DISS. ETH NO. 16168, 2005

Smallest enclosing balls of balls

Combinatorial structure & algorithms

A dissertation submitted to the
Swiss Federal Institute of Technology, ETH Zürich
for the degree of Doctor of Technical Sciences

presented by
Kaspar Fischer
Dipl. Ing. Inf. ETH
born 23rd May 1975
citizen of Switzerland

accepted on the recommendation of
Prof. Dr. Emo Welzl, ETH Zürich, examiner
Dr. Bernd Gärtner, ETH Zürich, co-examiner
Prof. Dr. Jiří Matoušek, co-examiner

Acknowledgments

I would like to thank all the people without whom this thesis would not have been written. In the first place, my thanks go to Emo Welzl for giving me the opportunity to work in his research group and the graduate program *Combinatorics, Geometry, and Computation*.

I would like to express my deep gratitude to my advisor Bernd Gärtner. His support, character, and working style were a warm companion during my stay at ETH. And without the interesting time I spent doing my diploma thesis under his guidance I would not have started my PhD in the first place.

I also want to warmly thank Jirka Matoušek for being my third supervisor and for taking the time to read and provide feedback.

Oscar Chinelato, Menelaos Karavelas, Sylvain Pion and others on the CGAL mailing list helped me a lot with ideas, discussions, and programming knowledge. I also owe a big thank you to Martin Kutz; working with him was inspiring.

Good memories come from and go to the members of the Gremo group: Andreas Razen, Bettina Speckmann, Dieter Mitsche, Eva Schubert, Falk Tschirschnitz, Floris Tschurr, Frans Wessendorp, Franziska Hefti, Joachim Giesen, Leo Rüst, Matthias John, Michael Hoffmann, Miloš Stojaković, Philipp Zumstein, Robert Berke, Shankar Ram Lakshminarayanan, Tibor Szabó, Uli Wagner, and Yoshio Okamoto. In particular, I am very thankful to Ingo Schurr (for tons of good moments) and Péter Csorba (for the best humour on the floor).

Behind the scenes there were, of course, other people who helped and influenced me substantially during my PhD time. I thank my girlfriend Nicole Schmidlin, my best friends Klaus Labhardt and Florian Düblin, and my sister Alexandra Ryser—no, not just for feeding me. Katharina Adiecha, Béatrice Goetz, Joachim Dietiker and Miriam Kündig, together with a dozen of anonymous Tango dancers all over the world changed a lot inside me.

Last but not least, I would like to thank my parents, Elisabeth Zumthor and Franz Fischer, for their lively and lovely support.

Abstract

This subject of this thesis is the *miniball problem* which asks for the smallest ball that contains a set of balls in d-dimensional Euclidean space.

We try to answer three main questions concerning this problem. What structural properties does the 'miniball' exhibit? What practical algorithms can be used to compute it? And how efficiently can instances in high dimensions be tackled? In all these questions, it is of interest how the problem and algorithms for it compare to the more specific variant of the problem in which all input balls are *points* (all radii zero).

In connection with the first question, we show that many of the already known properties of the *miniball of points* translate also to the *miniball of balls*. However, some important properties (that allow for *subexponential algorithms* in the point case, for instance) do not generalize, and we provide counterexamples and appropriate new characterizations for the balls case.

The change in structure between the point and ball case also reflects itself in the algorithmic picture, and we demonstrate that *Welzl's algorithm* does not work for balls in general, and likewise a known reduction to *unique sink orientations* does not apply either. Our main result here is that under a simple general-position assumption, both the correctness of Welzl's algorithm and a reduction to unique sink orientations can be established. The result has an appealing geometric interpretation and practical significance in that it allows for *pivoting algorithms* to solve the problem; the latter have the potential of being very fast in practice. As a byproduct, we develop a deeper (but not yet complete) understanding of the general applicability of Welzl's algorithm to certain optimization problems on the combinatorial cube.

Our contributions concerning the last question are twofold. One the one side, we provide a combinatorial, simplex-like algorithm for the point case that turns out to be very efficient and robust in practice and for which we can show that in theory it does not cycle. On the other hand, we formulate the problem with balls as input as a *mathematical program* and show that the latter can be solved in *subexponential time* by using Gärtner's algorithm for *abstract optimization problems*. In fact, our method works for other convex mathematical programs as well, and as a second application of it we present a subexponential algorithm for finding the distance between two convex hulls of balls.

Zusammenfassung

Diese Arbeit beschäftigt sich mit dem *Miniballproblem*, welches nach der kleinsten Kugel verlangt, die eine gegebene Menge von Kugeln im d-dimensionalen Euklidschen Raum einschließt. Drei Fragen versuchen wir zu beantworten: Welche strukturellen Eigenschaften weist der 'Miniball' auf? Wie kann man ihn in der Praxis schnell berechnen? Und wie verhält sich die Komplexität des Problems in hohen Dimensionen? Von speziellem Interesse ist dabei, wie sich das Problem und die Algorithmen von der Variante unterscheiden, in welcher alle Eingabekugeln *Punkte* sind.

Im Zusammenhang mit der ersten Frage zeigen wir, daß sich viele der schon bekannten Eigenschaften des *Miniball von Punkten* auf den Ballfall übertragen lassen. Einige wichtige Merkmale jedoch—sie ermöglichen es zum Beispiel, das Problem im Punktfall in *subexponentieller Zeit* zu lösen—verallgemeinern sich nicht, was wir anhand von Gegenbeispielen und angepaßten Charakterisierungen aufzeigen.

Der strukturelle Unterschied zwischen dem Punkt- und Ballfall drückt sich auch in den Algorithmen aus. Wir stellen fest, daß *Welzl's Punkt-Algorithmus* für Bälle nicht mehr funktioniert und gleichfalls eine Reduktion zu *Unique Sink Orientations* (USO) scheitert. Unser Beitrag dazu zeigt auf, daß eine einfache Annahme über die Lage der Eingabekugeln sowohl die Korrektheit von Welzl's Algorithmus garantiert als auch eine Reduktion zum USO-Problem ermöglicht. Das Resultat kommt mit einer anschaulichen geometrischen Erklärung und ist von praktischer Bedeutung, da damit *Pivotieralgorithmen* angewandt werden können, die in der Praxis oft sehr schnell sind. Als Nebenprodukt entwickeln wir ein tieferes (aber noch unvollständiges) Verständnis für jene Optimierungsprobleme, auf welche Welzl's Algorithmus angewandt werden kann.

Unser Beitrag zur dritten Frage ist zweigeteilt. Zum einen liefern wir einen kombinatorischen, Simplex-artigen Algorithmus für den Punktfall. Dieser ist in der Praxis sehr effizient und robust, und wir beweisen, daß er nicht 'zykeln' kann. Zum anderen formulieren wir dass ursprüngliche Problem als *mathematisches Programm* und zeigen, daß sich dieses mittels Gärtner's Algorithmus für *Abstract Optimization Problems* in subexponentieller Zeit lösen läßt. Die resultierende Methode erfaßt auch andere mathematische Programme, insbesondere erhalten wir einen subexponentiellen Algorithmus zur Berechnung der Distanz zwischen den konvexen Hüllen zweier Kugelmengen.

Contents

Notation

$\mathbb{N}$ .... the *natural numbers* including 0
$\mathbb{R}^d$ .... the d-dimensional *Euclidean space*
$\mathbb{R}^d_+$ .... the *positive orthant* $\{x \in \mathbb{R}^d \mid x \geq 0\}$ of $\mathbb{R}^d$
$\mathrm{im}(f)$ .... the *image* $\{f(x) \mid x \in \mathrm{dom}(f)\}$ of a function f
$\mathrm{sgn}(\alpha)$ .... the *sign* of a real number α (with $\mathrm{sgn}(0) = 0$)
$U_\delta(x)$ .... the *open δ-neighborhood* $\{x' \in \mathbb{R}^d \mid \|x' - x\| < \delta\}$ of the point $x \in \mathbb{R}^d$ for real radius $\delta > 0$
$\dot{U}_\delta(x)$ .... the *dotted neighborhood* $U_\delta(x) \setminus \{x\}$ of $x \in \mathbb{R}^d$
2^T .... $\{V \mid V \subseteq T\}$, i.e., the *power set* of the set T
$U \oplus V$ .... the *symmetric difference* of the sets U and V
$U \subseteq V$ .... U is any *subset* of V ($U = V$ is possible)
$U \subset V$ .... U is a *proper* subset of V ($U = V$ is not possible)
$\mathbb{E}[X]$ .... the *expectation* of the random variable X
Ω .... a *totally quasiordered set* (see p. 15)
$\pm\bowtie$ .... the *maximal* and *minimal* element in a quasiordered set Ω
$[a]$ .... the *equivalence class* of $a \in \Omega$ under relation $\sim$ (p. 16)
$[A, B]$ .... $\{X \mid A \supseteq X \supseteq B\}$, i.e., the *set interval* between A and B
$C^{[A,B]}$ .... the *cube* spanned by $A \supseteq B$ (p. 28)
$F(C)$ .... the set of all *faces* of the cube C (p. 29)
$\mathrm{conv}(P)$ .... the *convex hull* of a pointset $P \subseteq \mathbb{R}^d$
$\mathrm{aff}(P)$ .... the *affine hull* of a pointset $P \subseteq \mathbb{R}^d$
$B(c, \rho)$ .... the d-dimensional *ball* $\{x \in \mathbb{R}^d \mid \|x - c\|^2 \leq \rho^2\}$ (p. 49)
c_B, ρ_B .... the *center* and *radius*, respectively, of ball B
∂B .... the *boundary* of a ball B (p. 50)
C_T .... the set $\{c_B \mid B \in T\}$ of *centers* of the balls from the set T
$\mathrm{MB}(T)$ .... the *miniball* of a set T of balls (p. 49/98)
$\mathrm{B}(U, V)$ .... the set of balls that contain U and to which the

balls in V are internally tangent (p. 56/98)

$\mathrm{MB}(U,V)$ ....the set of smallest balls in $\mathrm{B}(U,V)$ (p. 56/98)

$\mathrm{MB}_p(U)$ .... the set $\mathrm{MB}(U \cup \{p\}, \{p\})$, p a point (p. 57/100)

$\mathrm{CB}(T)$ ....the *circumball* of a nonempty affinely independent pointset $T \subset \mathbb{R}^d$, or of a set T of balls (p. 58)

$\mathrm{CC}(T)$ ....the *circumcenter* of T, i.e., the center of the ball $\mathrm{CB}(T)$

$s_B(D)$ ....the *support point* of a ball B w.r.t. a larger ball $D \in \mathrm{B}(\{B\},\{B\})$, i.e., the single point in $\partial B \cap \partial D$ (p. 61)

$\mathrm{supp}_D(T)$ .... the set $\{s_B(D) \mid B \in T\}$ for a ball $D \in \mathrm{B}(T,T)$ larger than every ball in T (p. 61)

$\mathrm{tang}_D(T)$ ....the subset of the balls T that are internally tangent to the ball D (p. 67)

I_d .... the *identity matrix* (with d rows and d columns)

∇f ....the *gradient* (a column vector) of a function $f : \mathbb{R}^n \to \mathbb{R}$

Vectors, points, and scalars. In order to avoid confusion between vectors, points, and scalars, we try to stick to Arabic letters for points and vectors and use Greek ones for scalars.

Notation for mathematical programs. A *mathematical program* $\mathcal{P}$ is the problem of minimizing a real function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ over a domain $\mathcal{X} \subseteq \mathbb{R}^n$, which is called the *feasibility region* of $\mathcal{P}$. The problem $\mathcal{P}$ is called *convex* if the domain $\mathcal{X}$ is a convex set and f is a convex function over $\mathcal{X}$. A point $x \in \mathbb{R}^n$ is called *feasible* (or, a *solution* of $\mathcal{P}$) if $x \in \mathcal{X}$; it is called *finite* if $f(x) < \infty$.

