Helly Theorems and Generalized Linear Programming

by

Annamaria Beatrice Amenta

B.A. (Yale University) 1979

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Raimund Seidel
Professor John Canny
Professor William Thurston

1993

Helly Theorems

and Generalized Linear Programming

by

Annamaria Beatrice Amenta

Abstract

Helly Theorems and Generalized Linear Programming

by

Annamaria Beatrice Amenta

Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Raimund Seidel, Chair

This thesis establishes a connection between the Helly theorems, a collection of results from combinatorial geometry, and the class of problems which we call Generalized Linear Programming, or GLP, which can be solved by combinatorial linear programming algorithms like the simplex method. We use these results to explore the class GLP and show new applications to geometric optimization, and also to prove Helly theorems.

In general, a GLP is a set of constraints and a function to be minimized, which obey certain combinatorial conditions. Linear programming is an example. A Helly theorem is also defined by its combinatorial structure. We observe that there is a Helly theorem about any GLP, which is that the minimum is no greater than m if and only if the minimum of every subproblem with $d+1$ constraints is no greater than m.

We use this observation to prove Helly theorems. Then we give a paradigm which usually allows us to construct a GLP corresponding to a given Helly theorem. Most of our algorithmic results are based on this paradigm. We show that in there are GLPs in which the constraints or objective function are not only non-linear but also non-convex or disconnected.

We give numerous applications, concentrating on expected $O(n)$ time algorithms. Some examples are that the largest axis-aligned box in the the intersection of a family of convex sets in fixed dimension, and the translation and scaling which minimizes the Hausdorff distance between two convex polygons in the plane, can be found by GLP. An example of a second family of results is a GLP to find the smallest factor by which a family

of boxes can be scaled around their centers so as to admit a hyperplane transversal, thus fitting a hyperplane to the family of centers.

The author was supported by an National Science Foundation Graduate Fellowship, an AT&T Graduate Research Program for Women Grant, and a U.C. President's Dissertation Year Fellowship. Some of this work was done while visiting AT&T Bell Labs at Murry Hill and the Freie Universiatät, Berlin.

Committee Chairman: Raimund Seidel

Acknowledgments

I have many people to thank for the wonderful, formative years I spent at Berkeley. First among them is my advisor, Raimund Seidel. Raimund always treated me with respect, encouraging me to follow my inclinations while providing critical feedback. He was patient as I learned how to talk and write about math. Whenever I knocked on his door, I knew that he would be genuinely glad to see me; this made a tremendous difference.

I would never have gone to graduate school without the Reentry Program, and I thank everyone involved in it. It is a model affirmative action program.

I was lucky to be at a school that takes graduate instruction seriously. I especially profited from the teaching of professors Karp and Vazirani.

I am proud to have had access to the experience and insight of Nimrod Megiddo. Two summers at Bell Labs in Murry Hill, thanks to the Graduate Research Program for Women, were important in my development. I thank my mentor Steve Fortune, Ken Clarkson, and Peter Shor for crucial encouragement. Another summer at Thinking Machines in Cambridge was a valuable and pleasant experience. Finally, I thank Raimund for bringing me to the Freie Universiatät, Berlin, where Emo Welzl, Jirka Matoušek, Helmult Alt and their students made me feel welcome.

My fellow Berkeley students were delightful and fascinating. It was with the other theory girls - Ronitt Rubinfeld, Sandy Irani, Diane Hernek and Dana Randall - that I learned how to be a woman scientist, and enjoy it. Ronitt's faith in me has made the difference more than once, Sandy has been an ideal buddy in both work and play, and Diane, Dana and I giggled our way through the whole experience. Mike Luby has been invaluable as an honored friend and advisor. Will Evans has been a great friend, an inspiring example, and the perfect office-mate. Ashu Rege was an essential civilizing presence in the office.

Renny Lucas was my first Berkeley friend and still one of the best. I will always value her good advice and her sense of humor. Becky Gross's affection and enthusiasm touch everyone's heart, especially mine.

The graphicsboys - Paul Heckbert, Seth Teller, Henry Moreton, Mike Hohmeyer and Eric Enderton - taught me a great deal, and delighted me with their individual warmth and collective liveliness. I thank Carlo Sequin for fostering their environment.

Nervous for Nigel - Dan Jurafsky, DeKai Wu, Terry Regier, Erin Dare, and sometimes Eric Enderton - was the fundamental force that made Berkeley CS a “scene” rather than a mere department. I hope to always be their number one fan, and I thank them for keeping me up so late. Like the rest of the AI crowd, including Mike Schiff and the actual Nigel Ward, they always had something interesting to talk about.

One of the pleasures of academia is meeting people from all over. The Zlatevs from Bulgaria, and Monica from Berlin, were an education. My Napolitan friend, Geppino Pucci, can always make me laugh; I thank him from somewhere deep in the pit of my stomach.

Some people from my past made graduate school possible. Both Tom Hayes and Sally Calhoun gave me great jobs that pushed me forward. And the deluxe Jim McGuffin taught me two essential skills during our many years as compañeros de trabajo - how to hack, and how to party.

So many other people were important. Ramon Caceres, Fracesca Barrientos, Amie Wilkinson, Sally Floyd, Susan Mathews and Jane Edwards all helped in their very different ways. David Wolfe showed me the ropes. The other grad students included so many people who were both brilliant and truly nice. Marshall Bern, Leo Guibas and Marco Pellegrini welcomed me to computational geometry, and Phil Klein and Danny Sleator made me feel welcome in the larger theory community.

I was lucky to have family nearby; I benefitted from the support and perspective of my sister, Davida, all through my years in Berkeley. Her honesty always kept me from heading off into the ozone. I thank her, my parents and my brother Tom. In this moment of achievement, I'm proud to be an Amenta.

Contents

Chapter 1

Introduction

This thesis builds a connection between two old and distinct lines of research. The story begins in 1911, with

Helly's Theorem *Let K be a family of at least $d+1$ convex sets in E^d, and assume K is finite or that every member of K is compact. If every $d+1$ members of K have a point in common, then there is a point common to all the members of K.*

This is a fundamental result in convexity theory and combinatorial geometry. It gave rise to a whole family of theorems with the same logical structure, for objects other than convex sets, for properties other than intersection, or for special cases in which $d+1$ is replaced by some other number. We call these *Helly theorems*. An influential survey by Danzer, Grünbaum, and Klee appeared in 1963 [DGK63], and some more recent material is collected in [E93], [GPW93].

Meanwhile, computers were invented, and one of the first uses to which they were put was

Problem: Linear Programming

Input: *A finite family of closed linear halfspaces in d-dimensional Euclidean space, called the* constraints*, and a linear objective function on E^d.*

Output: *The minimum of the objective function over the intersection of the constraints.*

The earliest algorithm for this problem was the simplex algorithm, introduced by Dantzig in 1951 [D51]. The simplex algorithm is essentially combinatorial, in that it searches the d element subfamilies of constraints for one which determines the minimum. Although it is arguably still the most efficient algorithm in practice, most variants have been shown to require exponential time in the worst case [KK87]. There is a well-developed theory of linear programming, but problems in which the constraints or the objective function are non-linear are less well understood. Much linear programming theory carries over to

Problem: Convex Programming

Input: *A finite family of closed convex sets in d-dimensional Euclidean space, and a convex objective function on E^d.*

Output: *The minimum of the objective function over the intersection of the constraints.*

An objective function is convex when $f(\lambda a+(1-\lambda)b) \leq \lambda f(a)+(1-\lambda)f(b)$, forall $a, b \in E^d$ and $0 \leq \lambda \leq 1$.

More general classes of problems are called *mathematical programming* or *nonlinear programming*. Combinatorial approaches analogous to the simplex method are often applied to convex and other nonlinear problems [F87]. These also perform well in practice.

Computational geometry is the branch of theoretical computer science which is concerned with geometric problems. This new field borrowed from both these older lines of research. Helly's theorem is a basic combinatorial tool in computational geometry. Other Helly theorems have also found algorithmic applications. At least a few people [EW89], [GPW93] had the idea that when there is a Helly theorem about some property on a class of objects, there also should be a linear time algorithm to test a finite family of objects for the property. Examples of such algorithms were known, but no general result.

Linear programming, in computational geometry, was both used to solve geometric problems (e.g. [AD92]) and studied as a geometric problem itself. Computational geometers concentrated on the fixed-dimensional case, in which the number of variables, d, is assumed to be constant, and the goal is to optimize the running time with respect to the number of constraints, n. Early deterministic algorithms were linear in n but double exponential in d [D84], [M84]. Later, simple randomized algorithms with a better dependence on d were developed [C90], [S90], [SW92]. These algorithms are combinatorial in the sense given above, and closely related to the simplex algorithm. This research into the fixed dimensional case led to progress on the general problem. An exciting recent development is that the randomized simplex algorithms in [K92], and the re-analysis of [SW92] in [MSW92], give algorithms subexponential [1] in both d and n. [2]

An important feature of the randomized algorithms [C90], [S90], [SW92] is that they can be applied to certain nonlinear problems as well, as all the authors observed. In [SW92], Sharir and Welzl formalize this idea by giving a *abstract framework*, that is, a list of combinatorial conditions on the family of constraints and the objective function, under which these algorithms can be applied. This framework defines a class of problems, which we call *Generalized Linear Programming*, or GLP. In [MSW92] Matoušek, Sharir and Welzl list many problems which are GLP, almost all of which can be formulated as special cases of convex programming. No provably non-convex examples of GLP were known.

The work in this thesis springs from the observation that there is a Helly theorem about the constraint set of every GLP problem (for example, Helly's theorem is about the constraint family of convex programming). One consequence of this observation is that we can prove a Helly theorem by showing that the set family in question is the constraint family of a GLP. We use this idea to give a simple proof of a Helly theorem in which the constraint set is not only not convex, but disconnected. This immediately shows that the the class of GLP problems includes more than just convex programming.

The next natural question is, can we go in the other direction? Given a family of constraints about which there is a Helly theorem, is it always possible to construct an objective function which gives a GLP problem? The answer is no; we give an example for which there is no such function. But we also give a paradigm which *does* yield an appropriate objective function in almost every interesting case.

Applying this paradigm to the collections of Helly theorems gives new algorithms for a variety of geometric optimization problems. Some of these problems arose in appli-

[1] By subexponential in x, we mean $O(e^{o(x)})$. The exact bound is $O(e^{O(\sqrt{d \ln n})})$.

[2] We will survey the computational geometry algorithms in detail in Chapter 7.

cations in computer graphics, computer vision, and computer aided manufacturing.

The GLP algorithms handle the combinatorial aspects of these problems. Their running times are measured by the number of calls to *primitive operations*, which are required to solve subproblems of some fixed size d. When the time required for a primitive operation is unrelated to n, the number of constraints, the GLP algorithms run in expected time linear in n. This is applies to the fixed-dimensional versions of most of the problems listed below. When we can show that the primitive operations require at most subexponential time, the algorithm of [MSW92] is subexponential in the general dimensional case as well.

Here is a list of the problems we solve with GLP, with references to the relevant sections.

Homothet Cover(sections 9.1 and 9.3)
Find the smallest homothet of a convex object containing a family of sets in E^d, the largest homothet of a convex object contained in the intersection of a family of convex sets, or the smallest homothet of a convex object intersecting every member of a family of convex sets,

Convex Hausdorff Distance(section 9.4)
Find the minimum Hausdorff distance between two convex polygons in the plane, under translation and scaling,

Bounded Box(section 9.5)
Find the largest axis-aligned box in the intersection of a family of convex sets,

Line Transversal of Translates(section 10.1)
Find a line transversal of disjoint translates of a convex object in the plane,

Polytopal Hyperplane Fitting(section 10.2)
Find the hyperplane which minimizes the maximum distance to a family of points under any metric whose unit ball is a polytope (such as L^1 or L^∞),

Weighted L^∞ Hyperplane Fitting(section 10.4)
Find the hyperplane which minimizes the maximum distance to a family of points

in E^d under the *weighted* L^∞ *metric*, in which every coefficient of every point is equipped with a weight, [3]

Subset Line Fitting(section 11.1)
Find the line which minimizes the maximum distance to a family of points in E^d under any *subset metric*, for which the unit ball is not full dimensional,

Line Transversal for Separated Spheres(section 11.1)
Find a line transversal in E^d for a family of spheres such that the distance between any two spheres is at least the sum of their radii,

Weighted L^∞ Line Fitting(section 11.2)
Find the line which minimizes the maximum distance to a family of points in E^d under the weighted L^∞ metric.

Via GLP, this thesis ties together the algorithmic study of mathematical programming with the purely geometric results about Helly theorems. The offspring of this union is a variety of new algorithms, Helly theorems and proofs, which enrich both practice and theory.

[3]Each point coordinate p_i has a weight w_i. The weighted L^∞ distance from a hyperplane h to p is $\min_{x \in h} \max_i w_i(|x_i - p_i|)$, where x is a point in hyperplane h.

Chapter 2

Helly theorems

In this section we define Helly theorems and introduce notation, and give some background for our results.

2.1 Definitions

Let C be a family of objects, $\mathcal{P}$ a predicate on subfamilies of C, and k a constant. A *Helly theorem* for C is something of the form:

For all $H \subseteq C$, $\mathcal{P}(H)$, if and only if, $\mathcal{P}(B)$ holds for every $B \subseteq H$ with $|B| \leq k$.

A shorthand statement of a Helly theorem is that C has *Helly number* k with respect to $\mathcal{P}$. Although we need not think of k as a constant, we will restrict our attention to the case in which k is independent of $|H|$. We will use the following as a running example of a Helly theorem.

Theorem 2.1.1 (Radius Theorem) *A family of points in E^d is contained in a unit ball if and only if every subfamily of $d+1$ points are contained in a unit ball.*

Here the family of objects is the set of points in E^d, the predicate is that a subfamily is contained in a unit ball, and $k = d+1$. We will see in a moment that the Radius Theorem is a corollary of Helly's theorem proper.

2.2 Helly systems

The term *Helly-type theorem* is often used to describe a larger class of theorems, including ones in which the fact that every subfamily has some property $\mathcal{P}$ implies that the whole family has some other property $\mathcal{Q}$. We will not be concerned with this larger class. In fact, we will only be concerned with those theorems in which C is a family of sets and $\mathcal{P}$ is the property of having non-empty intersection. Fortunately many Helly theorems can be restated in this form.

We need some notation. A *set system* is a pair (X, C), where X is a set and C is a family of subsets of X. For $G \subseteq C$, we write $\bigcap G$ for $\{x \in X \mid x \in h, \forall h \in G\}$. We say that a family G of sets satisfies the intersection predicate, or simply *intersects*, when $\bigcap G \neq \emptyset$. A set system (X, C) is a *Helly system* if there is some k such that C has Helly number k with respect to the intersection predicate.

Sometimes we have to look closely to see that a particular Helly theorem is representable by a Helly system. Let's consider the Radius Theorem. Instead of a set of points, we can state it as a theorem about the intersection of sets of unit balls, where each set consists of all the unit balls containing a particular point. Formally, let X be the points of E^d, and let Y be the set of possible positions for the center of a a unit ball in E^d (which of course is another copy of E^d). Let the predicate $\mathcal{Q}(x, y)$ mean that the ball with center $y \in Y$ contains the point $x \in X$, let $c_x \subset Y$ be the set of centers of unit balls containing x, and let $C = \{c_x \mid x \in X\}$. Then (Y, C) is a Helly system, since any family $H \subseteq X$ of points corresponds to a family $C_H \subseteq C$, $C_H = \{c_x \mid x \in H\}$, such that there is a unit ball containing H if and only if C_H intersects.

This is a sort of dual transformation. We can apply it to theorems of the form "X has Helly number k with respect to $\mathcal{P}$", when X and $\mathcal{P}$ have the following special form. There has to be a set Y, and a predicate $\mathcal{Q}$ on pairs in $X \times Y$, such that $\mathcal{P}$ is defined in terms of $\mathcal{Q}$ as follows. For $A \subseteq X$, $\mathcal{P}(A)$ if and only if $\exists\ (y \in Y)\ \forall\ (x \in A) \mathcal{Q}(x, y)$. Then the theorem "$X$ has Helly number k with respect to $\mathcal{P}$" corresponds to the Helly system (Y, C), where $c_x = \{y \in Y \mid \mathcal{Q}(x, y)\}$, for $x \in X$, and $C = \{c_x \mid x \in X\}$.

2.3 More general Helly theorems

This duality transformation allows us to express certain Helly theorems in terms of the intersection predicate in some space Y. In the case of the Radius Theorem, Y is the space of unit balls in E^d, which we parameterize by identifying a unit ball with it's center. The sets C are convex under this parameterization (balls, in fact), so it is clear that the Radius Theorem is a special case of Helly's Theorem. Note that if we had used some other weird parameterization of Y, the sets C might *not* be convex. If, for some Helly system (Y, C), there is *any* way to parameterize Y as E^d so that the sets in C are all convex, then (Y, C) is a special case of Helly's Theorem proper.

But not all Helly theorems are special cases of Helly's Theorem. Rather, Helly's Theorem is itself a special case. Recall that a *cell*, in topological contexts, is (intuitively) something homeomorphic to a ball, with no holes or bubbles.

Helly's Topological Theorem[DGK63] (page 125) *Let K be a finite family of closed sets in R^d such that the intersection of every k members of K is a cell, for $k \leq d$ and is nonempty for $k = d+1$. Then $\bigcap K$ is a cell.*

This theorem says that the constraint sets do not have to be convex, only that they should intersect, topologically, as if they were. Many useful Helly theorems can be reduced to special cases of this theorem. A different generalization of Helly's theorem is

Morris' Theorem [Mo73]. *Let I_d^m be a finite family of sets in R^d, each of which is the disjoint union of at most m closed convex sets, with the special property that I_d^m is closed under intersection. Then $\bigcap I_d^m \neq \emptyset$ if and only if $\bigcap B \neq \emptyset$, for any $B \subseteq I_d^m$ with $|B| \leq m(d+1)$.*

Here, the intersection may have multiple convex components, and the Helly property is maintained so long as the number of components remains bounded by m.

One way to prove a Helly theorem is to reduce it to Helly's Theorem proper, or to one of its generalizations, by expressing it as a Helly system (Y, C) and showing that (Y, C) is a special case of one of the general theorems. Most, but not all, of the Helly theorems we will use can be proved in this way, although they were usually first proved by other techniques.

It is possible that there is some *very* general theorem which subsumes any Helly theorem which can be stated in terms of the intersection predicate. Such a general theorem would have to include both Morris' Theorem and Helly's Topological Theorem as special cases.

2.4 Helly system witness problems

One natural computational problem associated with a Helly system (X, C) is

Problem: Helly System Witness

Input: *A finite family* $H \subseteq C$.

Output: *An element* $x \in \bigcap H$, *or the special symbol* Ω *if* $\bigcap H = \emptyset$

For Helly's Theorem, [AH91] gave an algorithm for Helly System Witness which uses $O(n^{d+1})$ calls to a primitive which returns a point in the intersection of $d+1$ convex sets, or reports that the intersection is empty. Notice that the problem is non-trivial, since it is possible that none of the points returned by the primitive lie in $\bigcap H$. To find a point in

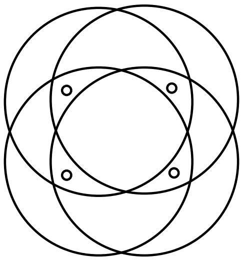

Figure 2.1: A point in every 3 sets but none in all 4

$\bigcap H$ they had to use other properties of convex sets, which is why the algorithm applies only to Helly's Theorem proper.

As we shall see later on, when we can introduce an objective function into the problem, we get much more efficient algorithms. For Helly's Theorem proper, introducing a convex objective function gives us the convex programming problem.

2.5 Helly theorems and VC-dimension

> *"We often ask our friends, "Any new Helly numbers?" " - Buchman and Valentine, beginning an article in American Mathematical Monthly.*

My friends often ask me, "Isn't the Helly number the same as the VC-dimension?" It is not. The VC (for Vapnik-Chervonenkis) dimension is another combinatorial property of set systems exploited by many algorithms in computational geometry and computational learning theory [HW87].

For a set system (X, C), we say a subset $B \subseteq X$ is *shattered* by C if C can select any subset of B, that is, $\{B \cap c \mid c \in C\} = 2^B$. The VC-dimension of (X, C) is the cardinality of the largest $B \subseteq X$ which is shattered by C. (If arbitrarily large sets of points can be shattered, we say that the VC-dimension is infinite.) For example, when C is all disks in the plane X, the VC-dimension of (X, C) is 3, because there are subsets of three points for which any subset can be selected by a circle, but no such set of four points.

The VC-dimension is a useful tool in developing randomized algorithms. In this thesis, we show that Helly theorems are too, which tends to increase the intuition that the two combinatorial properties might be related. But convex sets in R^d, the classic example

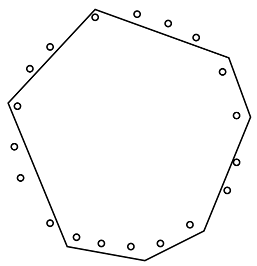

Figure 2.2: Convex sets have infinite VC-dimension

of a set system with constant Helly number, are well known to have infinite VC-dimension. This is shown by taking an arbitrarily large set of points on the surface of a sphere, as in figure 2.2; any subset of them can be selected by a single convex set. And conversely, the following example has infinite Helly number but constant VC-dimension.

Example 2.5.1 *Pairs of intervals*

Consider the set system (I, C), where I is the unit interval and C is the set of all pairs of connected subintervals. This set system has VC-dimension 4 [HW87] since any subset of four points on the line is shattered, but any subset of five points is not. But the Helly number for this set system is infinite, since for any n, we can construct a family of pairs of intervals for which every subfamily of size $n-1$ intersects but the entire family does not, as follows. Take any $n-1$ distinct points in the interior of the unit interval I. These divide I into n overlapping closed intervals I_j. Then let $C = \{c_j \mid c_j = I - I_j,\ 1 \leq j \leq n\}$. The entire family does not intersect since any point in I is in at least one of the I_j, while for any j, all the $c \in C$ except for c_j intersect in the interior of I_j.

This shows that the Helly number and the VC-dimension are completely independent.

