, 3$, consider an ordering of the points $p(s_i, f)$ on s_i, $f \in F$, and connect every two consecutive points by an edge, which is exactly the portion of s_i that lies between the two points. Moreover, for each cell f, we connect $p(f)$ to $p(s_i, f)$ by a curve that lies strictly in the interior of f (except at its endpoints) for all $i = 1, 2, 3$. Then, the edge set $E(H)$ of H consists of the set of all edges connecting the consecutive points as well as the curves $(p(f), p(s_i, f))$ for all $f \in F$ and $i = 1, 2, 3$.

If $|F| > 2$, then take any three cells $f_1, f_2, f_3 \in F$ and consider the subgraph H' of H induced by $\{p(f_i) : i = 1, 2, 3\} \cup \{p(s_i, f_j) : i, j = 1, 2, 3\}$. Now, consider the graph G constructed from H' as follows. For each s_i, $i = 1, 2, 3$, we place a new vertex $v(s_i)$ close to s_i and connect it to the three vertices $p(s_i, f_j)$ for all $j = 1, 2, 3$ such that the resulting graph remains planar. One can easily verify that this is doable since the three line segments are disjoint and so there are a few cases for where to place $v(s_i)$ depending on which side of s_i the three cells

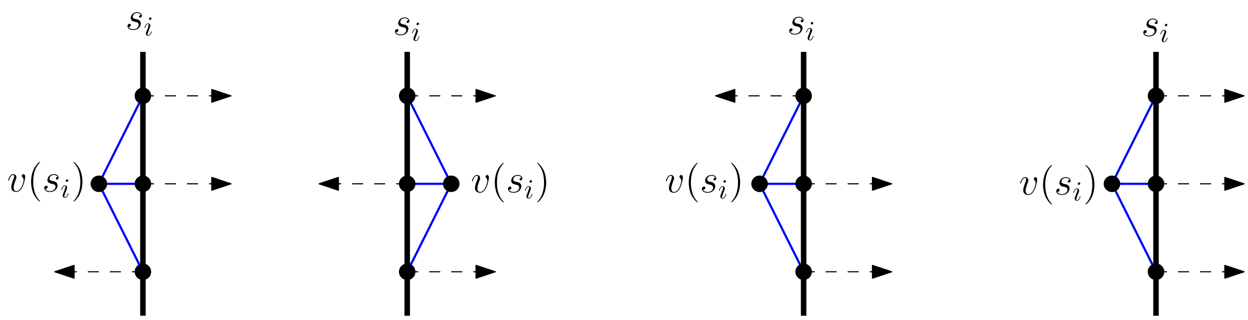


Figure 3.3: Placing a new vertex $v(s_i)$ close to s_i and connecting it to the vertices corresponding to the incident cells. The arrows indicate the side on which the cell lies.

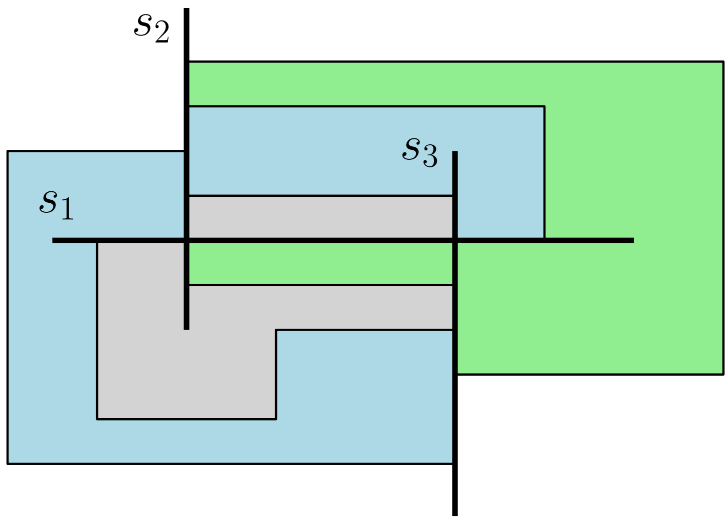


Figure 3.4: Three line segments s_1, s_2 and s_3 cover six cells.

lie; see Figure 3.3. Observe that the resulting graph G is a planar drawing of a subdivision of $K_{3,3}$, which is not possible. So, $|F| \leq 2$.

We now show that, in general, $|F| \leq 6$. Indeed, note that, since the line segments are axis-parallel, we may assume, without loss of generality, that s_1 does not intersect s_3, with s_2 being split into at most three parts by s_1 and s_3. If $|F|$ were more than 6, there would be at least one such part of s_2 covering at least three faces in F. By shortening s_2 to a segment s_2' that still covers these faces but that intersects neither s_1 nor s_3, we are left with three subsegments s_1, s_2' and s_3 that are pairwise disjoint but that simultaneously cover three faces of $\mathcal{A}(S)$, which contradicts what we just showed. □

We note that the upper bound in Lemma 7 is tight as Figure 3.4 shows an example with three line segments that cover six cells. We now apply Lemma 7 to obtain our FPT algorithm. We first formulate the LSC problem as a hitting set problem as follows. The ground set is S and for each cell in $\mathcal{A}(S)$, there exists a set that contains the line segments that cover the cell. Let $\mathcal{C}$ be the resulting set of subsets of S. Then, the LSC problem is equivalent to selecting a minimum number of elements from S such that each set in $\mathcal{C}$ is hit by at least one selected element.

We first reduce the set $\mathcal{C}$ to a set $\mathcal{C}_1$ as follows. For every pair of line segments $\{s_i, s_j\} \in S$, if they appear in more than $6k$ sets in $\mathcal{C}$, then we remove all such sets from $\mathcal{C}$ and add the set $\{s_i, s_j\}$ to $\mathcal{C}$. Let $\mathcal{C}_1$ be the resulting set.

Lemma 8. *A set $S' \subseteq S$ with $|S'| \leq k$ is a minimum-size cover of $\mathcal{C}$ if and only if it is a minimum-size cover of $\mathcal{C}_1$.*

Proof. We prove the lemma by an argument similar to the one by Korman et al. [33]. The lemma clearly follows if $\mathcal{C} = \mathcal{C}_1$. So, assume that $X = \mathcal{C}_1 \setminus \mathcal{C}$ and $Y = \mathcal{C} \setminus \mathcal{C}_1$ are two non-empty sets. Let S' with $|S'| \leq k$ be a minimum-size cover for $\mathcal{C}_1$. First, S' is also a cover for $\mathcal{C}$ because for every set $M \in Y$ there exists a pair of line segment s_i and s_j such that both s_i and s_j are in M and we have $\{s_i, s_j\} \in X$. We now prove that S' is also a cover of minimum size for $\mathcal{C}$.

Suppose for a contradiction that there exists a cover S'' for $\mathcal{C}$ such that $|S''| < |S'|$. Then, S'' cannot be a cover of $\mathcal{C}_1$ because S' is a cover of minimum size for $\mathcal{C}_1$. Since S'' covers $\mathcal{C} \cap \mathcal{C}_1$, there must exist $\{s_i, s_j\} \in X$ such that neither s_i nor s_j is in S''. But, we introduced the set $\{s_i, s_j\}$ into $\mathcal{C}_1$ because there were more than $6k$ sets containing both s_i and s_j. If neither s_i nor s_j is in S'', then every other line segment can cover at most six of such subsets by Lemma 7. Therefore, $|S''| > k$ — a contradiction. By a similar argument, we can show that a minimum-size cover of $\mathcal{C}$ is also a minimum-size cover for $\mathcal{C}_1$. This completes the proof of the lemma. ◻

Next, we construct a new set $\mathcal{C}_2$ from $\mathcal{C}_1$ as follows. For each line segment $s \in S$, we count how many sets in $\mathcal{C}_1$ contain s. If s appears in more than $6k^2$, then we remove all those sets and add the set $\{s\}$ to $\mathcal{C}_1$. Let $\mathcal{C}_2$ denote the resulting set.

Lemma 9. *A set $S' \subseteq S$ with $|S'| \leq k$ is a minimum-size cover for $\mathcal{C}_1$ if and only if it is a minimum-size cover for $\mathcal{C}_2$.*

Proof. The lemma follows if $\mathcal{C}_1 = \mathcal{C}_2$. So, assume that $X' = \mathcal{C}_2 \setminus \mathcal{C}_1$ and $Y' = \mathcal{C}_1 \setminus \mathcal{C}_2$ are two non-empty sets. Let S' with $|S'| \leq k$ be a minimum-size cover for $\mathcal{C}_2$. For any set $M \in Y'$, there exists a singleton set in X' whose member is in M. This means that S' is also a cover for $\mathcal{C}_1$. We next show that S' is also a minimum-size cover for $\mathcal{C}_1$.

Suppose for a contradiction that there exists a cover S'' for $\mathcal{C}_1$ such that $|S''| < |S'|$. Therefore, S'' is not a cover of $\mathcal{C}_2$. Since S'' covers $\mathcal{C}_1 \cap \mathcal{C}_2$, there must exist a set in X' that is not covered by S''. Notice that this set must be of size 1 from the construction of $\mathcal{C}_2$; let $\{s\}$ be such a set, where $s \in S$. The reason we have the set $\{s\}$ in $\mathcal{C}_2$ is that because there were more than $6k^2$ sets in $\mathcal{C}_1$ containing s. If s is not in S'', then all such sets of $\mathcal{S}_1$ must be covered by other line segments. But, from the construction of $\mathcal{C}_1$, every pair of line segments can appear in at most $6k$ sets. So, $|S''|$ must be greater than k, which is a contradiction. A similar argument can be used to show that a minimum-size cover for $\mathcal{C}_1$ is also is minimum-size cover for $\mathcal{C}_2$. This completes the proof of the lemma. ◻

Consider the set $\mathcal{C}_2$. By Lemma 9, no line segment of S appears in more than $6k^2$ sets in $\mathcal{C}_2$. Therefore, if $|\mathcal{C}_2| > 6k^3$, then the problem does not have a cover of size at most k. Since the construction of $\mathcal{C}_2$ can be done in polynomial time, we have the following result.

Lemma 10. *For the* LSC *problem on a set of axis-parallel line segments, in polynomial time, we can either obtain a kernel of size $O(k^3)$ or conclude that the problem does not have a cover of size at most k, where k is the size of an optimal cover.*

Since having a kernel of size $O(k^3)$ implies that the problem is FPT [22], we have the main result of this section.

Theorem 11. *The line segment covering* (LSC) *problem on a set of axis-parallel line segments is* FPT *with respect to the size of an optimal cover.*

3.5.1 Generalization to Arbitrary Segments

The requirement that the segments in S be horizontal or vertical stems from our previous inability to generalize Lemma 7 to arbitrary sets of non-overlapping segments. In this section, we lift that limitation by proving such a generalization of the lemma in a surprisingly simpler way. Note, however, that since our bound on the number of faces simultaneously covered by three segments increases from 6 to 8, the size of the kernel increases accordingly.

Lemma 3.1. *Let S be a set of non-overlapping line segments in the plane. Then the number of faces in $\mathcal{A}(S)$ covered simultaneously by three distinct segments in S is at most two plus twice the number of pairwise crossings between these segments.*

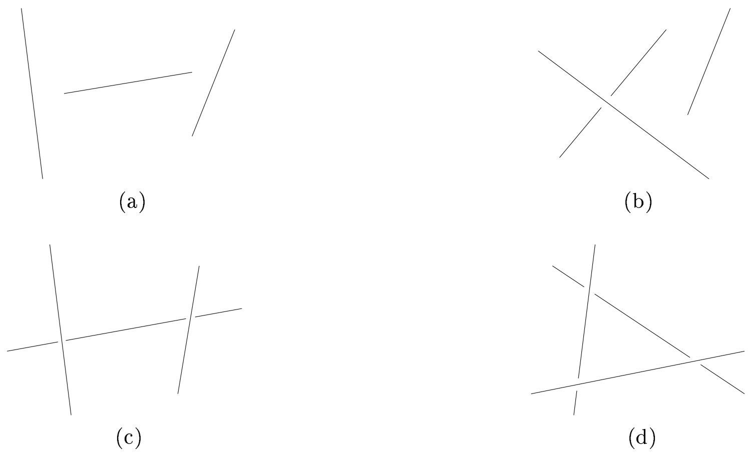


Figure 3.5

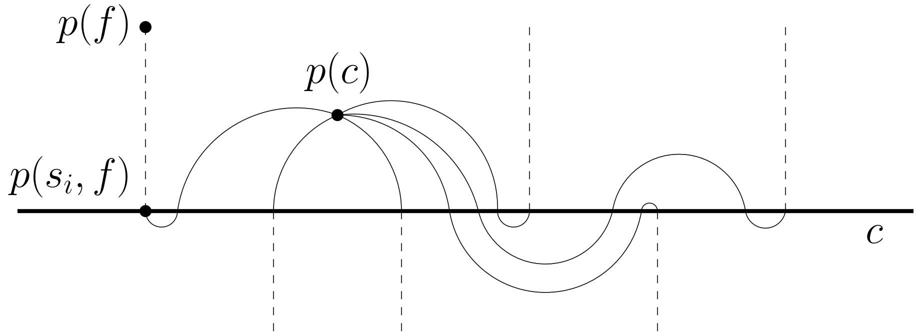


Figure 3.6

Proof. Let F be the set of faces in $\mathcal{A}(S)$ simultaneously covered by distinct segments $s_1, s_2, s_3 \in S$. For each face $f \in F$, pick a point $p(f)$ in the interior of f and, for each $i \in \{1, 2, 3\}$, pick a point $p(s_i, f)$ on the intersection of s_i and the boundary of f that does not belong to any other segment in S.

We shall construct a plane bipartite graph H as follows. For each face $f \in F$, find three paths inside f from $p(f)$ to $p(s_i, f)$ for $i \in \{1, 2, 3\}$ that are, save at $p(f)$, pairwise disjoint. Split s_1, s_2 and s_3 into $3 + k$ disjoint chunks containing all points $p(s_i, f)$ as in Figure 3.5, where k is the number of pairwise crossings among these segments. For each chunk c, say from segment s_i, pick a point $p(c)$ outside c but sufficiently close to it and find paths from $p(c)$ to the points $p(s_i, f)$ for faces $f \in F$ such that $p(s_i, f)$ lies in c (Figure 3.6). Make sure the paths for a chunk c are pairwise disjoint save at $p(c)$, that the paths from different chunks are pairwise disjoint and that chunk paths are disjoint from face paths save at common endpoints $p(s_i, f)$. The vertices of H are $p(c)$ for chunks c on one partition and $p(f)$ for $f \in F$ on another. To connect $p(c)$ to $p(f)$, find the segment s_i from which c originates and join the paths from $p(c)$ to $p(s_i, f)$ and from $p(f)$ to $p(s_i, f)$ into an edge.