A *(local) minimizer* of $\mathcal{P}$ is a point $x \in \mathcal{X}$ for which there exists a real $\delta > 0$ such that $x' \in U_\delta(x)$ implies

$$f(x') \geq f(x) \tag{1}$$

for all $x' \in \mathcal{X}$; x is called a *strict minimizer* if (1) holds with strict inequality. A *global minimizer* of $\mathcal{P}$ is a point $x \in \mathcal{X}$ such that (1) holds for all points $x' \in \mathcal{X}$. In this case, x is also called an *optimal solution* of $\mathcal{P}$. Whenever we say that a point (optimally) *solves* $\mathcal{P}$ we mean that x is a minimizer of $\mathcal{P}$; for convenience, we drop the word 'optimally.'

Chapter 1

Introduction

The subject of this thesis is the *miniball problem* which asks for the the smallest closed ball that contains a given finite set of objects (points or balls) in d-dimensional Euclidean space (Fig. 1.1). We focus on *exact* (i.e., not approximate) and *combinatorial* algorithms.

This chapter reviews previous results and lists the contributions contained in this thesis.

1.1 Background

History. The miniball problem has a long history dating back to 1857 when Sylvester posed it for points in the plane [83]. Many applications have popped up since then, including collision detection [48], the computation of bounding sphere hierarchies for the rendering of complex scenes, culling (e.g. for visualization of molecular models [87]), facility location and surveillance [58], automated manufacturing [46], similarity search in feature spaces [57], and medical irradiation [65]. Some applications require the problem to be solved in high dimensions, examples being tuning of support vector machines [14], high-dimensional clustering [7, 15], and farthest neighbor approximation [42].

On the algorithmic side many different approaches have been pursued, starting with Sylvester's geometric procedure [84] which he attributed to Pierce and which Chrystal rediscovered some years later [18].

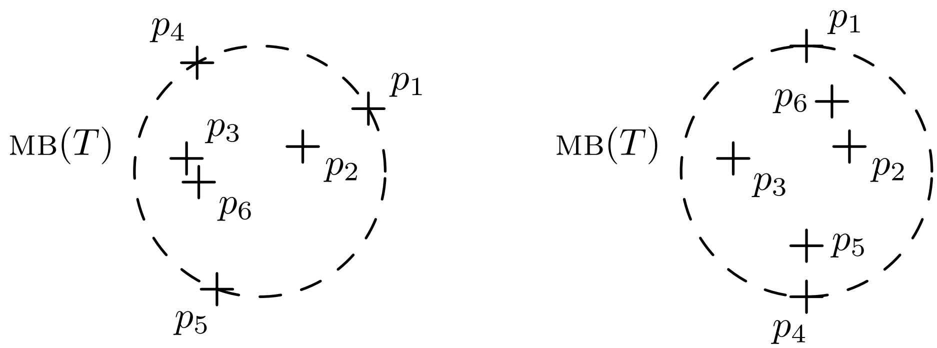


Figure 1.1. *Two examples in the plane $\mathbb{R}^2$ of the (dashed) miniball* MB(T) *for a pointset $T = \{p_1, \ldots, p_6\}$. Throughout the thesis, points and balls of zero radius are drawn as '$+$'.*

Many papers followed (e.g. [58, 4, 25], [76] with a fix [9], [49, 81]; see also the historical remarks in [12]). Of particular importance was Megiddo's algorithm [62] as it was the first to compute the miniball of a pointset in *linear time* for fixed dimension. Megiddo and Dyer later observed that the $\mathcal{O}(n)$-bound also applies to the case when the input objects are balls [63, 22]. In both cases, however, the algorithm does not work 'out of the box' because the *prune-and-search technique* underlying it requires systems of constant-degree algebraic equations to be solved, rendering an implementation difficult.

The first 'practical' algorithm to achieve a linear running time was a simple and elegant randomized procedure published by Welzl in 1991, computing the smallest enclosing ball of a pointset [86]. Welzl's algorithm was inspired by an idea Seidel [74] devised for solving *linear programming* (LP), the problem that asks for the 'lowest[1]' point in the intersection of a set of halfspaces. In contrast to other miniball algorithms for points, Welzl's algorithm is easy to implement, robust against degeneracies [33], and very efficient in small dimensions ($d \leq 20$, say).

In the following years, Matoušek, Sharir & Welzl enhanced the algorithm (its underlying method by Seidel, respectively) and developed and analyzed a new randomized algorithm for LP. The resulting *MSW-algorithm* achieved a subexponential expected running time for LP [61, 60] which was, together with an independently obtained result by Kalai [50], a great breakthrough in the area. Surprisingly, the algorithm only uses

[1] A precise definition of this and all other notions in this preamble (and in the preambles of subsequent chapters) will be given later, of course.

very little structure of LP itself and therefore works for other problems too, as long as they share a few basic properties with LP. The miniball problem in particular is such an *LP-type problem.*

So far, we have only mentioned *exact algorithms*—and we will restrict ourselves to such methods in this thesis. We would like to remark, however, that several very efficient *approximation algorithms* have been proposed in the literature. Some of these build on general, iterative optimization techniques, refer for instance to the papers [88, 89]. Recently, a new, very successful approach has been pursued: the *core set method* finds a small subset of the input objects—a *core set*—approximately spanning the same smallest enclosing ball as the input itself; for this, it repeatedly calls an (approximate) solver for small instances [56, 1, 3]. More concretely, this approach gives a polynomial-time algorithm of complexity $\mathcal{O}(dn/\epsilon + f(\epsilon))$ for computing an enclosing ball whose radius is by a factor of at most $1 + \epsilon$ larger than the radius of the optimal ball.

The combinatorial view. The emphasis in this thesis lies on (exact) *combinatorial* algorithms; to see what we mean by this, let us look at the miniball problem for points. We claim that instead of looking for a *ball* that encloses the pointset T, it suffices to find an inclusion-minimal[2] *set* $B \subseteq T$ whose miniball has the same radius as the miniball of T. It takes some simple calculations (we will do them in Sec. 5.2) in order to verify that the miniball of such a set B—we denote this ball by $\text{MB}(B)$—is easy to compute, and that $\text{MB}(B)$ coincides with the miniball of the points T. In Fig. 1.1 for instance, we want to find the set $\{p_1, p_4, p_5\}$ (for the left example) and $\{p_1, p_4\}$ (in case of the right example). Observe that this new formulation of the problem is *discrete* in the sense that the task is to select an 'optimal' subset among the *finitely many* subsets of T, in contrast to the original task of choosing the 'optimal' ball among all enclosing balls of T (which is an infinite set).

LP-type problems. The above combinatorial formulation of the miniball already makes the problem fit into Matoušek, Sharir & Welzl's *LP-type framework.* To illustrate this, we again stick (for the purpose of this introduction) to pointsets T as our input objects and observe that for any subsets $U' \subseteq U \subseteq T$ of the input points T,

[2] A set is *inclusion-minimal with some property P* if it fulfills P but none of its proper subsets does.

(i) the miniball of U has a radius that is at least as large as the radius of the miniball of U', and

(ii) if the miniball of U has a larger radius than the miniball of U' then there exists some point $p \in U$ not contained in the miniball of U'.

Given this, the miniball problem appears as follows. There is some *ground set* T (the input points) with some function $w : 2^T \to \mathbb{R}$ assigning to every subset U of the ground set a *value* (the radius of the miniball of U). The goal is to find an inclusion-minimal subset of the ground set with largest value, where we may exploit properties (i) and (ii) from above. The former says that the function w must increase if we add *constraints* (i.e., points) to its argument, and the latter requires that whenever we notice a 'global change' (the fact that $w(U') < w(U)$ for $U' \subseteq U$), there exists a single element that witnesses the change 'locally' (the fact that the gap $w(U') < w(U' \cup \{p\})$ opens up for some $p \in U$). Problems exhibiting such *monotonicity* and *locality* are called *LP-type* problems; besides LP and the miniball problem, the class of LP-type problems spans many more real-world optimization problems, some of which we will encounter later on.

LP-type problems. Matoušek, Sharir & Welzl's algorithm (combined with some other algorithms as in Lemma 2.11) solves any LP-type problem provided two 'subroutines' are available. These need to be devised and implemented for the specific problem and will be called at most

$$\mathcal{O}(\delta n + e^{\mathcal{O}(\sqrt{\delta \log \delta})}), \tag{1.1}$$

times in expectation overall [39]. Here, $n = |T|$ is the size of the ground set and δ is the problem's so-called *combinatorial dimension*, which is defined to be the maximal cardinality of a solution of any subset of T, i.e., the maximal size of an inclusion-minimal subset $U \subseteq T$ with value equal to $w(U)$. In the miniball problem from above, for instance, it can be shown that the combinatorial dimension is at most $d + 1$, a fact we have already witnessed in Fig. 1.1, where the desired sets $\{p_1, p_4, p_5\}$ and $\{p_1, p_4\}$ have size at most 3. Returning to the running time (1.1), we can see that if the subroutines can be implemented in time polynomial (or subexponential) in δ and n—as is the case for instance with LP—the whole problem can be solved in *subexponential* time.[3]

[3]This assumes we take the *real RAM model* as our model of computation; see Sec. 2.1 for more on this.

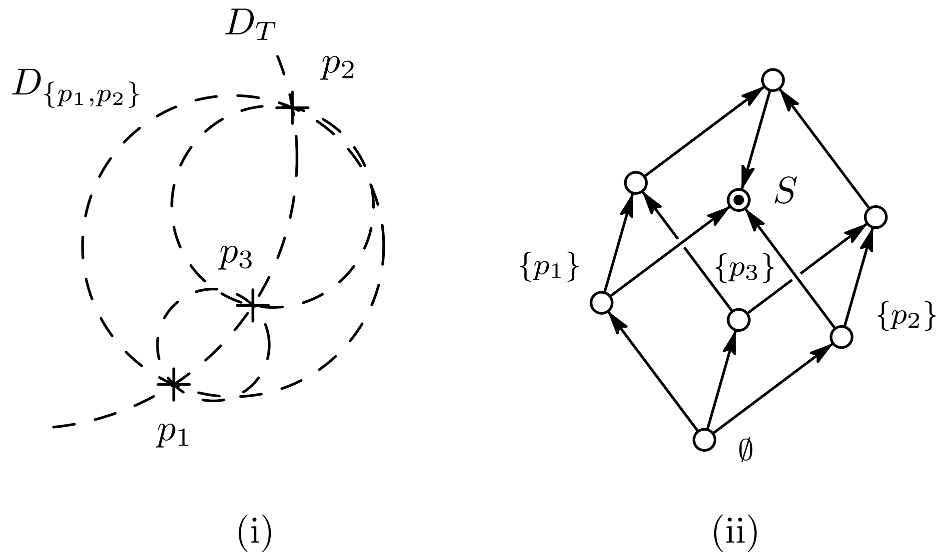


Figure 1.2. *(i) The balls D_U for $U \subseteq T := \{p_1, p_2, p_3\}$. (ii) The unique sink orientation induced by the pointset T.*

However, this is not so easy for general LP-type problems. To understand the issue, we need to take a closer look at the above problem-specific subroutines involved in the algorithm. Usually, the challenging one is the *basis computation*, which essentially asks to solve a subinstance T_{small} of the problem of size $|T_{\text{small}}| \leq \delta + 1$. For some problems, this turns out to be relatively easy: for instance, if all solutions have size *exactly* δ, we can simply check all $\delta + 1$ subsets of size δ, one after another, to see which one solves the problem. In LP, for instance, such a procedure works and we can solve the basis computation in $\mathcal{O}(\delta)$ 'trials.' In contrast to this, the situation is more difficult in the miniball problem, where the task in this step is to solve an instance consisting of at most $\delta + 1 = d + 2$ points. Here, not all solutions need to have cardinality δ (recall the solution $\{p_1, p_4\}$ of the right instance in Fig. 1.1), so (almost) *every single* of the exponentially many subsets is a candidate for the solution, and thus, the naive enumeration approach will take exponential time in the worst case. So if not 'all' solutions have size δ—we say that the problem violates *basis regularity*—it is not at all obvious whether a *subexponential* algorithm exists for solving it.