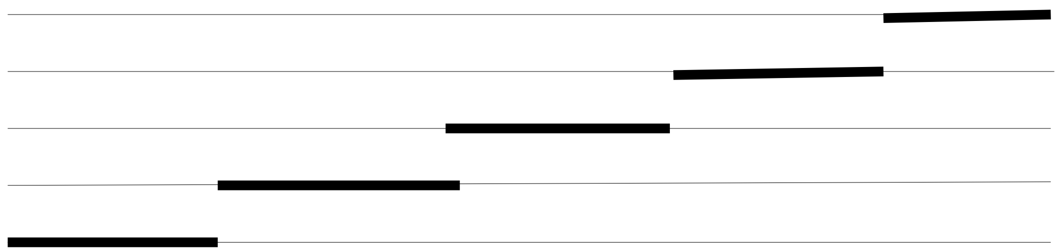

Figure 2.3: Pairs of intervals have infinite Helly number

Chapter 3

Generalized Linear Programming

In this chapter we introduce the idea of Generalized Linear Programming. We review the abstract framework, due to Sharir and Welzl, which defines GLP, and then we immediately embed this abstract framework in a more traditional mathematical programming context.

3.1 GLP framework

All of the following definitions are due to Sharir and Welzl, although the term GLP is mine. A *generalized linear programming* (or GLP) problem is a family H of *constraints* and an objective function w from subfamilies of H to some simply ordered set S [1]. The pair (H, w) must obey the following conditions:

1. Monotonicity: For all $F \subseteq G \subseteq H$: $w(F) \leq w(G)$

2. Locality: For all $F \subseteq G \subseteq H$ such that $w(F) = w(G)$ and for each $h \in H$:
$w(F + h) > w(F)$ if and only if $w(G + h) > w(G)$

By $F + h$, we mean $F \cup \{h\}$. The set S must contain a special maximal element Ω; for $G \subseteq H$, if $w(G) = \Omega$, we say G is *infeasible*; otherwise we call G *feasible*. A *basis* for

[1] A *simple* (or *linear*) order is like a total order, except with $<$ instead of $\leq$. Formally, a set S is *simply* (or *linearly*) ordered by a relation $<$ if, 1) for every $a, b \in S$, $a \neq b$, either $a < b$ or $b < a$, 2) there is no $a \in S$ such that $a < a$, and 3) if $a < b$ and $b < c$, then $a < c$.

$G \subseteq H$ is a minimal subfamily $B \subseteq G$ such that $w(B) = w(G)$. Here minimal is to be taken in the sense that for every $h \in B$, $w(B - h) < w(B)$. Any minimal subfamily B is a basis, if only for itself.

For instance, in linear programming, H is a finite set of closed halfspaces in E^d, and the function $w(G)$ returns the coefficients of the lexicographic minimum point in $\bigcap G$. If this intersection is non-empty, this point is determined by a basis of size no greater than d (we ensure that every subfamily has a minimum by surrounding the problem with a suitably large bounding box B, which is itself the intersection of halfspaces. We do not include halfspaces from B in the basis; so that if, for instance, $w(G)$ is the minimum point in B, then the empty set is a basis for G.). Notice that although more than d halfspaces may have the minimum point on their boundary, a subfamily of at most d of them are sufficient to determine the minimum. In the example below, the two solid lines are a basis. Notice also that a subfamily G may have more than one basis.

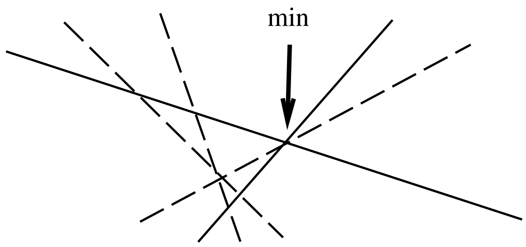

Figure 3.1: Basis for LP

Definition 3.1.1 *The* combinatorial dimension d *of a GLP problem is the maximum size of any basis for any feasible subfamily G.*

Definition 3.1.2 *A GLP problem of combinatorial dimension d, where d is independent of $|H| = n$, is called* fixed dimensional.

There are some simple consequences of this framework which drive the GLP algorithms. We say a basis B *violates* a constraint h when $w(B + h) > w(B)$. B is a basis for H if and only if B does not violate any element of H. Also, if $B \subseteq G \subseteq H$ is a basis for H, then B is a basis for G as well. Another property is that if B is a basis for G, and B violates h, then h is a member of any basis for $G+h$.

A somewhat less obvious consequence of the framework is

Theorem 3.1.1 *The maximum size of any basis for an infeasible problem of combinatorial dimension k is $k+1$.*

Proof: Let B be a basis for an infeasible problem, so that any proper subset of B is feasible. Let A be the subset $B-h$ such that $w(A) \geq w(B-h')$ for any $h' \in B$. Notice that A has to be feasible, since B is a basis. Let B' be a basis for A. Since $w(A+h) > w(A)$, the Locality Condition implies that $w(B'+h) > w(B') = w(A) \geq w(B-h')$ for any $h' \in B$. The Monotonicity Condition then implies that $(B'+h) \not\subseteq (B-h')$, for any $h' \in B$, that is, that $(B'+h) = B$. $|B'| \leq k$, so $|B| \leq k+1$.
■

We will use similar arguments later on, to prove the main theoretical results of the thesis.

3.2 Computational assumptions

It is not clear, of course, what computational operations are possible on an abstract object like (H, w). Sharir and Welzl assume two computational primitives, and analyze their algorithm by counting the number of calls to the primitives. The running time for a specific GLP problem then depends on how efficiently the primitives can be implemented.

Definition 3.2.1 *A* violation test *is a function $violation(B, h)$, which takes a basis B and a constraint h and returns* TRUE *if B violates h, and* FALSE *otherwise.*

Definition 3.2.2 *A* basis computation *is a function $basis(B, h)$, which takes a basis B and a constraint h and returns a basis B' for $B+h$.*

Sharir and Welzl note that only one primitive is required, since a violation test can be implemented by a basis computation (B violates h if $basis(B+h) \neq B$), but since the violation test can usually be implemented more efficiently, it is useful to count them separately.

These primitives are fundamentally different from the computational assumptions usually made in mathematical programming, which involve evaluating functions and taking derivatives, although they are analogous to the combinatorial primitive for the Helly System Witness problem employed by [AH91] .

3.3 GLP specialized to mathematical programming

In the GLP framework the constraints are just abstract objects, and the objective function applies to subfamilies of constraints. The traditional framework for mathematical programming is a little more concrete. It has three parts: an ambient space, or *ground set* X (usually E^d or the integer lattice), a set H of constraints, which are subsets of the ground set, and an objective function w' from X to some simply ordered set S. We call the elements of X *points*. The points in $\bigcap H$ are called *feasible*. The goal is to minimize w' over $\bigcap H$. To distinguish $w' : X \to S$ from $w : 2^H \to S$, we will call w' a *ground set objective function* and w a *subfamily objective function*. We represent a mathematical programming problem as a triple (X, H, w').

To simplify our proofs later, we will make a few observations about the GLP framework specialized to mathematical programming.

Definition 3.3.1 *Let (X, H, w') be a mathematical programming problem. For $G \subseteq H$, let $w(G) = \min\{w'(m) \mid m \in \bigcap G\}$, and $w(G) = \Omega$ when $\bigcap G = \emptyset$. Then $w : 2^H \to S$ is the* induced *subfamily objective function of (X, H, w'), and the pair (H, w) is the* induced *abstract problem.*

For example, in linear programming, the value of w on a subfamily of constraints is the minimum value that the linear objective function w' achieves on the feasible points.

There is a problem with this definition, however, since there may be no minimum point even when $\bigcap G \neq \emptyset$. For example, say X is E^d and w' is a linear function.

If the constraints in G are open sets, or if $\bigcap G$ is unbounded below, the minimum does not exist. One way to handle subfamilies whose intersection is unbounded is by somehow "compactifying" the space, for instance by representing points at infinity. Another common approach is to add constraints which bound the solution from below, for instance by putting a bounding box around a linear program. In this case we replace every constraint h by its intersection with the bounding box.

Definition 3.3.2 *When $w(G)$ is defined for every $G \subseteq H$, we say w is* well-defined*.*

Observation 3.3.1 *The induced abstract problem (H, w) satisfies the Monotonicity Condition of the GLP framework whenever w is well-defined.*

This follows from the fact that adding a constraint only eliminates feasible points, so the value of the minimum remaining feasible point can only go up. Certain functions w' also produce a function w which meets the Locality Condition.

Definition 3.3.3 *If w is well-defined, and $|\{x \in \bigcap G \mid w'(x) = w(G)\}| = 1$, for all $G \subseteq H$, then we say (X, H, w') satisfies the* Unique Minimum Condition*.*

This definition says that every subfamily not only has a minimum, but that this minimum is achieved by a unique point.

Observation 3.3.2 *If (X, H, w') meets the Unique Minimum Condition, then (H, w) satisfies the Locality Condition of the GLP framework.*

This is because if $w(G) = w(F)$, for $G \subseteq F$, is achieved only at a single point x, then $w(F + h) > w(F)$ only if $x \notin h$, in which case $w(G + h) > w(G)$. There is one easy way to satisfy the Unique Minimum Condition.

Observation 3.3.3 *If $w'(x) \neq w'(y)$ for any two distinct points $x, y \in X$, then (X, H, w') satisfies the Unique Minimum Condition.*

In general, however, the GLP we get when we use an arbitrary function w' to ensure the Unique Minimum Condition will *not* be fixed dimensional. For example, the most common function w' which ensures the Unique Minimum Condition on E^d is the *lexicographic objective function*, under which $x > y$ if $x_0 > y_0$, or $x_0 = y_0$ and $x_1 > y_1$, etc. [2] In the mathematical program in figure 3.2, X is a rectangle, and each constraint $h \in H$ is the complement of some translate of a downwards-directed cone. For any finite n, we can

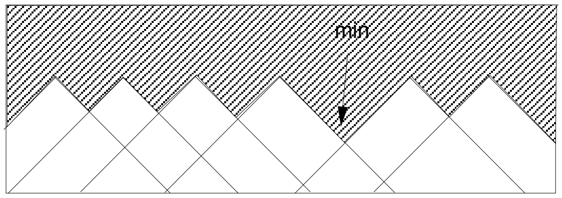

Figure 3.2: All constraints in the basis

construct an instance of this problem with n constraints, all of which are in the basis, when w is induced by the lexicographic objective function. Although it isn't worth coming back to this example, we note that the machinery developed in later chapters would allow us to construct an objective function w such that every (H, w) has combinatorial dimension at most two.

In most applications of mathematical programming, we are interested in finding not only an optimal value of S, but also in producing some point $x \in X$ which achieves this optimal value. In these cases we will use an *extended* version of the induced objective function w. The output of this extended function $w : 2^H \to S \times X$ is a pair (λ, x), where λ is the minimal feasible value of w', and x is a feasible point with $w'(x) = \lambda$.

3.4 Necessity of the GLP framework

The Monotonicity Condition and seems essential to our intuitive notions of what an LP-like problem ought to be, while the Locality Condition might appear to be an artifact of the algorithms. On the other hand, there are problems which meet the Monotonicity Condition, and have fixed combinatorial dimension, which do not seem at all LP-like. For example,

[2]We will study lexicographic objective functions in Chapter 6.

Problem: Point Set Diameter

Input: *A family of n points in E^d.*

Output: *The pair of points whose distance is maximum.*

We might represent this problem as a pair (H, w), where H is the set of points and $w(G)$

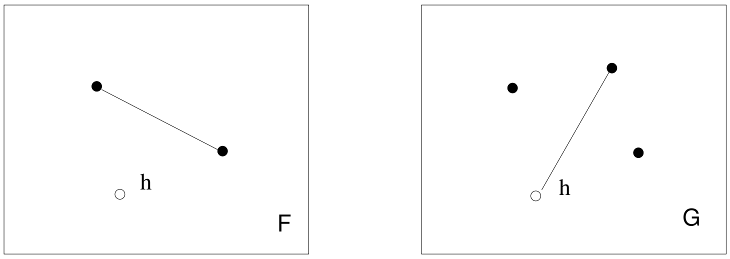

Figure 3.3: Adding h increases diameter of G but not F

is the maximum distance between any two points in G. This problem is monotone, and no basis has size greater than 2, but it fails to meet the Locality Condition, as illustrated in figure 3.3.

Certainly the diameter of H is less than any constant m if the diameter of any pair is less than m, but it is hard to see how to express this Helly theorem as a Helly system. Similarly there is no obvious mathematical program (X, H, w') which induces (H, w).

Chapter 4

Proving Helly Theorems with GLP

In this chapter we begin to connect Helly theorems and GLP. We observe that there is a Helly theorem about the constraint family of every GLP problem. This implies that we can prove Helly theorems using GLP. We use this idea to give a new, simple proof of an interesting Helly theorem.

First we define what we mean when we say there is a Helly theorem about the constraint family.

Definition 4.0.1 *A* Helly problem *is a pair (H, w), where H is a family of constraints and $w : 2^H \rightarrow S$ is a function to a simply ordered set S such that* there exists *a constant $m \in S$, with the property that $w(H) \leq m$ if and only if $w(B) \leq m$ for every $B \subseteq H$ with $|B| \leq k+1$.*

This defines a broad class of problems, since we only require one such constant m. If (H, w) is a GLP, the following easy theorem states that every $m \in S$ is such a constant. The Locality Condition is not needed, so in the following, just let H be a set of constraints and w an objective function on subsets of H, such that (H, w) meets the Monotonicity Condition and the maximum size of any feasible basis is k.

Theorem 4.0.1 *For all $m \in S$, H has the property $w(H) \leq m$ if and only if every $B \subseteq H$ with $|B| \leq k+1$ has the property $w(B) \leq m$.*

Proof: Let $w(H) \leq m$. By the Monotonicity Condition, every $B \subseteq H$ must have $w(B) \leq w(H) \leq m$. Going in the other direction, H must contain some basis B with $w(B) = w(H)$, with $|B| \leq k+1$. So if every subfamily B with $|B| \leq k+1$ has $w(B) \leq m$, then $w(H) = w(B) \leq m$.

■

So every GLP problem is a Helly problem. Replacing m with the special symbol Ω, we get

Corollary 4.0.1 *If H fails to intersect, then H contains a subfamily of size at most $k+1$ which also fails to intersect.*

The Helly number in Theorem 4.0.1 is $k+1$, while the combinatorial dimension is k. This is because every subfamily of size at most k may be feasible although H is infeasible. Given a guarantee that H is feasible, we get a slightly better result.

Corollary 4.0.2 *If H is feasible, H has the property $w(H) \leq m$ if and only if every $B \subseteq H$ with $|B| \leq k$ has the property $w(B) \leq m$*

In later chapters, we will go in the other direction; that is, we will use Helly theorems to construct GLP problems. But first, we will apply this observation to give a new, short proof of an interesting Helly theorem.

4.1 A short proof of Morris' theorem

The basic scheme for using GLP to prove a Helly theorem is this: given a set system which we suspect has a Helly theorem, we introduce an objective function, and then show that the resulting problem meets the Monotonicity and Locality Conditions and is fixed dimensional. We use this technique to give a simple proof of Morris's theorem (which we mentioned in chapter 2).

A family I of sets is *intersectional* if, for every $H \subseteq I$, $\bigcap H \in I$. Let C^d be the set of all compact convex sets in E^d, defined so as to include the empty set. Let Z_k^d be the family of all unions of k or fewer disjoint compact convex sets:

$$Z_k^d = \{ \bigcup_{i=1\ldots k} C_i \mid C_i \in C^d \text{ and } C_i \text{ disjoint}, \forall 1 \leq i \leq k\}$$

The whole family Z_k^d is not intersectional. But consider some subfamily $I_k^d \subset Z_d^k$ which *is* intersectional. This may "just happen" to be true, for instance for the family subsets if figure 4.1. Any intersection contains fewer than three connected components, although a single component in one intersection may be split in another. Or, I_k^d may be inherently

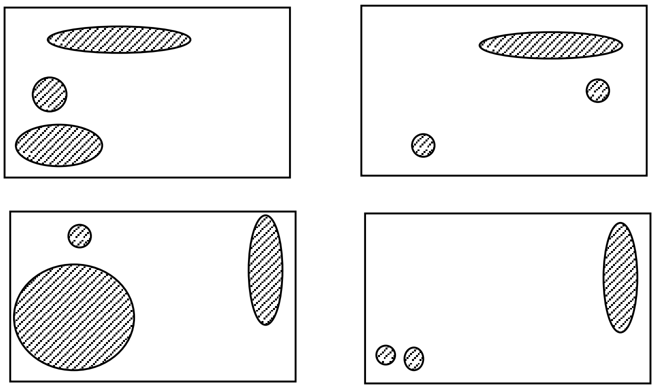

Figure 4.1: Just happens to be intersectional

intersectional, for example the family of subsets in figure 4.2, in which each subset is a pair of balls of diameter δ, separated by a distance of at least δ, kind of like dumbbells. Morris [Mo73] proved the following

Morris' Theorem *Any intersectional family $I_k^d \subseteq Z_k^d$ has Helly number $k(d+1)$.*

Grünbaum and Motzkin originally made this conjecture, [GM61], and proved the case $k = 2$. The case $k = 3$ was proved by Larman [L68]. Morris settled the conjecture in his thesis. His proof, however, is quite long (the thesis is 69 pages). In addition, Eckhoff notes [E93]

> ...Morris' proof of the lemma is extremely involved, and the validity of some of his arguments is, at best, doubtful. So an independent proof would be desirable.

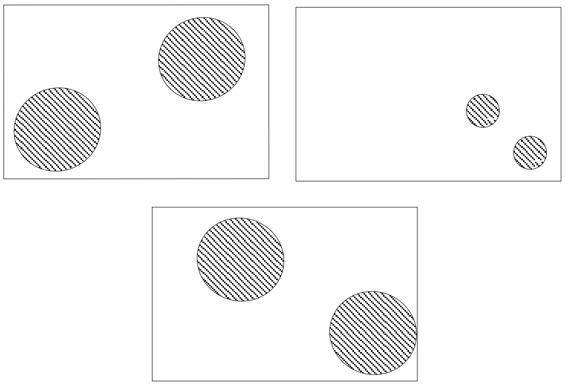

Figure 4.2: Inherently intersectional family

The GLP machinery leads to a much simpler proof.

In this proof we make use of the fact, which we shall prove in Section 6.2, that the following is a GLP problem of combinatorial dimension d.

Problem: Lexicographic Convex Programming

Input: *A finite family H of convex subsets of E^d, and a lexicographic function v' on E^d, such that $v'(x) = \langle v'_1(x), \ldots, v'_k(x)\rangle$, and and each v'_i is a convex function on E^d.*

Output: *The minimum of v' over $\bigcap H$.*

It is generally taken for granted by people familiar with GLP and its relations, although I cannot find a reference, that this problem is GLP.

We consider any finite family $H \subseteq I_d^k$ of constraints. As the objective function w, we use the function induced by the lexicographic objective function w' on the ground set E^d. For a subfamily H of constraints, $w(H) = \min\{x \mid x \in \bigcap H\}$. Since minima are identified with points, we will speak of $w(H)$ as a point and ignore the fine distinction between the point x itself and the value $w'(x)$.

Theorem 4.1.1 *(H, w) is a GLP problem of combinatorial dimension $k(d+1) - 1$.*

Proof: (H, w) satisfies the Monotonicity Condition, by Observation 3.3.1. And is satisfies the Locality Condition by Observation 3.3.2 since every point is assigned a different value by w'.

Recall that the combinatorial dimension is the largest cardinality of any feasible basis B. We will count the constraints in B by carefully removing selected constraints one by one, while building up a subfamily S of "sacred" constraints which may not be removed in later steps.

We will maintain two invariants. The first is that $w(B-h) < w(B)$ for all $h \in B-S$. Notice that if B is a basis, this is true for any $S \subseteq B$, and if $S \subseteq T$, the invariant holds for T whenever it holds for S. Now consider the point $m \in E^d$ which is the minimum $w(B)$. The second invariant is that for all $h \in B-S$, the minimum $m_h = w(B-h)$ lies in a convex component of $\bigcap(B-h)$ different from the one containing m.

First we choose the subfamily S so that the invariants are true initially. Since B is feasible, the minimum $m_0 = w(B)$ lies in some convex component of $\bigcap B$. Each $h \in B$ is the disjoint union of convex sets; the point m_0 is contained in exactly one of them. Call this C_h, and let $C = \{C_h \mid h \in B\}$. The pair (C, w) is an instance of Lexicographic Convex Programming, above, a GLP problem of combinatorial dimension d, with $w(C) = m_0$. So C must contain a basis B_C with $|B_C| \leq d$. We set $S = \{h \in B \mid C_h \in B_C\}$.

How does this ensure the invariants? Since B is a basis, the first invariant holds for any subset S. The second invariant holds because all the constraints which contributed a convex component to B_C are in S, and for any $h \in B-S$, $m_h < m_0 = w(B_C)$. That is, since m_0 is the lowest point in $\bigcap B_C$, and m_h is lower than m_0, then m_h cannot be in $\bigcap B_C$, and hence must be in a different convex component of $\bigcap(B-h)$.

Now we turn our attention to selecting a constraint to remove from B. We use the fact that all the $m_h = w(B-h)$ are distinct, for all $h \in B-S$. This is true because the point $m_h \notin h$, so that for any other $h' \in B$, $m_h \notin \bigcap(B-h')$ since $h \in (B-h')$. This fact implies that there is some $h_{max} \in B-S$ such that $w(B-h_{max}) > w(B-h)$ for all other $h \in B-S$.

So consider removing h_{max} from B. Since $w(B-h) < w(B-h_{max})$, for any other $h \in B-S$, certainly $w(B-h-h_{max}) < w(B-h_{max})$. So the first invariant is maintained for $B-h_{max}$ and S. To re-establish the second invariant, we have to add more elements to S. We do this in the same way as before, by finding the at most d constraints which determine the minimum of the convex component containing $m_{h_{max}}$. We add these constraints to S, and set $B = B - h_{max}$.

We iterate this process, selecting constraints to remove from B and adding constraints to S, until $B - S$ is empty. We now show that each removed constraint h accounts for at least one convex component $\mathcal{C}_h$ in the final feasible region $\bigcap S$. We associate h with the point $m_h = w(B - h)$, the new minimum point in $\bigcap(B - h)$. This point m_h is the minimum point in some convex component $\mathcal{C}_h$ of $\bigcap(B - h)$. We add the constraints determining m_h to S, so m_h will always be the minimum point in whatever convex component it lies in. So $\mathcal{C}_h$ cannot ever become part of some larger component with a lower minimum point m'. Each new component created will in fact have a lower minimum point m', so $\mathcal{C}_h$ will remain distinct from all later components. Since no earlier component $\mathcal{C}_{h_1}$ can every merge with a later component $\mathcal{C}_{h_2}$, every removed constraint h will account for a distinct component in $\bigcap S$.

Since I_k^d is an intersectional family, no subfamily of constraints can have more than k convex components in its intersection. Since B was feasible, we started with at least one convex component, so no more than $k - 1$ constraints were removed. Each constraint removed added at most d constraints to S, and we started with at most d constraints in S. So the total size of $|B| \leq (k-1) + |S| \leq k(d+1) - 1$.