Since there are three outgoing paths from $p(f)$ for $f \in F$, graph H has $3|F|$ edges. However, it has $3 + k + |F|$ vertices, so $3|F| \leq 2(3 + k + |F|) - 4$, or $|F| \leq 2 + 2k$, as H is plane and bipartite. $\square$

3.6 Conclusion

In this paper, we considered the problem of covering the cells in the arrangement of a set of line segments in the plane. We proved that the problem admits a PTAS when the covering line segments can be selected from only one orientation. We then showed that if we allow selecting the covering line segments from more than one orientation, then the problem is APX-hard when we are interested in covering the rectangular cells. Finally, we gave an FPT algorithm for the problem when the line segments have only two orientations, but the goal is to cover all the cells.

We conclude with several open problems. Our APX-hardness rules out the possibility of a PTAS for "covering rectangular cells" variant of the problem, but is there a 2-approximation algorithm for the problem? For the more general variant, where the line segments are in any orientation, covering line segments can be selected from any orientation and the goal is to cover all the cells, can we obtain a c-approximation algorithm for some small constant c?

Acknowledgements

The research of Paz Carmi is supported by Grant 2016116 from the United States-Israel Binational Science Foundation. The research of Anil Maheshwari is supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC). We thank the anonymous reviewers for their helpful comments, and for pointing us out to [44].

Chapter 4

Optimal Road Placement in a Square

Paz Carmi[1] Jean-Lou De Carufel[2]
Stephane Durocher[3] Darryl Hill[4]
Luís Fernando Schultz Xavier da Silveira[5]
Thomas Shermer[6]

Abstract

We present optimal solutions to the problem of placing roads of bounded total length in the plane so as to minimize the maximum travel time between two opposing sides of a square given that traveling on and off the roads is accomplished at different speeds. We also consider an open variant of this problem regarding circles instead of squares.

4.1 Introduction

We study a simple yet interesting problem. One has been assigned the task of laying down roads in the plane. The goal is to facilitate trips from two given opposing sides of a unit square: trips off the roads require one unit of time per unit of distance, while trips on the roads are faster, requiring only α units of time per unit of distance, where $0 \leq \alpha < 1$. The metric by which we judge road networks is the maximum (shortest) travel time between two points, one on each of the opposing sides. Furthermore, there is a budget constraint: the total length of the roads may not exceed a parameter $\beta > 0$.

In this paper, we show road networks that make this maximum travel time the least possible (Figure 4.1). This is for any choice of α, but limitations in our methodology demand the assumption that $\beta \leq \sqrt{2}$.

It is most natural for one to think of roads as rectifiable curves and of road networks as finite sets of such curves whose total length does not exceed the budget. Unfortunately, such a natural definition would burden our reasoning about travel times and fastest paths with real analysis technicalities. Therefore, we define roads as line segments and road networks as finite sets of non-overlapping line segments. There should be little objection to this definition because rectifiable curves can be arbitrarily well approximated by line segments whose total length is not greater.

[1]Department of Computer Science, Ben Gurion University of the Negev, Beer Sheva, Israel
[2]School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada
[3]Department of Computer Science, University of Manitoba, Winnipeg, Canada
[4]School of Computer Science, Carleton University, Ottawa, Canada
[5]School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada
[6]School of Computing Science, Simon Fraser University, Vancouver, Canada

The remainder of this paper is organized as follows. In Section 4.2, we argue that road networks admit polygonal fastest paths between any two points in the plane. We also argue that the objective value of road networks, i.e., their maximum travel time, is well defined. In Section 4.3, we prove the optimality of our road network designs. Finally, in Section 4.4, we consider a variant of this problem regarding circles which, as far as we know, remains open.

The following notation is used throughout the paper. The Euclidean norm of a vector $v \in \mathbb{R}^2$ is denoted by $|v|$. The length of a rectifiable curve γ is denoted by $|\gamma|$. The length of a set $S \subseteq \mathbb{R}^2$ that is a finite union of line segments is denoted by $|S|$. The *reverse* of a curve $\gamma : [0;1] \to \mathbb{R}^2$ is the curve $\gamma^\top : [0;1] \to \mathbb{R}^2$ given by $\gamma^\top(t) = \gamma(1-t)$.

4.2 Road Networks and Fastest Paths

For a given finite set R of non-overlapping line segments (the road network), the goals of this section are: to show that there exists an injective polygonal fastest path (with respect to R) between any two points in the plane; to define a metric space $(\mathbb{R}^2, \delta_R)$ of fastest path (time) costs; and to prove that the maximum travel time between two opposing sides of a square (with respect to R) is well defined. However, this is a rather technical section and we recommend readers to whom these goals seem obvious skip to Section 4.3.

Let $p, q \in \mathbb{R}^2$ be two points in the plane and let γ be a path between them, i.e., $\gamma : [0;1] \to \mathbb{R}^2$ is a rectifiable curve with $\gamma(0) = p$ and $\gamma(1) = q$. In some order, for each road $s \in R$, if s intersects $\mathrm{Im}(\gamma)$, let $t_s = \min\{t : \gamma(t) \in s\}$ and $t'_s = \max\{t : \gamma(t) \in s\}$ (these exist since s is a closed set of $\mathbb{R}^2$) and replace the restriction of γ to $[t_s; t'_s]$ with a straight-line path contained in s. The resulting path is not slower than the original one as each part that is replaced becomes a straight-line path fully sped up by a road. Furthermore, for any $t \in [0;1]$ that has not been replaced, $\gamma(t)$ can not be on any road. Indeed, if it were on a road s, then when our procedure processed s, since t was never replaced, $\gamma(t) \in s$ and so $t \in [t_s; t'_s]$, which would get it replaced, a contradiction. Therefore, the finitely many parts of γ that were not replaced can also be replaced by line-segments as the original parts were not sped up. This yields a polygonal path γ' between p and q that is not slower than γ.

We also run a procedure to refine γ': while there is some road $s \in R$ that intersects $\mathrm{Im}(\gamma')$ but is such that the restriction of γ' to $\{t : \gamma'(t) \in s\}$ is not a straight-line path (for example if $\{t : \gamma'(t) \in s\}$ is not a connected set), we replace the restriction of γ' to $[\min\{t : \gamma'(t) \in s\}; \max\{t : \gamma'(t) \in s\}]$ by a straight-line path contained in s. This procedure ends since each iteration must reduce the number of bends in the path. Since each replacement is by a straight-line path fully sped up by a road, the resulting path γ'' is not slower than γ'. Furthermore, since the parts of γ' outside the roads were all straight lines and since no parts outside the roads were introduced, γ'' is in the set $\mathscr{P}$ of polygonal paths from p to q that do not re-enter roads and travel in straight lines between p, q and entry and exit points of roads. Therefore, to accomplish the first goal of this section, we need only show that there is a fastest path among the paths in $\mathscr{P}$.

To that end, we introduce the notion of path encodings. Pick a subset $S \subseteq R$ of size k, a permutation $\pi : \{0, \ldots, k-1\} \to S$ and, for every $i \in \{0, \ldots, k-1\}$, two coefficients $\lambda_i, \lambda'_i \in [0;1]$. Under some convention, define $a_{S,\pi,i}$ and $b_{S,\pi,i}$ as the endpoints of segment $\pi(i)$ and pack all the coefficients λ_i, λ'_i into a vector $\Lambda \in [0;1]^{2k}$. The encoded path $\gamma_{S,\pi,\Lambda}$ is constructed as follows: start at p; for each $i \in \{0, \ldots, k-1\}$ in increasing order, go to $p_{S,\pi,\Lambda,i} = (1-\lambda_i)a_{S,\pi,i} + \lambda_i b_{S,\pi,i}$ in a straight line and then to $q_{S,\pi,\Lambda,i} = (1-\lambda'_i)a_{S,\pi,i} + \lambda'_i b_{S,\pi,i}$ along the segment $\pi(i)$; go to q in a straight line. Note that all paths in $\mathscr{P}$ can be encoded in this manner: pick S as the set of roads in R that overlap with the path, π as the permutation that sorts S by entry order and the Λ which correctly encodes road entry and exit points. This is the *canonical encoding* of a path in $\mathscr{P}$.

Unfortunately, the cost of $\gamma_{S,\pi,\Lambda}$ may not be a continuous function of Λ due to unintended speed-ups. To rectify this, we define an intended path cost function

$$f(S, \pi, \Lambda) = |\gamma_{S,\pi,\Lambda}| - (1-\alpha) \sum_{i=0}^{k-1} |q_{S,\pi,\Lambda,i} - p_{S,\pi,\Lambda,i}|.$$

This function is continuous on Λ and, because there are only finitely many choices of S and π and because $[0;1]^{2k}$ is a compact set, we may apply Weierstrass' theorem (also known as the extreme value theorem) and conclude that there are S^*, π^* and Λ^* minimizing $f(S^*, \pi^*, \Lambda^*)$.

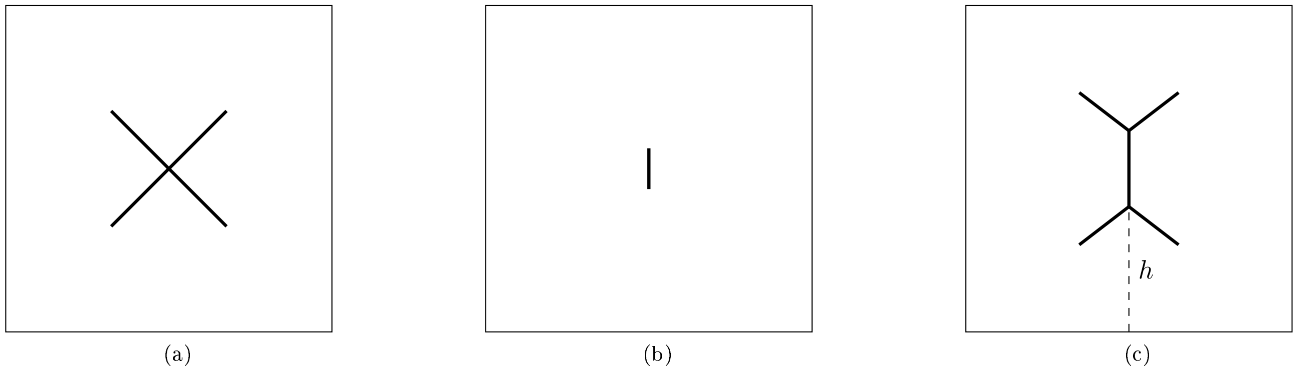


Figure 4.1: Optimal road network layouts when (a) $\alpha \geq \sqrt{2}-1$, when (b) $\alpha < \sqrt{2}-1$ and $\beta \leq 1-2h$ and when (c) $\alpha < \sqrt{2}-1$ and $\beta > 1-2h$.

Observe now that the value through f of the canonical encoding of a path in $\mathscr{P}$ is that path's cost, which implies that every path in $\mathscr{P}$ costs at least $f(S^*, \pi^*, \Lambda^*)$. On the other hand, recall that there is a path $\gamma'' \in \mathscr{P}$ whose cost is not greater than the cost of $\gamma_{S^*,\pi^*,\Lambda^*}$, which is upper-bounded by $f(S^*, \pi^*, \Lambda^*)$. It follows that this polygonal path γ'' costs no more than any path in $\mathscr{P}$ and is thus a fastest path. Furthermore, clipping any loops in γ'' does not make the path slower, leaving us with an injective polygonal fastest path.

Next we define the metric space $(\mathbb{R}^2, \delta_R)$ of fastest path costs with respect to R by letting $\delta_R(p,q)$ be the time a fastest path between p and q takes. That a metric space is properly formed easily follows from fastest path properties, such as the triangle inequality.

As a matter of fact, δ_R satisfying the triangle inequality implies it is continuous. Therefore, so is the function

$$\begin{array}{rccl} f: & [0;1]^2 & \to & \mathbb{R} \\ & (x,y) & \mapsto & \delta_R((1-x)A + xB, (1-y)D + yC), \end{array}$$

where $A, B, C, D \in \mathbb{R}^2$ are the corners of the unit square, as depicted in Figure 4.2. The maximum value attained by f, which exists due to Weierstrass' theorem, is therefore the maximum travel time of the road network R.

4.3 Optimal Road Networks across a Square

In this section, for any fixed parameters α and β with $0 \leq \alpha < 1$ and $0 < \beta \leq \sqrt{2}$, we construct a (line segment) road network and prove it has minimum maximum travel time between any point in the top side of the unit square and any point in its bottom side. More precisely, if $\alpha \geq \sqrt{2}-1$, the layout is a centered cross with legs of size $\beta/4$ (Figure 4.1a). Otherwise, let

$$h = \frac{1+\alpha}{2\sqrt{4-(1+\alpha)^2}} \in \left[\frac{1}{2\sqrt{3}}, \frac{1}{2}\right).$$

If $\beta \leq 1-2h$, then the layout is a centered vertical line segment of length β (Figure 4.1b). Finally, if $\alpha < \sqrt{2}-1$ and $\beta > 1-2h$, then the layout is as in Figure 4.1c.

To arrive at these network designs and to show their optimality, our strategy is to first optimize crossings rather than road networks. A *crossing* is a pair of paths between opposing corners of the unit square. More precisely, if $\Gamma(p,q)$ is the set of injective polygonal paths from point $p \in \mathbb{R}^2$ to point $q \in \mathbb{R}^2$, then the set of crossings is $\Gamma(A,C) \times \Gamma(B,D)$, which we denote simply by Γ^2, where $A, B, C, D \in \mathbb{R}^2$ are the corners of the unit square, as in Figure 4.2. In order to define the cost of a crossing, we start by defining its cost with respect to a particular road network R. To that end, note that, for any injective polygonal path γ, it is straightforward to define $c_R(\gamma)$, the cost of γ with respect to R: we simply let $c_R(\gamma) = |\gamma| - (1-\alpha)|\mathrm{Im}(\gamma) \cap R|$. The cost of a crossing $(\gamma_0, \gamma_1) \in \Gamma^2$ with respect to R is then defined as $c_R(\gamma_0, \gamma_1) = \max\{c_R(\gamma_0), c_R(\gamma_1)\}$.