Unique sink orientations. In some cases, the exponential worst-case running time of the basis computation can be improved by embedding the LP-type problem into the *unique sink framework* [85]. Let us again look at the construction in the case of the miniball problem, where we assume for the sake of simplicity that the at most $d+2$ input points T_{small} are *affinely independent*. (This can always be achieved by embedding the points in $\mathbb{R}^{d+1}$ and perturbing them afterwards.) In this case, there exists for every subset $U \subseteq T_{\text{small}}$ a unique smallest ball D_U with the points U on its boundary, see Fig. 1.2(i). Moreover, it is not difficult to see that the miniball $\text{MB}(T_{\text{small}})$ coincides with one of the balls D_U, $U \subseteq T_{\text{small}}$ (Lemma 3.21 will settle this). But which one is it? That is, which inclusion-minimal set $U \subseteq T$ induces it?

Given a candidate subset $U \subseteq T_{\text{small}}$ and a point $p \in T_{\text{small}} \setminus U$, we can ask ourselves whether U or $U \cup \{p\}$ is a (locally) better candidate. If D_U does not contain p, the set U is not a candidate for a solution because we seek for an *enclosing* ball (which D_U is not). If conversely D_U already contains the point p, the set U induces a ball which encloses $U \cup \{p\}$; thus, U is in this case a better candidate than $U \cup \{p\}$ in the sense that it already spans a ball enclosing $U \cup \{p\}$. We conclude that $U \cup \{p\}$ is preferable to U if and only if D_U does not contain p.

In this fashion we can ask $n := |T_{\text{small}}|$ questions for a given subset $U \subseteq T_{\text{small}}$, one for each pair $\{U, U \oplus \{p\}\}$ with $p \in T_{\text{small}}$. Geometrically, the situation matches a *cube*, with each vertex corresponding to a candidate set U and each edge $\{U, U \oplus \{p\}\}$ representing a question. By answering all questions, that is, by orienting all edges towards the preferable endpoint of an edge, we obtain an *edge-oriented* cube C that possesses a very useful property, the so-called *unique sink property* [40, 85] (see again Lemma 3.21). Namely, every nonempty subcube of the cube has in its induced orientation exactly *one* vertex with only incoming edges. In particular, this means that the whole cube has a *global sink* $S \subseteq T_{\text{small}}$, a set whose neighbors are all less preferable. It turns out that this set S is the desired inclusion-minimal set spanning $\text{MB}(T_{\text{small}})$ because all points not in S are contained in D_S (from which we see that D_U encloses the pointset S), and a point $p \in S$ cannot be dropped because it would not be enclosed anymore. Fig. 1.2(ii) for instance shows the unique sink orientation of the points in part (i) of the figure. The sink is $S = \{p_1, p_2\}$ which corresponds to the ball D_S, and indeed, this ball coincides with the miniball of $\{p_1, p_2, p_3\}$.

This view of the miniball problem has a practical significance. It allows us to employ one of the many algorithms designed for finding the global sink in an *unique sink orientation* (USO), i.e., in an oriented cube fulfilling the unique sink property. Such a *USO-algorithm* takes as input an *orientation oracle* for a USO ϕ, that is, a subroutine returning the orientations w.r.t. ϕ of all edges incident to a given cube vertex, and outputs the vertex of the cube that is the sink in ϕ. As the orientation of a given edge (and hence the oracle itself) can be computed efficiently in the above oriented cube for the miniball problem, we can employ the currently best USO-algorithm to solve the basis computation of the miniball problem with $\mathcal{O}(1.44^n)$ trials. This is an improvement of approximately a square-root factor over the naive enumeration approach. Moreover, the reduction to the USO problem allows for *pivoting methods* (like the USO-algorithms random-facet [35] or Murty's pivoting method [66]) to be applied; for these we may not be able to give worst-case performance guarantees, but they have the potential of being fast for almost every input.

Abstract optimization problems. Closely related to LP-type problems is Gärtner's framework of *abstract optimization problems*. In fact, the above bound (1.1) for LP-type problems uses Gärtner's algorithm [32] internally to solve small instances, and it turns out that the bound even applies if the basis computation does not return the *optimal* solution of the small problem but merely a *better* one (if there is a better one at all). This means that any LP-type problem can be solved if a *basis improvement* routine is available, which for a given small instance T_{small} (of size at most $\delta + 1$) and a candidate solution $V \subseteq T_{\text{small}}$ either asserts that $w(V) = w(T_{\text{small}})$ or computes a better solution $V' \subseteq T_{\text{small}}$ otherwise; again, the number of basis improvement is bounded by (1.1). Of course, devising a basis improvement for an arbitrary LP-type problem is difficult in general; for our running example, the miniball problem of points, Gärtner showed how this can be achieved in polynomial time [32].

1.2 Contributions and outline of the thesis

In Chap. 2 we review the combinatorial methods we have encountered in the above outline. In particular, we discuss Gärtner's *strong LP-type problems* [36] which provide a link between LP-type problems and unique sink orientation. Also, we introduce the new class of *weak LP-type problems* which captures for instance the miniball problem (with points as input) and polytope-distance problem without any general-position assumptions (as are needed in strong LP-type formulations and reductions to USO). We show that if an additionally property, the so-called *reducibility*, holds, a weak LP-type problem can be solved using Welzl's algorithm, which in this case produces a *strong basis*. In the miniball problem, for instance, this implies that Welzl's algorithm computes an *inclusion-minimal* subset of the input points T whose miniball coincides with $\text{MB}(T)$.

As a byproduct we observe in Chap. 2 that the *total order* underlying the monotonicity and locality axioms of (original, weak, and strong) LP-type problems can be relaxed to a *total quasiorder*, without affecting existing algorithms (in particular, without worsening the upper bound (1.1)). In case of miniball, for instance, this allows us to work with *balls* instead of arguing with radii (as we did in the introduction). As far as LP-type problems are concerned, this observation is probably negligible. However, when we study weak and strong LP-type problems, it becomes essential that we work with *unique objects* and not merely with one of their parameters (refer to the example on page 45 for more on this).

In Chap. 3 we develop properties of SEBB, the problem of computing the smallest enclosing ball of a set of balls. Restricted to points as input, we provide a mathematical program to compute $\text{MB}(U, V)$, the smallest ball enclosing U that goes through the points V and present a similar program for the problem of computing the smallest 'superorthogonal' ball of a pointset.

Building on the basics, Chap. 4 presents a new combinatorial algorithm for SEBP, the miniball problem for points. In contrast to Welzl's algorithm from above, our method is very fast also in (moderately) high dimensions ($d \leq 10{,}000$, say), outperforming even dedicated interior-point solvers. On the theoretical side, we adopt *Bland's rule* [19] from the simplex algorithm to prevent the algorithm from cycling.

Chapter 5 resumes problem SEBB. It starts off by showing that the problem distinguishes itself from SEBP in that it is neither a reducible nor a strong LP-type problem. In particular, Welzl's algorithm does not work for it. Using the geometric *inversion transform* and a suitable general-position assumption, we can establish a strong LP-type problem satisfying reducibility. It follows that Welzl's algorithm *does work* if the balls have affinely independent centers, and that such SEBB instances can be reduced to the problem of computing the sink in a unique sink orientation. For the latter, it is essential that we overcome the possible nonexistence of the balls D_U from Fig. 1.2 when U is not a pointset but a set of balls (Fig. 3.4 illustrates the issue). We do this by generalizing $\textsc{mb}(U,V)$ appropriately, and the resulting 'generalized balls' have the nice property that they can be computed from the solutions of (at most two) convex programs with *linear* constraints. In particular, this implies that not only SEBP can be tackled by solving a linearly constrained convex program (which was already known [25, 38]) but also SEBB, provided an input ball internally tangent to $\textsc{mb}(T)$ is known (which can be guessed if need be). We emphasize that the general position assumption for the unique sink orientation can be handled efficiently in practice so that it is not necessary to resort to general symbolic perturbation methods.

Chapter 6 addresses the question whether SEBB, too, can be solved in *subexponential* time (recall that Gärtner has shown this for SEBP). We generalize a method by Gärtner & Schönherr for solving convex quadratic mathematical programs [72, 38], and use Gärtner's algorithm for abstract optimization problems to show that a wider class of convex programs can be solved in subexponential time. What we require for this is a certain computational primitive that needs to be devised for the mathematical program at hand. In case of SEBB and also for the problem of computing the distance between two convex hulls of balls, the respective primitive is easy to realize, entailing the existence of subexponential algorithms for both problems. The resulting method solves problems that do not fit into the *convex linear programming* framework presented by Amenta [2].

This thesis is accompanied by three software packages [28, 30, 27] which have (or will) become part of the *Computational Geometry Algorithm Library* CGAL,[4] a C++ library developed in a joint project by several universities in Europe and Israel. All these implementations follow the *generic programming paradigm* and are carefully designed to be at the same time efficient and easy to use. In case of the codes for SEBB and SEBP, the implementations use dedicated techniques to ensure robustness against degeneracies in almost all cases. Also, the code for solving SEBB can be driven with *arbitrary precision arithmetic* instead of floating-point arithmetic, allowing for an *exact* solution to be computed.

[4] Visit `http://www.cgal.org` for further information.

Chapter 2

Combinatorial frameworks

In this chapter, we review the *LP-type framework* introduced by Matoušek, Sharir & Welzl, including their *MSW-algorithm* for solving such problems and its forerunner algorithm by Welzl. We explain how Gärtner's algorithm for *abstract optimization problems* can be used to solve small LP-type problems in subexponential time, and link a certain subclass of LP-type problems, the so-called *strong LP-type problems*, to the *unique sink framework*. Furthermore, we introduce the less restrictive *weak LP-type problems* for which Welzl's algorithm produces an appealing solution. The problem SEBP of finding the smallest enclosing ball of a pointset and the polytope-distance problem, for instance, can be modeled as weak LP-type problems without any general position assumptions.

Overall, this chapter settles the basic notions and gives the details to the overview taken in the introduction. SEBP will accompany us throughout this exposition and will serve to illustrate the concepts. However, no properties of SEBP are proven here; we will do this in later chapters.

2.1 Complexity model

When we talk about the 'running time' of a certain algorithm, we mean the number of steps it takes until it completes with a result. Clearly, this measure of efficiency depends on what basic operations we allow to be counted as a single 'step,' i.e., which *model of complexity* we adopt.

Our model of computation is the *real RAM model* [68, p. 109] in which every memory cell can store a *real number* (and not only one value from a *finite* alphabet) and in which every basic arithmetic operation (be it an addition, subtraction, multiplication, division, square root, or comparison) counts as one 'step,' regardless of the 'sizes' of the operands.[1] (What is the size of a real number, anyway!) As an example, finding the largest among n real numbers takes time $\mathcal{O}(n)$ in this model. If an algorithm is *polynomial* (i.e., takes polynomial time in the *Turing machine model* [80]) and runs (in the real RAM model) in polynomial time in the number of input numbers only, we say that it is *strongly polynomial*. Linear programming for instance, is a problem for which polynomial algorithms are known [55, 52], but the question whether a strongly polynomial algorithm exists is still open at present.