■

Theorem 4.1.4 implies Morris' Theorem as a corollary of Theorem 4.0.1. Later, we will see other examples of Helly theorems which arise as byproducts of the construction of GLP problems.

This is an example of a GLP problem in which the feasible region is disconnected. We immediately get the interesting

Corollary 4.1.1 *The feasible region of a GLP problem need not be connected.*

Chapter 5

Constructing GLP problems using Helly theorems

We observed in the last chapter that there is a Helly theorem associated with every value m of the objective function of a GLP problem. If the problem is a mathematical programming problem (X, H, w'), then Theorem 4.0.1 says that the set $X_m = \{x \in X \mid w'(x) \leq m\}$, and the constraints restricted to that set, $H_m = \{h \bigcap X_m \mid h \in H\}$, form a Helly system (X_m, H_m). Now we want to go in the other direction: given a Helly system (X', H'), we want to use it somehow to construct a GLP problem. By analogy, we should try making the assumption that $(X', H') = (X_m, H_m)$ for some GLP (X, H, w').

In this chapter we formally define a paradigm for the construction of such a GLP based on a Helly theorem, and give some results about when the paradigm can be applied. In later chapters we will apply the paradigm to various Helly theorems to produce algorithms for some interesting geometric optimization problems.

5.1 Constructing a GLP objective function

At the heart of this paradigm is the construction of a subfamily objective function w, which usually, although not always, gives a GLP problem. The basic idea is to extend the Helly system to one which has a natural objective function.

We introduce a set S with a simple order $<$ to be the range of the objective function. Now we define the extension of a Helly system (X, H) to a *parameterized Helly system* $(X \times S, \overline{H})$. $(X \times S, \overline{H})$ is a set system, that is, the constraints $\overline{h} \in \overline{H}$ are subsets of $X \times S$. For a particular constraint $\overline{h}$, we write $h_\lambda = \{x \in X \mid \exists\, \mu \leq \lambda \; s.t.\; (x, \mu) \in \overline{h}\}$, for the projection into X of the part of $\overline{h}$ with S-coordinate no greater than λ. Also, for a subfamily of constraints $\overline{G} \subseteq \overline{H}$, we write G_λ as shorthand for $\{h_\lambda \mid \overline{h} \in \overline{G}\}$.

Definition 5.1.1 *An indexed family of subsets $\{h_\lambda \mid \lambda \in S\}$, such that $h_\alpha \subseteq h_\beta$, for all $\alpha, \beta \in S$ with $\alpha < \beta$, is a* nested family.

Definition 5.1.2 *A set system $(X \times S, \overline{H})$ is a* parameterized Helly system *with Helly number k, when*

1. *$\{h_\lambda \mid \lambda \in S\}$ is a nested family, for all $\overline{h} \in \overline{H}$, and*
2. *(X, H_λ) is a Helly system, with Helly number k, $\forall \lambda$.*

Figure 5.1 shows an example of a parameterized Helly system. The whole stack represents

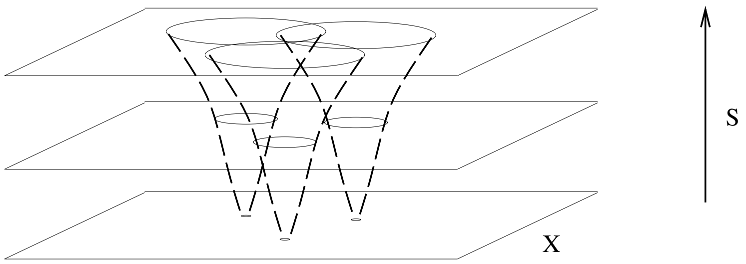

Figure 5.1: A parameterized Helly system

$X \times S$, and each of the fluted things is a set $\overline{h} \in \overline{H}$. Notice that because all the $\overline{h}$ are monotone with respect to S, the cross-section at λ (represented by one of the planes) is equivalent to the projected Helly system (X, H_λ).

We say that $\overline{G} \subseteq \overline{H}$ *intersects at* λ if $\bigcap G_\lambda \neq \emptyset$. Notice that in a parameterized Helly system, if $\overline{G}$ does not intersect at some value λ_2, then $\overline{G}$ also fails to intersect at all $\lambda_1 < \lambda_2$, and if $\overline{G}$ intersects at λ_1, then $\overline{G}$ also intersects at all $\lambda_2 > \lambda_1$.

Definition 5.1.3 *The* natural ground set objective function w' *of a parameterized Helly system is the projection into S, that is, $w'(x, \lambda) = \lambda$, where $(x, \lambda) \in X \times S$.*

The induced *natural objective function* w on subfamilies of constraints is then the least value of S at which the subfamily intersects, so that $w(\overline{G}) = \lambda^* = \inf\{\lambda \mid \bigcap G_\lambda \neq \emptyset\}$, and $w(\overline{G}) = \Omega$ if $\overline{G}$ does not intersect at any $\lambda \in S$ (recall that Ω is a symbolic maximal element of S).

It is almost always useful to think of S as *time* [1], so that a subfamily G_λ is a "snapshot" of the situation at time λ. Usually we can think of some initial time 0 at which G_0 does not intersect, and then envision the h_λ growing larger with time, so that $\lambda^* = w(\overline{G})$ is the first "moment" at which G_λ intersects.

As an example, let us consider how the Helly system (X, C) for the Radius Theorem can be extended to a parameterized Helly system. Recall that the ground set X is the set of centers of unit balls in E^d, and let $S = \mathcal{R}^+$ be the set of radii. Define $(X \times S, \overline{C})$ so that each $p_\lambda \in C_\lambda$ is the set of centers at which a ball of radius at most λ contains a particular point $p \in C$. The nested family $\overline{p}$ is the set of *all* balls containing p. The ground set $X \times S$ is the set of all balls in E^d, and $\overline{C}$ is the family of nested families for all points.

The natural objective function for this parameterized Helly system, $w(\overline{G})$, returns the smallest radius at which there is a ball containing all the points corresponding to constraints $\overline{p} \in \overline{G}$. So for any finite family $\overline{H} \subseteq \overline{C}$, $(X \times S, \overline{H}, w')$ is the following mathematical programming problem:

Problem: Smallest Enclosing Ball

Input: *A finite family H of points in E^d.*

Output: *The smallest ball enclosing H.*

Smallest Enclosing Ball is actually the classic GLP problem.

[1] although it need not be isomorphic to, well, whatever time is isomorphic to.

5.2 The main theorem

Theorem 5.2.1 (Main Theorem) *If $(X \times S, \overline{C})$ is a parameterized Helly system with Helly number k, then $(\overline{H}, w)$ is a GLP problem of combinatorial dimension k for any finite $\overline{H} \subseteq \overline{C}$, whenever $(X \times S, \overline{H}, w')$ meets the Unique Minimum Condition.*

Proof: Since w is induced by the natural ground set objective function w' on the space $X \times S$, by Observation 3.3.1, $(\overline{H}, w)$ obeys the Monotonicity Condition. Observation 3.3.2 tells us that $(\overline{H}, w)$ satisfies the Locality Condition whenever $(X \times S, \overline{H}, w')$ satisfies the Unique Minimum Condition. [2]

To prove that $(\overline{H}, w)$ has combinatorial dimension k, we have to show that the size of any basis is at most k. Consider any $\overline{G} \subseteq \overline{H}$ and a basis $\overline{B}$ for $\overline{G}$. The definition of a basis says that for any $\overline{h} \in \overline{B}$, $w(\overline{B} - \overline{h}) < w(\overline{B})$. Let $\lambda^{max} = max\{w(\overline{B} - \overline{h}) \mid \overline{h} \in \overline{B}\}$. Since $\overline{H}$ is finite, so is $\overline{B}$, and this maximum is guaranteed to exist.

The basis $\overline{B}$ does not intersect at λ^{max}, but for any $\overline{h} \in \overline{B}$, $w(\overline{B} - \overline{h}) \leq \lambda^{max}$, which means that $\overline{B} - \overline{h}$ intersects at λ^{max}. Since $(X, H_{\lambda^{max}})$ is a Helly system with Helly number k, $\overline{B}$ must contain some subfamily $\overline{A}$ with $|\overline{A}| \leq k$, such that $\overline{A}$ does not intersect at λ^{max}. Every $\overline{h} \in \overline{B}$ must be in $\overline{A}$, since otherwise it would be the case that $\overline{A} \subseteq (\overline{B} - \overline{h})$ for some $\overline{h}$. This cannot be, because $\overline{A}$ does not intersect at λ^{max} while every $(\overline{B} - \overline{h})$ does. Therefore $\overline{B} = \overline{A}$ and $|\overline{B}| \leq k$.

■

5.3 Parameterizing by intersection

In the Smallest Enclosing Ball example, we constructed a parameterized Helly system in a natural way using the radius as the parameter. This idea is very useful and can be applied to many transversal and covering problems. In fact, it suffices for all of our applications. But is it possible to parameterize *any* Helly system? In this section, we

[2] Recall that the Unique Minimum Condition (Definition 3.3.3) says that w is well-defined on any $\overline{G} \subseteq \overline{H}$ and that the minimum value of w' is achieved by a unique point in $\bigcap \overline{G}$.

give an existential argument which shows that it is, although we cannot guarantee that we produce a problem which obeys the Locality Condition.

We use the idea from a construction by Hoffman [H79]. He assumes the existence of one nested family of constraints and uses it to build an objective function. In this way he relates the Helly number k to something he calls the *binding constraint number*; in our terms, he shows that the combinatorial dimension is $k-1$. We use another fairly common idea from convexity theory to show that this construction can be applied to any Helly system.

Definition 5.3.1 *The* closure *of a set system (X, C) is a set system (X, C') where $C' = \{g = \bigcap G \mid , G \subseteq C\}$.*

The following lemma will come in handy in other contexts.

Lemma 5.3.1 (Intersection Lemma) *The closure (X, C') of a Helly system (X, C) of Helly number k also has Helly number k.*

Proof: We will prove this Helly theorem by proving the contrapositive; so we need to show that if a subfamily $G' \subseteq C'$ fails to intersect, then it contains a subfamily B' with $|B'| \leq k$ which also fails to intersect. Notice that $\bigcap G' \in C'$. Let

$$F_{G'} = \{h \in C \mid h \supseteq h', h' \in G'\}$$

Notice that $\bigcap F_{G'} = \bigcap G'$, so if $\bigcap G' = \emptyset$, then there is some $B \subseteq F_{G'}$ with $|B| \leq k$ such that $\bigcap B = \emptyset$. Each $h \in B$ must contain at least one $g \in G'$. We construct a corresponding set $B' \subseteq G'$ by choosing a single such g for each h. Then B' cannot intersect, and $|B'| \leq k$. So (X, C') also has Helly number k.

■

We use this lemma to show

Lemma 5.3.2 *Every Helly system can be extended to one which includes a nested family.*

Proof: Let (X, C) be a Helly system. We begin by using the intersection lemma to extend (X, C) to (X, C').

The maximum principal from set theory (e.g. [Mu75], page 69) says that any set with a partial order on its elements has a maximal simply ordered subset. So the set C', partially ordered by inclusion, contains some, not necessarily unique, maximal sequence $\overline{P}$. We extend (X, C) to $(X, C + S)$, where S is the family $\emptyset + \overline{P} + X$.

■

This lemma is non-constructive, since we do not say how to find a maximal sequence $\overline{P}$. Now we show that we can use S to parameterize the original Helly system (X, C).

Theorem 5.3.1 *Let (X, C) be a Helly system with Helly number k, with $S \subset C$ such that S is a nested family and $\emptyset, X \in S$. (X, C) can be extended to a parameterized Helly system $(X \times S, \overline{C})$ with natural objective function w. If, for a finite subfamily $\overline{H} \subseteq \overline{C}$, $(X \times S, \overline{H}, w')$ meets the Unique Minimum Condition, then $(\overline{H}, w)$ is a GLP problem of combinatorial dimension $k - 1$.*

Proof: The range of our objective function will be the nested family S itself. So $\lambda \in S$ is a subset of X, rather than, for instance, a real number.

We extend every element $h \in C$ of our original family of constraints to a nested family $\overline{h}$. For $\lambda \in S$, we let $h_\lambda = h \cap \lambda$. Again we write $\overline{C} = \{\overline{h} \mid h \in C\}$, and $C_\lambda = \{h_\lambda \mid h \in C\}$.

Assuming that $(X \times S, \overline{H}, w')$ meets the Unique Minimum Condition, the Main Theorem tells us that $(\overline{H}, w)$ is a GLP problem of combinatorial dimension no greater than k. It remains therefore to be shown that the combinatorial dimension is in fact no greater than $k - 1$, that is, that any feasible basis $\overline{B} \subseteq \overline{H}$ has $|\overline{B}| \leq k - 1$.

We use another version of the argument in the proof of the Main Theorem. Let $\lambda^* = w(\overline{B})$. There is some element $\overline{h}_{max} \in \overline{B}$, such that $\lambda_{max} = w(\overline{B} - \overline{h}_{max}) \geq w(\overline{B} - \overline{h})$ for all $\overline{h} \in \overline{B}$. Remember that $\lambda_{max} \in C$.

Now let $B = \{h \in C \mid \overline{h} \in \overline{B}\}$. $B_{\lambda_{max}}$ fails to intersect, which means that $A = B + \lambda_{max}$ fails to intersect. Since $\overline{B}$ is a basis, $B - h + \lambda_{max}$ does intersect, for every $h \in B$, and we know that B intersects because B_{λ^*} intersects.

So all of A's proper subsets do intersect, while A does not. Since $A \subseteq C$ and

(X, C) is a Helly system with Helly number k, $|A| \leq k$. This means $|B| = |\overline{B}| \leq k - 1$. $\blacksquare$

The tricky part here is using the maximal simply ordered sequence S itself as the parameter of the nested family. We have to do it this way because S may have any order type; it may not be isomorphic to some convenient simply ordered set like the real line.

Notice that the Main Theorem used the fact that the Helly number of each (X, G_λ) was k to show that the combinatorial dimension was bounded by k. Here, although the combinatorial dimension is $k - 1$, the Helly number of each (X, G_λ) can still be as great as k.

Lemma 5.2.2 and Theorem 5.2.1 together imply

Theorem 5.3.2 *Every Helly system can be extended to a parameterized Helly system.*

This theorem is non-constructive because Lemma 5.5.2 was non-constructive.

We reiterate that although we can parameterize any Helly system by intersection, we cannot guarantee that the natural objective function always gives a GLP. In the next section, however, we get a useful GLP by a careful choice of the nested family S.

5.4 Convex Programming

We now have the machinery to show that Convex Programming is GLP. Remember that a d-dimensional Convex Program is a mathematical program (X, H, w'), where the ground set X is E^d, H is a family of convex sets, and w' is a convex function from X to $\mathcal{R}$. That means that all the sets $w'_\lambda = \{x \in X \mid w'(x) \leq \lambda\}$ are convex.

Theorem 5.4.1 *Any d-dimensional Convex Program which meets the Unique Minimum Condition is a GLP of combinatorial dimension d.*

Proof: The important observation is that the family of sets $\overline{P} = \{w'_\lambda \mid \lambda \in \mathcal{R}\}$ is a nested family of of convex sets, isomorphic to $\mathcal{R}$. Remember that (X, H) is a Helly system with Helly number $d + 1$ (an instance of Helly's Theorem proper, since H is a finite family of convex sets). We let $S = \emptyset + \overline{P} + X$. Then we just apply Theorem 5.2.1 to $(X, H + S)$ and S.

■

Since we know, from Corollary 4.1.1 that there are GLP problems whose constraints are not even connected, let alone convex, we can infer from Theorems 5.3.1 that

Corollary 5.4.1 *The class of GLP problems strictly contains the class of convex programming problems.*

It cannot be the case that all Helly systems produce a GLP when parameterized by intersection, for as we shall see in the next section, there are some Helly systems for which no function w gives a fixed dimensional GLP problem.

5.5 A Helly system with no fixed combinatorial dimension

In this section we exhibit a set system with a fixed Helly number which *cannot* be turned into a fixed dimensional GLP problem.

Theorem 5.5.1 *For all $n > 1$, there is a family H of $2n$ sets with Helly number 2 such that for any valid GLP objective function w the combinatorial dimension of (H, w) is n.*

Proof: Let the universe X consist of the 2^n points at the vertices of an n dimensional hypercube, and let the constraint family H be the $2n$ subsets each of which lies in a facet of the hypercube. If a subfamily $G \subseteq H$ includes any pair of opposite facets, then G fails to intersect, and otherwise G does intersect. So the Helly number of (X, H) is 2.

Any valid objective function w must assign $w(G) = \Omega$ to the infeasible families G which contain a pair of opposite facets. Meanwhile any feasible G which does not contain a pair of opposite facets will have $w(G) = \lambda \in S$, with $\lambda < \Omega$. Let

$$\lambda^{max} = \max\{\lambda \in S | \lambda < \Omega \text{ and } \lambda = w(G) \text{ for some } G \subseteq H\}$$

and consider some G with $w(G) = \lambda^{max}$. If $|G| < n$, then there exists some pair (h^+, h^-) of facets, such that G contains neither h^+ nor h^-. This means that $G + h^+$ is also feasible. By the Monotonicity Condition, $w(G + h^+) \geq w(G)$; and since $w(G)$ is maximal, we can conclude that $w(G + h^+) = w(G) = \lambda^{max}$. This argument shows that there must be a subfamily G of size n with $w(G) = \lambda^{max}$.

Now we show that there is no basis B for such a subfamily G such that $B \neq G$. Assume, for the purpose of contradiction, and without loss of generality, that there is some element $h^+ \in G$ such that $h^+ \notin B$. Then $B + h^-$ is still feasible, so $w(B + h^-) = w(B) = \lambda^{max}$. But $w(G + h^-) = \Omega$. Since $B \subset G$ and $w(B) = w(G)$, this means that w is not a valid objective function because it fails to satisfy the Locality Condition. So any valid objective function w must have $B = G$, and (H, w) must have combinatorial dimension n.

■

For any valid objective function w, there is a Helly theorem about the constraints of (H, w) for the value $m = \Omega$ (see definition 4.0.1). Together with Theorem 4.0.1, then, this theorem implies

Corollary 5.5.1 *The class of GLP problems is strictly contained in the class of Helly problems.*

In this example, we increase the combinatorial dimension by increasing the dimension of the ground set X. It is also possible to present this example combinatorially, as a function on family of abstract constraints, or on a family of (perhaps oddly shaped) sets in some constant dimensional space. But is it possible to construct a family of sets in a constant dimensional space which has the same intersection pattern as this example, in which every set, or intersection of sets, is the union of a constant number of cells? We make the following

Conjecture 5.5.1 *Let (X, C) be a set system, closed under intersection, such that X is R^d, and any element $h \in C$ is the disjoint union of at most m cells. Then for any finite $H \subseteq C$, there is an objective function w such that (H, w) is a GLP problem of combinatorial dimension $f(d, m)$, where $f(d, m)$ is a function independent of $|H|$.*

This would imply that the Helly number of such a family is no greater than $f(d,m)+1$, a very general Helly theorem which would subsume both Helly's Topological Theorem and Morris' Theorem (see Section 2.3).

Chapter 6

Lexicographic objective functions and quasiconvex programming

So far we have just assumed that we can satisfy the Unique Minimum Condition. This is not immediate even for linear programming. The objective function

$$\textit{minimize}\ x_0$$

fails to ensure a unique minimum for every finite subset of constraints, since the minimal value of x_0 might be achieved over a face of any dimension in the feasible polytope. One way to fix this is to rotate the polytope, or equivalently the objective function, slightly so that no facet is parallel to the hyperplane $x_0 = 0$. This is an example of a *perturbation* technique. Although conceptually simple, these can be messy in practice. Instead, we usually use a *lexicographic* objective function

$$\textit{minimize}\ \langle x_0, x_1, \ldots, x_{d-1} \rangle$$

where the minimum point is the one which minimizes x_{d-1}, over all which minimize x_{d-2}, etc. In this section we use this idea to build a GLP objective function for other Helly systems.

We find that the lexicographic objective functions interact strangely with the combinatorial dimension. Here is one way of looking a problem with a lexicographic objective function v. Say v has k parameters. The most significant parameter is determined by some simple one-parameter objective function w_1. At each value λ of w_1, H_λ is the constraint set of another problem whose lexicographic objective function has $k-1$ parameters. [1] By analogy with linear programming, if the original problem has combinatorial dimension d, we expect the new problem to have combinatorial dimension $d-1$, but in fact this is not always the case; surprisingly, the combinatorial dimension can go down by as much as a half.

This strange behavior does not occur for GLP problems where we could have used a perturbed one-parameter objective function instead of the lexicographic function. But some non-convex problems can *only* be formulated as GLP using a lexicographic objective function. These include a class of problems from the mathematical programming literature called *quasiconvex programs*. Some of our applications will fall into this class.

6.1 Constructing a lexicographic objective function

We construct a lexicographic objective function recursively. If the natural objective function on a parameterized Helly system fails to meet the Unique Minimum Condition, the most significant parameter λ is minimized over a region $R \subseteq X$, rather than at a single point. The recursive assumption we make is that minimizing the other parameters over R is a GLP problem of combinatorial dimension d. Given this assumption, we show that the problem of finding the overall minimum is also GLP.

Let $(X \times S, \overline{H})$ be the Helly system parameterized by λ with Helly number k and natural objective function w. For all λ, we assume that there is an objective function $v_\lambda : G_\lambda \to S'$, where S' is some totally ordered set containing a maximum element Ω', such that (G_λ, v_λ) is a GLP problem. We will define a *lexicographic objective function* $v : 2^{\overline{H}} \to S \times S'$ in terms of w and the functions v_λ. So that the range of this function is a simply ordered set, we impose a lexicographic order on the pairs in $S \times S'$, where $(\lambda, \kappa) > (\lambda', \kappa')$ if $\lambda > \lambda'$, or $\lambda = \lambda'$ and $\kappa > \kappa'$.

[1] This idea is sort of like building the problem up using parametric search.

Since the parameterized system $(X \times S, \overline{H})$ has Helly number k, so does every (X, G_λ). But a fixed Helly number does not necessarily imply a bound on the combinatorial dimension. So we will simply assume that the combinatorial dimension of every (G_λ, v_λ) is bounded above by some other constant d.