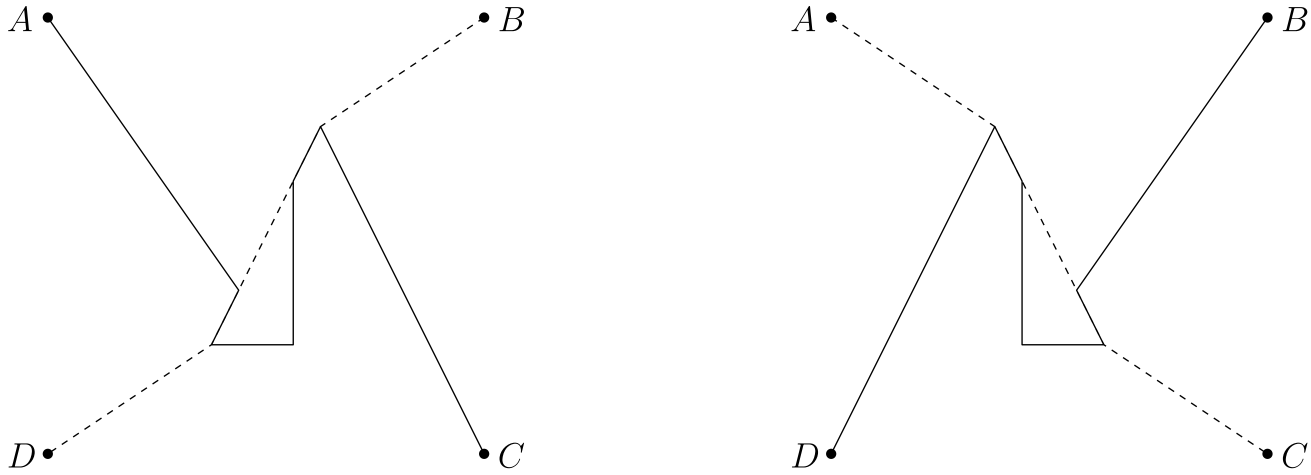


Figure 4.2: A crossing $(\gamma_0, \gamma_1) \in \Gamma^2$ with $|\gamma_0| > |\gamma_1|$ and the crossing $(T \circ \gamma_1, T \circ \gamma_0)$.

We would like to define the cost $c(\gamma_0, \gamma_1)$ of a crossing $(\gamma_0, \gamma_1) \in \Gamma^2$ as the minimum of $c_R(\gamma_0, \gamma_1)$ over all road networks R. While demonstrating that this minimum is indeed achievable, we will compute it explicitly in terms of $|\gamma_0|$, $|\gamma_1|$ and $|\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)|$. We assume that $|\gamma_0| \leq |\gamma_1|$, abbreviate $|\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)|$ as ℓ and consider three cases. First, if $\beta \leq \ell$, it is clear that the best we can do is place all the roads in $\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)$, where we speed up both paths, yielding a cost of $|\gamma_1| - (1-\alpha)\beta$. For larger budgets, the remainder can only speed up γ_0 or γ_1 individually, so our first priority must be to bring down the cost of γ_1 to that of γ_0. Therefore, if $\ell < \beta \leq \ell + (|\gamma_1| - |\gamma_0|)/(1-\alpha)$, we should first fill $\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)$ with roads and then put the remaining roads on $\mathrm{Im}(\gamma_1) \setminus \mathrm{Im}(\gamma_0)$, also yielding a cost of $|\gamma_1| - (1-\alpha)\beta$. Finally, if $\beta > \ell + (|\gamma_1| - |\gamma_0|)/(1-\alpha)$, then, after filling $\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)$ and equating the costs of γ_0 and γ_1, we should evenly divide the remaining budget between the unpaved parts of γ_0 and γ_1, resulting in a cost of

$$|\gamma_0| - (1-\alpha)\ell - \frac{1-\alpha}{2}\left(\beta - \ell - \frac{|\gamma_1| - |\gamma_0|}{1-\alpha}\right) = \frac{|\gamma_0| + |\gamma_1|}{2} - (1-\alpha)\frac{\beta+\ell}{2}.$$

Note that there is no danger of oversaturating the paths with roads because $\beta \leq \sqrt{2} \leq |\gamma_0| \leq |\gamma_1|$. Putting all this together, we define, for $|\gamma_0| \leq |\gamma_1|$,

$$c(\gamma_0, \gamma_1) = \begin{cases} |\gamma_1| - (1-\alpha)\beta, & \beta \leq \ell + \dfrac{|\gamma_1| - |\gamma_0|}{1-\alpha} \\ \dfrac{|\gamma_0| + |\gamma_1|}{2} - (1-\alpha)\dfrac{\beta+\ell}{2}, & \text{otherwise.} \end{cases}$$

To define $c(\gamma_0, \gamma_1)$ when $|\gamma_0| > |\gamma_1|$, first consider the linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ that orthogonally reflects a point around the bisector of A and B. A crossing $(\gamma_0, \gamma_1) \in \Gamma^2$ with $|\gamma_0| > |\gamma_1|$ then has a corresponding crossing $(T \circ \gamma_1, T \circ \gamma_0) \in \Gamma^2$ with $|T \circ \gamma_1| = |\gamma_1| < |\gamma_0| = |T \circ \gamma_0|$ (Figure 4.2). Exploiting the symmetry of the transformation, we define $c(\gamma_0, \gamma_1) = c(T \circ \gamma_1, T \circ \gamma_0)$.

Having defined a cost function $c : \Gamma^2 \to \mathbb{R}$, we now turn our attention to finding a crossing of minimum cost. In that pursuit, the following lemma will prove itself very useful.

Lemma 4.1. *Let $(\gamma_0, \gamma_1), (\gamma_0', \gamma_1') \in \Gamma^2$ be two crossings satisfying:*

- $\max\{|\gamma_0'|, |\gamma_1'|\} \leq \max\{|\gamma_0|, |\gamma_1|\}$*; and*
- $|\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| - |\mathrm{Im}(\gamma_0') \cap \mathrm{Im}(\gamma_1')| \leq \dfrac{|\gamma_0| + |\gamma_1| - |\gamma_0'| - |\gamma_1'|}{1-\alpha}$.

Then $c(\gamma_0', \gamma_1') \leq c(\gamma_0, \gamma_1)$.

Proof. The key observation is that the definition of the cost of a crossing only depends on the length of its paths and the length of the intersection of their images. Acting on this observation, for non-negative real numbers x, y and z, we define

$$c(x,y,z) = \begin{cases} y-(1-\alpha)\beta, & \beta \le z + \dfrac{y-x}{1-\alpha} \\ \dfrac{x+y}{2} - (1-\alpha)\dfrac{\beta+z}{2}, & \text{otherwise.} \end{cases}$$

This is a continuous function on x, y and z since, when $\beta = z + (y-x)/(1-\alpha)$,

$$y-(1-\alpha)\beta = x-(1-\alpha)z = \frac{x+y}{2} - (1-\alpha)\frac{\beta+z}{2}.$$

Note that $c(\gamma_0,\gamma_1) = c(x,y,z)$ for $x = \min\{|\gamma_0|,|\gamma_1|\}$, $y = \max\{|\gamma_0|,|\gamma_1|\}$ and $z = |\mathrm{Im}(\gamma_0)\cap\mathrm{Im}(\gamma_1)|$. Similarly, $c(\gamma_0',\gamma_1') = c(x',y',z')$ for $x' = \min\{|\gamma_0'|,|\gamma_1'|\}$, $y' = \max\{|\gamma_0'|,|\gamma_1'|\}$ and $z' = |\mathrm{Im}(\gamma_0')\cap\mathrm{Im}(\gamma_1')|$. Therefore, we need only show that $c(x',y',z') \le c(x,y,z)$ given that $y' \le y$ and $z - z' \le (x+y-x'-y')/(1-\alpha)$.

Consider the closed 3-dimensional half-space $H = \{(x,y,z) \in \mathbb{R}^3 : \beta \le z + (y-x)/(1-\alpha)\}$ and the 3-dimensional path

$$\begin{aligned} f:\quad [0;1] &\to \mathbb{R}^3 \\ t &\mapsto (1-t)\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + t\begin{bmatrix} x \\ y \\ z \end{bmatrix}. \end{aligned}$$

If $f(U) \subseteq H$ for some open set U of $[0;1]$, then we can compute the derivative $(c\circ f)'(t) = y - y' \ge 0$ for $t \in U$. Furthermore, since $\mathbb{R}^3 \setminus H$ is open, whenever $f(t) \in \mathbb{R}^3 \setminus H$, we can also compute the derivative

$$(c\circ f)'(t) = \frac{x-x'+y-y'}{2} - (1-\alpha)\frac{z-z'}{2} \ge 0.$$

Because $\mathrm{Im}(f)$ can either be contained in the boundary of H or intersect this boundary in at most one point, the derivative of the continuous function $c\circ f$ exists and is non-negative in all points of $[0;1]$ except possibly for one. Therefore we are allowed to conclude that $(c\circ f)(0) \le (c\circ f)(1)$ or, equivalently, $c(x',y',z') \le c(x,y,z)$. ☐

In our first application of Lemma 4.1, we show that we may disregard crossings that do not effectively cross in our search for an optimal crossing. Indeed, if $(\gamma_0,\gamma_1) \in \Gamma^2$ is such that $\mathrm{Im}(\gamma_0)\cap\mathrm{Im}(\gamma_1) = \{\}$, then consider the crossing $(\overrightarrow{AC},\overrightarrow{BD}) \in \Gamma^2$. We have that $|\overrightarrow{AC}| = \sqrt{2} \le |\gamma_0|$, that $|\overrightarrow{BD}| = \sqrt{2} \le |\gamma_1|$ and that $|\mathrm{Im}(\overrightarrow{AC})\cap\mathrm{Im}(\overrightarrow{BD})| = 0 = |\mathrm{Im}(\gamma_0)\cap\mathrm{Im}(\gamma_1)|$, so the lemma allows us to conclude that $c(\overrightarrow{AC},\overrightarrow{BD}) \le c(\gamma_0,\gamma_1)$.

Consider then a crossing $(\gamma_0,\gamma_1) \in \Gamma^2$ with $\mathrm{Im}(\gamma_0)\cap\mathrm{Im}(\gamma_1) \ne \{\}$. Since $\mathrm{Im}(\gamma_1)$ is a closed set of $\mathbb{R}^2$, we can define the points

$$X = \gamma_0(\min\{t : \gamma_0(t) \in \mathrm{Im}(\gamma_1)\}) \quad \text{and} \quad Y = \gamma_0(\max\{t : \gamma_0(t) \in \mathrm{Im}(\gamma_1)\}).$$

Because γ_0 is injective, the points X and Y determine a piece of γ_0. Likewise, a piece of γ_1 is determined, but while γ_0 reaches X before Y, γ_1 may reach Y before X. However, if that is the case, we can use the linear transformation $T' : \mathbb{R}^2 \to \mathbb{R}^2$ that orthogonally reflects points around the line $\overleftrightarrow{AC}$ to obtain a crossing $(T'\circ\gamma_0, T'\circ\gamma_1^\top) \in \Gamma^2$, as depicted in Figure 4.3. Since $|T'\circ\gamma_0| = |\gamma_0|$, $|T'\circ\gamma_1^\top| = |\gamma_1|$ and $|\mathrm{Im}(T'\circ\gamma_0)\cap\mathrm{Im}(T'\circ\gamma_1^\top)| = |\mathrm{Im}(\gamma_0)\cap\mathrm{Im}(\gamma_1)|$, we have $c(\gamma_0,\gamma_1) = c(T'\circ\gamma_0, T'\circ\gamma_1^\top)$ by Lemma 4.1. Therefore, we may assume, without loss of generality, that γ_1, like γ_0, moves from X to Y.

In order to successfully apply Lemma 4.1 again, we introduce a few definitions. The saturation of a point $p \in \mathbb{R}^2$ to a (closed) half-plane $H \subseteq \mathbb{R}^2$ is p itself if $p \in H$ and the orthogonal projection of p on the line at the boundary of H if $p \notin H$. The saturation of a point $p \in \mathbb{R}^2$ to a convex polygon is the point in that polygon obtained by iteratively saturating p to each of the finitely many half-planes that contain the polygon and whose boundaries contain one of its edges (under some order convention). Because saturating a point to a half-plane does not increase its distance to points in that half-plane and because simultaneously orthogonally projecting two points on a line does not increase their distance to each other, the points X' and Y' obtained by respectively saturating X and Y to the square $ABCD$ satisfy $|X'Y'| \le |XY|$ and, for each corner $Q \in \{A,B,C,D\}$ of the square $ABCD$, both $|QX'| \le |QX|$ and $|QY'| \le |QY|$ hold.

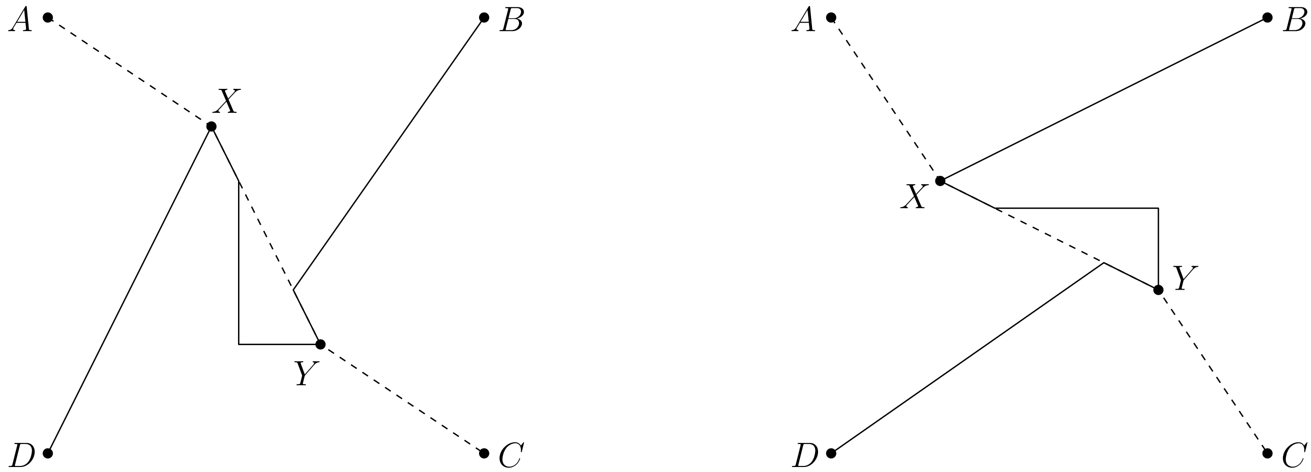


Figure 4.3: Example of crossings (γ_0, γ_1) and $(T' \circ \gamma_0, T' \circ \gamma_1^\top)$.