The reason why for the problems and algorithms in this thesis we prefer the real RAM over the usually adopted *Turing machine model* [80] are the following. First of all, for many of the problems we will encounter, algorithms with polynomial running time (i.e., that complete in polynomial time in the Turing machine model) are already known, and it is therefore an interesting question what their complexity is if we express it in terms of the input parameters only—parameters like for instance the number 'n' in the above problem of finding the largest among n numbers—and *not* in the input size.

The second and more important reason is that our interest lies in the combinatorics and structure of the problems, and for the resulting combinatorial algorithms, a running time statement for the real RAM model seems more adequate. To make our point, let us look at the miniball problem again, and suppose we solve an instance involving points whose Euclidean coordinates are numbers in $\{0,\dots,10\}$. If we scale the pointset by a factor of thousand, say, the structure of the problem does not change at all—referring to the introduction, the very same inclusion-minimal subset of the input points spans the miniball—and we want the algorithm *and its running time* to behave identically for both the unscaled and the scaled input. Such insensitivity to input representation can be achieved in the real RAM model whereas the bit-complexity usually changes (due to the different input size). We will therefore express the running times of the algorithms in this thesis in the real RAM model.

[1] The real RAM is usually equipped with a *fairness assumption* to disallow arithmetic operations on numbers that are 'much larger' than the input numbers.

Randomized algorithms. Some algorithms we will deal with are randomized in nature, so let us clarify what we mean by this. For our purposes, a *randomized algorithm* is an algorithm that uses random decisions internally, i.e., that queries at certain times a source of random numbers in order to decide which action to take next. We assume that such a *random number generator* (in form of a routine that for given k returns a natural number uniformly random in $\{0,\dots,k\}$) is available and that a single access to it costs unit time. Moreover, all algorithms we are going to see are *Las Vegas algorithms* without exception, that is, they return the correct result in *all* cases, independent of the randomization. (This is in contrast to so-called *Monte Carlo methods* that use randomization in such a way that their output is correct with a certain, hopefully high probability $p > 0$.)

We will want to express the running time of an algorithm $\mathcal{A}$ with respect to some parameter(s): in the above problem of determining the largest among n numbers, for instance, it is natural to express the running time in terms of n; for the miniball problem, we might want to parameterize by the number n of input objects and the dimension d of the ambient space. That is, the (possibly infinite) set $\mathcal{I}$ of all instances the algorithm $\mathcal{A}$ may run on is usually partitioned into classes, $\mathcal{I} = \bigcup_{p\in\mathcal{P}} \mathcal{I}_p$, where $\mathcal{P}$ is the set of all possible parameters (for instance, $\mathcal{P} = \{(n,d) \mid n,d \in \mathbb{N},\, d \geq 1\}$ in the miniball problem). For a given parameter value $p \in \mathcal{P}$ and an instance $I \in \mathcal{I}_p$, we denote by $t_{\mathcal{A}}(I)$ the random variable for the running time of $\mathcal{A}$ on the instance I. Notice here that the probability space on which the variable $t_{\mathcal{A}}(I)$ is defined consists of all possible sequences of outcomes of the random source the algorithm $\mathcal{A}$ queries internally—the events of the probability space are *not* the instances of the problem. The *maximal expected running time* (or *expected running time* for short) of algorithm $\mathcal{A}$ on instances of parameter $p \in \mathcal{P}$ is defined as

$$t_{\mathcal{A}}(p) := \max_{I\in\mathcal{I}_p} \mathbb{E}[t_{\mathcal{A}}(I)].$$

This worst-case analysis stands in contrast to an average-case statement about the running time in which the instances in $\mathcal{I}_p$ are chosen according to some predefined probability distribution and the algorithm achieves a certain running time on average. In this latter case, there might be an instance for which the algorithm (always) takes much longer than the average running time. Average- and worst-case analyses are different, useful concepts, but we will stick to worst-case analysis in the sequel.

2.2 The LP-type framework

Let us start with the running example of this chapter, problem SEBP of computing the smallest enclosing ball: for a finite nonempty pointset $T \subseteq \mathbb{R}^d$, we define $\text{MB}(U)$, the *miniball of* U, to be the smallest ball that contains the balls $U \subseteq T$, and we write 'SEBP' for the problem of computing $\text{MB}(T)$ for a given input pointset $T \subseteq \mathbb{R}^d$.

Later on, we will see that SEBP is something like the 'mother' of all LP-type problems because it exhibits most properties a 'general' LP-type problem possesses. As almost all of them are geometrically appealing, SEBP will accompany us through the chapter, serving as an illustration and motivation of the concepts. Most of the time, however, we will not (yet!) be able to formally prove things about it—in fact, we have not even bothered to give the precise definition of the problem at this stage but hope that the introduction has given you the picture.[2] We will provide the precise definition and proofs in the following chapters for the more general problem SEBB of computing the miniball of a set of balls.

From the computational point of view, a first property of SEBP that stands out is that the knowledge of an inclusion-minimal set $V \subseteq T$ with $\text{MB}(V) = \text{MB}(T)$ already allows us to 'cheaply' compute $\text{MB}(T)$: as we are going to prove in Lemma 5.2 (and as has already been mentioned in the introduction), the ball $\text{MB}(V)$ can in this case be computed from V in time $\mathcal{O}(d^3)$. A second simple observation we can make is the following: if $U' \subseteq U$ and the balls $\text{MB}(U')$ and $\text{MB}(U)$ have identical radii then $\text{MB}(U') = \text{MB}(U)$. This is an immediate consequence of the fact that the miniball is unique (which we prove in Lemma 3.1 to come).

We take these two properties as the motivation for the definition of a *quasiorder problem.*

Definition 2.1. *A* quasiorder problem *is a tuple* $(T, \leq, \Omega, w)$ *where* T *is a finite set,* $\leq$ *is a total quasiorder on* Ω*, and* $w : 2^T \to \Omega$ *is such that*

(i) $w(U') \leq w(U)$ *and* $w(U') \geq w(U)$ *for* $U' \subseteq U \subseteq T$ *implies* $w(U') = w(U)$ (uniqueness)*, and*

(ii) Ω *contains a minimal and a maximal element under* $\leq$*.*

The goal of the problem is to compute an inclusion-minimal set $V \subseteq U$ *with* $w(V) = w(U)$*.*

[2] It did not? Feel free to check out the details on page 49.

Here, a *quasiorder* on some set Ω is a binary relation $\leq \subseteq \Omega \times \Omega$ that is reflexive and transitive; it is called *total* if $x \leq y$ or $y \leq x$ holds for all $x, y \in \Omega$. We write '$x < y$' for the fact that $x \leq y$ and $x \neq y$. Notice that $x \not< y$ implies $x \geq y$ through totalness (where we take the freedom to write '$x \geq y$' for '$y \leq x$').

Given a quasiorder problem $(T, \leq, \Omega, w)$, we call the set T the problem's *ground set* and its members are said to be *constraints*. Furthermore, we call $w(U)$ for $U \subseteq T$ the *value of* U. Also, if the set Ω and the quasiorder $\leq$ are clear from the context, we denote the quasiorder problem by (T, w) for convenience. Similarly, we write $(T, \leq, w)$ for the quasiorder problem $(T, \leq, \operatorname{im} w, w)$, provided that the quasiorder on $\operatorname{im}(w) = \{w(U) \mid U \subseteq T\}$ is clear from the context.

Please notice that condition (ii) is merely a technical requirement: we can always add two special symbols, $\pm\bowtie$, say, to the set Ω and define $-\bowtie \leq x$ and $x \leq \bowtie$ for all $x \in \Omega$ (the function w then never attains any of these special values).

Clearly, problem SEBP is a quasiorder problem: take Ω_{MB} as the set of all d-dimensional balls, including the *empty ball* $\emptyset$ of radius $-\infty$ and the *infeasible ball* $\bowtie$ (which we will use later) of radius ∞, and define $\text{MB}(\emptyset)$ to be the empty ball. We order the elements of Ω_{MB} by their radii, i.e., for $B, B' \in \Omega_{\text{MB}}$ we define $B \leq B'$ if and only if the radius of B is at most the radius of B'. Clearly, $\leq$ is a total quasiorder on Ω_{MB}, and by adding (as described above) another special symbol $-\bowtie$ to it, we obtain a total quasiorder with a maximal and a minimal element. It now easily follows from the two properties discussed above that the tuple $(T, \leq, \Omega_{\text{MB}}, \text{MB})$ is a quasiorder problem. (Notice that the function $\text{MB} : 2^T \to \Omega_{\text{MB}}$ never attains value $\pm\bowtie$.)

The quasiorder $\leq$ on Ω_{MB} shows a peculiarity that you at first might not associate with the symbol '$\leq$.' Namely, $B \leq B'$ and $B \geq B'$ can hold at the same time but still $B \neq B'$ (take any two different balls of identical radius).

2.2.1 LP-type problems

An LP-type problem is a quasiorder problem that fulfills two additional conditions (the conditions (i) and (ii) from the introduction).

Definition 2.2. *A quasiorder problem* $(T, \leq, \Omega, w)$ *is an* LP type problem *if for for all* $U' \subseteq U \subseteq T$ *the following conditions hold, where* $\bowtie \in \Omega$

is the maximal and $-\bowtie \in \Omega$ *the minimal element of* $\leq$*.*

(i) $w(U') \leq w(U)$ (monotonicity)*, and*

(ii) $-\bowtie < w(U') < w(U)$ *implies the existence of a constraint* $x \in U$ *with* $-\bowtie < w(U') < w(U' \cup \{x\})$ (locality)*.*

A set $U \subseteq T$ *is called* infeasible *if* $w(U) = \bowtie$*, it is called* unbounded *if* $w(U) = -\bowtie$ *and* bounded *if* $w(U) > -\bowtie$*.*

Observe in statement (ii) of locality that x in fact lies in the set $U \setminus U'$; if $x \in U'$ then $U' \cup \{x\} = U'$ and hence $w(U') = w(U' \cup \{x\})$. Also, the converse of the implication (ii) is true, too, as $w(U') < w(U' \cup \{x\})$ implies $w(U) > w(U')$ via monotonicity and uniqueness (if $w(U) = w(U')$ then uniqueness and $w(U') = w(U) \geq w(U' \cup \{x\}) \geq w(U')$ shows $w(U' \cup \{x\}) = w(U')$, a contradiction). Moreover, the special element $-\bowtie$ from Ω relaxes the requirement that locality hold for *all* subsets of the ground set: if U' is unbounded, locality need not hold for the pairs (U', U), $U' \subseteq U$.

The above definition of an LP-type problem differs from the original one given in [77] in one minor point: we do not require $\leq$ to be a total order on Ω (i.e., we do not require $\leq$ to be antisymmetric), but demand antisymmetry only if $U' \subseteq U$. We will see later (page 19/45) how this slightly simplifies some LP-type formulations. Clearly, if $\leq$ is in fact a total order, the uniqueness property is automatically satisfied, and we are back at the original definition. Even in the general case, there is an easy way to obtain a *total* order from the quasiorder $\leq$. The following shows this, and we will make use of it later.