Theorem 6.1.1 *Let $(X \times S, \overline{H})$ be a parameterized Helly system with Helly number k such that, for all $\overline{G} \subseteq \overline{H}$,*

1. *$\lambda^* = w(\overline{G})$ exists, and*

2. *for all λ, there is a function $v_\lambda : 2^{G_\lambda} \rightarrow S'$, such that (G_λ, v_λ) is a GLP problem of combinatorial dimension at most d.*

Let $v(\overline{G}) = (\lambda^*, \kappa^*)$, where $\lambda^* = w(\overline{G})$ and $\kappa^* = v_{\lambda^*}(G_{\lambda^*})$. Then $(\overline{H}, v)$ is a GLP problem of combinatorial dimension at most $k + d$.

Proof: As in the Main Theorem, we note that the constraints $\overline{H}$ are subsets of $X \times S$, so $(\overline{H}, v)$ obeys the Monotonicity Condition.

To establish locality, consider $\overline{G} \subseteq \overline{F} \subseteq \overline{H}$, with $v(\overline{G}) = v(\overline{F}) = (\lambda^*, \kappa^*)$ and $v(\overline{F} + \overline{h}) = (\lambda, \kappa) > v(\overline{F})$. Then $v_{\lambda^*}(F_{\lambda^*} + h_{\lambda^*}) > v_{\lambda^*}(F_{\lambda^*})$, since either $\lambda = \lambda^*$ and $\kappa > \kappa^*$, or $\lambda > \lambda^*$ and $v_{\lambda^*}(F_{\lambda^*} + h_{\lambda^*}) = \Omega'$. In either case, since v_{λ^*} is a GLP objective function obeying the Locality Condition, $v_{\lambda^*}(G_{\lambda^*} + h_{\lambda^*}) > v_{\lambda^*}(G_{\lambda^*})$, and $v(\overline{G} + \overline{h}) > v(\overline{G})$. So the lexicographic function v also satisfies the locality condition.

Finally we consider the combinatorial dimension. Let $\overline{B}$ be a basis for any $\overline{G} \subseteq \overline{H}$. Then $v(\overline{B} - \overline{h}) = (\lambda, \kappa) < v(\overline{B}) = (\lambda^*, k^*)$, for any $\overline{h} \in \overline{B}$. Let the subset $\overline{B}_1 = \{\overline{h} \in \overline{B} \mid v(\overline{B} - \overline{h}) = (\lambda, \kappa)$ and $\lambda < \lambda^*\}$. Since the combinatorial dimension of $(\overline{B}_{\lambda^*}, v_{\lambda^*})$ is d, $\overline{B} - \overline{B}_1$ contains at most d constraints.

Again following the proof of the Main Theorem, we let $\lambda^{max} = max\{\lambda \mid v(\overline{B} - \overline{h}) = (\lambda, \kappa), \overline{h} \in \overline{B}_1\}$. $\overline{B}$ fails to intersect at λ^{max} and hence must contain some set $\overline{A}$ of size at most k which also fails to intersect. And again, every $\overline{h} \in \overline{B}_1$ must also be in $\overline{A}$, since $\overline{B} - \overline{h}$ intersects at λ^{max} and $\overline{A}$ does not, so $\overline{A} \not\subseteq \overline{B} - \overline{h}$. So $|\overline{B}_1| \leq |\overline{A}| \leq k$, and $|\overline{B}| \leq k + d$.

■

6.2 Some lexicographic functions are equivalent to perturbation

Certainly the bound on the combinatorial dimension which we get using this theorem is not always tight. For instance, consider the lexicographic objective function v for d-dimensional linear programming which we mentioned at the beginning of the chapter, induced by the lexicographic order on E^d, which we shall write as $v' = \langle x_0, x_1, \ldots, x_{d-1}\rangle$. We know the combinatorial dimension of (H, v) is d, but Theorem 6.1.1 tells us only that the combinatorial dimension is at most $d(d+1)/2$, since each G_λ is the constraint set of a $(d-1)$-dimensional linear program, and so on.

But notice that for any specific finite family H of constraints, there is a single-parameter GLP objective function u which has combinatorial dimension d, and which behaves on H exactly like the lexicographic function v. This function u is the one induced by the linear function

$$u'(x) = x_0 + \epsilon x_1 + \epsilon^2 x_2 + \ldots + \epsilon^{d-1} x_{d-1}$$

for some infinitesimal $1 > \epsilon > 0$. For linear programming, we can actually find some ϵ small enough so that for any pair of subsets $F, G \subseteq H$, if $v(F) > v(G)$ then $u(F) > u(G)$, and if $v(F) = v(G)$, then $u(F) = u(G)$.

Observation 6.2.1 *If (H, v) is a GLP problem, and there is some other objective function w, such that v and w impose the same total order on the subsets of H, then the combinatorial dimension of (H, v) is the same as the combinatorial dimension of (H, w).*

Notice that we are not required to actually produce such a one-parameter function u by finding some small enough ϵ; the fact that it exists is enough to tell us the combinatorial dimension of (H, v).

In general, given a lexicographic objective function v' on X with k parameters $\langle v'_0 \ldots v'_{k-1}\rangle$, we can construct the single parameter function $u'(x, \epsilon) = v'_0(x) + \epsilon v'_1(x) + \ldots + \epsilon^{k-1} v'_{k-1}(x)$. Call the corresponding induced objective functions v and u. We can think of u' as a family of functions parameterized by ϵ, any member of which we will write as u'_ϵ. We can also think of u' as a perturbation of the most significant parameter v'_1 of v'.

Let us consider using this construction with the objective function of

Problem: Lexicographic Convex Programming

Input: *A finite family H of convex subsets of E^d, and a lexicographic function v' on E^d, such that $v'(x) = \langle v'_1(x), \ldots, v'_k(x) \rangle$, and and each v'_i is a convex function on E^d.*

Output: *The minimum of v' over $\bigcap H$.*

We can define a one-parameter ground set objective function u', as above, corresponding to v', but there might be no ϵ small enough that v and u_ϵ behave exactly the same on every subfamily of constraints. For instance, in the Lexicographic Convex Program below, $v(\{a, b\}) = v(\{a\})$, where v is the function induced by the usual lexicographic order on E^2. This is not true of the corresponding function u_ϵ for any $\epsilon > 0$.

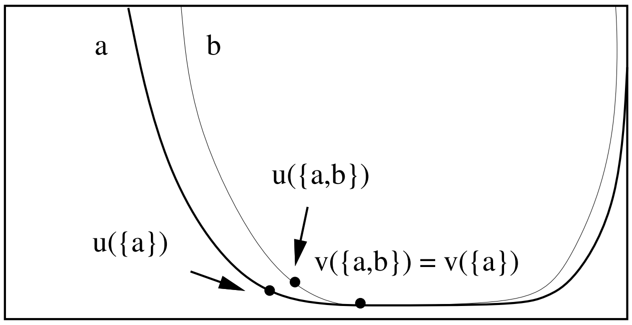

Figure 6.1: Minima are different under u_ϵ

As ϵ goes to 0, however, the points realizing u_ϵ converge to the points realizing v, so that if $v(F) > v(G)$, for infinitesimal ϵ, $u_\epsilon(F) > u_\epsilon(G)$. We make a slightly stronger

Observation 6.2.2 *Let (H, v) and (H, u) be GLP problems. If $v(F) > v(G)$ implies that $u(F) > u(G)$, then the combinatorial dimension of (H, v) is no greater than the combinatorial dimension of (H, u).*

To see this, let B be any basis under v. Then B is also a basis under u, since $u(B) > u(B - h)$ for all $h \in B$. So the maximum cardinality basis of (H, v) is no larger than the maximum cardinality basis of (H, u).

Observation 6.2.2 now tells us that the combinatorial dimension of (H, u_ϵ) is an upper bound on that of (H, v). Finally, notice that u' is a convex function because it is the sum of positive convex functions. So (H, u_ϵ) is a convex program with a single-parameter objective function, which, as we established in Theorem 5.3.1, has combinatorial dimension d. This implies the following

Theorem 6.2.1 *Any d-dimensional Lexicographic Convex Program can be solved by a GLP of combinatorial dimension d.*

6.3 An example which requires a lexicographic function

There are some problems, however, for which the lexicographic objective function is *not* equivalent to any one parameter function.

Theorem 6.3.1 *There is a parameterized Helly system $(X \times S, \overline{H})$ with Helly number $d+1$, and a family of functions v_λ where every (G_λ, v_λ) is a GLP problem with combinatorial dimension d, such that (H, v) has combinatorial dimension $2d+1$, where v is the lexicographic objective function.*

Notice that the combinatorial dimension in this theorem meets the upper bound of theorem 6.1.1 for $k = d+1$. **Proof:** We define a problem in which every G_λ is a d-dimensional linear programming problem. Every nested family $\overline{h} \in \overline{H}$ is a subset of $E^d \times \mathcal{R}$ of the form $a \cdot x \geq b_\lambda$, where a is a vector, $\cdot$ is dot-product, $x \in E^d$ and b_λ is a scalar quantity that varies with λ as follows. Let b and c be constants. For $\lambda < c$, $b_\lambda = b$, and for $\lambda \geq c$, $b_\lambda = -\infty$. Essentially, a constraint remains in force below c, and recedes to $-\infty$ above. The function v_λ can be any linear function. Notice that $(E^d \times \mathcal{R}, \overline{H})$ is a parameterized Helly system with Helly number $d+1$, and that every (G_λ, v_λ) has combinatorial dimension d.

We can construct an instance of this problem for which some basis $\overline{B} \subseteq \overline{H}$ has size $|\overline{B}| = 2d+1$. Figure 6.2 is an example for the case $k = 1$. The constraints h_1 and h_2 determine the minimum value of λ, while constraint h_3 determines the minimum value of x.

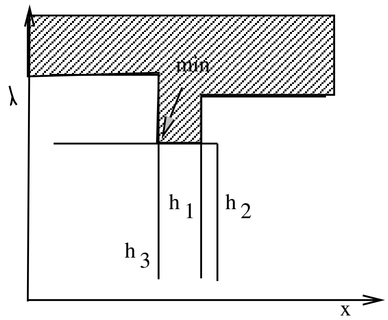

Figure 6.2: Size of basis is three

To construct such an example in general dimension, let u be any linear objective function on E^d. Construct a tiny simplex s by intersecting some family A' of positive halfspaces with $|A'| = d+1$. Let A be the corresponding family of negative halfspaces; A is the constraint set of an infeasible d-dimensional linear program, with A a basis for (A, u). Now let B be the basis of a feasible d-dimensional linear programming problem, with the same objective function u, such that $|B| = d$ and the simplex s is strictly contained in the interior of $\bigcap B$. We parameterize A and B by λ, as above, assigning some arbitrary value c_a to every $\overline{h} \in \overline{A}$, and some other value $c_b > c_a$ to every $\overline{h} \in \overline{B}$. Let the constraint set of the whole problem be $\overline{H} = \overline{A} \cup \overline{B}$. For every $\lambda \in S$, let the function v_λ be u.

Then $v(\overline{H}) = (\lambda^*, \kappa^*)$, with $\lambda^* = c_a$, and $\kappa^* = u(B)$. So what is (λ^*, κ^*), and how big is a basis for $(\overline{H}, v)$? Since A is the basis for an infeasible linear program, λ^* is at least c_a. Since the little simplex s is strictly contained in $\bigcap B$, if we remove any $\overline{h} \in \overline{A}$, we get $w(\overline{H} - \overline{h}) = -\infty$. Since $\kappa^* = u(B)$, then $v_{\lambda^*}(H_{l^*} - h_{l^*}) < k^*$ for all $\overline{h} \in \overline{B}$. So removing any element of $\overline{H}$ causes the minimum to go down, and $\overline{H}$ is a basis of size $2d+1$.

■

6.4 Quasiconvexity

The strange constraints in the GLP problem of Theorem 8 are certainly not convex. They belong to a category from the mathematical programming literature known as *quasiconvex* functions. Like a convex program, the only minimum of a quasiconvex program is the global minimum. We will show that like convex programs, quasiconvex programs can be solved by GLP, using lexicographic objective functions. [2]

Consider functions from $\mathcal{R}^d$ to $\mathcal{R}$. Remember that a function f is convex if $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$, for any x, y in $\mathcal{R}^d$, with $0 \leq \lambda \leq 1$. A function f is quasiconvex if $f(\lambda x + (1-\lambda)y) \leq \max\{f(x), f(y)\}$, and *strictly* quasiconvex when $f(\lambda x + (1-\lambda)y) < \max\{f(x), f(y)\}$, with $0 \leq \lambda \leq 1$.

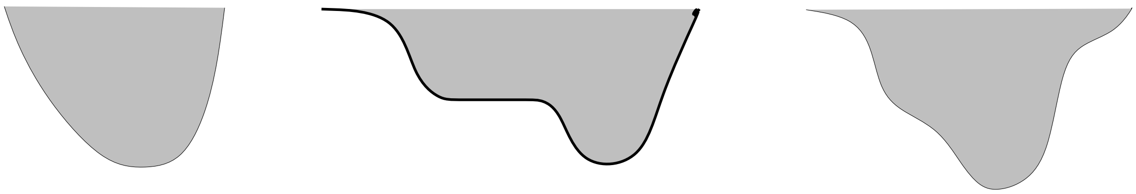

Figure 6.3: convex, quasiconvex, strictly quasiconvex

Define the $(\leq k)$-level set of f to be the $\{x \mid f(x) \leq k\}$. An alternative definition of quasiconvexity is that a quasiconvex function is one whose $(\leq k)$-level sets are convex. Notice that all convex functions are quasiconvex. Also notice that if a quasiconvex function f is constant along any line segment, it is *not* strictly quasiconvex.

A $(d+1)$-dimensional *quasiconvex program* is a problem of the form

$$\text{minimize } \lambda, \text{ subject to}$$

$$f_i(x) - \lambda \leq 0, \quad \text{for } i = 1, \ldots n, n \geq 1$$

$$c_j(x) \leq 0, \quad \text{for } j = 1, \ldots m, m \geq 0$$

where the f_i are quasiconvex functions, the c_i are convex functions, and x is a d-vector of variables. A two-dimensional example might look like figure 6.4.

Programs in which all the constraints are convex and the objective function $g(x)$ is quasiconvex are also common, but they can be expressed in the more general form above

[2] I am indebted to Nimrod Megiddo for telling me that the functions I was interested in are called quasiconvex.

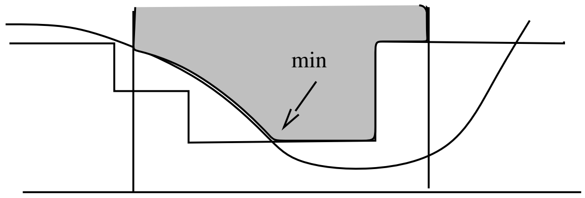

Figure 6.4: Quasiconvex program

by adding λ as a new dimension and $g(x) - \lambda \geq 0$ as a new constraint [3].

Many interesting and useful functions are quasiconvex (see [Mag69], page 148). Ratios of linear functions

$$f(x) = \frac{a \cdot x}{b \cdot x}$$

are quasiconvex over any domain where the denominator $b \cdot x$ is never 0. Also, functions of the form

$$f(x) = \frac{g(x)}{b \cdot x}$$

over any domain where $g(x)$ is convex and $b \cdot x$ is positive are quasiconvex.

We now proceed to show that quasiconvex programming is GLP by constructing nested families parameterized by λ.

For a quasiconvex constraint f_i, define

$$h_\lambda = \{x \mid f_i(x) \leq \lambda\}$$

Each h_λ is a convex subset of E^d. The h_λ form a nested family, $\overline{h_i}$, indexed by λ. For a convex constraint, c_j, define

$$g_\lambda = \{x \mid c_j(x) \leq 0\}$$

The g_λ also (trivially) form a nested family $\overline{g}_j$.

The constraints of our GLP are $\overline{H} = \{\overline{h}_i \mid i = 1 \ldots n\} \cup \{\overline{g}_j \mid j = 1 \ldots m\}$. $(E^d \times \mathcal{R}, \overline{H})$ is a parameterized Helly system, and the natural objective function w' gives us a mathematical programming problem $(E^d \times \mathcal{R}, \overline{H}, w')$.

[3] In this case a *strictly* quasiconvex function $g(x)$ can often be replaced with a convex objective function guaranteed to have the same minimum.

As we do for linear or convex programs, we can ensure that the objective function is well defined by adding constraints to the problem which bound the solution from below. In this case any single quasiconvex constraint v will do, so long as $w(\{v\}) < w(\overline{G})$ for all subfamilies $\overline{G} \subseteq \overline{H}$ such that $w(\overline{G}) = \lambda$ exists. We redefine $\overline{H}$ by replacing every constraint $\overline{h}$ or $\overline{g}$ with $\overline{h} \cap v$ or $\overline{g} \cap v$. If it is not the case that all the f_i are strictly quasiconvex, we also require v to bound the minimum of every G_λ as well as of $\overline{G}$.

Theorem 6.4.1 *A quasiconvex program $(E^d \times \mathcal{R}, \overline{H}, w')$ in which all the f_i are strictly quasiconvex, is a GLP problem of combinatorial dimension d, whenever the subfamily objective function w is well-defined.*

Proof: We need to show that $(\overline{H}, w)$ is a GLP problem. Since $(\mathcal{R} \times E^d, \overline{H})$ is a parameterized Helly system, and we assume that $w(\overline{G})$ is defined for all $\overline{G} \subseteq \overline{H}$, it suffices to show that $|\bigcap G_{\lambda^*}| = 1$, and then apply the Main Theorem.

This condition follows from the definition of strict quasiconvexity. Assume for contradiction that $\bigcap G_{\lambda^*}$ contains two points x, y. Any point on the segment between them, for instance $(x/2+y/2)$, is in $\bigcap G$. Because f is strictly quasiconvex, $\mu = f(x/2+y/2) < \max\{f(x), f(y)\} = \lambda^*$, so the point $(x/2+y/2) \in \bigcap G_\mu$. This contradicts the definition of λ^* as the minimum value λ at which G_λ is non-empty. So $\bigcap G_{\lambda^*}$ must consist of exactly one point.

■

For quasiconvex, but not strictly quasiconvex, functions, we need to use a lexicographic objective function.

Theorem 6.4.2 *A quasiconvex program $(E^d \times \mathcal{R}, \overline{H}, w')$ is a GLP problem of combinatorial dimension d, whenever the subfamily objective function w is well-defined.*

Proof: Since $\bigcap G_{\lambda^*}$ is convex, it has a unique minimum with respect to any convex objective function g on R^d. So the parametric objective function $v = (w, w')$ where w is the natural objective function on $\overline{G}$, and w' is the function on subsets of R^d induced by

g, gives a GLP problem of combinatorial dimension $\leq 2d$.

■

In fact, the idea of a parameterized Helly system generalizes the idea of a quasiconvex program. In a quasiconvex program, each family H_λ if a convex family of sets, obeying Helly's Theorem proper, whereas in a general parameterized Helly system, H_λ can be any family of sets which obeys some Helly theorem.

Chapter 7

GLP algorithms

"I think you have just rediscovered the simplex algorithm." - Raimund Seidel, after a research talk.

This chapter contains a survey of GLP algorithms. It establishes that there are algorithms for the entire class of GLP problems, and gives a high-level overview of their relative merits in terms of simplicity and asymptotic analysis. Such a survey is interesting beyond the context of this thesis, since the field has been very active in the past few years. Most of the GLP algorithms can be seen as variants of the simplex algorithm, the oldest and perhaps the most efficient linear programming algorithm. We will begin by showing that *any* simplex algorithm can be generalized to GLP.

7.1 The simplex algorithm

Most of the GLP algorithms, when applied to linear programming, are what is called *dual-simplex* algorithms. Remember that any linear program can be put into a *standard form*:

$$\text{maximize } c^T x, \text{subject to}$$

$$Ax = b$$

$$x_i \geq 0, \; i = 1, \ldots, n$$

Lets say the A is a matrix with n columns and d rows (so the dimension of the primal space is n). Geometrically, the affine subspace defined by the intersections of the hyperplanes $Ax = b$ intersects the positive orthant in an $(n-d)$-dimensional polytope, over which we maximize. The maximum is achieved at a vertex, which lies in the d hyperplanes $Ax = b$ and in $n-d$ of the coordinate hyperplanes $x_i = 0$. The maximum is determined by the d coordinate hyperplanes which are *not* involved in the intersection, which select some d columns of A. We'll call the standard form the *primal*.

The *simplex algorithm* was introduced by Dantzig in 1951 [D51]. The simplex algorithm begins at some vertex of the feasible polytope [1], and finds the maximum vertex by "walking" along a path of adjacent feasible vertices, increasing the value of the objective function at each step. Let H be the family of coordinate hyperplanes. A vertex is determined by the subset $V \subseteq H$ of coordinate hyperplanes that do not meet at the vertex, with $|V| = d$. Define a function $u(V)$ on a subset of coordinate hyperplanes to be the value of the objective function at the vertex, if V defines a feasible vertex, and $-\infty$ otherwise. Using this (somewhat nonstandard) terminology, we can give this very high-level description of the simplex algorithm:

choose a feasible vertex V
while $\exists h \in V$ and $\exists h' \in H$ such that $u(V - h + h') > u(V)$
    do $V = V - h + h'$
return V

Each iteration of the **while** loop is called a *pivot step*. At any time there may be many allowable pivot steps, and a simplex algorithm must provide a rule for choosing which to take. So it is more precise to speak of the *family* of simplex algorithms.

The linear programming *dual* of the standard form is:

$$\text{minimize } b^T y, \text{subject to}$$

[1] The problem can always be extended to one which has a vertex which is guaranteed to be feasible. It will be easy to see this in the dual.

$$A^T y \geq c$$

Here, geometrically, we are minimizing over the full-dimensional polytope whose facets are the inequalities $A^T y \geq c$, and the minimum is determined by d halfspaces, which we shall call a *basis*. We shall also call the d halfspaces determining the minimum of any subproblem defined by $G \subseteq H$ a basis. This is the geometric form in which we introduced the linear programming problem.

Any vertex of the arrangement in the dual corresponds to some subset of d coordinate planes in the primal, since a subfamily of rows of A in the dual is a subfamily of columns in the primal. But notice that not every vertex in the dual arrangement is a basis, just as not every d coordinate planes define a feasible vertex of the polytope in the primal. In fact, a basis in the dual problem corresponds to a feasible vertex in the primal, and the value of the dual objective function at the basis is the value of the primal objective function at the vertex.

Let's look at this idea in a little more detail. We know that the minimum of the dual problem is the maximum of the primal; this is the Strong Duality Theorem of linear programming. Any other basis in the dual is the minimum of another linear program with fewer constraints. Removing a constraint in the dual corresponds, in the primal, to fixing the variable corresponding to the omitted constraint to zero, or, geometrically, restricting the problem to the facet of the feasible polytope supported by the corresponding coordinate hyperplane. We know there is such a facet because the restricted primal problem is feasible, since the new dual problem is bounded. So every basis in the dual corresponds to a feasible vertex in the primal, and again by the Strong Duality Theorem, the values of the objective functions are the same.