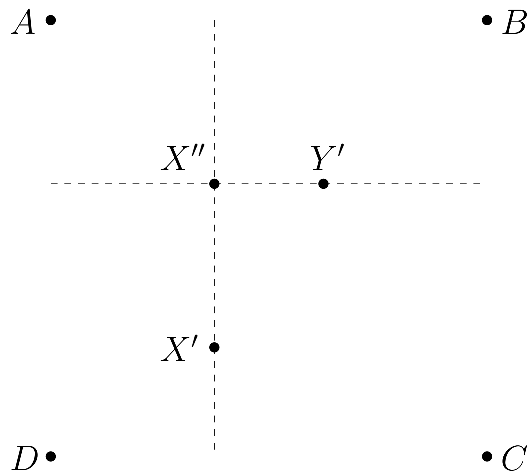


Figure 4.4: Rectifying the situation where X' is closer to $\overleftrightarrow{DC}$ than Y'.

It is rather inconvenient that X' could be closer to the line $\overleftrightarrow{DC}$ than Y', so we define the point $X'' \in \mathbb{R}^2$ as X' if this is not the case and as the intersection of the line parallel to $\overleftrightarrow{AD}$ containing X' and the line parallel to $\overleftrightarrow{AB}$ containing Y' otherwise (Figure 4.4). Since X' and Y' are in the square $ABCD$, it is straightforward to verify that $|AX''| \leq |AX'|$, $|BX''| \leq |BX'|$ and $|X''Y'| \leq |X'Y'|$.

Next, we apply Lemma 4.1 with $(\gamma_0', \gamma_1') = (\overrightarrow{AX''Y'C}, \overrightarrow{BX''Y'D})$. Since both γ_0 and γ_1 visit X then Y, we have that

$$\begin{aligned} |\gamma_0| &\geq |AX| + |XY| + |YC| \\ &\geq |AX'| + |X'Y'| + |Y'C| \\ &\geq |AX''| + |X''Y'| + |Y'C| \\ &= |\gamma_0'| \end{aligned}$$

and

$$\begin{aligned}
|\gamma_1| &\geq |BX| + |XY| + |YD| \\
&\geq |BX'| + |X'Y'| + |Y'D| \\
&\geq |BX''| + |X''Y'| + |Y'D| \\
&= |\gamma_1'|,
\end{aligned}$$

from where we have the first hypothesis of the lemma satisfied. To verify the second one, observe that

$$|\gamma_0'| + |\gamma_1'| = |AX''| + |BX''| + 2|X''Y'| + |Y'C| + |Y'D|$$

while, since $\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)$ is contained in the piece of γ_0 from X to Y,

$$\begin{aligned}
|\gamma_0| + |\gamma_1| &\geq |AX| + |\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| + |YC| + |BX| + |XY| + |YD| \\
&\geq |AX'| + |\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| + |Y'C| + |BX'| + |X'Y'| + |Y'D| \\
&\geq |AX''| + |\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| + |Y'C| + |BX''| + |X''Y'| + |Y'D|,
\end{aligned}$$

so

$$\begin{aligned}
\frac{|\gamma_0| + |\gamma_1| - |\gamma_0'| - |\gamma_1'|}{1-\alpha} &\geq |\gamma_0| + |\gamma_1| - |\gamma_0'| - |\gamma_1'| \\
&\geq |\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| - |X''Y'| \\
&= |\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| - |\mathrm{Im}(\gamma_0') \cap \mathrm{Im}(\gamma_1')|.
\end{aligned}$$

This step was quite important as now we can restrict our attention to crossings of the form $(\overrightarrow{AXYC}, \overrightarrow{BXYD})$ with (X, Y) in the set $\mathcal{XY}$ of pairs of points in the square $ABCD$ whose first point is not closer to the line $\overleftrightarrow{DC}$ than its second one (note that this condition implies that $\overrightarrow{AXYC}$ and $\overrightarrow{BXYD}$ are injective paths). Our optimization problem over pairs of paths has become one over pairs of points, so, for convenience, let us define the cost $c(X, Y)$ of a pair of points $(X, Y) \in \mathcal{XY}$ as $c(\overrightarrow{AXYC}, \overrightarrow{BXYD})$.

Before continuing, let us consider a brief lemma.

Lemma 4.2. *Let ℓ and ℓ' be two parallel lines in $\mathbb{R}^2$ and let A and B be two points in ℓ. Then a point C in ℓ' that minimizes $|AC| + |BC|$ is the one on the bisector of A and B.*

Proof. If $\ell = \ell'$ then clearly the optimal points are the ones on the segment AB. Otherwise, consider the ellipse with foci A and B whose boundary contains C (the point on ℓ' that is on the bisector of A and B). The ellipse is tangent to ℓ' so it contains no other points of ℓ' (Figure 4.5). However, it is the locus of points $P \in \mathbb{R}^2$ with $|AP| + |BP| \leq |AC| + |BC|$, which shows that, for every point $C' \in \ell' \setminus \{C\}$, $|AC'| + |BC'| > |AC| + |BC|$. $\square$

We now proceed to show that we may assume that X and Y lie in the bisector line m of A and B. To that end, for an arbitrary $(X, Y) \in \mathcal{XY}$, define the pair $(X', Y') \in \mathcal{XY}$ by letting X' and Y' be the orthogonal projections of X and Y onto m, respectively. We shall apply Lemma 4.1 to show that $c(X', Y') \leq c(X, Y)$, i.e., $c(\gamma_0', \gamma_1') \leq c(\gamma_0, \gamma_1)$ where $(\gamma_0', \gamma_1') = (\overrightarrow{AX'Y'C}, \overrightarrow{BX'Y'D})$ and $(\gamma_0, \gamma_1) = (\overrightarrow{AXYC}, \overrightarrow{BXYD})$.

Applying Lemma 4.2 to $\overleftrightarrow{AB}$ and the line parallel to it that contains X gives us $|AX'| + |BX'| \leq |AX| + |BX|$. Likewise, applying the lemma to $\overleftrightarrow{DC}$ and the line parallel to it that contains Y yields $|DY'| + |CY'| \leq |DY| + |CY|$. Furthermore, orthogonally projecting points simultaneously on a line never increases their distance, so $|X'Y'| \leq |XY|$.

To verify the hypotheses of the lemma, first note that since $X', Y' \in m$,

$$|\gamma_0'| = |AX'| + |X'Y'| + |Y'C| = |BX'| + |X'Y'| + |Y'D| = |\gamma_1'|,$$

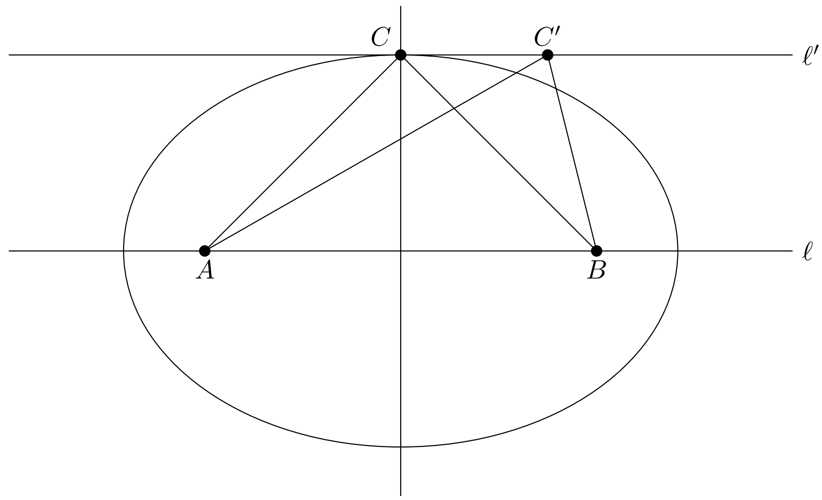


Figure 4.5

so

$$\begin{aligned}
2\max\{|\gamma_0'|, |\gamma_1'|\} &= 2\min\{|\gamma_0'|, |\gamma_1'|\} \\
&\leq |\gamma_0'| + |\gamma_1'| \\
&= |AX'| + |BX'| + 2|X'Y'| + |Y'D| + |Y'C| \\
&\leq |AX| + |BX| + 2|XY| + |YD| + |YC| \\
&= |\gamma_0| + |\gamma_1| \\
&\leq 2\max\{|\gamma_0|, |\gamma_1|\}.
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\frac{|\gamma_0| + |\gamma_1| - |\gamma_0'| - |\gamma_1'|}{1-\alpha} &\geq |\gamma_0| + |\gamma_1| - |\gamma_0'| - |\gamma_1'| \\
&= \begin{array}{c} |AX| + |BX| + 2|XY| + |YD| + |YC| \\ -|AX'| - |BX'| - 2|X'Y'| - |Y'D| - |Y'C| \end{array} \\
&\geq 2|XY| - 2|X'Y'| \\
&\geq |XY| - |X'Y'| \\
&= |\mathrm{Im}(\gamma_0) \cap \mathrm{Im}(\gamma_1)| - |\mathrm{Im}(\gamma_0') \cap \mathrm{Im}(\gamma_1')|.
\end{aligned}$$

With its hypotheses verified, the lemma allows us to conclude that $c(X', Y') = c(\gamma_0', \gamma_1') \leq c(\gamma_0, \gamma_1) = c(X, Y)$ and so we may assume, without loss of generality, that $X, Y \in m$.

Given now a particular $(X, Y) \in \mathcal{XY}$ with $X, Y \in m$, define $(X', Y') \in \mathcal{XY}$ in such a way that $X', Y' \in m$, $|X'Y'| = |XY|$ and the middle point of $X'Y'$ is the center of the square $ABCD$. We wish to show that $c(X', Y') \leq c(X, Y)$, so we can assume that $|XY| < 1$ as $(X', Y') = (X, Y)$ if $|XY| = 1$. Figure 4.6 shows two triangles obtained from XY and $X'Y'$ which, by Lemma 4.2, show that $|AX'| + |Y'D| \leq |AX| + |YD|$. Because m is the bisector of AB and hence also of DC and because $|X'Y'| = |XY|$, Lemma 4.1 can be easily applied to obtain $c(X', Y') \leq c(X, Y)$.

This is an important realization since (X', Y') has one less degree of freedom than (X, Y): it is uniquely determined by the value $|X'Y'|$ or, equivalently, by the angle $\widehat{BAX'}$. Our optimization problem can therefore be reformulated as: find an angle $\theta \in [0; \pi/4]$ for which $c(\theta) = c(X(\theta), Y(\theta))$ is minimized, where $(X(\theta), Y(\theta))$ is the only pair in $\mathcal{XY}$ with $X(\theta), Y(\theta) \in m$ and both angles $\widehat{BAX(\theta)}$ and $\widehat{CDY(\theta)}$ equal to θ (Figure 4.7).

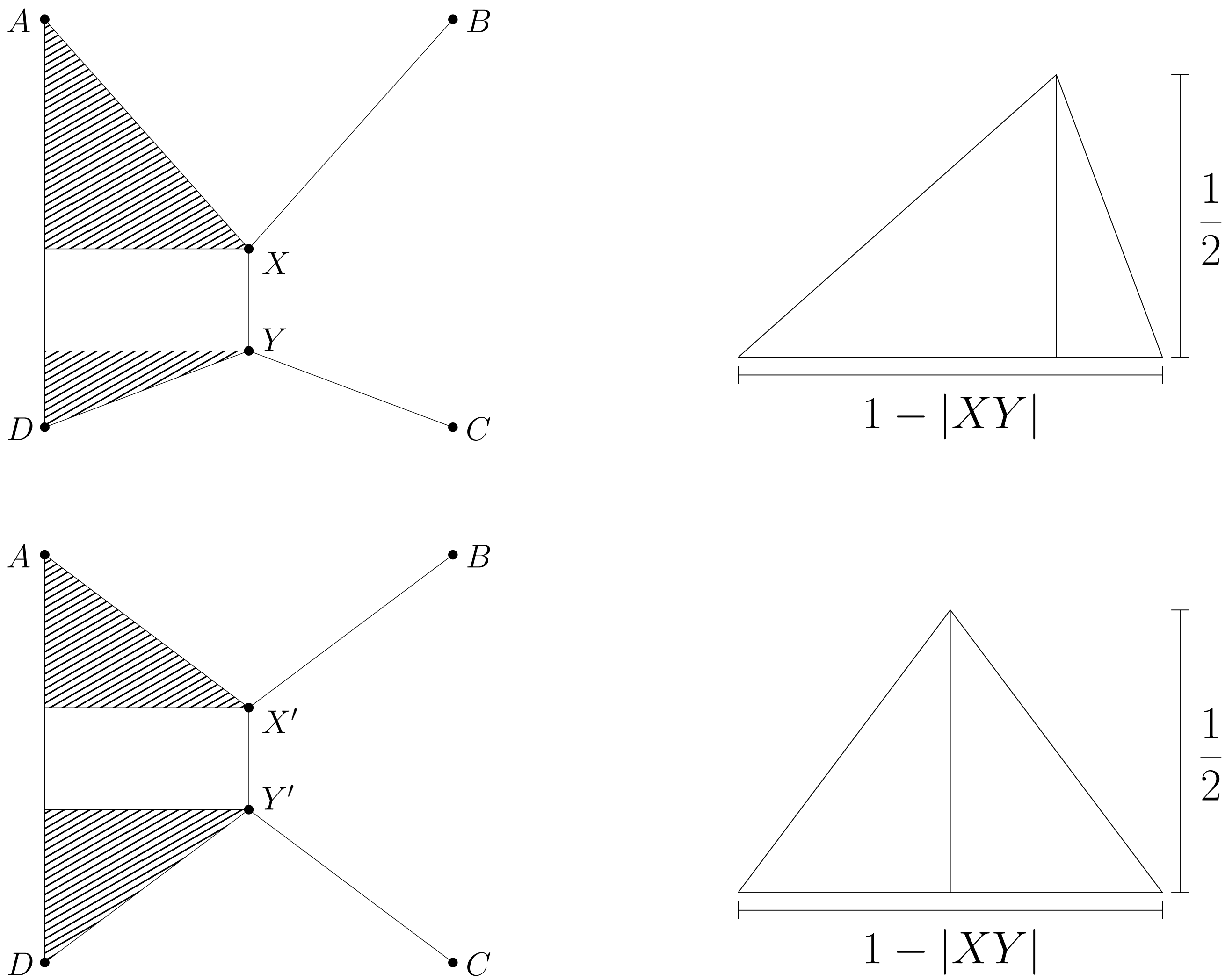


Figure 4.6: Extracting triangles from XY and $X'Y'$.