Lemma 2.3. *Let* $\leq$ *be a total quasiorder on* Ω*.*

(i) *The relation* $\sim$ *defined for* $a, b \in \Omega$ *via* $a \sim b$ *iff* $a \leq b$ *and* $b \leq a$ *is an equivalence relation over* Ω*.*

(ii) *The relation* $[a] \leq [b] \Leftrightarrow a \leq b$ *on the equivalence classes* $[a]$*,* $a \in \Omega$*, of* $\sim$ *is a total order.*

Proof. (i) is obvious and for (ii), let $a, a' \in [a]$ and $b, b' \in [b]$. Then we have $a \leq a' \leq a$ and $b \leq b' \leq b$, and thus

$$[a] \leq [b] \Leftrightarrow a \leq b \Leftrightarrow a' \leq b' \Leftrightarrow [a'] \leq [b'],$$

which shows that the relation defined in (ii) is well-defined. As it clearly is reflexive, antisymmetric, transitive, and total, the claim follows. □

The first defining property of an LP-type problem, monotonicity, says that the function w increases if we add constraints to its argument. In particular, we can see from it that the value of any subset of the ground set T is at most $w(T)$.

Definition 2.4. *Let (T, w) be an LP-type problem.*

(i) A set $V \subseteq T$ is called a basis *if V is bounded and $w(V') < w(V)$ for all $V' \subset V$.*

(ii) For $U \subseteq T$, a set $V \subseteq U$ is called a basis of U *iff V is a basis and $w(V) = w(U)$. (Notice that a basis is a basis of itself.)*

In other words, a basis is a bounded subset V of the ground set that is inclusion-minimal with the property of achieving value $w(T)$. Given this, we can rephrase the goal of an LP-type (T, w) as follows: find a basis with value $w(T)$. Or even simpler: find a basis of T. (Notice that such a basis exists if and only if $w(T) > -\bowtie$.)

Here is a different but equivalent formulation of locality; in the original papers introducing LP-type problems [61, 60, 77] you will find this one being employed.

Lemma 2.5. *Locality holds iff for all $x \in T$ and for all $U' \subseteq U$ with*

$$-\bowtie < w(U') = w(U),$$

the fact $w(U \cup \{x\}) > w(U)$ implies $w(U' \cup \{x\}) > w(U')$.

By monotonicity, the 'implies' in the statement is actually an 'iff.'

Proof. We first show that our definition implies the condition given in the lemma. If $w(U \cup \{x\}) > w(U) = w(U') > -\bowtie$ then locality yields an element $y \in (U \cup \{x\}) \setminus U'$ with $w(U' \cup \{y\}) > w(U')$. Furthermore, y cannot lie in U because for all $z \in U$ we have

$$w(U) \geq w(U' \cup \{z\}) \geq w(U') = w(U)$$

and hence $w(U' \cup \{z\}) = w(U')$ by uniqueness. So $y = x$ follows.

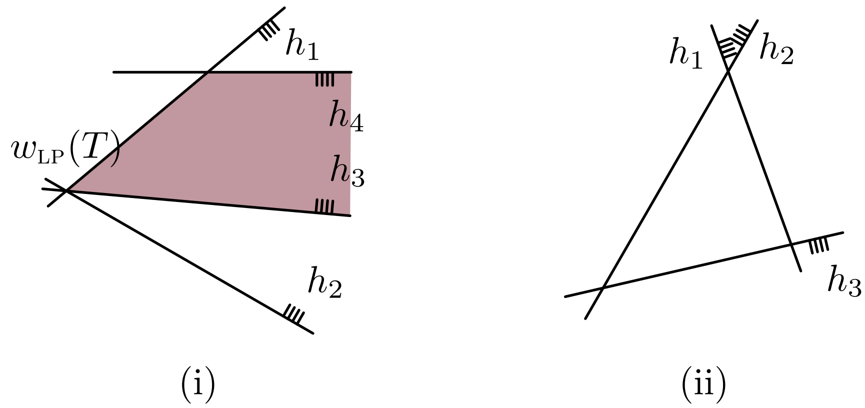


Figure 2.1. *(i) An instance T of the linear programming problem* (LP) *in the plane. (ii) An example with $w_{\text{LP}}(\{h_1, h_2, h_3\}) = \infty$.*

For the other direction assume that the alternative locality from the lemma holds and that $w(U) > w(U')$. Take any sequence

$$U' = U_0 \subset U_1 \subset U_2 \subset \ldots \subset U_m = U,$$

where $|U_i| = |U_{i-1}| + 1$ for all $i \in \{1, \ldots, m\}$. Let k be the smallest index such that $w(U') = w(U_k) < w(U_{k+1})$; such an index exists because $w(U') < w(U)$. Writing $U_{k+1} = U_k \cup \{x\}$, we get $w(U_k \cup \{x\}) > w(U_k) = w(U')$ and thus $w(U' \cup \{x\}) > w(U')$ by alternative locality. □

The need for $-\bowtie$. For many LP-type problems (in particular for SEBP discussed below) locality holds *without* the precondition that the involved subset U' be bounded. That is, there is no need to 'break' locality for subsets U' of the groundset that are unbounded as is done in the definition. Nonetheless, some problems can be formulated much more naturally if we allow such 'exceptions' to locality. One example is linear programming (LP), in which we ask for the lexicographically smallest[3] point, denoted by $w_{\text{LP}}(T)$, in the intersection $\bigcap_{h \in T} h$ of a set T of closed halfspaces in $\mathbb{R}^d$. Refer to Fig. 2.1(i) for an example in the plane.

Let us see why for LP locality does not hold for all pairs (U', U) if the precondition '$-\bowtie < w(U')$' is dropped in the definition of locality. First of all, Fig. 2.1 shows that $\bigcap_{h \in U} h$ is lexicographically unbounded

[3]A point $x \in \mathbb{R}^d$ is *lexicographically smaller* than a point $y \in \mathbb{R}^d$ iff there exists an index $i \in \{1, \ldots, d\}$ such that $x_j = y_j$ for all $j < i$ and $x_i < y_i$.

for $U = \emptyset$ (or for $|U| < d$, more generally), and thus $w_{\text{LP}}(U)$ is not well-defined. To remedy this, we define $w_{\text{LP}} : 2^T \to \mathbb{R}^d \cup \{\pm\bowtie\}$ to assign to a subset $U \subseteq T$ the lexicographically smallest point if it exists, $-\bowtie$ if $\bigcap_{h \in U} h$ is lexicographically unbounded, and $\bowtie$ in case the latter region is empty (see Fig. 2.1(ii)). Given this, we can read

$$w_{\text{LP}}(\emptyset) = w_{\text{LP}}(\{h_1\}) = -\bowtie$$

off Fig. 2.1(i), and since $w_{\text{LP}}(\{h_1\} \cup \{h_2\}) > -\bowtie$ does not imply $w_{\text{LP}}(\emptyset \cup \{h_2\}) > -\bowtie$, locality does not always hold for pairs (U, U') that attain value $-\bowtie$. Refer to [61, 77] for more information on LP.

We turn again to SEBP and see how it fits into the LP-type framework. We have already seen that $(T, \leq, \Omega_{\text{MB}}, \text{MB})$ is a quasiorder problem, and so it remains to verify that monotonicity and locality hold. Monotonicity is obvious because if the miniball of a set had a smaller radius than the miniball of a subset, the former would be a smaller ball enclosing the subset, contradiction. Similarly, $\text{MB}(U \cup \{x\}) > \text{MB}(U)$ implies that $x \notin \text{MB}(U)$ (otherwise $\text{MB}(U)$ would already be enclosing). So if in addition $\text{MB}(U) = \text{MB}(U')$ for $U' \subseteq U$ then x is not contained in $\text{MB}(U')$ either, and it follows $\text{MB}(U' \cup \{x\}) > \text{MB}(U')$. Thus, locality holds, too, and SEBP in the form of (T, MB) is hence an LP-type problem. (Here, locality as defined on p. 16 holds without the precondition '$-\infty < \text{MB}(U')$;' in other words, we could add an artificial minimal element $-\bowtie$ to Ω_{MB} and the resulting quasiorder problem would still be LP-type.)

We remark here that had we clung to total orders instead of quasiorders in the definition of LP-type problems, we would have had to introduce a value function w_{SEBP} that maps a subset $U \subseteq T$ to the *radius* of $\text{MB}(U)$ (see e.g. [61]). The result is the same, and by working with values $\text{MB}(U)$ we only gain the minor advantage that in *all* our (upcoming) formulations of SEBP as variants of LP-type problems, the value of a subset is a ball (and not a number). (For the formulation of SEBP as a *strong LP-type problem* one *has* to work with balls instead of radii, see p. 45.)

2.2.2 Basis-regularity

In SEBP, a basis $U \subseteq T$ is an inclusion-minimal subset spanning the same miniball as U. From geometry it seems clear (Lemma 3.6 to come) that points that are properly contained in the miniball do not affect it and can thus be removed. Hence, if V is a basis of $U \supseteq V$, all points in V

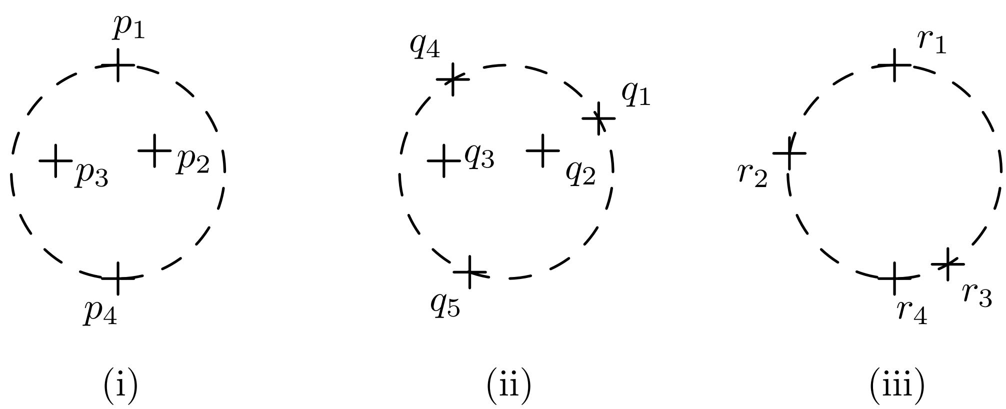


Figure 2.2. *Three pointsets P, Q, R in the plane, together with their miniballs, which are spanned by bases of different cardinalities. Thus,* SEBP *is not basis-regular.*

must lie on the boundary of the ball $\text{MB}(V)$. Here are some examples of bases: in Fig. 2.2, the sets

$$\{p_1, p_4\}, \{q_1, q_4, q_5\}, \{r_1, r_4\}, \{r_1, r_2, r_3\},$$

are some (but not all) bases; in fact, every single one of these bases is a basis of the respective pointset P, Q, or R, respectively, and these are all such bases. From this we see that not all bases need to have identical cardinality in an LP-type problem, and that there may be several bases that span a certain value, as is the case in Fig. 2.2(iii).

When we later encounter the MSW-algorithm for solving LP-type problems, we will learn that its running time heavily depends on two factors, namely on the maximal size of a basis, and on whether for a maximal-size basis V, the number $|V|$ equals the size of every basis of $V \cup \{x\}$ or not. This motivates the following definitions.

Definition 2.6. *Let (T, w) be an LP-type problem.*

(i) The combinatorial dimension *of (T, w), written as* $\dim(T, w)$*, is the maximal cardinality of a basis.*

(ii) (T, w) is basis-regular *if for every basis $V \subseteq T$ of size $\delta := \dim(T, w)$ and every $x \in T$, all bases of $V \cup \{x\}$ have size δ.*

In particular, (T, w) is basis-regular if all bases have size $\delta = \dim(T, w)$ (which happens iff all sets smaller than δ are unbounded).

As we have already hinted at, an LP-type problem (T, w) can be solved 'efficiently' using the MSW-algorithm from the next section if its combinatorial dimension is small; if it is large (e.g., $\delta = \Omega(|T|)$), we might be unlucky, though. We note that basis-regularity can be *enforced* using a trick by Gärtner, see [61, p. 511]. However, this does not imply that basis-irregular problems can be solved as efficiently as basis-regular ones. In fact, the enforcement usually shifts the difficulty to one of the primitives which the MSW-algorithm invokes, and thus the so-called 'basis computation' becomes more involved.