In the primal, two vertices are *adjacent* if they differ in one constraint. Similarly in the dual, two bases are adjacent if they differ in one constraint. Most of the GLP algorithms walk along a path of adjacent infeasible bases. During this walk the value of the objective function increases, until finally the algorithm reaches the single feasible basis. A walk on increasing adjacent bases in the dual is exactly the same computation as a walk on increasing adjacent feasible vertices in the primal. Any such algorithm *is* a simplex algorithm, when applied to linear programming; we are just watching its progress in the dual. Same computation, different graphics.

This dual point of view seems simpler in many ways. For instance, we can add a bounding box to the dual problem, so that any subset of constraints has a feasible basis, which may include some of the constraints from the box. We can then start the walk from the basis of the bounding box. This corresponds to the somewhat more abstract idea of adding variables to the primal to ensure a feasible vertex from which to start.

More importantly, the dual viewpoint is more useful than the primal for extending the simplex algorithm to convex and other non-linear problems. Using the simplex algorithm for non-linear problems is an old idea, and examples of such algorithms can be found in nonlinear programming textbooks [BS79], [F87]. Let's consider the simple case of minimizing a convex function over a polytope. The minimum might lie in any face of the feasible polytope, not only at a vertex. The primal-simplex algorithm becomes rather clumsy, since it keeps track of the coordinate planes in which the current point does *not* lie, and in the interior of some face, there may be as many as n of these. A better idea is an *active set algorithm*. This is a primal simplex algorithm which keeps track of the coordinate planes in which the current point *does* lie. The size of the active set can be as great as $n - d$, which is reasonable only when d is not much smaller than n.

We don't have to make any such modifications to the dual-simplex algorithm, since there we always keep track of of the constraints whose boundaries contain the current point. If again we minimize a convex function over a polytope, the minimum point of the feasible polytope for any subfamily of constraints will lie on some face, and hence be contained in at most d constraints. The amount of combinatorial information needed remains bounded by d. Since the Strong Duality Theorem of linear programming carries over to convex programming (although perhaps not to all GLP problems), we can find the solution to a problem given in the primal by solving the dual.

We can use Sharir and Welzl's GLP framework to give a generalized version of the simplex algorithm which applies to any GLP problem. As usual, (H, w) is a GLP problem of combinatorial dimension d, and we assume a function *basis* which returns a basis for a subset of at most $(d + 1)$ constraints. We also assume that we can find an arbitrary basis at the beginning.

choose any basis B
while $\exists h \in H$ such that $w(B+h) > w(B)$
  do $B = basis(B+h)$
return B

Applied to any GLP problem, this generalized simplex algorithm returns a basis for H, since for any basis B with $w(B) < w(H)$, there is some $h \in H$ with $w(B+h) > w(B)$. And it terminates, since $w(B)$ increases with each pivot step. When applied to a linear program, this is just the standard simplex algorithm, since the function *basis* will always swap h into the basis. Since we have not specified how to choose the pivots, all we can say about the running time is that it is $O(n^d t_b)$, where t_b is the time required for a basis computation, since there are $O(n^d)$ fesible bases and each feasible basis is visited at most once.

7.2 Deterministic algorithms

The first linear time fixed dimensional linear programming algorithms [M84],[D84], [C86], were deterministic and do not follow the simplex model. They all rely on finding vertical [2] hyperplanes through the intersection of a pair of constraints, which makes them difficult to generalize to non-linear problems. Megiddo and others have generalized his approach to specific non-linear problems [M89]. Dyer [D92] gave a deterministic algorithm for the general class of non-linear fixed dimensional problems which have an arbitrary number of linear constraints and only a constant number of convex constraints. In particular, a problem with linear constraints and a convex objective function can be expressed in this form.

The linear programming algorithms of Megiddo, Dyer and Clarkson had running times of $O(2^{2^d} n)$ [M84] ,$O(3^{d^2} n)$ [D84] , and $O(3^{d^2} n)$ [C86] , respectively. Although

[2] parallel to the direction of optimization

these algorithms are linear in n, the dependence on d is usually unacceptable in practice. Another deterministic algorithm is derived from Clarkson's algorithm, below.

7.3 Clarkson's algorithm

Clarkson's randomized algorithm [C90] improved the running time by separating the dependence on d and n. [3] He used a three-level algorithm, with a "base-case" algorithm at the lowest level solving subproblems of size $6d^2$.

The higher two levels reduce the problem to smaller problems using the following idea. Take a sample $R \subseteq H$, find a basis B' for R by calling the next lower level algorithm, and then find the subset $V_R \subseteq H$ of constraints which violate B'. A simple consequence of the GLP framework is

Fact 7.3.1 *V_R contains at least one constraint from any basis B for H.*

The purpose of the top level is to get the number of constraints down so that we can apply the second level, which is more efficient in d but less efficient in n. We take a random sample R, with $|R| = d\sqrt{n}$ so that $E[|V_R|] = O(\sqrt{n})$, that is, we take a big random sample which gives an expected small set of violators. We iterate, keeping the violators in a set G, and finding a basis B' for $(R+G)$. At every iteration we add the violators to G, so that after d iterations G contains a basis B for H and $E[|G|] = d\sqrt{n}$. Solving the subproblem on G then gives the answer.

All recursive calls from the first level call the second level algorithm, which uses small random samples of size $6d^2$. Initially the sample R is chosen using the uniform distribution, but then we double the weights of elements in V and iterate. Since at least one basis element always end up in V, eventually they all become so heavy that $B \subseteq R$. The analysis shows that the expected number of samples before $B \subseteq R$ is at most $6d \lg n$. Since we need $O(n)$ work at each iteration to compare each constraint with the basis B'

[3] I benefit in this section from the insights in an enlightening lecture by Mike Luby and a graceful survey by Emo Welzl [W93].

of R, without the first phase this algorithm alone would be $O(n \lg n)$. All the recursive calls from this reweighting algorithm are made to some "base-case" algorithm.

The running time comes to $O(d^2 n + d^4 \sqrt{n} \lg n + d^2 (\lg n) s(d^2))$, where $s(d^2)$ is the time required for a call to the base-case algorithm with $O(d^2)$ constraints. So the algorithm is not polynomial in d only because s is not a polynomial function. We can see Clarkson's algorithm as a tool for reducing a GLP problem with many constraints to a collection of small problems with few constraints.

Clarkson noted that his algorithm could also be applied to Smallest Enclosing Ball, and, interestingly, to integer programming in fixed dimension.

The first level algorithm, if it called itself recursively instead of the second level algorithm, would be a simplex algorithm (at least if we add the specification that the current basis B' be included in G at every step). The second reweighting algorithm is not, since the current vertex does not change at each iteration. We can see the second level algorithm as a series of experiments, which eventually finds a short sequence of pivot steps from the current vertex to the top.

Clarkson's algorithm can be derandomized [4] to give a deterministic algorithm for a broad subclass of GLP problems. Let $\mathcal{B}$ be the family of bases for a GLP problem (H, w), let V_B be the violators of basis $B \in \mathcal{B}$, and let $\mathcal{V} = \{V_B \mid B \in \mathcal{B}\}$. Chazelle and Matoušek [CM93] show that if the set system $(H, \mathcal{V})$ has constant VC-dimension D (see Section 2.4), then Clarkson's random sample R can be replaced by a deterministicly constructed sample R' (called an ϵ-net), such that $|V_{R'}| \leq E[|V_R|]$. Unfortunately, as we observed before, there are set systems with finite Helly number which have infinite VC-dimension, and as a result this algorithm does not apply to all GLP problems. Constructing R' requires $(O(D)^{3D} r^{2D} \lg^D Dr)n$ time, where D is the VC-dimension of $(H, \mathcal{V})$ and $r = \frac{|H|}{|V_{R'}|}$, for the desired $|V_{R'}|$, so that even when r is fixed, the constant on n is usually unacceptably large.

[4] that is, the random sample can be replaced by a deterministicly chosen one.

7.4 Seidel's algorithm

As originally presented, Seidel's algorithm relied heavily on the geometry of (dual) linear programming. The idea was to select a random halfspace $h \in H$, and recursively find the minimum point x in $H - h$. If $x \in h$, then output x. If not, the true minimum must lie on the boundary of h, so we find it by solving the $(d-1)$-dimensional linear programming problem within that hyperplane. These recursions bottom out when either H is empty or the minimum is constrained to lie in d hyperplanes, at which point we just take the intersection.

Notice [5] that it is not really necessary to construct each $(d-1)$-dimensional problem. Instead, let the recursive call include a list of hyperplanes in which the current minimum must lie, called the *tight constraints*. Each successive minimum that we actually compute in the base case is the minimum of a problem consisting of just tight constraints. Looking at it this way frees the minimum from actually having to lie on the boundaries of the tight constraints, and makes this a GLP algorithm. The only condition on the problem is that if a constraint h violates a basis B of a subfamily G, then any basis B' for $G + h$ has $h \in B'$, and this follows from the GLP framework.

The analysis of Seidel's algorithm is extremely simple. The probability that we begin by removing an element of the basis is at most d/n. If we do have to do a recursive call on h, then there are at most $d-1$ remaining basis elements to find. So the expected running time satisfies

$$T(n,d) \leq T(n-1,d) + \frac{d}{n}T(n-1,d-1) + O(1)$$

which is $O(d!n)$, that is, $O(2^{d \lg d}n)$.

7.5 Sharir and Welzl's Algorithm

Notice that Seidel's algorithm is not exactly a simplex algorithm, since the minimum might go down at the beginning of a recursive call. The algorithm of Sharir and

[5] As Seidel and Mike Hohmeyer did, while implementing the algorithm.

Welzl is similar to Seidel's algorithm, except that it *is* a simplex algorithm. Again we remove a random constraint h, and then recursively find a basis B for $H - h$. But now, if h violates B, we solve the problem recursively starting from a basis for $B + h$. Although the statement of the algorithm does not include a set of tight constraints, you can show, using the GLP framework, that any basis found in the recursive call will include h. So, as in Seidel's algorithm, the dimension of the problem is effectively reduced. They call this the "hidden dimension" of the recursive call.

Matoušek, Sharir and Welzl gave a careful and complicated analysis of this algorithm and showed that it requires expected $O(e^{O(\sqrt{d \ln n})})$ calls to the *basis* subroutine on subproblems with $d+1$ constraints, and expected $O(ne^{O(\sqrt{d \ln n})})$ violation tests. Since, for linear programming, both the violation test and the basis computation can be performed in time polynomial in both n and d, this gives a subexponential algorithm for linear programming. Using this as the base-case algorithm at the third level of Clarkson's algorithm gives expected $O(e^{O(\sqrt{d \ln d})} \lg n)$ basis computations and expected $O(d^2(n + e^{O(\sqrt{d \ln d})}))$ violation tests. Let t_b be the time required for a basis computation and t_v be the time required for a violation test. When d is constant, the running time of the combined algorithm is $O(t_v n + t_b \lg n)$. We will use this expression in the analysis of the running times of many of our applications.

7.6 Kalai's algorithm

Matoušek, Sharir and Welzl were inspired to reexamine the analysis of the algorithm of Sharir and Welzl in response to the discovery of a subexponential randomized simplex algorithm by Kalai [K92]. Like any simplex algorithm, we can generalize this to a GLP algorithm.

Kalai actually gives three pivot rules, one of which is a variant of Sharir and Welzl's. He writes:

> Starting from a vertex v of the feasible polyhedra, choose a facet F containing v at random, find the top vertex w in F (using the algorithm recursively), set $v =: w$ and repeat.

In dual-simplex language, this rule is to start with some basis B, remove a random constraint h from $H - B$, and find a basis B' for $H - h$ recursively, starting from B. We repeat only if h violates B' (since otherwise we are done), in which case we find a basis for H recursively, starting from B'. The difference between this and the algorithm of Sharir and Welzl is that Kalai starts the recursion from B, rather than $basis(B + h)$. In a non-degenerate linear program, the only possible next step is from B to $basis(B + h)$, since (thinking about this in the primal for once) B is at the top of a facet, and hence has $d - 1$ incident polytope edges going down and only one going up. Kalai's algorithm will discover this edge in at most $O(n)$ time, so for linear programming, the two algorithms are virtually identical, at least in the general dimensional case where the extra $O(n)$ factor is unimportant. To generalize Kalai's algorithm to GLP, we replace the the pivot steps with calls to $basis(B + h)$, which makes his algorithm and that of Sharir and Welzl completely identical.

Chapter 8

Subexponential Algorithms

We now turn out attention to the question of which of the GLP problems can be solved in expected subexponential time. Recall that [MSW92] showed that their algorithm runs in expected $O(e^{O(\sqrt{d \lg n})})$ time for linear programming by showing that it requires that many basis computations and violation tests. This is fine for linear programming, since a basis computation can be done in $O(d^4)$ time (brute force). We use the fact that the minimum must lie at a vertex, so all we have to do is compute the value of the objective function at each of the vertices. This is not true for all GLP problems, of course; all we know in general is that $basis(B, h)$ includes h, when h is a constraint violating a basis B. The minimum may be determined by any size subfamily of constraints, and not every minimum determined by a subfamily is feasible. For many GLP problems we know of no better way to find a basis for a subfamily $B + h$ of size $d + 1$ than to test each of its proper subsets which contain h. In this case the basis computation takes $\Omega(2^d)$ time, so the overall running time of the algorithm has to be exponential in d.

Gärtner [G92] shows that the basis computation for the following problem requires expected subexponential time.

Problem: Minimum Separation Distance

Input: *Two polytopes, A and B, given as families of vertices.*

Output: *The minimum distance between any pair of points (a, b), where $a \in A$ and $b \in B$.*

Since, as Gärtner points out, Smallest Enclosing Ball is reducible to Minimum Separation Distance, this gives a subexponential algorithm for the classic GLP problem as well.

In this chapter we explore how far these results generalize, in mathematical programming terms. We show that, using Gärtner's approach, the *basis* function can be computed in subexponential time for any problem which can be formulated as the minimization of a smooth, strictly convex function over the intersection of a family of halfspaces. We call this problem *Convex Linear Programming*, or CLP. On the other hand, his approach does not generalize, at least immediately, to general convex programming.

8.1 Gärtner's abstract optimization problem

Gärtner [G92] considered an *abstract optimization problem* which takes as input a set H of constraints and a function $\phi : 2^H \rightarrow S$, where S as usual is a totally ordered set. Unlike GLP, there are no conditions on Gärtner's (H, ϕ). He assumes that the following oracle can be implemented in polynomial time:

Improvement Oracle
Input: (H, F)*, where* $F \subseteq H$
Output: $G \subset H$*, with* $\phi(G) < \phi(F)$*, if such an* G *exists, and* F *otherwise.*

Using calls to this oracle, Gärtner gives a randomized subexponential time algorithm to find the subfamily $B \subseteq H$ which minimizes ϕ. What does this have to do with finding a basis in a GLP problem?

Definition 8.1.1 *Let* (H, w) *be a GLP problem.* $\phi : 2^H \rightarrow S$ *is an* abstract objective function *for* (H, w) *when the* $B \subseteq H$ *that minimizes* ϕ*, over all subfamilies of* H*, is a basis for* (H, w)*,*

So Gärtner's algorithm finds a basis when ϕ is an abstract objective function for (H, w).

8.2 The abstract objective function

Every (H, w) has some abstract objective function, for instance

Definition 8.2.1 *The* trivial abstract objective function ψ *assigns the special maximal symbol* Ω *to every subfamily of* H *except for one, which is a basis for* H.

Implementing an improvement oracle for this trivial function is equivalent to finding a basis. Using the structure of a mathematical program, we can define an abstract optimization function ϕ which is sometimes *not* trivial. Like w, ϕ is defined in terms of w'.

Definition 8.2.2 *Let* (X, H, w') *be a mathematical program, where all of the* $h \in H$ *are closed sets. For any subfamily* $F \subseteq H$*, let* m_F *be the minimum point, with respect to* w'*, in the intersection of the* boundaries *of the constraints in* F*, or* $\emptyset$ *if the boundaries fail to intersect. If* m_F *is a feasible point, with respect to every constraint in* H*, let* $\phi(F) = w'(m_F)$*. Otherwise let* $\phi(F) = \Omega$*.* ϕ *is the* primal objective function *for* (X, H, w') [1].

Notice that m_F is usually different from the minimum point m'_F in the intersection of the constraints.

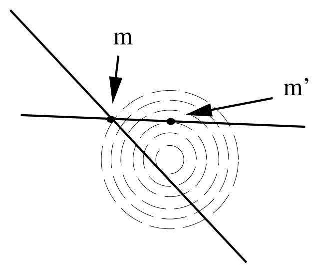

Figure 8.1: $m_{\{a,b\}} \neq m'_{\{a,b\}}$

For instance, in figure 8.1, $m_{\{a,b\}}$ is the *only* point in the intersection of the boundaries, and it differs from $m'_{\{a,b\}}$ which is the minimum point in $\bigcap\{a, b\}$. Notice that

[1] Just as w encapsulates the behavior of the dual simplex algorithm on subfamilies of constraints, ϕ encapsulates the action of the primal simplex algorithm.

there may be many subfamilies $F \subseteq H$ such that m_F is feasible.

Observation 8.2.1 *When (X, H, w') is a convex program, where all of the $h \in H$ are closed sets, the primal objective function ϕ is an abstract objective function.*

The subset which minimizes ϕ is a basis for (H, w), since the minimum feasible point in a convex program is the minimum point in the intersection of some (possibly empty) subfamily of constraints.

For CLP, ϕ is always non-trivial because the feasible region is a polytope, and the subfamily V of constraints at any vertex of this polytope will have $\phi(V) \neq \Omega$. However,

Theorem 8.2.1 *Given any $0 \leq i \leq d$, there is a convex program (E^d, H, w') with a basis of size i, such that the primal objective function ϕ is trivial.*

Proof: We construct a convex program, using a linear function w', and a family H of $n \geq d+1$ balls. Arrange i little balls so that the minimum of their intersection is the minimum of the intersection of their boundaries, and this point is not the minimum of the intersection of the boundaries for any subfamily of size $j < i$. Arrange the remaining $n - i$ large balls so that they all contain the i little balls. Then the only subfamily F for which m_F is a feasible point is the family of small balls.

■

The figure provides an example.

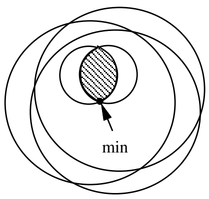

Figure 8.2: Only the the minimum is feasible

Since we know of no non-trivial abstract objective function for convex programming, we are not in a position to apply Gärtner's algorithm. For CLP, however, the primal objective function is non-trivial, so the next step is to give an algorithm for the improvement oracle.

8.3 A subexponential primitive for CLP

We now focus on the CLP problem. Let P be a polytope, given as a family H of n halfspaces, and w' a smooth strictly convex function on E^d. We need to assume some computational primitive for the oracle itself. We assume

Subroutine: min
Input: $F \subseteq H$.
Output: *The point* m_F *and the value* $w'(m_F)$.

For reasonable functions w' this subroutine requires polynomial time.

Theorem 8.3.1 *There is an improvement oracle for* d*-dimensional CLP, on inputs of the form* F, H *with* $F \subset H$ *and* $|H| \leq d+1$*, which requires* $O(n^2)$ *calls to min, where* $n = |H|$.

Proof: The input to the oracle is a subfamily F, with $|F| = d - k$, $0 \leq k \leq d$. We first call *min* to find m_F. If m_F is infeasible, return any subfamily defining a vertex of the feasible polytope. Since $|H| \leq d+1$, we can find such a subfamily in $O(d^3)$ time. Otherwise, let the k-flat f_h be the intersection of the boundaries of the constraints in F. m_F lies in some face f of of the feasible polytope P, a subset of f_h. The face f lies in $d-k$ polytope faces of dimension $k+1$, each of which is supported by an affine subspace of dimension $k+1$, which contains and is divided by f_h. Call *min* on the subfamilies defining these supporting subspaces, and get the minimum for each of them. Select any $k+1$ dimensional face - call it g - for which the minimum m_G of the supporting subspace g_h lies in the same side of f_h as the polytope.

Claim: If there is no such supporting subspace, then m_F is the global minimum. Consider the $w'(m_F)$-level set of w'. This is some smooth strictly convex body k_F, which touches f_h in the single point m_F. So f_h is tangent to k_F, and lies in the tangent plane τ at m_F.

If there is a separating plane which puts P in one closed halfspace and k_F in the other, it must go through m_F. If not, then P and k_F must intersect in some other point besides m_F. In this case some $(k+1)$-face g adjacent to f must intersect the closed halfspace of

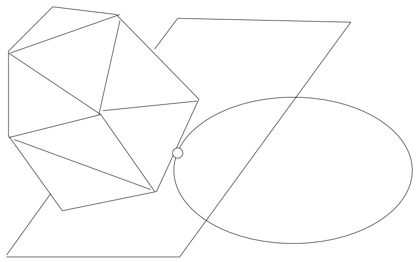

Figure 8.3: When a separating plane exists

τ containing k_F. Consider the situation in a tiny neighborhood around m_F. There k_F is approximated by τ, so g also intersects k_F. Therefore the minimum of w' within g_h lies on the same side of of f_h as P. This establishes the claim.

We will return the minimum of w' over g as the answer to the oracle. Let G be the subfamily of constraints supporting g, and m_G the point of g achieving the minimum. Connect m_F and m_G by a line segment s, and find the intersection of s with every facet of g. If s fails to intersect any facet, then m_G must be in the interior of g, and we return G. Otherwise, the closest intersection point along the line segment, p' is on the boundary of g, in some facet g'. Find the minimum $m_{G'}$ in the supporting subspace g'_h. Now recur

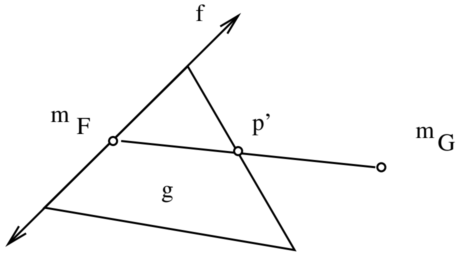

Figure 8.4: When m_G is not in g

on p',m' and g'. Since g' is always one dimension less than g, we find a smaller basis when we get down to a 0 dimensional subspace, if not before.

■

Together with Gärtner's result, this implies

Corollary 8.3.1 *A basis for a CLP* (X, H, w') *with* $|H| \leq d + 1$ *can be found in expected subexponential time using an expected subexponential number of calls to a subroutine min which takes a subfamily* F *and returns the point* m_F *and* $w'(m_F)$.