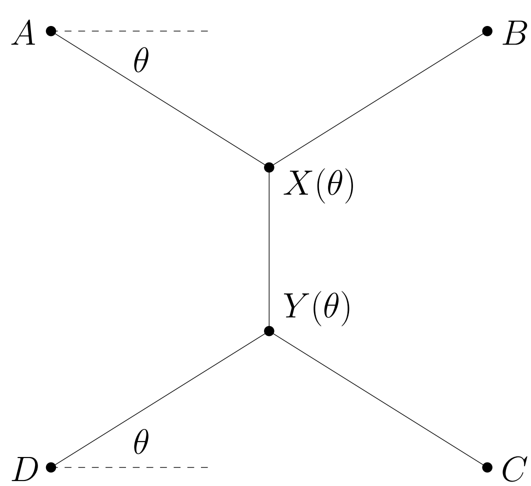


Figure 4.7: A depiction of the crossing induced by a particular angle $\theta \in [0; \pi/4]$.

Recall that $c(X(\theta), Y(\theta))$ is the minimum maximum cost of the paths $\overrightarrow{AX(\theta)Y(\theta)C}$ and $\overrightarrow{BX(\theta)Y(\theta)D}$ that can be obtained by placing at most β units of road in the plane. Recall also that the best way to achieve this is to place as much road as possible on the segment $X(\theta)Y(\theta)$ and to distribute the remainder evenly between the two paths. Therefore, since $|X(\theta)Y(\theta)| = 1 - \tan\theta$ and

$$|AX(\theta)| = |BX(\theta)| = |Y(\theta)D| = |Y(\theta)C| = \frac{1}{2\cos\theta},$$

we may express $c(\theta)$ explicitly as

$$c(\theta) = \begin{cases} \dfrac{1}{\cos\theta} + 1 - \tan\theta - (1-\alpha)\beta, & \theta \le \arctan(1-\beta) \\ \dfrac{1}{\cos\theta} + 1 - \left(\dfrac{1+\alpha}{2}\right)\tan\theta - \left(\dfrac{1-\alpha}{2}\right)(\beta+1), & \theta \ge \arctan(1-\beta) \end{cases}$$

since the expression in the second case is a simplification of

$$\frac{1}{\cos\theta} + 1 - \tan\theta - (1-\alpha)\left(1 - \tan\theta + \frac{\beta - (1-\tan\theta)}{2}\right)$$

and it is easy to verify that both cases meet when $\beta = 1 - \tan\theta$.

The derivative of the first case with respect to θ is $(\sin\theta - 1)/\cos^2\theta$, which is always non-positive, implying that $c(\theta)$ is non-increasing for $\theta \le \arctan(1-\beta)$. The derivative of the second case is $(\sin\theta - (1+\alpha)/2)/\cos^2\theta$, so this second case is non-increasing for $\theta \le \arcsin((1+\alpha)/2)$ and non-decreasing for $\theta \ge \arcsin((1+\alpha)/2)$. Therefore, since $\arctan(1-\beta) < \pi/4$, there are three mutually exclusive possibilities:

- $\arcsin((1+\alpha)/2) \ge \pi/4$, wherein both cases in the definition of $c(\theta)$ are non-increasing and therefore $\pi/4$ is optimal;
- $\arcsin((1+\alpha)/2) \le \arctan(1-\beta)$, wherein the first case is non-increasing and the second one non-decreasing, which makes $\arctan(1-\beta)$ optimal (note that $\pi/6 \le \arcsin((1+\alpha)/2) \le \arctan(1-\beta) < \pi/4$); and
- $\arctan(1-\beta) < \arcsin((1+\alpha)/2) < \pi/4$, wherein the first case and the second case up to $\arcsin((1+\alpha)/2)$ are non-increasing and the second case after $\arcsin((1+\alpha)/2)$ is non-decreasing, making $\arcsin((1+\alpha)/2)$ optimal.

These three possibilities correspond to the ones in Figure 4.1. Indeed, since $0 \le \alpha < 1$, $\arcsin((1+\alpha)/2) \ge \pi/4$ is equivalent to $\alpha \ge \sqrt{2} - 1$. Furthermore, $\arcsin((1+\alpha)/2) \le \arctan(1-\beta)$ is equivalent to $\tan\arcsin((1+\alpha)/2) \le 1 - \beta$, which is equivalent to

$$\frac{(1+\alpha)/2}{\sqrt{1 - ((1+\alpha)/2)^2}} \le 1 - \beta \quad \Longleftrightarrow \quad \beta \le 1 - \frac{1+\alpha}{\sqrt{4 - (1+\alpha)^2}} = 1 - 2h.$$

Likewise, $\arctan(1-\beta) < \arcsin((1+\alpha)/2) < \pi/4$ is equivalent to $\alpha < \sqrt{2} - 1$ and $\beta > 1 - 2h$.

So far, for every $\alpha \in [0; 1)$ and every $\beta \in (0; \sqrt{2}]$, we have found a crossing $(\gamma_0^*, \gamma_1^*) \in \Gamma^2$ that minimizes $c(\gamma_0^*, \gamma_1^*)$, which in turn is the minimum over all line-segment road networks R of $\max\{c_R(\gamma_0^*), c_R(\gamma_1^*)\}$. This is quite an achievement as not only we have shown how to obtain a particular line-segment road network R^* of minimum $\max\{c_{R^*}(\gamma_0^*), c_{R^*}(\gamma_1^*)\}$, but also this road network can be made the same as the ones depicted in Figure 4.1.

Furthermore, for any line-segment road network R, we have $\max\{c_{R^*}(\gamma_0^*), c_{R^*}(\gamma_1^*)\} \le \max\{\delta_R(A, C), \delta_R(B, D)\}$ as otherwise fastest paths with respect to R from A to C and from B to D would contradict the optimality of (γ_0^*, γ_1^*) and R^*. Therefore, if we define the *crossing time* of a line-segment road network R as $\max\{\delta_R(A, C), \delta_R(B, D)\}$, then R^* is a line-segment road network of minimum crossing time since $\delta_{R^*}(A, C) \le c_{R^*}(\gamma_0^*)$ and $\delta_{R^*}(B, D) \le c_{R^*}(\gamma_1^*)$.

Recall that our original problem is to find a line-segment road network of minimum maximum travel time. Recall also that the maximum travel time of a line-segment road network R is the maximum value of $\delta_R(p, q)$ for points p in AB and q in DC. It is thus clear that the crossing time of any line-segment road network is a lower bound on its maximum travel time. Therefore, if we are able to show that the crossing time of R^* equals

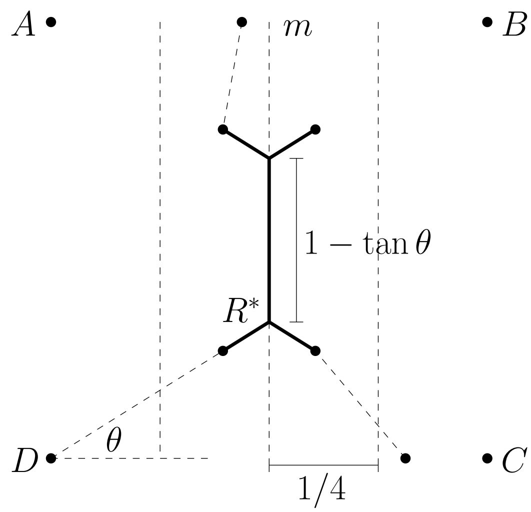


Figure 4.8: An example of a path from a point in AB to a point in DC that is at least as fast as γ_0^* and γ_1^*.

its maximum travel time, we would have shown R^* to be of minimum maximum travel time as well, establishing the advertised optimality of the road networks in Figure 4.1.

It is enough to show that the points where γ_0^* and γ_1^* enter and leave the road network R^* are at an orthogonal distance of at most $1/4$ from m (Figure 4.8). Indeed, then every point in AB can reach one of these entry points at least as fast as γ_0^* and γ_1^* and every point in DC can be reached from one of these exit points at least as fast as D and C can. Fastest paths between points in AB and DC can thus not take longer than γ_0^* and γ_1^*.

To conclude the proof, observe that R^* is as in Figure 4.8 for a particular value of $\theta \in [\pi/6; \pi/4]$. Each leg of the road network has length $(\beta - (1 - \tan\theta))/4$, so the orthogonal distance from the entry and exit points to m is

$$\frac{\beta - (1 - \tan\theta)}{4} \cos\theta = \frac{(\beta - 1)\cos\theta + \sin\theta}{4} \leq \frac{(\sqrt{2} - 1)\cos\theta + \sin\theta}{4}.$$

As $\theta \leq \pi/4$, we have $\cos\theta \geq \sqrt{2}/2$ and $\sin\theta \leq \sqrt{2}/2$, so the derivative of this expression with respect to θ is

$$-(\sqrt{2} - 1)\sin\theta + \cos\theta \geq -(\sqrt{2} - 1)\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} = \sqrt{2} - 1 > 0$$

and therefore the maximum orthogonal distance is achieved by $\theta = \pi/4$ (and $\beta = \sqrt{2}$), which yields $1/4$, as desired.

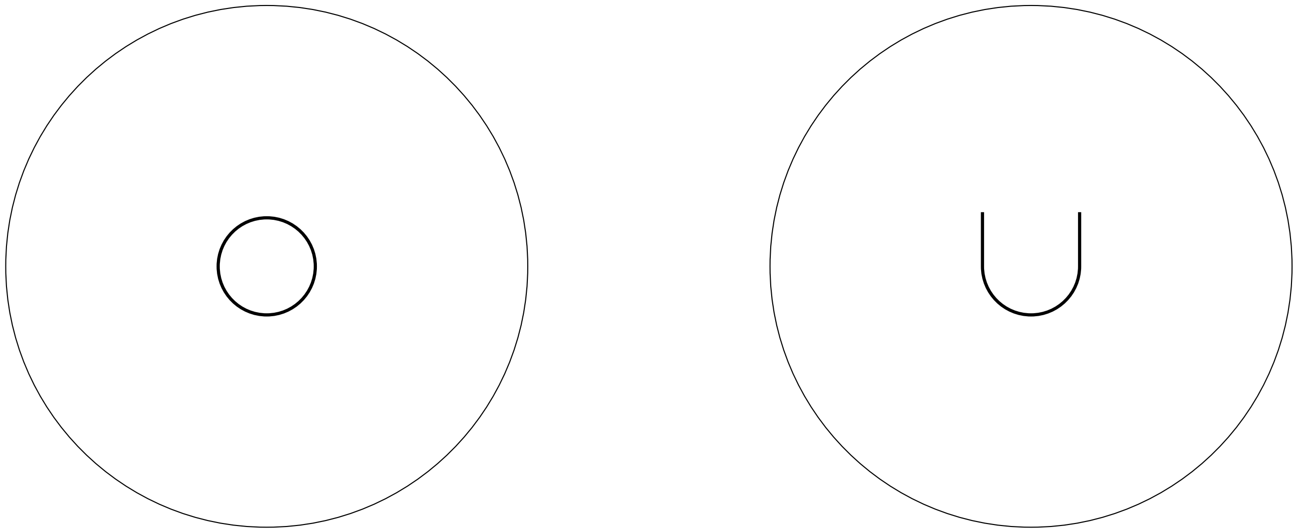
4.4 Road Placement in a Circle

In this section we briefly consider an open variant of this problem and show a counterexample to an intuitive conjecture that might be made regarding it. This time we are given a unit circle and would like to place roads in the plane in order to minimize the maximum travel time between any two points in the boundary of the circle. The total length of the roads should not exceed a budget of 1 and there is no cost to traveling on the roads.

With these simplifying assumptions, one might wonder whether a concentric circle (Figure 4.9a) is an optimal way to lay down these roads. Any point in the boundary of the unit circle requires $1 - 1/2\pi$ time to reach the road network, so antipodal points in the boundary of the circle require $2 - 1/\pi$ time to reach each other (as not using any road would take 2 units of time). However, since the road network is connected and free to traverse, every pair of points in the boundary of the unit circle can reach each other through the road network in time $2 - 1/\pi$, so this is the maximum travel time of this particular network.

Figure 4.9b shows a better approach. We let

$$r = \frac{4\sqrt{\pi^2 + 2\pi + 4} - 2\pi - 8}{2\pi^2} \approx 0.1861$$



(a) (b)

Figure 4.9: Two possible network layouts for the unit circle.

and construct a concentric semi-circle of radius r and two equally sized tangent segments using up the rest of the budget.

Figure 4.10 illustrates how we go about proving that the maximum travel time for this network is at most $1.63 < 1.68 < 2 - 1/\pi$. Since the total length of the network is 1, the distance from the origin to B is $(1 - \pi r)/2$, so $|BC| = (1 + \pi r)/2$. As $|AB| = r$, applying the Pythagorean theorem on the triangle ABC yields $|AC| = \sqrt{r^2 + (1 + \pi r)^2/4}$, which can easily be computationally verified to be $1 - r$, though a rather laborious manual verification is also feasible. It is evident that all points in the boundary of the circle and below line ℓ can reach the network in time at most $1 - r$. However, because $|AC| = 1 - r > |AD|$, all boundary points above ℓ and to the left of the y-axis can reach the network at point A in at most $1 - r$ time. Due to the construction being symmetric, the boundary points to the right of the y-axis and above ℓ can also reach the network in such time. Therefore, with all the boundary points being able to reach the network, which is connected, in time at most $1 - r$, the maximum travel time of this network is at most $2 - 2r < 1.63$.

4.5 Conclusions

We studied a road placement problem in the unit square and in the unit circle. We were able to design road network layouts that minimize the maximum travel time between opposing sides of the unit square for any road velocity and budgets up to $\sqrt{2}$. On the circle version, however, we were much less fortunate, being unable to find optimal layouts but disproving a natural conjecture.

Apart from the two obvious open problems of finding optimal layouts for the circle and for the unit square when $\beta > \sqrt{2}$, we pose the open problems of designing directed road networks. A directed road can be traversed at a faster speed, but only in the direction it was built. Building a bidirectional road thus costs twice as much as a directed road of same geometry. We conjecture that the concentric layouts in Figure 4.11 minimize the maximum travel times between points in the boundary of a unit square and of a unit disk, respectively.

Acknowledgement

Some of this work was carried out at the Sixth and Seventh Annual Workshops on Geometry and Graphs, held at the Bellairs Research Institute in Barbados, 2018 and 2019. The authors are grateful to the organizers and to the other participants of this workshop for providing a stimulating working environment.