We will see later that the combinatorial dimension of $(T, \textsc{mb})$ is at most $d+1$. Looking at Fig. 2.2(i) again, we may observe that the basis $V := \{p_2, p_3, p_4\}$ has $d+1 = 3$ elements whereas the (only) basis of $V \cup \{p_1\}$ has cardinality two. So our LP-type formulation of SEBP is not basis-regular. We note here that LP can be written as a basis-regular LP-type problem.

Violation. For an LP-type problem (T, w) and $U \subseteq T$, we say that $x \in T$ *violates* U if $w(U \cup \{x\}) > w(U)$; in this case, x is also called a *violator* of U. In the context of SEBP, a point $x \in T$ violates $U \subseteq T$ if and only if x is not contained in $\textsc{mb}(U)$. Observe here that a *violation test* (i.e., checking whether violation applies or not) is easy if U is a basis. In this case, $\textsc{mb}(U)$ can be computed in $\mathcal{O}(d^3)$ (see comment on p. 14) and the violation check boils down to a containment check between a point and a ball.

2.2.3 The MSW-algorithm

The algorithm proposed by Matoušek, Sharir & Welzl [61, 60, 77] to solve LP-type problems is shown in Fig. 2.3; we refer to it as the *MSW-algorithm* for the sake of simplicity. Just like any method that solves LP-type problems in general, it makes some assumptions on how the concrete LP-type problem can be accessed. In this case, two 'primitive routines' must be available that depend on the specific LP-type problem (T, w) to be solved. They are the following.

- *Violation test:* Given a basis $V \subseteq U \subseteq T$ and a constraint $x \in U \setminus V$, the primitive violates(x, V) returns 'yes' if and only if x is a violator of V, and 'no' otherwise.

```
procedure msw(U, V)
{ Computes a basis of U }
{ Precondition: V ⊆ U is a basis }
begin
    if U = V then
        return V
    else
        choose x ∈ U \ V uniformly at random
        W := msw(U \ {x}, V)
        if violates(x, W) then
            return msw(U, basis(W, x))
        else
            return W
end msw
```

Figure 2.3. *The MSW-algorithm for solving an LP-type problem* (T, w)*. The solution is obtained by* msw(T, V_{init})*, where* V_{init} *is some initial basis.*

- *Basis computation:* Given a basis $V \subseteq U \subseteq T$ and a violator $x \in U \setminus V$, the primitive basis(V, x) returns a basis of $V \cup \{x\}$.

As a side remark, we mention here that the correctness of algorithm msw and its analysis do not rely on basis(V, x) returning a basis *of* $V \cup \{x\}$: it is sufficient for what follows that the result of such a call is a basis $V' \subseteq V \cup \{x\}$ that *improves* over V in the sense that $w(V') > w(V)$.

Procedure msw(U, V) computes a basis of U, given the set $U \subseteq T$ and an arbitrary basis $V \subseteq U$ which we refer to as the call's *candidate basis*. In order to solve the LP-type problem (T, w), we call msw(T, V_{init}) where V_{init} is some initial basis. (Observe that the existence of an initial basis already implies that $w(T) \geq w(V_{\text{init}}) > -\bowtie$, so T is bounded.)

If $U = V$, the algorithm immediately returns V, as V is a basis of $V = U$ by precondition. Otherwise, a basis W of $U \setminus \{x\}$ is computed recursively after having chosen a random element x from $U \setminus V$; the set $U \setminus \{x\}$ is bounded (because $w(U \setminus \{x\}) \geq w(V) > -\bowtie$) and hence a basis W of it exists. Now if x does not violate W then $w(U) = w(U \setminus \{x\})$ by locality (Lemma 2.5), meaning that W is not only a basis of $U \setminus \{x\}$ but also of U. In this case the algorithm stops with W as the result. If on the other hand x violates W, the algorithm invokes itself again to compute a basis of U, passing a basis of $W \cup \{x\}$ as the subcall's candidate basis. We see from this that the algorithm is correct. Moreover, whenever the procedure msw is called to recurse, either the cardinality of the first argument drops by one or the value $w(V)$ of the second argument V strictly increases. This together with the fact that the function w only attains finitely many values proves that the algorithm eventually terminates.

The choice of the constraint x to drop is critical because if it turns out 'bad,' we are forced to do additional work while for a 'good' choice we can exit immediately. More precisely, the probability that the second recursive subcall is taken equals the probability that x is contained in every basis of U; the latter however is obviously bounded by $\delta/|U|$, where δ is the problem's combinatorial dimension. Using this, one can prove the following bound on the running time of msw.

Lemma 2.7 (Sharir & Welzl)**.** *Algorithm* msw *solves any LP-type problem of size n and combinatorial dimension δ with a maximal expected number of at most*

$$2^{\delta+2}(|T| - \delta)$$

basis computations and a maximal expected number of at most this many violation tests, provided an initial basis V_{init} is given.

The proof from [77, 61, 31, 39] of this is formulated for the case when the quasiorder $\leq$ of the LP-type problem $(T, \leq, \Omega, w)$ is an *order*. However, it is easily verified that the proof also works when $\leq$ is a total quasiorder. Moreover, it is clear from the algorithm that the number of basis computation is dominated by the number of violation tests so that it actually suffices to count the latter; one can also show that the number of invoked basis computations is $\mathcal{O}(\log(|T|)^{\delta})$ [40].

In particular, the lemma shows that any LP-type problem of fixed, finite combinatorial dimension can be solved with a linear number of basis computations and violation tests.

Subexponential running time. Although the maximal expected number of primitive calls in the above lemma is linear in n, the dependency on the combinatorial dimension is exponential. Is this a weakness of the analysis of msw, or is the exponential behavior inherent to the algorithm? The answer is that *basis-regular* LP-type problems (like e.g.

LP) are solved with *subexponentially*[4] many basis computations and violation tests. To show this, one uses the fact that for such problems, the recursion of algorithm $\mathsf{msw}(U, V)$ ends as soon as the input set U consists of only δ constraints (in which case U is a basis). The analysis under this assumption yields a bound of

$$(n - \delta)\, e^{\mathcal{O}(\sqrt{\delta \log n})} \tag{2.1}$$

on the maximal expected number of violations tests and basis computations, respectively. For the proof of this, we refer the reader to [31, 61].

It is at present time not clear whether there are (basis-irregular) LP-type problems for which algorithm msw needs exponentially many primitive calls (see also the subexponential lower bound [59] by Matoušek). The only currently known method to obtain an algorithm that is subexponential also for basis-irregular LP-type problems is the following variation of msw: in spirit of Gärtner's trick to enforce basis-regularity (see above), we mimic the algorithm's behavior for basis-regular problems and stop the recursion $\mathsf{msw}(U, V)$ as soon as $|U| \leq \delta$ (namely, at the moment the recursion would stop for a basis regular problem); for the remaining small instance, we employ some other algorithm. Thus, we assume the availability of a *solver for small problems*, i.e., of a routine $\mathsf{small}(U)$ that returns a basis of an instance $U \subseteq T$ of (T, w) for $|U| \leq \delta$.[5] The resulting algorithm msw-subexp is identical to algorithm msw from Fig. 2.3, except that the statement '**if** $U = V$ **then return** V' is replaced by '**if** $|U| \leq \delta$ **then return** $\mathsf{small}(U)$.' You can verify its correctness along the same lines as in case of the original algorithm msw.

Since msw-subexp stops recursing as soon as the problem size reaches the combinatorial dimension, the analysis leading to (2.1) applies to it as well, except that we now have a multiplicative overhead of $t_{\mathrm{small}}(\delta) + 1$, where $t_{\mathrm{small}}(\delta)$ denotes the maximum expected number of violation tests carried out by $\mathsf{small}(U)$ over all instances $U \subseteq T$ of size $|U| \leq \delta$. In this way one obtains

Theorem 2.8 (Matoušek, Sharir & Welzl)**.** *Algorithm* msw-subexp *solves any LP-type problem (T, w) of combinatorial dimension δ and size $|T| =$*

---

[4] A function is said to be *subexponential* if its logarithm is sublinear.

[5] If one assumes that the primitive $\mathsf{small}(U)$ can solve small instances of size up to 2δ, a subexponential bound can be obtained, too, and the analysis [40] is much easier than the one from [31, 61] which we mentioned earlier.

n with at most

$$(n-\delta)\, e^{\mathcal{O}(\sqrt{\delta \log n})}(t_{\mathrm{small}}(\delta)+1)$$

primitive calls in expectation, provided an initial basis is available. For a basis-regular problem, $t_{\mathrm{small}}(\delta) = 0$*.*

Again, the theorem was proven for LP-type problems $(T, \leq, \Omega, w)$ where $\leq$ is a total order (and not a total quasiorder as in this thesis), but it is easy to verify that the proof works with total quasiorders, too.

2.3 The AOP framework

In the last section we saw that any LP-type problem can be solved with a subexponential number of primitive calls, provided there is an algorithm $\mathsf{small}(U)$ that solves small instances with subexponentially many calls. The only known algorithm that achieves the latter is Gärtner's *AOP-algorithm* [31, 32]. Surprisingly, this method uses *less* structure than is available in LP-type problems. Because of this, the algorithm can be formulated not only for LP-type problems but more generally, for so-called *abstract optimization problems*, or AOP's for short.

The problems in the AOP framework are optimization problems of the following kind. There is a *ground set* H, some of whose subsets are potential candidates for a solution of the problem. These subsets are called *abstract bases*[6] and the set of all abstract bases is denoted by $\mathcal{B} \subseteq 2^H$. The abstract bases are ordered via some total quasiorder $\preceq \subseteq \mathcal{B} \times \mathcal{B}$, and the goal of the problem is to find the 'best,' i.e., the $\preceq$-largest abstract basis in the ground set. (To be precise, there might be *several* $\preceq$-largest abstract bases; we want to find one of them.)

An optimization problem of this sort is an AOP if a primitive is available that produces for some given abstract basis $F \in \mathcal{B}$ and for some superset $G \supseteq F$ a better abstract basis in G, if possible. That is, if we denote by

$$\mathcal{B}(G) := \{F \subseteq G \mid F \in \mathcal{B}, \forall\, F' \in \mathcal{B},\, F' \subseteq G : \, F \succeq F'\}$$

the set of $\preceq$-largest abstract bases in G, the primitive either reports that $F \in \mathcal{B}(G)$, or it produces an abstract basis $F' \subseteq G$ that is better than F:

[6] This is not to be confused with the notion of a 'basis' in the LP-type framework.

Definition 2.9. *Let H be a set, $\preceq$ a total quasiorder on 2^H, and $\mathcal{B} \subseteq 2^H$. A quadruple $(H, \mathcal{B}, \preceq, \Phi)$ is an* abstract optimization problem *iff*

$$\Phi : \{(G, F) \mid F \subseteq G \subseteq H,\ F \in \mathcal{B}\} \to \mathcal{B}$$

is an improving oracle*, i.e., a function satisfying*

$$\Phi(G, F) = \begin{cases} F, & \text{if } F \in \mathcal{B}(G), \\ F' \succ F,\ F' \subseteq G,\ F' \in \mathcal{B}, & \text{otherwise.} \end{cases}$$

Notice that in the context of the previous section, the improving oracle replaces locality.—The goal of an AOP $(H, \mathcal{B}, \preceq, \phi)$ is to find a $\preceq$-largest abstract basis (i.e., an element in $\mathcal{B}(H)$) by only performing queries to the oracle ϕ. In other words, we assume that the order $\preceq$ is unknown and information about it can only be gathered by querying the oracle. The question then is: how many times do we have to access the oracle in order to find one of the abstract bases in $\mathcal{B}(H)$? Trivially, $2^{|H|} - 1$ accesses suffice in the worst case, as one can see for instance from the algorithm that iterates '$F := \Phi(H, F)$' until no progress is achieved anymore. Gärtner's randomized algorithm [31, 32] performs much better:

Theorem 2.10 (Gärtner)**.** *Any AOP on a ground set H of n elements can be solved with an expected number of $\exp(\mathcal{O}(\sqrt{n}))$ oracle calls.*

We refer to [39] for an introduction to Gärtner's algorithm and its underlying ideas, and to [31, 32] for the proof of the above statement. We again point out that Gärtner's algorithm (in the formulation in his thesis [31]) works with quasiorders although some papers only define AOPs with total orders.