Notice that it remains possible that there is a polynomial time and/or deterministic implementation of the *basis* function for CLP problems. Another interesting open question is whether there is a subexponential *basis* function for general convex programming.

The Bounded Box problem, which we shall consider in the next chapter, is an example of CLP.

Chapter 9

Covering and filling

We now begin to work through the practical implications of the theory we have developed. We have a paradigm for constructing an objective function so that a Helly problem can be solved by a GLP algorithm. Using this paradigm with known Helly theorems produces a variety of algorithms for geometric optimization problems, most of which run in expected linear time in fixed dimension.

By *geometric optimization problem*, we mean a problem in which the goal is to select an element from a family of geometric *output objects* which is somehow optimal with respect to a finite family H of geometric *input objects*. The geometric optimization problems in this chapter will involve finding the smallest output object which somehow covers the input family, or the largest output object which somehow fills the intersection of the input family. Smallest Enclosing Ball is an example of a covering problem, and a few other covering and filling problems which were known to be GLP are discussed at the end of the chapter.

A common feature of these problems is that the ground set X is the space of output objects, so that the objective function w on subfamilies of input objects is induced (see definition 3.3.1) by some function w' on X.

9.1 Translates and homothets

As this is the first application, we treat it in great detail and address, in the process, some issues that will recur in other applications.

We use the following Helly theorem about translates of a convex object to show that a number of problems are GLP. This is theorem 2.1 from [DGK63],

Theorem 9.1.1 (Vincensini and Klee) *Let K be a finite family of at least $d+1$ convex sets in E^d, and let O be a convex set in E^d. Then there is some translate of O which [intersects/ is contained in/contains] all members of K if and only if there is such a translate for every every $d+1$ members of K.*

Recall that a *homothet* of an object is a scaled and translated copy.

Theorem 9.1.2 *Let K be a finite family of at least $d+1$ convex sets in E^d, and let O be a convex set in E^d. The smallest homothet of O which contains $\bigcup K$, or the largest homothet of O contained in $\bigcap K$, or the smallest homothet of O which intersects every member of K, can be found by a GLP of combinatorial dimension $d+1$.*

Proof: We consider carefully the problem of finding the largest homothet of O contained in $\bigcap K$; the others are analogous, and also similar to Smallest Enclosing Ball. Informally, the idea is to construct an objective function for the problem by "shrinking" the enclosed homothet with "time".

We pick a distinguished point in O. A translate of O can then be expressed as a translate of this distinguished point. So it is easy to see that the set of translates of O which lie inside a convex body k form a convex set.

The theorem of Vincensini and Klee, above, can be expressed as a Helly system (T, H), where T is the set of translates of O and each $h^k \in H$ is the set of translates which are contained in a particular convex set $k \in K$. We use the scale factor $\lambda \in \mathcal{R}^+$ to produce a parameterized Helly system $(T \times \mathcal{R}^+, \overline{H})$. Each $\overline{h}^k \in \overline{H}$ is the set of all homothets of O which are contained in k. Each h^k_λ is the set of homothets of $-\lambda O$, O scaled by $-\lambda$, which are contained in k. We use $-\lambda$ rather than λ so that we can find the largest homothet by minimizing $-\lambda$, which is consistent with the GLP framework. Notice that $\overline{h}^k$ is a cone over a convex set, and hence is always convex.

$(T \times \mathcal{R}^+, \overline{H})$ is a valid example of a parameterized Helly system, according to definition 5.1.2, since $h^k_\alpha \subseteq h^k_\beta$ for $\alpha < \beta$, and each (X, H_λ) is a Helly system. So it has a natural ground set objective function w', which is just the scale factor λ, and an induced natural objective function w on subfamilies of $\overline{H}$.

Now to show that $(\overline{H}, w)$ is GLP, we have to ensure that it meets the conditions of the Main Theorem. To guarantee that every $w(\overline{G})$ is well defined, we add a big bounding box as an implicit constraint to every subfamily $\overline{G}$, so that there is always a largest homothet in $\bigcap G$. And to ensure the Unique Minimum Condition, we use a lexicographic objective function v, with λ as the most significant parameter, followed by the translation coordinates in any order. This is equivalent to a perturbation of the linear objective function (see Section 6.2), and since the constraints $\overline{h}^k$ are convex , Theorem 6.2.1 implies that the combinatorial dimension is $d + 1$.

■

All of the problems treated by Theorem 9.1.2 are special cases of convex programming, so we don't really need the Main Theorem to show show that they are GLP. Presumably anyone tinkering with one of them for long enough would find the convex programming formulation. But since we already have the relevant Helly theorem, the convex formulation becomes obvious.

9.2 Simple objects

We have not said anything yet about the running time of the GLP algorithms on the three problems above. That is because it depends on how efficiently we can perform the basis computations and violation tests, which in turn depends on the representation and complexity of the convex objects in the particular problem instance. This issue will arise again, so we consider it in general before we return to the specific applications.

In order to state our algorithmic results in their full generality, we resort to the traditional trick (eg. [GPW93] page 188, [EW89]) of defining the objects we can use in terms out our computational requirements. Recall that in Chapter 7 we said that the combined algorithm from [MSW92] runs in time $O(t_v n + t_b \lg n)$, where t_v is the time required to perform a violation test and t_b is the time required for a basis computation. This is $O(n)$ when $t_v = O(1)$ and $t_b = O(n/\lg n)$. We define a *simple* object to be one which has a constant-size representation and for which we can compute the primitive operations within these time bounds. So a GLP problem involving convex sets in fixed dimension requires expected linear time when the convex sets are simple.

9.3 Homothet covering problems in the plane

We consider a particular homothet covering problem in which we can give a better running time than was previously known.

As far as I can determine, only planar versions of any of the homothet covering and filling problems have been considered. There is an algorithm for a dynamic version of the problem of maintaining the smallest homothet of $\mathcal{P}$ containing K in [BEI93]. There is an $O(k + n \lg^2 n)$ algorithm in [T91] for finding the largest homothet of a k-vertex polygon P inside an n-vertex polygon Q in E^2. This problem arises as a restriction of a pattern matching problem; we can think of Q as input from a low-level vision system and P as a model of an object we'd like to match. This is a restriction of the problem because we allow P to scale and translate, but not to rotate.

For this problem of finding the largest homothet of a polygon in another polygon, the GLP formulation, after preprocessing, is just linear programming. We use the halfspaces supporting Q as constraints. Let us identify translates of P with translates of an arbitrary distinguished point $q \in P$. For a fixed scale factor λ, the possible translates which put P on the correct side of some halfspace h lie in a halfspace h' parallel to h and offset by the distance from q to some vertex v of P.

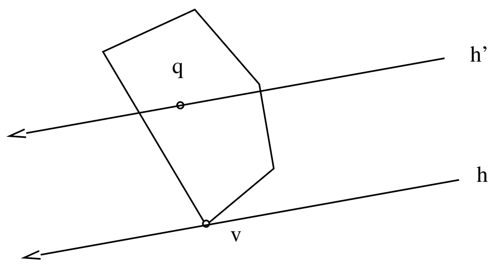

Figure 9.1: Constraint h'

In the plane, we can assign the correct vertex v to each halfspace h in $O(n+k)$ time by merging the lists of face normals for P and Q, as first observed in [C83]. As we increase the scale factor λ, the distance from q to v changes linearly, and h' traces out a three-dimensional halfspace in $\lambda \times E^2$. The linear program maximizes λ over the intersection of the h' in $O(n)$ time, so the total time is $O(n+k)$.

9.4 Minimum Hausdorff distance

Now we turn our attention to a similar covering problem, which has received a lot of attention in the literature, and show that a special case is much simpler than the general problem. Let's say that the distance between a point $x \in E^d$ and a body B is $dist_1(x, B) = \min\{dist_0(x, y) | y \in B\}$, where $dist_0(x, y)$ is the usual Euclidean metric. The distance from a body A to B is

$$dist_2(A, B) = sup\{dist_1(x, B) \mid x \in A\}$$

and the *Hausdorff distance* between A and B is $max\{dist_2(A, B), dist_2(B, A)\}$. The Hausdorff distance between two bodies is one measure of the difference between their shapes, and is used in pattern recognition and computer vision. We can consider applying a group of transformations to one of the bodies, and ask for the the transformation which minimizes the resulting Hausdorff distance. There has been a lot of work in computational geometry on computing the minimum Hausdorff distance between objects of various sorts, under various groups of transformations. For the problem of finding the minimum Hausdorff distance between two sets of k points in the plane under translation and rotation, [HKK92] give an $O(k^6 \lg k)$ algorithm. For simple polygons, an approximation algorithm for the minimum Hausdorff distance under translation and rotation appears in [ABB91]. For convex polygons fixed in the plane, [A83] just computes the Hausdorff distance in time $O(n)$.

Surprisingly, none of these algorithms consider scaling. Also, there are no results in higher dimensions or using metrics other than L^2 as $dist_0$.

We consider the problem of finding the minimum Hausdorff distance under translation and scaling of two convex polytopes A and B in E^d, using any quasimetric as $dist_0$. We show that this can be formulated as finding the minimum point in the intersection of a family of convex constraints. For polygons in the plane, this leads to an expected linear time algorithm.

The set of points in E^d at distance at most λ from a polytope A form the λ-offset surface, which we shall write $F^\lambda(A)$. The figure shows an offset surface for a polygon in the plane. $F^\lambda(A)$ is the Minkowski sum of A with the unit ball of the L^2 metric, scaled

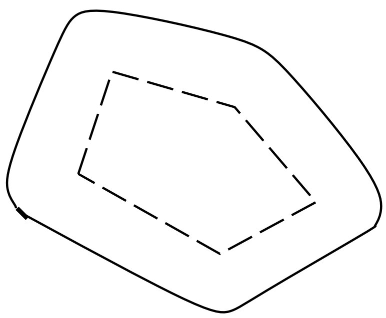

Figure 9.2: Offset surface

by λ. We can replace the sphere with any other convex body c. Call the resulting offset surface $F_c^\lambda(A)$. Since c and A are convex, $F_c^\lambda(A)$ is convex. Using the quasimetric whose unit ball is c as $dist_0$ in the original definition of the Hausdorff distance gives us a more general function which we shall call the *c-Hausdorff distance.*

Theorem 9.4.1 *Let c be a convex body in E^d, and let A and B be polytopes, with vertex sets $V(A)$ and $V(B)$ respectively. The minimum c-Hausdorff distance, under scaling and* *translation, between A and B is no greater than λ, if and only if, for every family of points $P_A \cup P_B$, with $P_A \subseteq V(A)$, $P_B \subseteq V(B)$ and $|P_A \cup P_B| \leq d+2$, there is a translation and scaling such that $p_A \in F_c^\lambda(B)$ for all $p_A \in P_A$, and $p_B \in F_c^\lambda(A)$ for all $p_B \in P_B$.*

Proof: Consider the *transformation space* in which each point is a $(d+2)$-tuple consisting of a translation vector in E^d, a scale factor, and an offset λ. $F_c^\lambda(A)$ is convex at a fixed λ, so for a fixed vertex $p \in V(B)$, the set of transformations which produce homothets of $F_c^\lambda(A)$ containing p is also convex. Call this set of transformations h_p^A, and define h_p^B analogously for $p \in V(A)$. The family $\{h_p^A | p \in V(B)\} \cup \{h_p^B | p \in V(A)\}$ is a finite family of convex sets in E^{d+1}, and so has Helly number $d+2$.

■

This theorem corresponds to a Helly system (X, H) where X is the transformation space and H is an infinite family of subsets of X, with one element for every point in either A or B. When A and B are polytopes, we can reduce H to a finite family, since a polytope is contained in a convex set if all of it's vertices are. We can use this Helly system to define a GLP problem, much as we did in the previous application. Informally, we grow the offset distance with time, to produce a nested family of constraints.

Theorem 9.4.2 *The minimum c-Hausdorff distance, under scaling and translation, between two polytopes A and B in E^d can be found by a GLP of combinatorial dimension $d+2$.*

Proof: In this case we construct an objective function by "growing" the offset with "time". We parameterize the Helly system (X, H) by the offset λ, giving a parameterized Helly system $(X \times \mathcal{R}, \overline{H})$. Each $\overline{h}$ is a nested family because the α-offset surface of a convex body is contained in the β-offset surface if $\alpha < \beta$. In fact each $\overline{h}$ is a convex subset of $X \times \mathcal{R}$, since it is the Minkowski sum of a cone over the convex body c that defines the distance function, with the convex constraint $h \in H$. The translation and scaling which achieves the minimum Hausdorff distance is in general unique; degeneracies can be eliminated by using the lexicographic objective function $< \lambda, s, \tau >$, where s is the scaling dimension and τ is the translation, without increasing the combinatorial dimension, by Theorem 6.2.1. The minimum value of λ is determined by a basis of size $d+2$, by the

Main Theorem.

■

The general dimensional GLP of Theorem 9.3.1 is not very useful since the constraints are not simple. In the plane, we can improve this so that we have a linear number of simple constraints.

Theorem 9.4.3 *The minimum c-Hausdorff distance between two convex polygons A and B in the plane, under scaling and translation, can be found in expected $O(n)$ time, where A and B have a total of n vertices.*

Proof: We break the offset surfaces up into pieces of constant complexity. Each piece consists of the offset surface of a vertex of one of the polygons, plus the rays supporting the adjacent sides, which we shall call an *angle*. If every vertex of A is within the λ-offset surface of every angle from B, and visa versa, the Hausdorff distance is no greater than λ. This gives a GLP with $O(n^2)$ constraints, since we have to pair every vertex of A with every angle from B, and visa versa.

We get a linear number of constraints by noting that for an angle α from B, there is a *critical subset* V of vertices from A such that A is within the λ-offset surface of α if every vertex from V is. Furthermore, every vertex of A is in at most two such sets,

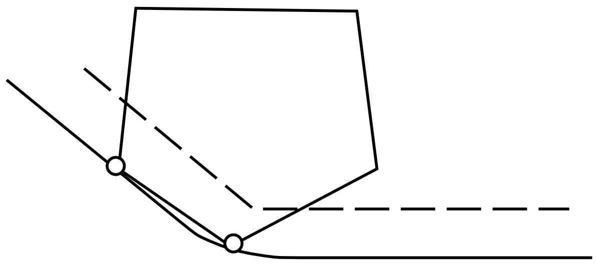

Figure 9.3: Critical subset

and the critical subsets for every angle of B can be found by merging the circular lists of face normals of A and B. This all applies, of course, to angles from A and vertices from B as well.

There is one constraint in our GLP for each vertex in the critical subset of each angle from either A or B. Since the constraints are simple, this gives an expected linear time algorithm.

■

These arguments can also be applied to versions of the problem in which we restrict the groups of transformations or compute the one-way Hausdorff distance from a convex body A to a convex body B, or even the one-way Hausdorff distance from a convex body A to a family of points in E^d. The important fact in the argument, that the set of transformations which puts a point into an offset surface is convex, remains the same.

9.5 Boxes

Perhaps the most common approximating volume used in practice is the bounding box, because it is so easy to compute and to compute with. Consider the analogous notion of a *bounded box* for an object, that is, the largest volume axis-aligned box which is completely contained in the object. Like the bounding box, the bounded box can be used as an approximation of the object. [DMR93] introduce the problem of finding the bounded box in the context of a heuristic for packing clothing pattern pieces. They give an $O(n\alpha(n)\log n)$ algorithm which finds the maximum area axis-aligned rectangle in an arbitrary polygon in E^2, and cite some results for special cases of 2-dimensional polygons. There are no results that I know of in higher dimensions, although the problem may have some practical importance. Three dimensional objects are frequently decomposed into collections of axis-aligned boxes (eg. in ray tracing, [AK89],pp 219-23), which should approximate the original object as well as possible. We show

Theorem 9.5.1 *Finding the largest volume axis-aligned box in the intersection of a family of convex bodies in E^d is a convex linear programming problem in dimension $2d$.*

Proof: We need to search the space of all axis-aligned boxes in E^d, which we shall call box-space. We think of a box as a point in E^d, and a positive offset in each coordinate direction. We use the parameters $x, a \in R^d$, where $x_1, \ldots, x_d$ are the coefficients of the point, and $a_1, \dots, a_d$ are the positive offsets. Notice that box-space is therefore equivalent to the union of those orthants of E^{2d} in which the a_i are all positive.

Now we construct a convex program in box-space. A box is contained in a convex body C if and only if all of its vertices are contained in C. We add one convex constraint to our problem for each vertex. Let us label the vertices with 0-1 vectors in the natural way, so that $(0, \dots, 0)$ is the vector at the minimum corner of the box. The box-space constraint corresponding to the vertex with label u is the set $\{x, a \mid x + (u \otimes a) \in C\}$ (here $u \otimes a$ is coordinate-wise multiplication). This is a cylinder with base C, whose slope in each of the a_i directions is either 0 or 1, depending on u_i. Each cylinder is convex, so the intersection of the cylinders for all the vertices, which is the feasible region, is convex.

It remains to show that maximizing the volume of the box corresponds to minimizing some convex function over this convex feasible region. The volume, negated, is

$$-\prod_{i=1}^{d} a_i$$

with all the a_i constrained to be positive. This function is not convex, only quasiconvex [1]. However, the function

$$-\log(\prod_{i=1}^{d} a_i) = -\sum_{i=1}^{d} \log(a_i)$$

is convex. This is easy to see. The sum of convex functions is convex, and $-\log(a_i)$ is a convex function of a_i. Also, $\log(-\prod a_i)$ is minimized over any polytope exactly where $-\prod a_i$ is minimized, so solving the problem with the convex objective function gives the maximum volume box.

■

Theorem 9.4.1 implies of course that finding the largest volume box can be done with GLP, and also,

Corollary 9.5.1 *For a family H of convex bodies in E^d, $\bigcap H$ contains an axis-aligned box of volume 1 if and only if every subfamily $B \subset H$ with $|B| \leq 2d$ contains an axis-aligned box of volume 1.*

[1] By Theorem 6.4.1, this suffices to show that the problem is GLP.

We get a Helly number of $2d$ here, rather than $2d+1$, because the premise that every subfamily of size $2d$ contains a box of volume one implies that every subfamily of size $d+1$ is non-empty, so every subfamily contains *some* box and the problem is guaranteed to be feasible. This allows us to apply Corollary 4.0.2.

9.6 Extensions of known GLP problems

Finally, we mention some simple extensions of earlier results on covering and filling problems which are GLP. Recall that the deterministic fixed-dimensional LP algorithms [M84],[D84] relied on the constraints being linear. Thus it was difficult and significant for Megiddo to produce a linear-time algorithm to find the smallest ball containing a family of balls [M89], although Dyer had already given an algorithm for finding the smallest ball containing a family of points. With GLP, however, such generalizations are easy.

Say (X, C) is a Helly system with Helly number d. The intersection lemma (section 5.2) tells us that the set system (X, C'), where C' is the collection of all intersections of subfamilies of C, also has Helly number d. Finally, (X, K), where $K \subseteq C'$, has Helly number no greater than d. For example, the set of balls in E^d containing an input ball b is the intersection, over all points $p \in B$, of the set of balls containing p. So the Radius Theorem implies that balls in E^d have Helly number $d+1$ with respect to the property of being contained in a unit ball.

If we can find an objective function w for a Helly system (X, C), so that (H, w) is a GLP problem for any finite subfamily $H \subseteq C$, then usually there is an analogous function for (X, K). We have to be careful, however, since if C is infinite, the intersections which contribute elements to K may also be infinite, and the existence of w only testifies that the Unique Minimum Condition is met for *finite* subfamilies.

The following problems were listed as GLP problems in [MSW92]:

Smallest enclosing ellipsoid: Find the smallest volume ellipsoid containing a family of points in E^d,

Smallest volume annulus: Find the smallest volume annulus [2] containing a family of points in E^d,

Largest volume ellipsoid in polytope: Find the largest volume ellipsoid contained in a family of halfspaces in E^d.

The word "points" can be replaced by "compact sets" in the first two, and "halfspaces" by "closed convex sets" in the third. It is important that the sets be closed, so that there is always a unique optimal output object.

[2] an *annulus* is the region between two concentric spheres

Chapter 10

Hyperplane transversals and hyperplane fitting

A transversal of a family H of input objects is some element x of a family X of output objects, such that x intersects every $h \in H$. There are a lot of Helly theorems about transversals. By defining an appropriate objective function, we can use such a Helly theorem to construct a GLP which actually finds a transversal. These GLPs find an *optimal* transversal with respect to the objective function, which is sometimes useful. In particular we will find that we can often interpret these same algorithms as *minimax fitting* algorithms, which find some output object x which minimizes the maximum distance from an input family H of points under some interesting distance function.

In the last chapter, our objective functions were induced by a function w' on the ground set X of output objects. This corresponded to the intuitive notion of shrinking or growing an output object until it satisfied the constraints. Here, our ground set will be the product of the space X of output objects with a scale parameter $\lambda \in \mathcal{R}^+$ of the input objects. This corresponds to the equally natural notion of growing the input objects until they admit a transversal.

10.1 Transversal of translates in the plane

One of the most interesting Helly theorems is about line transversals of a family T of disjoint translates of a single convex object O in E^2. Tverberg [T89] showed that if every family $B \subseteq T$ with $|B| \leq 5$ admits a line transversal, then T also admits a line transversal. Egyed and Wenger [EW89] used this theorem to give a deterministic linear time algorithm to find a line transversal. Here, we show that the problem can be formulated as GLP, implying the existence of a number of simpler, although randomized, linear time algorithms.

This problem is interesting in relation to the theory of GLP. Recall that using our usual transformation we are working with a Helly system (X, T') where X is the space of lines in the plane, and each $t' \in T'$ is the set of lines intersecting a translate from T. X is a two-dimensional space. If there were some affine structure on X such that the constraints T' were convex subsets of X, then the Helly number of the system (X, T') would be 3. But examples show that the bound of 5 is in fact tight, which means that this is a GLP which is *not* a convex program. As we shall see, this is also a natural example of a GLP problem in which the minimal object does not "touch" every constraint in the basis.

We assume that the family of translates is in general position (we will define general position in a moment); if not, we use a standard perturbation argument. Let $S = [0, 1]$. A subfamily $G' \subseteq T'$ intersects when there is a line which intersects every translate $t \in G$, where G is the family of translates corresponding to G'. We pick a point q in the interior of the object O. For a particular translate t, let λt be the homothet of O which results from scaling translate t by a factor of λ, keeping the point in t corresponding to q fixed in the plane. Let the family $\overline{t} = \{\lambda t \mid 0 \leq \lambda \leq 1\}$, and $\overline{T} = \{\overline{t} \mid t \in T\}$.