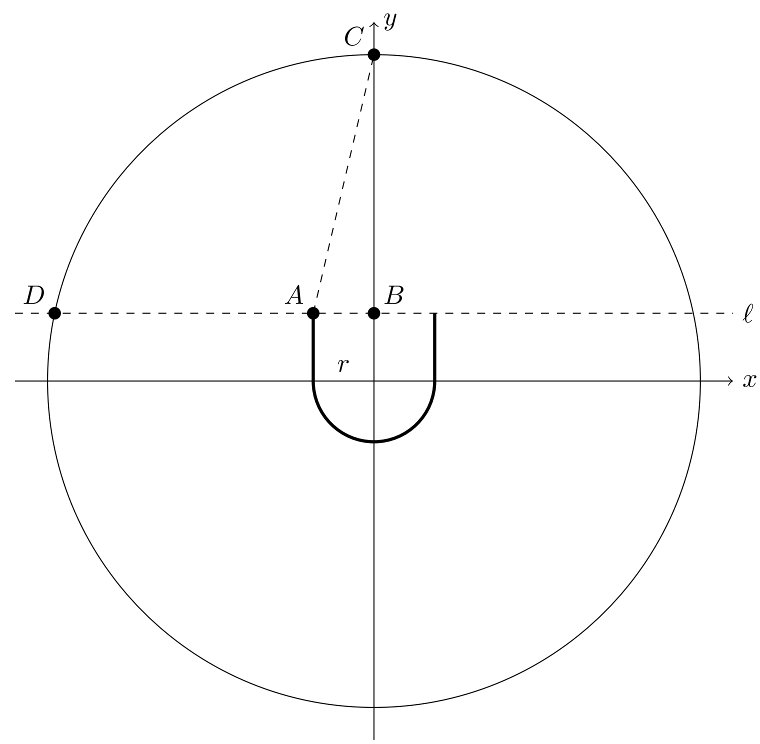


Figure 4.10

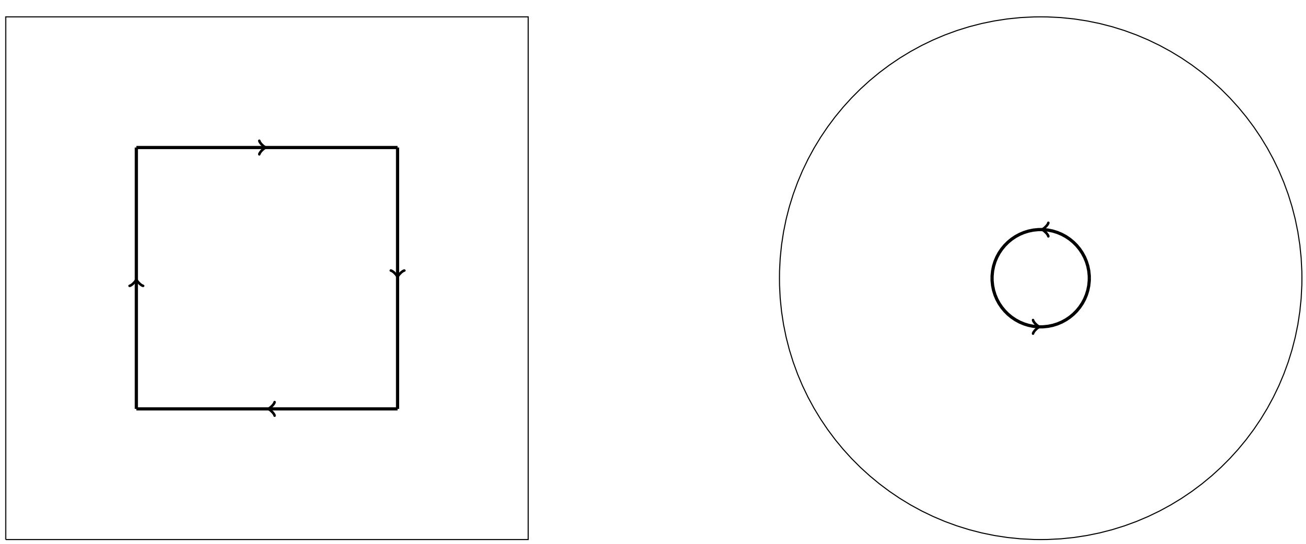


Figure 4.11

Chapter 5

Constant Delay Lattice Train Schedules

Jean-Lou De Carufel[1] Darryl Hill[2]
Anil Maheshwari[3] Sasanka Roy[4]
Luís Fernando Schultz Xavier da Silveira[5]

Abstract

Ajaykumar et al. (Problems on one way road networks, Proceedings of the CCCG 2016, pages 303–308) considered the following vehicle scheduling problem: given continuous curves $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}^2$, find non-negative delays $t_1, \ldots, t_n$ minimizing $\max\{t_1, \ldots, t_n\}$ while such that, for every distinct i and j and every time t, $|f_j(t - t_j) - f_i(t - t_i)| > \ell$, where ℓ is a given safety distance.

We study a simpler variant of this problem where we consider trains of fixed length ℓ that move at constant speed and sets of train lines, each of which consisting of an axis-aligned line-segment track with endpoints in the integer lattice $\mathbb{Z}^d$ and of a direction of movement (towards ∞ or $-\infty$). We are interested in upper bounds on the maximum delay we need to introduce on any line to avoid collisions, but more specifically on universal upper bounds that apply no matter the set of train lines.

We show small universal constant upper bounds for $d = 2$ and any given ℓ and also for $d = 3$ and $\ell = 1$. Through clique searching, we were also able to show that several of these upper bounds are tight.

5.1 Introduction

Ajaykumar et al. [2] considered the following problem: given a set $S = \{f_1, \ldots, f_n\}$ of curves, find a maximum subset $S' \subseteq S$ such that, for any two distinct $f_i, f_j \in S'$, there is no time $0 \le t \le 1$ such that $f_i(t) = f_j(t)$. Though there are some approximation algorithms, it is shown that most versions of these problems are NP-hard. This is true even when the curves are same-sized L-shapes and trajectories have the same constant speed. During a talk by Sasanka Roy on this problem at Stony Brook University, Joseph S. B. Mitchell and Esther M. Arkin posed the following variation: given $S = \{f_1, \ldots, f_n\}$ where $f_i : \mathbb{R} \to \mathbb{R}^2$ is a continuous function, we wish to assign delays

[1] School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada
[2] School of Computer Science, Carleton University, Ottawa, Canada
[3] School of Computer Science, Carleton University, Ottawa, Canada
[4] Indian Statistical Institute, Kolkata, India
[5] School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada

$t_i \geq 0$ so that, for any two distinct $f_i, f_j \in S$, there is no time $t \in \mathbb{R}$ such that $|f_j(t-t_j) - f_i(t-t_i)| \leq \ell$; we also wish that the maximum delay is minimized.

Automated guided vehicles (AGVs) and automated vehicle routing are currently some of the most prolific fields in the domain of motor vehicle technology, enjoying a large body of research addressing a large number of related algorithmic and optimization problems. The field of automated guided vehicles has a long and rich history and is closely related to our work. For an elaborate review, the reader is referred to a survey by Vis [56], though [28] is a more recent one. A central topic in this field is collision avoidance. Kim and Tanchoco [31] propose an algorithm based on Dijkstra's that facilitate conflict-free routing of AGVs. It runs in $\mathcal{O}(v^4n^2)$, where v is the number of vehicles and n is the number of nodes, i.e., path intersections. Arora et al. [53] used techniques based on game theory to design a methodology for AGV traffic control. Yan et al. [57] studied collision-free routing of AGVs on both unidirectional and bidirectional paths using digraphs. For more information on conflict-free routing of AGVs, the reader is referred to [32, 38, 34]. For an elaborate survey on routing and scheduling algorithms for AGVs, see [36]. Other related work in this area includes the study of One Way Road Networks (OWRNs) by Ajaykumar et al. [29], its motivation [45] and [39, 20, 48].

We study a simple vehicle scheduling problem in which many times we are guaranteed schedules with at most a constant delay.

5.2 Preliminaries

Definition 5.1. A (d-dimensional) *train line* consists of a *track*, which is a line segment in $\mathbb{R}^d$ with distinct endpoints distinguished between a *departure point* and an *arrival point*, of a *car length* and of a *velocity*, which are positive real numbers. A set of train lines with non-overlapping (but possibly crossing) tracks is called a *train network*.

Definition 5.2. For a set of real numbers X and a real number y, we denote $\{x+y : x \in X\}$ by $X+y$.

Definition 5.3. A (collision-free) *schedule* for a train network is an assignment of a *departure time*, which is a non-negative real number, to each of its lines with the property that every pair of cars (on lines whose tracks cross) do not *collide*, i.e., the open intervals $(0;\ell/v)+(t+\delta/v)$ and $(0;\ell'/v')+(t'+\delta'/v')$ do not intersect, where t and t' are the respective departure times of the lines, δ and δ' are the respective distances between the departure points of the lines and the crossing point, ℓ and ℓ' are the respective car lengths of the lines and v and v' their respective velocities. The (departure) *makespan* of the schedule is the maximum departure time it assigns. An *integer* schedule is one that only assigns integer departure times.

An alternative way to understand conflict-free schedules is to imagine each line's car starts "underground" with the front at the departure point and moves towards the arrival point at the line's velocity, where it moves "underground" again (Figure 5.1). A departure time assignment is then a conflict-free schedule if, and only if, the line segments that correspond to the "above-ground" parts of these cars never cross. For an example, see Figure 5.2.

It is straightforward that every train network admits a schedule: simply wait for the previous train to finish its course before sending the next one.

Definition 5.4. A train network is *regular* if all its lines' velocities are 1, all its car lengths are the same and integer, all its tracks are axis-aligned and all its departure and arrival points are in $\mathbb{Z}^d$.

Theorem 5.5. *All regular train networks admit an integer schedule of minimum makespan (among all schedules, integer or otherwise).*

Proof. It is enough to show that, for an arbitrary schedule for a regular train network, assigning to each line a departure time equal to the floor of the original time will not result in any collisions. Indeed, take two lines whose tracks cross and define t, t', δ, δ', v, v', ℓ and ℓ' as in definition 5.3. We thus have $\delta, \delta' \in \mathbb{Z}$, $v = v' = 1$ and $\ell = \ell' \in \mathbb{N} \setminus \{0\}$.

The schedule being collision-free ensures that $((0;\ell)+(t+\delta)) \cap ((0;\ell)+(t'+\delta')) = \{\}$, which is equivalent to $(0;\ell) \cap ((0;\ell)+(t'-t+\delta'-\delta)) = \{\}$. Because $t'-t$ and $\lfloor t' \rfloor - \lfloor t \rfloor$ differ by less than 1, $\lfloor t' \rfloor - \lfloor t \rfloor$ is either $\lfloor t'-t \rfloor$ or $\lceil t'-t \rceil$. Therefore, since ℓ and $\delta'-\delta$ are integers, the open interval $(0;\ell)+(\lfloor t' \rfloor - \lfloor t \rfloor + \delta' - \delta)$ can not intersect $(0;\ell)$, which implies our assignment is collision-free. $\square$

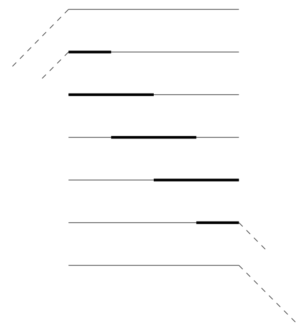
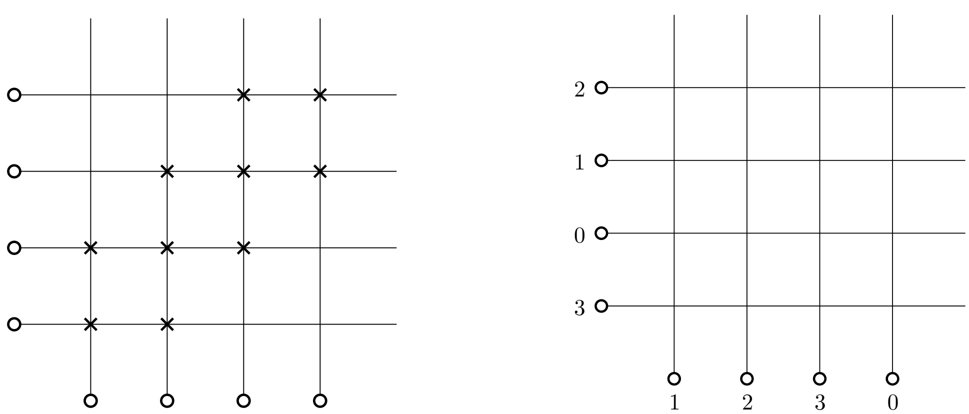


Figure 5.1: An illustration of the journey of a train.



(a) (b)

Figure 5.2: If unit velocity trains of length 2 depart simultaneously from the circles in (a), there will be collisions at the marked intersections. However, if delays are introduced as in (b), no collisions will occur.

Definition 5.6. Consider a train line with an axis-aligned track that departs from $p \in \mathbb{R}^d$ and arrives at $p + \alpha e_i$, where e_i is the i-th vector of the canonical basis of $\mathbb{R}^d$ ($0 \leq i < d$) and $\alpha \in \mathbb{R} \setminus \{0\}$. The *axis* of the line is the number i. Furthermore, we say this line is *positive* if $\alpha > 0$ and *negative* if $\alpha < 0$, with its *sign* being 1 or -1, respectively.

Definition 5.7. For any modulus $k \in \mathbb{N} \setminus \{0\}$, we extend the modulo k function to real numbers as follows: if $x \in \mathbb{R}$, then $x \bmod k = (\lfloor x \rfloor \bmod k) + (x - \lfloor x \rfloor)$. Furthermore, for a set $X \subseteq \mathbb{R}$, we denote $\{x \bmod k : x \in X\}$ by $X \bmod k$.