Reduction from LP-type problems. We now return to the question how one can solve small instances of an LP-type problem with a subexponential number of violation tests and basis computations. For this, we reduce the LP-type problem to an AOP and run Gärtner's algorithm [39].

So let (T, w) be an LP-type problem of combinatorial dimension δ. From it, we define the following AOP

$$\mathcal{P}(T, w) := (T, \mathcal{B}, \preceq, \phi_{\text{small}}),$$

with the parameters explained next: as the AOP's ground set we take the constraints T of the LP-type problem, and we define the AOP's

```
procedure Φ_small(G, F)
{ Computes a basis F' ⊆ G with w(F') > w(F) }
{ or asserts that w(G) = w(F). }
{ Precondition: F ⊆ G, F is a basis }
begin
    forall x ∈ G \ F do
        if violates(x, F) then
            return basis(F, x)
    return F
end Φ_small
```

Figure 2.4. *The improving oracle for the AOP that solves the LP-type problem (T, w).*

abstract bases to be the bases of (T, w). In this way, a solution of the AOP is a basis of the original LP-type problem. Moreover, we order the abstract bases according to their value: if $F, F' \subseteq \mathcal{B}$, we set $F \preceq F'$ if and only if $w(F) \leq w(F')$. Through this, the solution of the AOP is a $\preceq$-largest LP-type basis and thus a basis of T. Finally, we realize the AOP's oracle $\phi_{\text{small}}(G, F)$ as shown in Fig. 2.4. The correctness of the routine follows trivially from locality.

Invoking Theorem 2.10, the problem (T, w) with $n := |T|$ can be solved using at most $\exp(\mathcal{O}(\sqrt{n}))(\delta + 1)$ primitive calls in expectation, the additional factor $\delta + 1$ stemming from the fact that Φ_{small} performs at most δ violation tests and one basis computation. Since this bound is exponential in n, we should not use this reduction for the initial (large) LP-type problem. Instead, we run algorithm msw-subexp and invoke Gärtner's algorithm on $\mathcal{P}(U, w)$ whenever msw-subexp calls small(U). Using this approach and two algorithms by Clarkson [20], one obtains the currently best bounds for LP-type problems (please refer to [39] for the details):

Lemma 2.11. *Any LP-type problem (T, w) of combinatorial dimension δ and size $n = |T|$ can be solved with an expected number of at most*

$$\mathcal{O}(\delta n + e^{\mathcal{O}(\sqrt{\delta \log \delta})})$$

violation tests and an expected number of at most $e^{\mathcal{O}(\sqrt{\delta \log \delta})}$ basis computations, provided some initial basis $B \subseteq T$ is available.

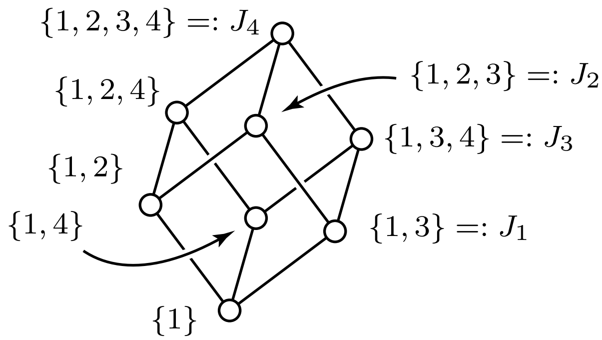


Figure 2.5. *The cube $C^{[A,B]}$ spanned by $A = \{1,2,3,4\}$ and $B = \{1\}$. The subgraph of $C^{[A,B]}$ induced by the vertices $\{J_1,\ldots,J_4\}$ is a face.*

We note here that instead of basis computations (as we use them for Φ_{small}) it might be more efficient in some cases to implement the AOP's oracle directly, using some sort of 'basis improvement.' The AOP algorithm clearly continues to work if we do so, and also the MSW-algorithm's analysis remains valid as pointed out on page 22.

For the moment, this concludes our overview of LP-type problems. We now turn to unique sink orientations and return to LP-type problems when we discuss a link between the latter and so-called reducible strong LP-type problems.

2.4 The USO framework

In this section we consider special optimization problems on cubes. For this purpose, we regard cubes (as we know them from geometry) as graphs whose vertices are sets. More precisely, we define for any two sets A and B satisfying $A \supseteq B$,

$$[A,B] := \{X \mid A \supseteq X \supseteq B\},$$

and denote by $C^{[A,B]}$ the *cube spanned by* $A \supseteq B$, that is, the graph of vertex set $[A,B]$ and edge set

$$\{\{J, J \oplus \{x\}\} \mid J \in [A,B],\, x \in A \setminus B\}.$$

The *dimension* of a cube $C^{[A,B]}$ is the number $|A \setminus B|$. An example of a 3-dimensional cube is shown in Fig. 2.5.

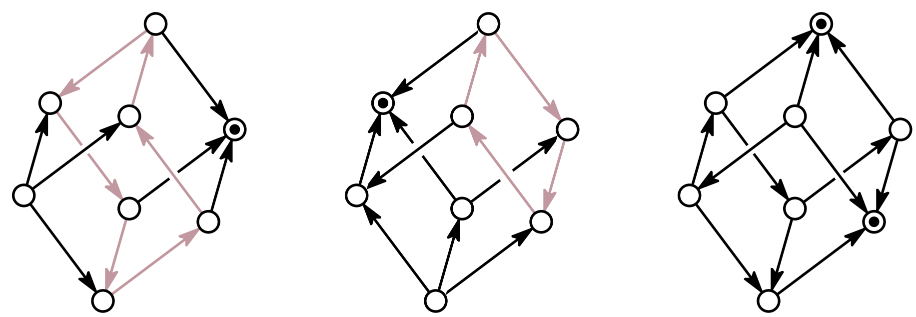


Figure 2.6. *Three orientations of a 3-dimensional cube: (i) is a unique sink orientation (with a cycle), (ii) and (iii) are not.*

Observe that the subgraphs of a cube $C = C^{[A,B]}$ induced by vertex sets $[F, G] \subseteq [A, B]$ are cubes again, and we call them the *faces* of C. The set of faces of a cube C is denoted by $F(C)$. For convenience, we identify cubes and faces, i.e., both $C^{[F,G]}$ (a cube) and $[F, G]$ (its vertex set) will be called faces (and cubes) in the sequel. Faces of dimension zero are called *vertices*, 1-dimensional faces are *edges*, and faces of dimension $\dim(C) - 1$ are called *facets* of C. Notice that C itself is a face of C.

By orienting the edges of C in an arbitrary way we obtain an *oriented cube*. If a vertex J of an oriented cube C has no outgoing edges (i.e., there is no edge in the graph C that is oriented away from J), it is called a *sink*.

Definition 2.12. *A* unique sink orientation (USO) *is an orientation of the edges of a cube such that every face of the cube has in its induced subgraph a unique sink.*

Figure 2.6 shows three orientations of a 3-dimensional cube C. The first one is a USO as is easily checked by applying the definition. The orientation (ii) is not a USO as it contains no sink in the highlighted face, and neither is (iii) since it contains two sinks. As you can see from the highlighted edges in Fig. 2.6(i), a unique sink orientation may contain *cycles*, i.e., closed oriented paths.

Finding the sink. Unique sink orientations appear in many contexts. Certain linear complementarity problems [82, 73], linear programs [73], certain strictly convex quadratic programs [73, 34], and (as we will see in the next section) also certain LP-type problems 'induce' unique sink orientations. In all these applications, the sink of the USO captures sufficient information to reconstruct the original problem's optimal solution. That is, knowing the sink solves the problem. However, the orientation is very expensive to compute (and also very large), so it is not explicitly available to us. What is desired is a way to 'query' a part of the orientation (as few times as possible!) in such a way that eventually we query the sink (and hence solve the original problem). This leads to the following model for finding the sink in a unique sink orientation [85].

We assume that a unique sink orientation ϕ is given implicitly through a *vertex evaluation oracle*, $\mathsf{evaluate}(J)$, which returns the orientations of all edges incident to a vertex J of the orientation's underlying cube. That is, if $C = C^{[A,B]}$ is ϕ's underlying cube and $J \in [A, B]$ then $\mathsf{evaluate}(J)$ returns a list of $|A \setminus B| = \dim(C)$ pairs

$$(x, o) \in (A \setminus B) \times \{\text{in}, \text{out}\},$$

where x identifies the edge $\{J, J \oplus \{x\}\}$ and o denotes its orientation relative to J. The goal of the problem is to query (i.e., evaluate) the sink of ϕ with as few vertex evaluations as possible. (One could also consider *edge evaluations*, but historically, vertex evaluation was first.)

An algorithm which solves this so-called *USO problem* is called a *USO-algorithm*. Its *running time* is the maximal (expected) number of vertex evaluations it needs to query the sink of any given USO. Currently, the best known randomized algorithm [85] has the following performance.

Theorem 2.13 (Szabó & Welzl)**.** *The sink of any unique sink orientation on a d-dimensional cube can be evaluated with a maximal expected number of $\mathcal{O}(1.44^d)$ vertex evaluations.*

2.5 Weak LP-type problems

We stay with problems defined on cubes and assume in contrast to the previous section that every face of the cube has an associated value. With a suitable monotonicity and locality of the value function, our goal will be—similarly to the original LP-type framework—to find a subcube of minimal dimension that spans the same value as the whole cube.

```
procedure welzl(U, V)
{ Computes a strong basis of [U, V] }
{ Precondition: V ⊆ U, w(U, V) < ⋈ }
begin
    if U = V then
        return V
    else
        choose x ∈ U \ V uniformly at random
        I:= welzl(U \ {x}, V)
        if infeasible(x, I) then
            return welzl(U, V ∪ {x})
        else
            return I
end welzl
```

Figure 2.7. *Welzl's algorithm for solving a reducible primal weak problem (T, w). The solution is obtained by calling* welzl$(T, \emptyset)$.

This is the setting of *weak* and *strong LP-type problems*; the latter were introduced by Gärtner [36] and the former are inspired by them.

In order to motivate the definition of weak LP-type problems we turn to *Welzl's algorithm* [86] which is listed in Fig. 2.7. Originally developed by Welzl for solving the miniball problem for points, the algorithm can be used to solve other problems as well: as we will learn later, the *polytope distance problem* (Sec. 6.6.2), problem SEBB (under some preconditions), and moreover also LP can be tackled using it.