Every line which intersects the homothet $\lambda_1 t$ also intersects $\lambda_2 t$ for any $\lambda_2 > \lambda_1$. So each $\overline{t}$ corresponds to a nested family $\overline{t}'$ of lines. At a particular scale factor $0 \leq \lambda \leq 1$, the translates t_λ are always disjoint, so every $(X, \lambda T')$ is a Helly system with Helly number 5, and $(X \times S, \overline{T}')$ is a parameterized Helly system.

The natural objective function $w(\overline{G}')$ is the minimum λ such that G'_λ intersects.

In the case where $\overline{G}' \subseteq \overline{T}'$ consists of a single translate, we define $w(\overline{G}') = 0$. Notice that for certain degenerate placements of the translates (see figure) it is possible for there to be two or even three distinct line transversals at $\lambda^* = w(\overline{G}')$.

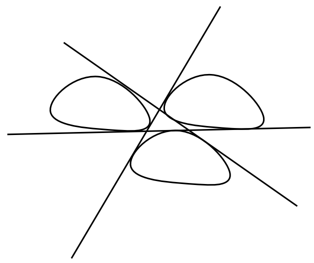

Figure 10.1: Degenerate input

The general position assumption is that the line transversal at λ^* is always unique.

Under this general position assumption, $(\overline{T}', w)$ is a GLP problem with combinatorial dimension 5, since each T'_λ has Helly number 5, a minimum λ^* which admits a line transversal always exists, and the line transversal at λ^* is unique. Either we find a line transversal at some value of $\lambda \leq 1$, or no line transversal of the input exists.

To find whether this implies an expected linear time algorithm we need to figure out how to implement the primitive operations. Here, a violation test determines whether the current minimum line m intersects a new homothet λt. For any m, there is pair of lines tangent to O and parallel to m. These lines support a pair of vertices on O, which we shall call an *antipodal pair*. The line m intersects a homothet λt if and only if m passes between the pair of vertices on λt. Figure 10.2 shows an example.

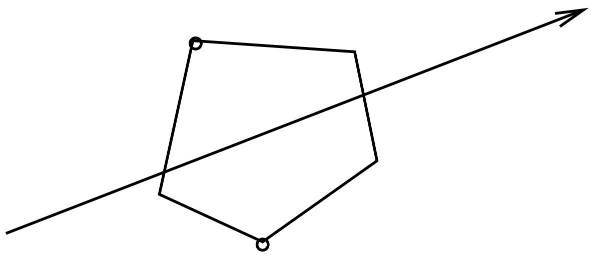

Figure 10.2: Antipodal pair

We can implement the basis computation so that whenever we find a new minimum line m we also find the corresponding antipodal pair. This lets us perform a violation test in time $O(1)$. The running time of the whole algorithm is then limited by t_b (see the end of Section 7.5); when the complexity of O is such that t_b is $O(n/\lg n)$, we get an expected linear time algorithm. So we get

Theorem 10.1.1 *A line transversal for a family of n disjoint translates of a convex object in the plane can be found by a GLP of combinatorial dimension 5, in expected $O(n)$ time.*

A point in X which lies on the boundary of one of the sets t'_λ corresponds to a line which is tangent to t_λ. We say such a line "touches" t_λ. The line transversal at λ^* is tangent to at most three translates, for any non-degenerate example. Since the size of a basis can be as great as 5, it is easy to construct situations in which the line transversal does not "touch" every element of the basis. The figure below shows an example.

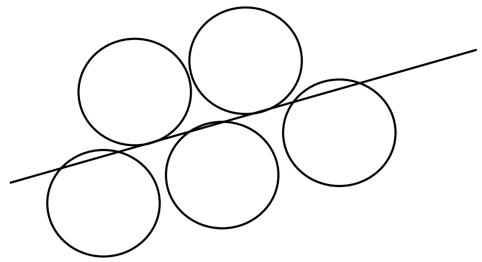

Figure 10.3: Transversal not tangent to every basis element

Consider the line transversals of the three central balls, as they grow with time. Assuming that the balls are in general position, there will be three distinct moments at which a new connected subset of line transversals becomes feasible. The two remaining balls are positioned so that only the last of these subsets contains a transversal of the whole family.

Curiously, it is not known whether the Helly theorem about line transversals of disjoint translates can be generalized to a theorem about hyperplane transversals in higher dimensions, even for the case of disjoint unit balls.

10.2 Hyperplane transversals of polytopes

There *is*, however, a Helly theorem about hyperplane transversals of polytopes in higher dimensions. This theorem is a byproduct of a reduction by Avis and Doskas [AD92] of the problem of finding such a transversal to a collection of linear programs. We will review their reduction, and then show how the same family of constraints can be used with a different objective function to give our first hyperplane fitting application.

Think of partitioning the hyperplanes in E^d into equivalence classes by partitioning the $(d-1)$-sphere of hyperplane normals by the hyperplanes through the origin normal to the polytope edges. For each equivalence class C there is an *antipodal pair* of vertices v_1, v_2 on the polytope such that a hyperplane $h \in C$ intersects the polytope if and only if $v_1 \in h^+$ and $v_2 \in h^-$. The figure illustrates that the critical values of the hyperplane normal at which antipodal pair changes are those normal to polytope edges.

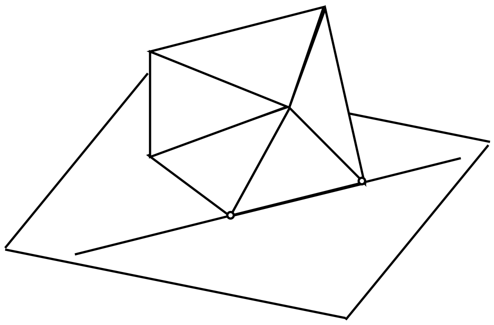

Figure 10.4: Hyperplane at critical inclination

Within one equivalence class, then, finding a hyperplane transversal for a family of translates is simply a matter of finding a hyperplane which is correctly oriented with respect to each of the pairs of antipodal points. This is the dual [1] of the problem of finding a point in the intersection of a family of halfspaces, which can be solved by linear programming. There are m^{d-1} equivalence classes, where m is the number of polytope

[1] Here we mean geometric, or projective, rather than linear programming duality: the vector inequality $a \cdot x > 0$ can be interpreted, geometrically, as saying that the point whose homogeneous coordinates are a vector a is on the positive side of an oriented hyperplane with coordinates x, or, as the equivalent dual statement that the point with coefficients x is on the positive side of the oriented hyperplane a.

edges, and the equivalence classes can be identified in $O(m^{d-1})$ time [EOS86]. The result of Avis and Doskas is that we can find a hyperplane transversal of a family of n polytopes by solving a linear program with $2n$ constraints for each equivalence class, in a total of $O(nm^{d-1})$ time, when d is a constant. By Theorem 4.0.1, we have the following

Corollary 10.2.1 *Let K be a family of polytopes in E^d with a total of m edge directions. Then the space X of hyperplanes in E^d can be divided into $O(m^{d-1})$ equivalence classes $X_1, \ldots, X_{m^{d-1}}$, such that for each class X_i, K has a hyperplane transversal $x \in X_i$ if and only if each subfamily $B \subseteq K$ with $|B| \leq d+1$ has a hyperplane transversal $x \in X_i$.*

10.3 Hyperplane fitting with the L^∞ and L^1 metrics

Once again, this Helly theorem can be expressed as a Helly system. The ambient space is X_i, and each constraint $h \in H$ is the subset of hyperplanes in X_i which pass between the antipodal points for a particular polytope. Following the example of the previous application, we parameterize this Helly system by selecting a fixed point p inside each polytope c, and then scaling each c around p. The natural objective function $w(\overline{G})$ for this parameterized Helly system $(X_i \times \mathcal{R}, \overline{H})$ returns the minimum scale factor by which the polytopes contributing constraints $\overline{h} \in \overline{G}$ can be scaled so as to admit a hyperplane transversal from the set X_i. The function w is inherently well-defined, and we use a perturbation argument to ensure the Unique Minimum Condition. The Main Theorem now tells us that $(\overline{H}, w)$ is a GLP problem. We solve this GLP for every X_i. If for *any* of these problems $w(\overline{H}) = (\lambda, x)$ with $\lambda \leq 1$, then x is a hyperplane transversal of the family of polytopes, and if not, no transversal exists. So

Theorem 10.3.1 *A hyperplane transversal for a family of n polytopes with at most m total edge directions can be found by a collection of $O(m^{d-1})$ GLPs of combinatorial dimension $d+1$ and $2n$ constraints.*

This is not very interesting, since we already know that the problem of finding a hyperplane transversal is a special case of d-dimensional linear programming, which has combinatorial dimension d. But notice that the algorithm actually solves a slightly more general problem.

Problem: Polytopal Hyperplane Fitting

Input: *A finite family* P *of pairs* (p, c_p)*, where* p *is a point,* c_p *is a polytope, and* $p \in c_p$.

Output: *A pair* (λ, x)*, where* $\lambda \in \mathcal{R}$ *is the smallest scale factor such that* $\lambda C_P = \{\lambda c_p \mid (p, c_p) \in P\}$ *admits a hyperplane transversal, and* x *is a hyperplane transversal of* λC_P.

The notation λc_p, $\lambda \in \mathcal{R}$, means c scaled by λ around p. We can now state the following stronger

Theorem 10.3.2 *Polytopal Hyperplane Fitting in* E^d*, where the family of* $C_P = \{c_p \mid (p, c_p) \in H\}$ *has at most* m *total edge directions, can be solved by a collection of* $O(m^{d-1})$ *GLPs of combinatorial dimension* $d+1$*, each with* $2n$ *constraints, where* $n = |H|$.

We can think of c_p as *measuring* the distance between p and any hyperplane h, so that $dist_p(p, h)$ is the smallest factor λ by which c_p can be scaled around p so that $\lambda c_p \cap h$ is non-empty. This is distance function induced by the quasi-metric d_p on E^d whose unit ball is c_p; $dist_p(p, h) = \min_{x \in h} d_p(p, x)$ (d_p is a quasi-metric because c_p is not necessarily centrally symmetric). The collection of GLPs in this theorem finds the hyperplane which minimizes the maximum $dist_p(p, h)$, over all $p \in P$.

Although this problem sounds obscure, it has a number of interesting special cases. The only useful cases are those in which m, the total number of polytope edge directions, is constant. This is certainly true when all the polytopes c_p are translates of a single polytope c with a constant number of edge directions. In this case the distance function is induced by the quasimetric with unit ball c, $dist_c$. This general family includes L^∞ $(m = d)$ and L^1 $(m = d^2 - d)$ [2]. Also notice that allowing the polytopes c_p to be different homothets of c is equivalent to using $dist_c$ with multiplicative weights on the points.

[2] The unit ball of the L^1 metric is the dual of a hypercube (sort of a hyper-octahedron, known as a *cross-polytope*). It has $2(d^2 - d)$ edges, but each edge is parallel to one other.

When the polytopes are translates, the antipodal pairs form two rigid families, moving in different directions. Factoring out the parallel component of their direction vectors leaves us with a special case of the simple

Problem: Point Set Separation

Input: *Two rigid finite families of points, P and Q, and a direction vector v.*

Output: *A pair (λ, x), where $\lambda \in \mathcal{R}$ is the smallest scale factor such that there is a hyperplane x separating $P + \lambda v$ (ie. the point set P translated by λv) and $Q - \lambda v$.*

This problem looks so simple that sometimes people assume that it is possible to reparameterizing it as a linear program in dimension $d+1$, especially when they notice that we can fix one family, so that the relative motion is all accounted for by the translation of the other family by $2\lambda v$. In fact it *is* possible to simultaneously linearize all the constraints, but the objective function then becomes quasiconvex. Here is the transformation.

We rotate the problem, so that v is pointing in the x_0 direction. We can then write a constraint due to a point $q \in Q$ on a hyperplane x as

$$(2\lambda + q_0)x_0 + q_1x_1 + \ldots + q_dx_d + 1 \geq 0$$

Making a new variable $\lambda' = \lambda x_0$ linearizes all the constraints, but then the objective function becomes

$$minimize\ \lambda'/x_0$$

This is quasiconvex, if we restrict our attention to hyperplanes for which x_0 is positive (see section 6.4, or [Mag69], page 149). This restriction is no problem, since x_0 is the direction in which the family of points is moving, so the separating hyperplane must have a positive x_0 coefficient anyway.

We note that this same GLP may be used to determine when a polytope P translated along a linear trajectory will become separated from a fixed polytope Q, since the two sets of vertices first admit a separating hyperplane at that moment.

10.4 The weighted L^∞ metric

The collection of GLPs in theorem 10.3.2 are also useful when each of the n points is surrounded by a *different* polytope, so long as all the edges of all the polytopes have no more than m directions total, where m is constant. In this case we cannot linearize all the constraints simultaneously, since the members of each family of antipodal points cannot be divided into two rigid sets which only move with respect to each other.

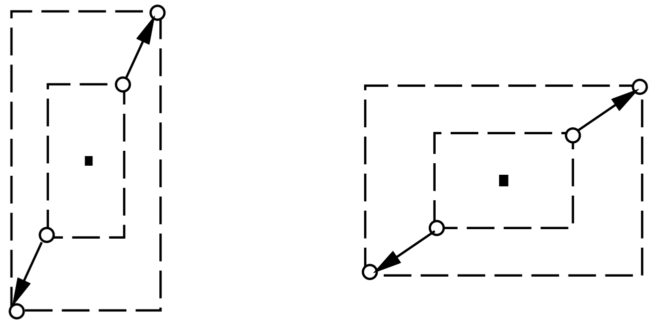

Figure 10.5: Antipodal pairs in the weighted L^∞ metric

This problem is interesting because it has an important special case. When each coefficient of each point is given a weight, each point is surrounded by a box-shaped unit ball. This situation arises when the coefficients represent heterogeneous quantities (height, age, annual income), and when different measurements of the same variable are made with different amounts of error. It also comes up when the coefficients are calculated and error is bounded using interval arithmetic, or when the unit balls of more complicated error metrics are approximated by bounding boxes. When every coefficient of every point is given the same weight, the unit ball is a hypercube; this is the L^∞ metric. When we allow the weights to vary, we call this the *weighted L^∞ metric*.

The general-dimensional version of the problem of finding the hyperplane which minimizes the maximum distance to a set of points under the weighted L^∞ metric has been considered in [R89], [D92], and in [PR92], where it is shown to be NP-hard. This is the first treatment of the fixed dimensional case, where GLP gives an expected linear time algorithm. Finding the minimizing hyperplane requires solving 2^{d-1} GLP problems (as we shall see in a moment), so the NP-hardness result has no bearing on the complexity of GLP.

The input to the problem is a pair of $n \times d$ matrices, one supplying the points in E^d and the other giving the weights. The boxes defined by the weights have d edge directions, partitioning the sphere of hyperplane normals into 2^d orthants. We only need to consider one member of each pair of opposite orthants in order to find the minimizing hyperplane, so the number of GLP problems we have to solve is 2^{d-1}. There is an antipodal pair of vertices on each box associated with each orthant, and the hyperplanes separating these antipodal pairs form the nested families. This gives us

Theorem 10.4.1 *The hyperplane which minimizes the maximum distance to a family of points in E^d under the weighted L^∞ metric can be found by 2^{d-1} GLPs with combinatorial dimension $d+1$, in expected $O(n)$ time for fixed d.*

10.5 Remarks on fitting problems

How do these algorithms compare to other algorithms which fit a hyperplane to a family of points? What merits do these algorithms have in comparison with linear regression, the classic hyperplane fitting algorithm? In this section we compare the paradigm used by the GLP fitting algorithms with that used by linear regression. The results that we present apply to the fitting algorithms in the following chapter as well.

A general structure for fitting problems involves a finite input family p of n points, a class X of possible *output objects*, a metric on R^d, and a norm on R^n. Using metric on R^d, we define the distance of an output object x from an input point p to be the minimum distance p to any point in x. For n input points, this assigns a vector of n distances to every output object. We measure the distance from x to the entire input family P by combining these individual distances using the norm on R^n. The fitting problem is to find the $x \in X$ with minimum norm.

In linear regression, both the metric and the norm are L^2. This is nice because the distance of a hyperplane x from a family H of points can be written down as a analytic function of the hyperplane coefficients and then minimized. When either the metric or the norm is not smooth, like L^1 (which uses absolute value) or L^∞ (which uses the maximum function), the problem becomes more combinatorial and the techniques of computational geometry come into play [3]. Our fitting algorithms minimize the maximum distance from the output object to any $p \in P$, so they use the L^∞ norm.

One important feature of our fitting algorithms is that a single "bad" data point, one far from any output object which is close to the rest of the data, strongly affects the choice of the output object. These bad data points, or *outliers*, are often just noise or errors in the data. In statistics lingo, our algorithms are not *robust*. This is bad in some applications, when we want to to choose a reasonable output object regardless of the presence of outliers [4]. But it is good when we want to detect and eliminate the outliers, since they show up in the basis. Linear regression, on the other hand, is not very good for finding outliers since the output objects it selects are often closer to the outliers than to other, valid data points [5].

Another positive feature of our algorithms is that we can make a strong *distribution free* predictive claim about the optimal output objects. This claim will apply to the common situation in which the input family of points is a sample drawn at random from some unknown distribution D. To avoid having to introduce a lot of notation, we will state the result for the simple Point Set Separation problem, although an analogous theorem holds for all of our fitting algorithms.

We define the appropriate distance function for Point Set Separation. The distance $dist(p, x)$ between a hyperplane x and a point $p \in P$ is the distance from p to x in the v direction, and $dist(q, x)$ for $q \in Q$ is the distance from q to x in the $-v$ direction. Notice that the optimal hyperplane x is in fact the one which minimizes the maximum distance from any point.

Theorem 10.5.1 *Let (λ, x) be the solution to an instance of Point Set Separation, in which the points in P and Q were drawn at random from a distribution D on points*

[3] A survey of combinatorial fitting algorithms can be found in [KM93].

[4] Using L^1 as the collective metric gives more robust algorithms.

[5] I thank David Jacobs for this observation

labeled either P or Q. The probability that a new point p drawn at random from D has $dist(p, x) > \lambda$ is at most $d/(n+1)$, where $n = |A + B|$.

Proof: Note that although we call the new point p, it may belong to either P or Q. The theorem follows from the fact that Point Set Separation is GLP. The argument is an example of Seidel's *backwards analysis*. We choose n points from D (to make $P+Q$), and then one more (p). We may as well think of choosing a set P' of $n+1$ elements, and then choosing one $p \in P'$ to be the "last" one. The probability that we choose a particular p is $1/(n+1)$.

As a consequence of the GLP framework, if $dist(p, x) > \lambda$, then p is violated by any basis for the set of constraints due to $P+Q$, and p is a member any basis B for the set of constraints due to P'. Since $|B| \leq d$, the probability that a member of B was chosen last is $d/(n+1)$. So the probability that $dist(p, x) > \lambda$ is no greater than $d/(n+1)$.
■

All our fitting algorithms return an output object x and a maximum distance λ. And for all of them, a theorem analogous to this one says that the probability that a new point will be further than λ from x is less than $d/(n+1)$, where d is the combinatorial dimension and n is the size of the input. Notice this is true for any distribution D, and requires no information about the distribution.

Chapter 11

Line transversals in three or more dimensions

In the plane, a line transversal is a hyperplane transversal. In E^d, $d > 2$, a hyperplane transversal is a point in the space of hyperplanes, which is also E^d, and the geometric structure of the hyperplane transversal problem remains pretty simple. A line transversal, on the other hand, is a point in the space of lines, which is a non-Euclidean space of dimension $2(d-1)$ (a *Grassmanian*). The geometric structure there is more complicated, and GLP can only be used to find line transversals in special cases.

Finding line transversals in dimension greater than two is none the less an important problem. In fact, it was an application of this problem which led me to study GLP. A friend in computer graphics [Te91] asked for an algorithm to find a line transversal through a family of axis-aligned rectangles in E^3. Megiddo showed that, in any dimension, this problem is reducible to a linear program. Like the LP for finding hyperplane transversals in the last chapter, this LP for finding line transversals implies a Helly theorem which leads to a GLP for finding the line which minimizes the maximum distance from a set of points under the weighted L^∞ metric. This problem had been brought to my attention long ago by [Pon 91]. In the last section, I summarize an application of a further extension of this algorithm to the *correspondence problem*, due to [J].

11.1 Helly theorems about line transversals

We begin with some Helly theorems about line transversals in E^d due to Grünbaum, [G60], which lead to GLPs which can be used either for finding line transversals or for the corresponding fitting problems. The first concerns a family of "flat" polytopes, which lie in parallel hyperplanes.

Theorem 11.1.1 (Grünbaum) *A family K of convex bodies in E^d which are contained in family of at least two parallel hyperplanes has a line transversal if and only if every $B \subseteq K$ with $|B| \leq 2d-1$ has a line transversal.*

The second is about a family of widely-spaced balls.

Theorem 11.1.2 (Grünbaum) *Let K be a family of balls in E^d, such that the distance between any two balls is greater than or equal to the sum of their radii. Then K has a line transversal if and only if every $B \subseteq K$ with $|B| \leq 2d-1$ has a line transversal.*

Grünbaum's idea was to apply Helly's Topological Theorem to the sets of lines which intersect these objects. Since the space of lines in E^d is $2(d-1)$-dimensional, the conditions of the topological theorem are that the intersection of any $2(d-1)$ of these sets is a cell, and that the intersection of any $2d-1$ is non-empty. These conditions are met in these special cases.

Theorem 11.1.1 is, in fact, a special case of Helly's Theorem proper. When the convex bodies are $(d-1)$-dimensional *polytopes*, given by their facets, the problem of finding a line transversal can be reduced to linear programming, as follows. Without loss of generality, let the at most n parallel hyperplanes be $h_i = \{x \in R^d \ : \ x_d = c_i\}$, where x_d is the dth coordinate of x. Describe the line as $u + x_d(v)$, where the vector u is the intersection of the line with the plane $x_d = 0$, and v is the "slope" of the line, normalized so that its x_d component is 1. The condition that the polytopes lie in at least two hyperplanes ensures that the line intersects each hyperplane in a point, at which $x_d = c_i$. The facets of polytope p lying in h_i can be oriented cyclically, so that the line intersects p if and only if this point lies on the positive side of all the halfspaces supporting the facets. Each facet then corresponds to a linear equation

$$a \cdot (u + c_i v) > 0$$

where the constant vector a specifies the coefficients of the facet. A solution to this system of equations gives the line transversal. There are $2(d-1)$ unspecified coefficients in the normalized vectors u and v, so this is the dimension of the resulting linear program.