5.3 Results

Theorem 5.8. *Any regular (d-dimensional) train network with cars of length ℓ and only positive lines has a schedule with makespan at most $d\ell - 1$.*

Proof. Assign departure time $\left(\ell a + \sum_{i=0}^{d-1} p_i\right) \bmod (d\ell)$ to lines departing from $p = (p_0, \ldots, p_{d-1}) \in \mathbb{R}^d$ with axis $a \in \{0, \ldots, d-1\}$. Consider then a crossing between a line departing from $p = (p_0, \ldots, p_{d-1}) \in \mathbb{R}^d$ with axis a and another line departing from $p' = (p'_0, \ldots, p'_{d-1}) \in \mathbb{R}^d$ with axis $a' \neq a$. In the notation of definition 5.3, we have $t = \left(\ell a + \sum_{i=0}^{d-1} p_i\right) \bmod (d\ell)$, $t' = \left(\ell a' + \sum_{i=0}^{d-1} p'_i\right) \bmod (d\ell)$, $\delta = p'_a - p_a$, $\delta' = p_{a'} - p'_{a'}$ and $v = v' = 1$. Thus, modulo $d\ell$,

$$\begin{aligned} t' - t + \delta' - \delta &\equiv \ell(a' - a) + \sum_{i=0}^{d-1} (p'_i - p_i) - (p'_{a'} - p_{a'}) - (p'_a - p_a) \\ &\equiv \ell(a' - a) + \sum_{\substack{i = 0 \\ i \notin \{a, a'\}}}^{d-1} (p'_i - p_i) \\ &\equiv \ell(a' - a) \end{aligned}$$

since, for axes $i \notin \{a, a'\}$, we have $p_i = p'_i$ as the lines cross. However, $((0;\ell) + (t' - t + \delta' - \delta)) \bmod (d\ell)$ can not intersect $(0;\ell) \bmod (d\ell) = (0;\ell)$ because $a \neq a'$. Therefore, $(0;\ell) \cap ((0;\ell) + (t' - t + \delta' - \delta)) = \{\}$ and there is no collision. □

Theorem 5.9. *Any regular 2-dimensional train network with car length ℓ has a schedule with makespan at most $M - 1$, where*

$$M = \begin{cases} 2, & \ell = 1 \\ 8, & \ell = 2 \\ 6\ell, & \ell \geq 3. \end{cases}$$

Proof. Assign departure time

$$\sigma\Big(x + y + (1 - a)\big(-2(y \bmod \ell) - \ell + 1\big) + a\big(-2(x \bmod \ell) + 2\ell - 1\big)\Big) \bmod M$$

to lines departing from (x, y) with axis a and sign σ. Consider then a horizontal line departing from (x, y) with sign σ that crosses a vertical line departing from (x', y') with sign σ'. In the notation of definition 5.3, we have $t = \sigma(x + y - 2(y \bmod \ell) - \ell + 1)$, $t' = \sigma'(x' + y' - 2(x' \bmod \ell) + 2\ell - 1)$, $\delta = \sigma(x' - x)$, $\delta' = -\sigma'(y' - y)$ and $v = v' = 1$.

Note that $M \geq 2\ell$ and so it is enough to show that $((t' - t) + (\delta' - \delta)) \bmod M \in \{\ell, \ldots, M - \ell\}$, as then $(0;\ell) \bmod M = (0;\ell)$ does not intersect $((0;\ell) + (t' - t + \delta' - \delta)) \bmod M = (0;\ell) + ((t' - t + \delta' - \delta) \bmod M)$, which implies $(0;\ell) \cap ((0;\ell) + (t' - t + \delta' - \delta)) = \{\}$, i.e., a lack of collisions. We prove this by cases starting when $\sigma = \sigma' = 1$, which gives us, modulo M,

$$\begin{aligned} t' - t + \delta' - \delta &\equiv \begin{array}{c} x' + y' - 2(x' \bmod \ell) + 2\ell - 1 - y' + y \\ -x - y + 2(y \bmod \ell) + \ell - 1 - x' + x \end{array} \\ &\equiv 2(y \bmod \ell) - 2(x' \bmod \ell) + 3\ell - 2. \end{aligned}$$

Because $(y \bmod \ell), (x' \bmod \ell) \in \{0, \ldots, \ell - 1\}$, we have that $2(y \bmod \ell) - 2(x' \bmod \ell) + 3\ell - 2 \in \{\ell, \ldots, 5\ell - 4\} \subseteq \{\ell, \ldots, M - \ell\}$.

When $\sigma = -1$ and $\sigma' = 1$, we have, modulo M,

$$\begin{aligned} t' - t + \delta' - \delta &\equiv \begin{array}{c} x' + y' - 2(x' \bmod \ell) + 2\ell - 1 - y' + y \\ +x + y - 2(y \bmod \ell) - \ell + 1 + x' - x \end{array} \\ &\equiv 2(x' - (x' \bmod \ell) + y - (y \bmod \ell)) + \ell. \end{aligned}$$

Because $x' - (x' \bmod \ell)$ and $y - (y \bmod \ell)$ are multiples of ℓ, $2(x' - (x' \bmod \ell) + y - (y \bmod \ell)) + \ell$ is an odd multiple of ℓ. However, since M is an even multiple of ℓ, we do have $(t' - t + \delta' - \delta) \bmod M \in \{\ell, \ldots, M - \ell\}$.

To complete the remaining two cases, note that if σ and σ' were simultaneously multiplied by -1, so would be t, t', δ and δ', also multiplying $t' - t + \delta' - \delta$ by -1. However, $\{\ell, \ldots, M - \ell\}$ multiplied by -1 is $\{\ell, \ldots, M - \ell\}$ modulo M. $\square$

Theorem 5.10. *Any regular 3-dimensional train network with car length 1 has a schedule with makespan at most 5.*

Proof. A line gets assigned as departure time the only integer in $\{0, \ldots, 5\}$ that is equivalent to $\sigma(p_0 + p_1 + p_2 + a)$ modulo 3 and equivalent to $p_0 + p_1 + p_2 + (\sigma + 1)/2$ modulo 2, where $p = (p_0, p_1, p_2)$ is its departure point, $a \in \{0, 1, 2\}$ is its axis and $\sigma \in \{-1, 1\}$ is its sign (the existence and uniqueness of this integer is assured by the Chinese Remainder Theorem). Consider then two crossing lines departing respectively from $p = (p_0, p_1, p_2) \in \mathbb{R}^3$ and $p' = (p'_0, p'_1, p'_2) \in \mathbb{R}^3$, with respective signs $\sigma, \sigma' \in \{-1, 1\}$ and with respective axes $a \in \{0, 1, 2\}$ and $a + 1 \in \{0, 1, 2\}$ (axis arithmetic is done modulo 3). Thus, in the notation of definition 5.3, we have $\delta = \sigma(p'_a - p_a)$, $\delta' = -\sigma'(p'_{a+1} - p_{a+1})$ and $v = v' = \ell = \ell' = 1$.

If $\sigma \neq \sigma'$, then, modulo 2, $t \equiv p_0 + p_1 + p_2 + (\sigma + 1)/2$ and $t' \equiv p'_0 + p'_1 + p'_2 + (\sigma' + 1)/2$, so, also modulo 2,

$$\begin{aligned} (t' - t) + (\delta' - \delta) &\equiv \sum_{i=0}^{2} (p'_i - p_i) + (\sigma' + 1)/2 - (\sigma + 1)/2 \\ &\quad -\sigma'(p'_{a+1} - p_{a+1}) - \sigma(p'_a - p_a) \\ &\equiv \sum_{i=0}^{2} (p'_i - p_i) - (p'_{a+1} - p_{a+1}) - (p'_a - p_a) + (\sigma' - \sigma)/2 \\ &\equiv (p'_{a+2} - p_{a+2}) + (\sigma' - \sigma)/2. \end{aligned}$$

However, since the lines cross, $p_{a+2} = p'_{a+2}$ and, since $\sigma, \sigma' \in \{-1, 1\}$ and $\sigma \neq \sigma'$, $(\sigma' - \sigma)/2 \equiv 1$, so $(t' - t) + (\delta' - \delta) \equiv 1$ modulo 2. The collision avoidance condition is $(0; 1) \cap ((0; 1) + (t' - t + \delta' - \delta)) = \{\}$, which is true since $(0; 1) \bmod 2 = (0; 1)$ but $((0; 1) + (t' - t + \delta' - \delta)) \bmod 2 = (1; 2)$.

On the other hand, if $\sigma = \sigma'$, then, modulo 3, $t \equiv \sigma(p_0 + p_1 + p_2 + a)$ and $t' \equiv \sigma(p'_0 + p'_1 + p'_2 + a + 1)$, so modulo 3,

$$\begin{aligned} (t' - t) + (\delta' - \delta) &\equiv \sigma\left(\sum_{i=0}^{2} (p'_i - p_i) + a + 1 - a\right) \\ &\quad -\sigma(p'_a - p_a) - \sigma(p'_{a+1} - p_{a+1}) \\ &\equiv \sigma((p'_{a+2} - p_{a+2}) + 1) \\ &\equiv \sigma \end{aligned}$$

as, again, $p_{a+2} = p'_{a+2}$. As before, $(0; 1) \bmod 3 = (0; 1)$ but $((0; 1) + (t' - t + \delta' - \delta)) \bmod 3 = (0; 1) + (\sigma \bmod 3)$, which is either $(1; 2)$ or $(2; 3)$. No collisions are therefore possible. $\square$

5.4 Clique Searching and Lower Bounds

Whether a train network admits an integer schedule with makespan at most $s \in \mathbb{N}$ can be decided using clique search: create a graph G containing a vertex $v_{L,t}$ for each line L and time $t \in \{0, \ldots, s\}$, put edges $v_{L,t} v_{L,t'}$

d	ℓ	σ	Makespan	Tight?	Figure
2	1 to 10	$+$	$2\ell - 1$	Yes, Theorem 5.8	5.3a
2	1	$\pm$	1	Yes, Theorem 5.9	5.3a
2	2	$\pm$	7	Yes, Theorem 5.9	5.3b
3	1	$+$	2	Yes, Theorem 5.8	5.3c

Table 5.1: A report on some useful regular train networks containing: its dimension d, its car length ℓ, whether its lines are all positive or of unrestricted sign, its minimum makespan, our knowledge on whether this makespan matches an upper bound for regular networks of the same type and a reference to a figure describing it.

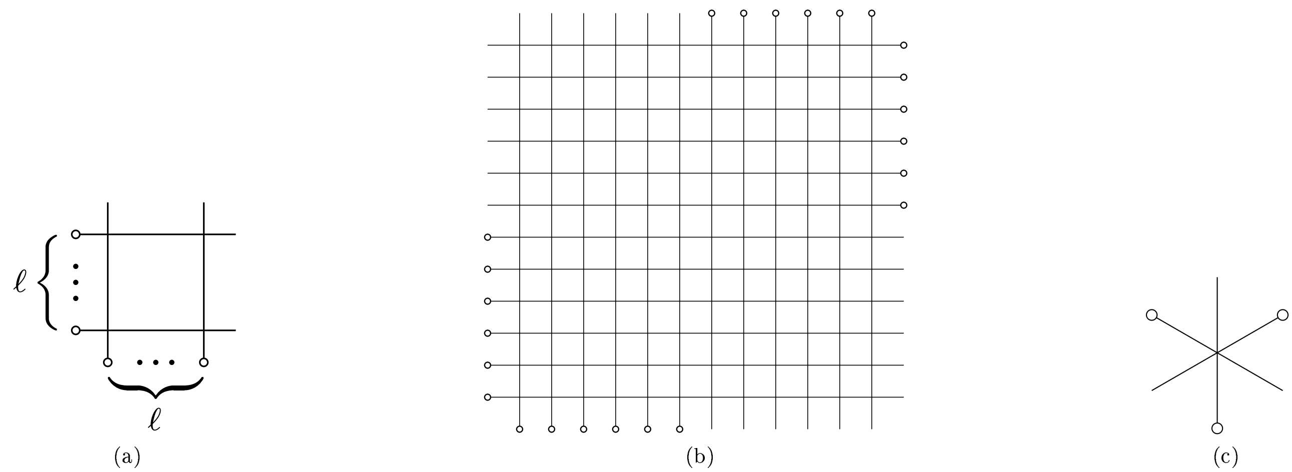


Figure 5.3: Regular train networks ($\circ$ denotes the departure point of a line).

in G for every line L and times $t, t' \in \{0, \dots, s\}$ with $t \neq t'$ and then put edges $v_{L,t}v_{L',t'}$ in G for every two crossing lines L and L' and every integers $t, t' \in \{0, \dots, s\}$ satisfying the collision condition $((0; \ell/v) + (t + \delta/v)) \cap ((0; \ell'/v') + (t' + \delta'/v')) \neq \{\}$, where δ and δ' are the respective distances from the lines' departure points to the crossing point, ℓ and ℓ' are their respective car lengths and v and v' are their respective cars' velocities. An integer schedule with makespan s exists if, and only if, the complement of G has a clique with as many vertices as there are lines, since the edges we put prevent two vertices associated with the same line from being in a clique and we encoded the collision conditions in edges between vertices from two crossing lines in such a way that such a clique corresponds to a collision-free departure time assignment.

With Theorem 5.5, we can decide whether a regular train network has a schedule with makespan at most a given number $s \in \mathbb{R}$. We have implemented this algorithm with the `cliquer` clique solver [42] and recorded some lower bounds for regular train networks in Table 5.1.

5.5 Conclusions

We were able to show that several types of regular train networks admit a constant upper bound on its minimum schedule makespan (Theorems 5.8, 5.9 and 5.10). We were also able to prove many of these upper bounds are tight through computational experiments.

The main problem that remains open is whether, for values of $\ell \geq 2$, every three-dimensional regular train network with cars of length ℓ admits a schedule with makespan bounded by a constant. However, we are also interested in extending the lower bounds in Table 5.1, in particular: finding a three dimensional regular train network with unit length cars that does not admit a schedule with makespan less than 5; and determining whether, for every ℓ, there exists a regular two-dimensional train network with only positive lines and cars of length ℓ that can not be scheduled with makespan $2\ell - 2$.

Acknowledgements

We would like to thank Joseph S. B. Mitchell, Esther M. Arkin, Sándor Fekete and members of the Computational Geometry group at Carleton University.

Chapter 6

Conclusion

We close this thesis with a summary of our results and directions for future research. In Chapter 2, we have answered, up to a logarithmic factor, 248 out of 256 questions regarding the asymptotics of $\mathrm{ex}(n, X)$ and tied the answer of the remaining 8 to that of the tripod packing problem (Table 2.1). Progress in the tripod packing problem and the elimination of these logarithmic gaps, preferably through a common framework, are the main objects of future research regarding this chapter.

In Chapter 3, we provided a PTAS for the version of the line segment cover problem (LSC) in which segments can only be chosen from one orientation (Theorem 4) and a FPT algorithm for the general LSC problem (Theorem 11 and Section 3.5.1). We have also shown that the variation of the LSC problem in which the goal is to cover all rectangular faces is APX-hard (Theorem 6). The main question that remains is whether these two variants of the LSC problem admit polynomial time constant-factor approximation algorithms.