Theorem 11.1.3 *A line transversal for a family of polytopes in E^d that lie in a family of at least two parallel hyperplanes can be found by a linear program in dimension $2(d-1)$.*

A general $(d-1)$-dimensional convex body can be thought of a polytope with an infinite number of facets, so the set of lines which pass through it is convex in the parameterization above. So a line transversal for a general family of convex bodies in parallel hyperplanes can be found by a convex program.

As before, we can also construct an objective function by selecting a fixed point inside each convex body, and scaling the bodies around the fixed points. This gives a GLP which can be used to fit a line to the family of points, as well as to find a transversal for the family of convex bodies.

This fitting problem has a reasonable statistical interpretation. Say we want to fit a line to data with one independent and $d-1$ dependent variables. We determine the distance from a data point to a line, as before, by putting a (possibly different) convex body around every point which is the unit ball of some (quasi-)metric. Since we assume that the error on the measurements of the independent variable is zero, these unit balls will lie in a family of hyperplanes normal to the coordinate representing the independent variable.

Statisticians call the problem of fitting a linear subspace using such a metric and the L^2 norm (see Section 10.5) *subset regression*. So we call the problem of finding the line which minimizes the maximum distance under such a distance function

Problem: Subset Line Fitting

Input: *A finite family P of pairs (p, c_p), where p is a point, c_p is a convex body, and $p \in c_p$,*

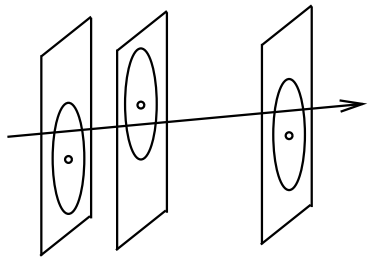

Figure 11.1: Unit balls for Subset Line Fitting

such that all the c_p *lie in a family of at least two parallel hyperplanes.*
Output: *A pair* (λ, x)*, where* $\lambda \in \mathcal{R}$ *is the smallest scale factor such that* $\lambda C_P = \{\lambda c_p \mid (p, c_p) \in P\}$ *admits a line transversal, and* x *is a line transversal of* λC_P*.*

Theorem 11.1.4 *Subset Line Fitting can be solved by a GLP of combinatorial dimension* $2d-1$ *with* n *constraints, where* $n = |P|$*.*

Constructing a GLP based on Theorem 11.1.2 is also straightforward. We again select a fixed point inside each ball, an "grow" the ball around the fixed point to produce an objective function. Thus,

Theorem 11.1.5 *A line transversal for a family of balls in* E^d *such that the distance between any two balls is greater than the sum of their radii can be found with a GLP of combinatorial dimension* $2d-1$*.*

This GLP, unlike most of the others, cannot be interpreted as a fitting algorithm as well as a transversal algorithm. The reason is that the Helly theorem holds, and the GLP "works" only so long as the balls remain widely separated. Running a GLP algorithm while ignoring this restriction produces some line, but not necessarily the one which minimizes the maximum distance from the points.

A construction using intersecting unit balls in the plane [D57] shows that there can be no Helly theorem about line transversals of balls in general. [1] Because the Helly

[1] The reader will probably also be interested to know, in this context, that there is an $\Omega(n \lg n)$ lower bound for the problem of finding a line transversal for unit balls in the plane [LW86], via reduction to the

theorem breaks down when the spheres get too close together, there is no affine structure that we may impose on the space of lines such that the set of lines through any ball forms a convex set. So Theorem 11.1.5 is another example (like Line Transversal of Translates, which was also based on a Helly theorem with a disjointness condition) of a GLP problem in which the constraints are not convex.

11.2 The weighted L^∞ metric

The first result on line transversals of boxes appears in [S40]:

Theorem (Santaló): *A family of axis-aligned boxes in E^d has a line transversal if and only if every subfamily of size $2^{d-1}(2d-1)$ has a line transversal.*

Unaware of this result [2], Teller raised the problem of finding a line transversal for a family of n boxes in E^3. The problem arose in the context of a computer graphics simulation of an architectural model [Te91]. During a pre-processing phase, the model is analyzed to determine which rooms can see into which others, to reduce the amount of visibility checking during the interactive simulation. Each physically connected sequence of doorways, windows and stairwells is tested to see if it admits a line of sight. The geometric formulation of this test is to find a line transversal for a family of axis-aligned rectangles, if one exists.

In [HT91], Hohmeyer and Teller gave an $O(n \lg n)$ algorithm. I then [A92] showed that a line transversal directed into the positive octant of E^3 (or a *positive line transversal*) could be found using a generalization of Seidel's linear programming algorithm, so that if a transversal exists, it can be found in expected $O(n)$ time by searching every octant in turn. A modification of that proof shows that the problem of finding a positive transversal is, in fact, GLP. Megiddo then [M] showed that the constraints in this GLP can be linearized, reducing he problem to linear programming. Furthermore, his linear formulation extends to any fixed dimension. Like the hyperplane fitting results, this technique is only useful in

Max-Gap problem on a circle, which is $\Omega(n \lg n)$ in the algebraic decision tree model of computation.

[2] Well, who among us spends enough time perusing articles in Spanish in back issues of Publicaciones del Instituto de Matematicas del Universidad Nacional del Litoral, Rosario, Argentina?

fixed dimension, since we need to solve 2^{d-1} linear programs, one for each pair of opposite orthants in E^d.

IBM is applying for a patent on Megiddo's reduction, and we are not at liberty to explain it here, which will unfortunately make the rest of this discussion rather cursory [3].

Like the reduction of Avis and Doskas in section 10.2, Megiddo's reduction implies the following

Theorem 11.2.1 *For a family H of axis-aligned boxes in E^d, there is a positive line transversal of the family if and only if there is a line transversal for every subfamily of size $\leq 2d - 1$.*

Instead of finding a line transversal with linear programming, we can use this Helly theorem in conjunction with the scaling objective function. By growing the boxes around a family of fixed points, we get another GLP algorithm for fitting a line to a set of points, this time using the weighted L^∞ metric introduced in Chapter 11.

Problem: Weighted L^∞ Line Fitting

Input: *A finite family P of pairs (p, c_p), where p is a point, c_p is an axis-aligned box, and p is at the center of c_p.*

Output: *A pair (λ, x), where $\lambda \in \mathcal{R}^+$ is the smallest scale factor such that $\lambda C_P = \{\lambda c_p \mid (p, c_p) \in P\}$ admits a positive line transversal, and x is a positive line transversal of λC_P.*

We find the smallest λ at which the boxes admit *any* line transversal by solving 2^{d-1} problems, one for each pair of opposite orthants of E^d. The transversal with the smallest λ is the one which minimizes the maximum distance from any point in the weighted L^∞ metric. This gives us

Theorem 11.2.2 *Weighted L^∞ Line Fitting can be solved by a family of 2^{d-1} GLP problems of combinatorial dimension $2d - 1$, each with $O(n)$ constraints, where $n = |P|$.*

---

[3] This seems to me to be evidence that patenting algorithms is a bad idea.

11.3 Application to the correspondence problem

Here is a problem from computer vision. We are given a collection of similar images, taken by a moving camera, in which our low-level vision system identifies critical points, which we call *features*. A rigid object appears in all the images as a set of corresponding features. If we can figure out the correct correspondence between the features from image to image, we can recover the motion of the camera. This is the *correspondence problem*.

Various heuristics are used to select possible correspondences. Given a purported correspondence, we have some hope of checking it, since a rigid set of points in E^3 can only produce a limited number of two-dimensional images. Jacobs has formulated a version of the testing process which can be solved as a GLP [J].

Say we are given n images of a rigid object in each of which we identify k features. In [J92], Jacobs represented the k features from each image as a pair of points, one in each of two separate $k-3$ dimensional *feature spaces*. He showed that all possible pairs of points produced by a particular rigid object lie on two parallel lines, one in each feature space. So if we have a correct correspondence across the n images, we should get a set of n points in each space, lying on two parallel lines [4].

But of course there is some unknown amount of error in the measurements of the image feature locations. Jacobs assumes a uniform error on the measurements, and then tracks this through the transformation to the feature space. This gives an oddly-shaped error function around each point, which he bounds with a box-shaped error function. There is some *pair* of parallel lines, one in each space, such that the maximum distance from a point in either space to the line is minimized. This maximum distance is an approximate measure of the quality of the proposed correspondence.

So we are left with the problem of finding the pair of parallel lines which minimizes the weighted L^∞ distance to two sets, each of n points, in E^{k-3}. To handle it in the

[4] To get this formulation he has to make the simplifying approximation that the images were produced by orthogonal projection and scaling, rather than ordinary projection. Apparently this is common in computer vision and does not introduce too much error.

same manner as the usual Weighted L^∞ Line Fitting problem, we need a Helly theorem. Fortunately Megiddo's reduction, to which we referred above, is easily generalized to the problem of finding a pair of parallel line transversals for two families of boxes. The fact that this problem can be solved by a linear program implies the following Helly theorem, which again, regrettably, we have to state without proof.

Theorem 11.3.1 *For two families H_1, H_2 of axis-aligned boxes in E^d, there are two parallel lines l_1, l_2 such that l_1 is a line transversal of H_1 and l_2 is a line transversal of H_2, if and only if, for every pair of subfamilies B_1, B_2, $B_1 \subseteq H_1$, $B_2 \subseteq H_2$, with $|B_1 \cup B_2| \leq 3(d-1)+1$, there is a pair of parallel lines l_1, l_2 such that l_1 is a line transversal of B_1 and l_2 is a line transversal of B_2.*

Once again we construct a parameterized Helly system by scaling the boxes around their center points. Considering each orthant of E^d in turn gives us an algorithm to find *any* parallel pair.

Theorem 11.3.2 *For two families P_1, P_2 of points in E^d, with weights W_1, W_2 on their coefficients, the pair of lines which minimizes the maximum distance from any point under the weighted L^∞ metric can be found with 2^{d-1} GLPs of combinatorial dimension $3(d-1)+1$, each with $O(n)$ constraints.*

So using GLP, we can get an upper bound on the quality of a purported correspondence for a constant number of feature points, in expected linear time.

This approach could also be applied to fitting k parallel lines to k families of points with weighted coefficients, or to the associated transversal problem.

Chapter 12

Conclusions and Questions

In this thesis we have explored the class of GLP problems and its intimate relationship with the Helly theorems. Our theoretical results can be summarized as

$$Helly \supset GLP \supset CP$$

that is, the class of problems for which there is a Helly theorem about the constraints strictly contains the class of GLP problems, which strictly contains the class of CP convex programming problems. Previously, all that was known was $GLP \supseteq CP$, and that only as folklore. Although we have shown that GLP is a broader class than some people expected, we have not completely characterized it, since we only give a paradigm for the construction of an objective function. Conjecture 5.4.1 suggests a characterization of the GLP problems, and, if settled in the affirmative, would imply a general Helly theorem subsuming many others.

The connection between Helly theorems and GLP theory has been mutually profitable. We have greatly simplified the proof of Morris' theorem, and proved two interesting new Helly theorems (one about the largest box in the intersection of a family of convex sets, and the other about the minimum Hausdorff distance between two convex bodies under scaling and translation). Going in the other direction, we have shown many geometric optimization problems to be GLP by using Helly theorems, including problems raised by workers in computer graphics [Te92], computer vision [J], and computer aided manufacturing [DMR93]. There are other interesting Helly theorems whose exploitation remains to be explored, for instance theorems concerning separators [H91].

We hope that both theoretically and practically the somewhat unexpected juxtaposition of Helly theorems and GLP will continue to be a source of useful results.

Bibliography

[A92] Nina Amenta. Finding a line transversal of axial objects in three dimensions, *Proceeding of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, (1992) pages 66-71.

[A93] Nina Amenta, Helly Theorems and Generalized Linear Programming, (1993) pages .

[ABB91] Helmut Alt, Bernd Behrends, and Johannes Blömer. Approximate matching of polygonal shapes, *Proceedings of the 7th Annual Symposium on Computational Geometry*, (1991), pages 186-93.

[AK89] James Arvo and David Kirk. A survey of ray tracing acceleration techniques, in *An Introduction to Ray Tracing* (Andrew Glassner, ed), Academic Press, San Diego, (1989), pages 201-262.

[A83] Mikhail J. Atallah. A linear time algorithm for the Hausdorff distance between convex polygons, *Information Processing Letters* **17** (1983), pages 207-9.

[AD92] David Avis and Mike Doskas. Algorithms for high dimensional stabbing problems, *Discrete Applied Mathematics*, vol. 27, (1990) pages 39-48.

[ARW89] David Avis, Jean-marc Robert, and Rephael Wenger. Lower bounds for line stabbing, *Information Processing Letters*, vol. 33, (1989) pages 59-62.

[AH91] David Avis and Micheal E. Houle. Computational aspects of Helly's theorem and its relatives, *Proceedings of the Third Canadian Conference on Computational Geometry*, (1991) pages 11-14.

[AGPW] Boris Aronov, Jacob E. Goodman, Richard Pollack and Rephael Wenger. There is no Hadwiger number for line transversals in higher dimensions. *unpublished manuscript*, cited in [GPW93], theorem 2.9.

[BEI93] Reuven Bar-Yehuda, Alon Efrat and Alon Itai. A Simple Algorithm for Maintaining the Center of a Planar Point-Set, *Proceedings of the Fifth Canadian Conference on Computational Geometry*, (1993) pages 252-257.

[BS79] Bazaraa, M.S. and Shetty, C.M. *Nonlinear Programming: Theory and Algorithms*, John Wiley and Sons, New York, (1979).

[C83] Bernard Chazelle. The polygon containment problem, in *Advances in Computing Research, Vol. 1: Computational Geometry*,(F.P. Preparata, ed.) JAI Press, Greewich, Connecticutt (1983), 1-33.

[CM93] Bernard Chazelle and Jiří Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension, *Proceeding of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, (1993) pages 281-90.

[BEHW86] A. Blumer, A. Ehrenfeucht, D. Haussler and M. Warmuth. Classifying learnable geometric concepts with Vapnik-Chervonenkis dimension, *Proceedings of the 24th Symposium on Theory of Computation* (1985), pages 175-85.

[C86] Kenneth L. Clarkson. Linear programming in $O(n \times 3^{d^2})$ time, *Information Processing Letters* **22** (1986) pages 21-4

[C90] Kenneth L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small, *manuscript*, 1990. An earlier version appeared in *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 452-5, 1988.

[DMR93] Karen Daniels, Victor Milenkovic and Dan Roth. Finding the maximum area axis-parallel rectangle in a polygon, *Proceedings of the Fifth Canadian Conference on Computational Geometry*, (1993) pages 322-327.

[D57] Ludwig Danzer. Über ein Problem aus der kombinatorishen Geometrie, *Archiv de Mathematik* **8**, (1957), pages 347-351.

[DGK63] Ludwig Danzer, Branko Grünbaum, and Victor Klee. Helly's Theorem and it's relatives, *Proceedings of the Symposium on Pure Mathematics*, Vol. 7, Convexity (1963) pages 101-180. American Mathematical Society, Providence, RI.

[D51] Dantzig, G. B. Maximization of a linear function of variables subject to linear inequalities, *Activity Analysis of Production and Allocation*, Koopman (Ed.), Cowles Commission Monograph, 13, John Wiley and Sons, New York, (1951).

[D92] James Demmel. The componentwise distance to the nearest singular matrix, *SIAM Journal of Matrix Analysis and Applications*, vol. 13, no. 1, (1992) pages 10-19.

[D84] Martin Dyer. On a multidimensional search technique and its application to the Euclidean one-center problem, *SIAM Journal on Computing* 15, (1986) pages 725-38.

[D92] Martin Dyer. A class of convex programs with applications to computational geometry, *Proceedings of the 8th Annual Symposium on Computational Geometry*, (1992) pages 9-15.

[E93] Jürgen Eckhoff. Helly, Radon and Carathody type theorems, Chapter 2.1 in *Handbook of Convex Geometry*, P.M. Gruber and J.M. Willis, eds., (1993) Elsevier Science Publishers B. V., Amsterdam.

[EOS86] Herbert Edelsbrunner, Joseph O'Rourke and Raimund Seidel. Constructing arrangements of lines and hyperplanes with applications, *SIAM Journal on Computing* **15** (1986) 341-63

[EW89] P. Egyed and Rephael Wenger. Stabbing pairwise disjoint translates in linear time, *Proceedings of the 5th Annual Symposium on Computational Geometry*, (1989), pages 364-9.

[F87] Roger Fletcher. *Practical Methods of Optimization*, John Wiley & Sons, New York (1987).

[G92] Bernd Gärtner. A subexponential algorithm for abstract optimization problems, *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science* (1992), pages 464-472.

[GPW93] Jacob E. Goodman, Richard Pollack and Rephael Wenger. Geometric Transversal Theory, in New Trends in Discrete and Computational Geometry, János Pach, ed., (1993) Springer Verlag, Berlin, pages 163-198.

[G58] Branko Grünbaum. On common transversals, *Archiv de Mathematik* **9**, (1958), pages 465-9.

[G60] Branko Grünbaum. Common transversals for families of sets, *Journal of the London Mathematical Society* **35** (1960), pages 408-16.

[GM61] Branko Grünbaum and Theodore S. Motzkin. On components in some families of sets, *Proceedings of the American Mathematical Society*, vol. 12, (1961) pages 607-13.

[H57] H. Hadwiger. Über Eibereich mit gemeinsamer Treffgeraden, *Portugal. Math.* **16** (1957) 23-29

[HW87] David Haussler and Emo Welzl. ϵ-nets and simplex range queries, *Discrete and Computational Geometry* **2** (1987), pages 127-151

[H79] Alan J. Hoffman. Binding constraints and Helly numbers, *Annals of the New York Academy of Sciences* **319**, (1979) pages 284-8.

[HT91] Michael Hohmeyer and Seth Teller. *Stabbing Isothetic Boxes and Rectangles in $O(n \lg n)$ Time*, Technical Report Number 91/634, University of California at Berkeley, 1991.

[H91] Michael E. Houle. Theorems on the existence of separating surfaces, *Discrete and Computational geometry*, **6**, (1991) pages 49-56.

[HKK92] Daniel P. Huttenlocher, Klara Kedam and Jon M. Kleinberg. On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane, *Proceedings of the 8th Annual Symposium on Computational Geometry*, (1992) pages 110-9.

[J] David Jacobs, *personal communication*, (1993).

[J92] David W. Jacobs. Space Efficient 3D Model Indexing, *IEEE Conference on Computer Vision and Pattern Recognition*, (1992), pages 439-44.

[K92] Gil Kalai. A subexponential randomized simplex algorithm, *24th Annual ACM Symposium on the Theory of Computation* (1992), pages 475-482.

[KK87] Victor Klee and Peter Kleinschmidt. The d-step conjecture and its relatives, *Mathematics of Operations Research* **12**:4 (1987), pages 719-755.

[KM93] Nikolai M. Korneenko and Horst Martini. Hyperplane approximation and related topics, in *New Trends in Discrete and Computational Geometry*, János Pach, editor, Springer-Verlag, Berlin (1993) pages 135-155

[L68] D.G. Larman. Helly type properties of unions of convex sets, *Mathematika* 15 (1968) pages 53-59.

[LW86] D.T. Lee and Y. F. Wu. Geometric complexity of some location problems, *Algorithmica* **1** (1986), pages 193-211.

[Mag69] Olvi L. Mangasarian. *Nonlinear Programming*, McGraw Hill, New York (1969).

[MSW92] Jiří Matoušek, Micha Sharir and Emo Welzl. A subexponential bound for linear programming, *Proceedings of the 8th Annual Symposium on Computational Geometry* (1992) pages 1-8.

[M84] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed, *Journal of the Association for Computing Machinery*, 31, (1984) pages 114-27.

[M83] Nimrod Megiddo. Linear-time algorithms for linear programming in R^3 and related problems, *SIAM Journal on Computing* 12, (1983) pages 759-76.

[M89] Nimrod Megiddo. On the ball spanned by balls, *Discrete and Computational Geometry* 4, (1989) pages 605-10.

[M] Nimrod Megiddo. *personal communication*, (1991)

[Mo73] Howard Cary Morris. *Two Pigeon Hole Principles and Unions of Convexly Disjoint Sets*, PhD thesis, California Institute of Technology, Pasadena, California (1973).

[Mu75] James R. Munkres. *Topology: A First Course* (1975), Prentice-Hall, Inc. Engelwood Cliffs, New Jersey.

[O89] Stephen M. Omohundro. Five Balltree Construction Algorithms, Technical Report Number 89-063, International Computer Science Institute, Berkeley, California (1989).

[Pon 91] Carl Ponder. A Search Approach to general Stabbing Problems, *Proceedings of the 3d Canadian Conference on Computational Geometry*, pages 195-202, 1991.

[PR92] Svatopluk Poljak and Jiří Rohn. Radius of nonsingularity, *Mathematics of Systems, Signals and Control*, to appear.

[R89] Jiří Rohn. Linear interval equations, *Linear Algebra and its Applications*, vol 126, (1989) pages 39-78.

[S40] L. A. Santaló. Un teorema sobre conjuntos de paralelepipedos de aristas paralelas, *Publicaciones del Instituto de Matematicas, Universidad Nacional del Litoral, Rosario, Argentina* **2**, (1940), pages 49-60.

[S90] Raimund Seidel. Linear programming and convex hulls made easy, *Proceedings of the 6th Annual Symposium on Computational Geometry*, (1990) pages 211-215.

[SW92] Micha Sharir and Emo Welzl. A combinatorial bound for linear programming and related problems, *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science* (1992), Lecture Notes in Computer Science 577, pages 569-579.

[Te91] Seth Teller and Carlo Sequin. Visibility Preprocessing for Interactive Walkthroughs, Computer Graphics, Volume 25, Number 4, SIGGRAPH '91 Conference Proceedings, pages 61-69, 1991.

[Te92] Seth Teller. *Visibility Computations in Densely Occluded Polyhedral Environments*, PhD thesis, Univerity of California, Berkeley, California (1992).

[T91] Sivan Toledo. Extremal Polygon Containment Problems, *Proceedings of the 7th Annual Symposium on Computational Geometry*, (1991) pages 176-185.

[T89] Helge Tverberg. Proof of Grünbaum's conjecture on common transversals for translates, *Discrete and Computational Geometry*, vol. 4, (1989) pages 191-203.

[W90] Rephael Wenger. A generalization of Hadwiger's theorem to intersecting sets. *Discrete and Computational Geometry* **5** (1990) 383-88

[W91] Emo Welzl. *Smallest enclosing disks (balls and ellipsoids)*, Technical report number B 91-09 Fachbereich Mathematik, Freie Universität Berlin, Berlin, Germany (1991)

[W93] Emo Welzl. *A Review of Recent Algorithms for Linear Programming and Related Problems*, manuscript, (1993).