In Chapter 4, we constructed provably optimal road network layouts to minimize the maximum travel time between opposing sides of a unit square for budgets up to $\sqrt{2}$ (Section 4.3) and disproved the natural conjecture that a concentric circle is an optimal layout for minimizing the maximum travel time between points in the boundary of a circle (Section 4.4). Optimal road networks for the circle and for unit square when $\beta > \sqrt{2}$, as well as optimal directed road networks for the square and circle (Section 4.5), are the next goals for research on this topic.

Finally, in Chapter 5, we found universal constant upper bounds on the maximum delay required to schedule regular train networks in two dimensions and, for unit length trains, in three dimensions (Theorems 5.8, 5.9 and 5.10). We also proved computationally that several of these upper bounds are tight (Section 5.4). There is plenty of room for future improvement on this work. We would mainly like to determine whether universal constant upper bounds on optimal schedule makespan exist in three dimensions for $\ell \geq 2$ and find (or refute the existence of) a three dimensional regular train network with unit length cars that does not admit a schedule with makespan below 5. We would also like to improve and expand Table 5.1, in particular determining whether, for every ℓ, there is a regular two-dimensional train network with only positive lines and cars of length ℓ that can not be scheduled with makespan $2\ell - 2$.

Bibliography

[1] Ankush Acharyya, Subhas C. Nandy, Supantha Pandit, and Sasanka Roy. Covering segments with unit squares. In *proceedings of the 15th International Symposium on Algorithms and Data Structures (WADS 2017), St. John's, NL, Canada*, pages 1–12, 2017.

[2] Jammigumpula Ajay and Sasanka Roy. Collision-free routing problem with restricted L-path. *Combinatorial Algorithms – 29th International Workshop, IWOCA 2018, Singapore*, pages 1–13, 2018.

[3] Paola Alimonti and Viggo Kann. Hardness of approximating problems on cubic graphs. In *Algorithms and Complexity*, pages 288–298, 1997.

[4] Boris Aronov, Vida Dujmović, Pat Morin, Aurélien Ooms, and Luís Fernando Schultz Xavier da Silveira. More Turán-type theorems for triangles in convex point sets. *The Electronic Journal of Combinatorics*, 26(1), 2019.

[5] Rom Aschner, Matthew J. Katz, Gila Morgenstern, and Yelena Yuditsky. Approximation schemes for covering and packing. In *proceedings of the 7th International Workshop on Algorithms and Computation (WALCOM 2013), Kharagpur, India*, pages 89–100, 2013.

[6] Pradeesha Ashok, Sudeshna Kolay, Neeldhara Misra, and Saket Saurabh. Unique covering problems with geometric sets. In *proceedings of the 21st International Computing and Combinatorics Conference (COCOON 2015), Beijing, China*, pages 548–558, 2015.

[7] Sayan Bandyapadhyay, Anil Maheshwari, Saeed Mehrabi, and Subhash Suri. Approximating dominating set on intersection graphs of rectangles and L-frames. In *proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018), Liverpool, UK*, pages 37:1–37:15, 2018.

[8] Felix Adalbert Behrend. On sets of integers which contain no three terms in arithmetic progression. *Proceedings of the National Academy of Sciences USA*, 32:331–332, 1946.

[9] Prosenjit Bose, Jean Cardinal, Sébastien Collette, Ferran Hurtado, Matias Korman, Stefan Langerman, and Perouz Taslakian. Coloring and guarding arrangements. *Discrete Mathematics & Theoretical Computer Science*, 15(3):139–154, 2013.

[10] Peter Braß. Turán-type extremal problems for convex geometric hypergraphs. *Contemporary Mathematics*, 342:25–34, 2004.

[11] Peter Braß, Günter Rote, and Konrad J. Swanepoel. Triangles of extremal area or perimeter in a finite planar point set. *Discrete & Computational Geometry*, 26(1):51–58, 2001.

[12] Valentin E. Brimkov, Andrew Leach, Jimmy Wu, and Michael Mastroianni. Approximation algorithms for a geometric set cover problem. *Discrete Applied Mathematics*, 160(7-8):1039–1052, 2012.

[13] Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

[14] Paz Carmi, Anil Maheshwari, Saeed Mehrabi, and Luís Fernando Schultz Xavier da Silveira. Approximability of covering cells with line segments. In Donghyun Kim, R. N. Uma, and Alexander Zelikovsky, editors, *Combinatorial Optimization and Applications - 12th International Conference, COCOA 2018, Atlanta, GA, USA, December 15-17, 2018, Proceedings*, volume 11346 of *Lecture Notes in Computer Science*, pages 436–448. Springer, 2018.

[15] Timothy M. Chan, Elyot Grant, Jochen Könemann, and Malcolm Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), Kyoto, Japan*, pages 1576–1585, 2012.

[16] Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

[17] Timothy M. Chan and Nan Hu. Geometric red-blue set cover for unit squares and related problems. *Comput. Geom.*, 48(5):380–385, 2015.

[18] Kenneth L. Clarkson and Kasturi R. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete & Computational Geometry*, 37(1):43–58, 2007.

[19] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press and McGraw–Hill, second edition, 2001.

[20] Philip Dasler and David M. Mount. On the complexity of an unregulated traffic crossing. *Algorithms and Data Structures*, pages 224–235, 2015.

[21] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity.* Springer, 1999.

[22] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

[23] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of ACM*, 45(4):634–652, 1998.

[24] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, 174:561–579, 2011.

[25] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[26] William Timothy Gowers and Jason Long. The length of an s-increasing sequence of r-tuples. 2016.

[27] Elyot Grant and Timothy M. Chan. Exact algorithms and apx-hardness results for geometric set cover. In *proceedings of the 23rd Annual Canadian Conference on Computational Geometry (CCCG 2011), Toronto, Ontario, Canada*, 2011.

[28] Pawel Hyla and Janusz Szpytko. Automated guided vehicles: the survey. *Journal of KONES*, 24, 2017.

[29] Navaneeta Saikia Jammigumpula Ajaykumar, Avinandan Das and Arindam Karmakar. Problems on one way road networks. *Proceedings of the 28th Canadian Conference on Computational Geometry*, pages 303–308, 2016.

[30] Thomas Jansen. *Introduction to the theory of complexity and approximation algorithms*, pages 5–28. 1998.

[31] Chang W. Kim and Jose M. A. Tanchoco. Conflict-free shortest-time bidirectional AGV routeing. *The International Journal of Production Research*, 29(12):2377–2391, 1991.

[32] Garry A. Koff. Automatic guided vehicle systems: applications, controls and planning. *Material flow*, 4(1–2):3–16, 1987.

[33] Matias Korman, Sheung-Hung Poon, and Marcel Roeloffzen. Line segment covering of cells in arrangements. *Inf. Process. Lett.*, 129:25–30, 2018.

[34] Hsu-Pin Wang Laiguang Zeng and Song Jin. Conflict detection of automated guided vehicles: a petri net approach. *The International Journal of Production Research*, 29(5):866–879, 1991.

[35] Stefan Langerman and Pat Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005.

[36] Shell-Ying Huang Ling Qiu, Wen-Jing Hsu and Han Wang. Scheduling and routing algorithms for agvs: a survey. *The International Journal of Production Research*, 40(3):745–760, 2002.

[37] Po-Shen Loh. Directed paths: from Ramsey to Ruzsa and Szemerédi. 2016.

[38] Charles J. Malmborg. A model for the design of zone control automated guided vehicle systems. *The International Journal of Production Research*, 28(10):1741–1758, 1990.

[39] Junichi Takeno Masayoshi Kakikura and Masao Mukaidono. A tour optimization problem in a road network with one-way paths. *Electrical Engineering in Japan*, 98(4):141–147, 1978.

[40] Saeed Mehrabi. Approximating domination on intersection graphs of paths on a grid. In *proceedings of the 15th International Workshop on Approximation and Online Algorithms (WAOA 2017), Vienna, Austria*, pages 76–89, 2017.

[41] Nabil H. Mustafa and Saurabh Ray. Improved results on geometric hitting set problems. *Discrete & Computational Geometry*, 44(4):883–895, 2010.

[42] Sampo Niskanen and Patric R. J. Östergård. Cliquer user's guide, version 1.0. Technical Report T48, Communications Laboratory, Helsinki University of Technology, Espoo, Finland, 2003.

[43] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

[44] Rajiv Raman and Saurabh Ray. Planar support for non-piercing regions and applications. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 69:1–69:14, 2018.

[45] Herbert Ellis Robbins. A theorem on graphs, with an application to a problem of traffic control. *The American Mathematical Monthly*, 46(5):281–283, 1939.

[46] Aniket Basu Roy, Sathish Govindarajan, Rajiv Raman, and Saurabh Ray. Packing and covering with non-piercing regions. *Discrete & Computational Geometry*, 60(2):471–492, 2018.

[47] Imre Z. Ruzsa and Endre Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976)*, 18:939–945, 1978.

[48] Christian Scheffer. Train scheduling hardness and algorithms. *The 13th International Conference and Workshops on Algorithms and Computation (WALCOM 2020)*, 2020.

[49] Neil James Alexander Sloane. The online encyclopedia of integer sequences. Sequence number A070214.

[50] Sherman Stein. Combinatorial packing of R^n by certain error spheres. *IEEE Transactions on Information Theory*, 30(2):364–368, 1984.

[51] Sherman K. Stein. Packing tripods. *Mathematical Intelligencer*, 17(2):37–39, 1995.

[52] Sherman K. Stein and Sándor Szabó. *Algebra and Tiling: Homomorphisms in the Service of Geometry*, volume 25 of *The Carus Mathematical Monographs*. The Mathematical Association of America, 1994.

[53] A. K. Raina Sudha Arora and A. K. Mittal. Collision avoidance among AGVs at junctions. *Intelligent Vehicles Symposium*, pages 585–589, 2000.

[54] Alexander Tiskin. Packing tripods: narrowing the density gap. *Discrete Mathematics*, 307(16):1973–1981, 2007.

[55] Kasturi R. Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010), Cambridge, Massachusetts, USA*, pages 641–648, 2010.

[56] Iris F. A. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3):677–709, 2006.

[57] Canrong Zhang Xuejun Yan and Mingyao Qi. Multi-AGVs collision-avoidance and deadlock-control for item-to-human automated warehouse. *Industrial Engineering, Management Science and Application (ICIMSA)*, pages 1–5, 2017.

Appendix A

Source code

In Chapter 5, we saw how to reduce regular train network scheduling to the maximum clique problem. Here we share our implementation of this reduction as a Python 3 program. The input network should be provided through the standard input and the output may be piped into “`cliquer -u -`”.

```
#!/usr/bin/env python3

import sys
import itertools

# Standard input: one train line per input line, each in the form
#     <carlen> <axis><dir> <x> <y> <z>
# where:
#     * <carlen> is the line's car length, a positive integer;
#     * <axis> is the axis the track is parallel to ("x", "y", or "z");
#     * <dir> is the line's direction of movement ("-" or "+"); and
#     * <x>, <y> and <z> are the line's departure point.
#
# Command line argument: a non-negative integer s.
# Output: a graph in a Cliquer-compatible format that has a clique of size
#     equal to the number of train lines (or input lines) if, and only if, the
#     lines form a network that has a schedule with makespan at most s.

class TrainLine:
    # carlen: a positive integer, the line's car length
    # start: a 3d point, the line's departure point
    # axis: the number of the axis the track is parallel to (0, 1 or 2)
    # dir: the line's direction of movement (-1 or 1)
    def __init__(l, carlen, start, axis, dir):
        l.carlen = carlen
        l.start = start
        l.axis = axis
        l.dir = dir

    # l[i] is the departure point's i-th coordinate (i is 0, 1 or 2)
    def __getitem__(l, i):
        return l.start[i]

    # Whether two line's tracks overlap
    def overlaps(l0, l1):
```

```
        if l0.axis != l1.axis:
            return False
        a = l0.axis
        if l0[(a+1)%3] != l1[(a+1)%3] or l0[(a+2)%3] != l1[(a+2)%3]:
            return False
        return l0.dir == l1.dir or l0[a]*l0.dir < l1[a]*l0.dir

    # Input: two train lines with non-overlapping tracks.
    # Output: None if the line's tracks do not cross and a pair with the
    # respective distances from the line's departure points to the
    # crossing point otherwise.
    def distances_to_crossing(l0, l1):
        other_axis = 0
        while other_axis == l0.axis or other_axis == l1.axis:
            other_axis+= 1
        if l0[other_axis] != l1[other_axis]:
            return None
        if l0[l1.axis]*l1.dir <= l1[l1.axis]*l1.dir:
            return None
        if l1[l0.axis]*l0.dir <= l0[l0.axis]*l0.dir:
            return None
        return (
            (l1[l0.axis]-l0[l0.axis])*l0.dir,
            (l0[l1.axis]-l1[l1.axis])*l1.dir
        )

    # Reads a train line from a text line
    @staticmethod
    def read(text_line):
        tokens = text_line.split()
        carlen = int(tokens[0])
        axis = {"x": 0, "y": 1, "z": 2}[tokens[1][0]]
        dir = {"-": -1, "+": 1}[tokens[1][1]]
        start = tuple(map(int, tokens[2:5]))
        return TrainLine(carlen, start, axis, dir)

# Whether two open intervals intersect
def intervals_overlap(I0, I1):
    (a0, b0) = I0
    (a1, b1) = I1
    return a0 < b1 and a1 < b0

# Command line argument
s = int(sys.argv[1])

# Read standard input into a list of train lines
L = [TrainLine.read(input_line) for input_line in sys.stdin.readlines()]

# Check input tracks are non-overlapping
for l0, l1 in itertools.combinations(L, 2):
    if l0.overlaps(l1):
        raise Exception("Input contains overlapping tracks")

```

```python
# Assign vertices to lines
for i in range(len(L)):
    L[i].vertices = range((s+1)*i, (s+1)*(i+1))

# Print the graph
print("p graph %d 0" % ((s+1)*len(L)))
for l0, l1 in itertools.combinations(L, 2):
    ds = l0.distances_to_crossing(l1)
    if ds is not None:
        d0, d1 = ds
    for t0, t1 in itertools.product(range(s+1), repeat=2):
        if ds is None or not intervals_overlap(
            (t0 + d0, t0 + d0 + l0.carlen),
            (t1 + d1, t1 + d1 + l1.carlen)
        ):
            print("e %d %d" % (l0.vertices[t0]+1, l1.vertices[t1]+1))
```