In the abstract setting, welzl is an algorithm that works on a cube $C = C^{[T,\emptyset]}$ whose faces have, as described above, associated *values*, i.e., there is a function $w : F(C^{[T,\emptyset]}) \to \Omega$, with Ω a quasiordered set, that assigns to any face $[A, B] \subseteq [T, \emptyset] = 2^T$ a *value* $w(A, B)$. (Recall that $F(C)$ is the set of the faces of the cube C.) If the function w fulfills certain conditions, Welzl's algorithm finds (as we will see) a subcube of minimal dimensions that spans the same value as the whole cube. More precisely, a call to welzl(U, V) for $[U, V] \subseteq 2^T$ then returns a *vertex* (a zero-dimensional cube) I with $w(I, I) = w(U, V)$. The goal of this section is to develop the conditions that need to hold for this. In doing so, we will not focus on a *minimal* set of conditions; rather, we will impose requirements onto w that ensure that a call to welzl(U, V) not merely computes *some* vertex $I \subseteq T$ with $w(I, I) = w(U, V)$ but one that is a *strong basis of* $[U, V]$, meaning that it fulfills

$$w(U, V) = w(I, I) = w(U, I) = w(I, V).$$

In case of problem SEBP, this shows a property of Welzl's algorithm that is not mentioned in the original paper [86], and which is automatically fulfilled for *every* problem in the weak LP-type framework

In the sequel we first introduce *weak LP-type problems*, then mention the concept of *reducibility* which ensures that every face $[U, V]$ indeed contains a vertex I (i.e., a minimal-dimension subcube) with $w(I, I) = w(U, V)$, and finally show that Welzl's algorithm computes a strong basis for reducible weak problems.

Weak LP-type problems. To get a feeling for LP-type problems we turn to problem SEBP once again and define $\text{MB}(U, V)$ for $V \subseteq U \subseteq T$ as the smallest ball that contains the points in U and goes through (at least) the points in V. (We say that a ball $B \subset \mathbb{R}^d$ *goes through* a point $p \in \mathbb{R}^d$ if p lies on the boundary of B.) Lemma 3.11(i) in the next chapter shows that this ball is unique provided *some* ball though V containing U exists; if it does not exist (which may happen), we set $\text{MB}(U, V)$ to the *infeasible ball* $\bowtie \in \Omega_{\text{MB}}$ of radius ∞, see page 15. (More details on $\text{MB}(U, V)$ can be found in Chap. 3.)

Consider now for a given input pointset $T \subset \mathbb{R}^d$ the cube $C^{[T,\emptyset]}$ where we assign to the face $[U, V] \subseteq 2^T$ the ball $\text{MB}(U, V)$ as its value. We can observe the following two simple properties of MB. First of all, $\text{MB}(U', V') \leq \text{MB}(U, V)$ for all $U' \subseteq U$ and all $V' \subseteq V$, so *monotonicity* holds. Second, $\text{MB}(U, V) = \text{MB}(U', V') =: D$ implies that the ball D contains *all* points in $U \cup U'$ and that *all* points in $V \cup V'$ lie on the ball's boundary. We refer to this property as *dual nondegeneracy* (and will explain the name in a minute). We take this as a motivation to define *primal weak LP-type problems* and *dual weak LP-type problems* as follows; for convenience, we drop the word 'LP-type' in these terms, and moreover call a problem a *weak (LP-type) problem* if it is a primal or dual weak problem.

Definition 2.14. *Let T be a finite set, $\leq$ a total quasiorder on some set Ω, and $w : F(C^{[T,\emptyset]}) \to \Omega$. The quadruple $(T, \leq, \Omega, w)$ is a* primal (dual) weak problem *if the following properties (i), (iii), and (iv) ((i), (ii), and (v), respectively) hold for all $[U', V'], [U, V] \subseteq 2^T$ and all $x \in U \setminus V$.*

(i) $w(U', V') \leq w(U, V)$ *for* $U' \subseteq U$, $V' \subseteq V$ (monotonicity), *and for* $U' \subseteq U$, $V' \subseteq V$, $w(U, V) \leq w(U', V)$ *implies* $w(U, V) = w(U', V)$ *and* $w(U, V) \leq w(U, V')$ *implies* $w(U, V) = w(U, V')$ (uniqueness),

(ii) *If* $w(U, V) = w(U', V') \leq \bowtie$ *holds then* $w(U \cap U', V \cap V') = w(U, V)$ (primal nondegeneracy),

(iii) *If* $-\bowtie < w(U, V) = w(U', V')$ *holds then* $w(U \cup U', V \cup V') = w(U, V)$ (dual nondegeneracy),

(iv) *If* J *is an inclusion-minimal strong basis of* $[U, V \cup \{x\}]$ *and* $\bowtie > w(U, V) > w(U \setminus \{x\}, V)$ *then* $w(J, J) = w(J, V)$ (primal optimality).

(v) *If* I *is an inclusion-minimal strong basis of* $[U \setminus \{x\}, V]$ *and* $-\bowtie < w(U, V) < w(U, V \cup \{x\})$ *then* $w(U, I) = w(I, I)$ (dual optimality),

Here, $\bowtie \in \Omega$ *is the maximal and* $-\bowtie \in \Omega$ *the minimal element of* $\leq$.

The goal of a primal (dual, respectively) weak problem is to find a smallest-dimensional subcube of $[T, \emptyset]$ with the property of spanning the whole cube's value $w(T, \emptyset)$. With the following definition, the objective is to find a *weak basis* of $[T, \emptyset]$.

Definition 2.15. *Given a weak LP-type problem* (T, w) *and* $[U, V] \subseteq 2^T$, *a face* $[U', V'] \subseteq [U, V]$ *is called a* weak basis of $[U, V]$ *if*

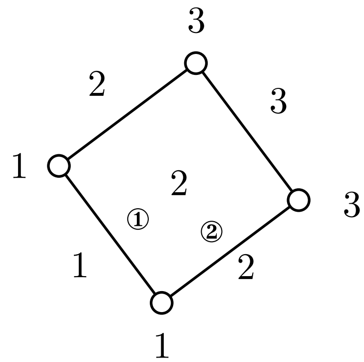
(i) $-\bowtie < w(U', V') = w(U, V)$, *and*

(ii) $w(U'', V'') \neq w(U', V')$ *for all* $[U'', V''] \subset [U', V']$.

A face $[U, V]$ *is called a* weak basis *if it is a weak basis of itself.*

Figure 2.8 shows an example of a weak LP-type problem on the groundset {①, ②}. As you can easily verify, the one-dimensional face $F = [\{②\}, \emptyset]$ is the only weak basis of F. In particular, there is no *vertex* $J \in F$ that spans the value $w(F)$. This shows that not every face of a weak problem needs to have a strong basis.

Miniball again. We will show in the next chapter that $(T, \leq, \text{MB})$ is a primal weak problem (Lemma 3.21). What use we can make of this? Since the points in a basis $V \subseteq U$ all lie on the boundary of $\text{MB}(V)$ (see page 20), we can observe that $\text{MB}(V) = \text{MB}(V, V)$ holds for any basis V,



U vs. V	$\emptyset$	$\{①\}$	$\{②\}$	$\{①, ②\}$
$\emptyset$	1			
$\{①\}$	1	1		
$\{②\}$	2		3	
$\{①, ②\}$	2	2	3	3

Figure 2.8. *The pair (T, w) with $T = \{①, ②\}$ and w as in the table is a weak LP-type problem (in fact, it is* strong*, see Sec. 2.6). The face $F = [\{②\}, \emptyset]$ has no strong basis (its only weak basis is F itself).*

in particular for a basis of the input pointset T. (This shows that in case of SEBP the weak bases of $[T, \emptyset]$ have dimension zero.) From this point of view, our goal is indeed to find a *weak basis* of the cube $[T, \emptyset]$ because any $[J, J]$ is automatically a weak basis.

This formulation of SEBP as a weak LP-type problem reveals a pattern that applies to all 'practical' weak LP-type formulations we have seen so far. Namely, the constraints in the set T manifest themselves in *two variants:* $x \in T$ can be 'weakly' active in which case it is a member of U but not of V, or it can be 'strongly' active in which case the constraint is listed in V and U. (And of course, a constraint can be inactive, in which case it is neither in V nor in U.) In the above formulation of SEBP, the weak version of a constraint $x \in T$ requires x to be contained in the miniball, while the strong version requires it to be *on the boundary.* More generally, a weak constraint may correspond to an inequality being satisfied while the corresponding strong constraint is the very same inequality with '$=$' in place of '$\leq$' (see for instance the formulation of SEBP based on quadratic programming [34]).

Interpretation. Let us try to get a feeling for the defining properties of a weak LP-type problem. First of all, monotonicity does not have an immediate interpretation in terms of cubes and subcubes: $[U', V']$ is not necessarily a face of $[U, V]$ if $U' \subseteq U$ and $V' \subseteq V$ as in the definition.

Primal nondegeneracy has an appealing interpretation: it guarantees *uniqueness of inclusion-minimal weak bases* (and in this sense the

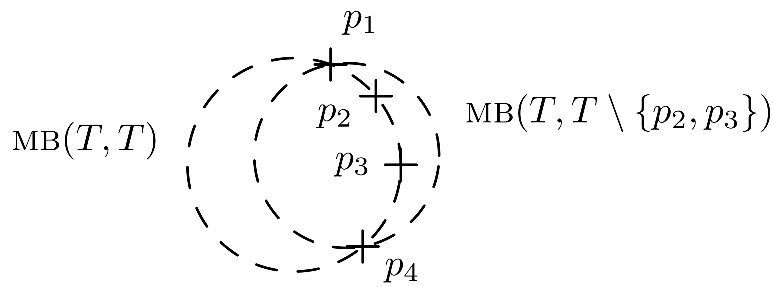


Figure 2.9. $(T, \leq, \text{MB})$ *is a primal weak problem for any finite* $T \subset \mathbb{R}^d$*, however, the induced problems* $(U, \geq, w^U)$*,* $U \subseteq T$*,* need not *be LP-type.*

problem is 'nondegenerate'): we say that a weak basis $[U, V]$ is *inclusion-minimal* if all weak bases $[U', V']$ with $U' \subset U$ and $V' \subset V$ have a strictly smaller value than $w(U, V)$. Now suppose $[U', V'], [U'', V''] \subseteq [U, V]$ are two different inclusion-minimal weak bases of $[U, V]$; primal nondegeneracy implies that $[U' \cap U, V' \cap V]$ attains value $w(U, V)$ as well, contradicting the inclusion-minimality of (at least one of) the weak bases $[U', V'], [U'', V'']$. Likewise, dual nondegeneracy implies uniqueness of inclusion-maximal weak bases, where a weak basis $[U, V]$ is *inclusion-maximal* if all weak bases $[U', V']$ with $U' \supset U$ and $V' \supset V$ have a strictly larger value than $w(U, V)$.

The significance of primal (dual, respectively) optimality will become clear later when we proof the correctness of Welzl's algorithm, to which they are tailored (see Lemma 2.19). We already mention that these two properties are 'stronger' than LP-type locality in the following sense. If we define for any fixed $V, U \subseteq T$ the two functions

$$\begin{aligned} w_V(X) &:= w(V \cup X, V), \qquad X \subseteq T \setminus V, \\ w^U(X) &:= w(U, U \setminus X), \qquad X \subseteq U, \end{aligned}$$

we can look at the quasiorder problems $(U \setminus V, \leq, w_V)$ and $(U, \geq, w^U)$, which we call the weak problem's *induced quasiorder problems* (and which will encounter again in Sec. 2.2.1). (We already know the problem $(T, \emptyset, \leq, \text{MB}_\emptyset)$, see page 19!) Now if we required monotonicity (as in the definition of weak problems), primal or dual nondegeneracy, and in addition that the above to quasiorder problems are LP-type (i.e., to fulfill locality) then the resulting structure *need not fulfill primal and dual optimality.* An example is Fig. 2.8 again, where it is easily verified that both quasiorder problems are LP-type, yet primal optimality is violated for $[U, V] = [\{②\}, \emptyset]$, $J = \{②\}$, and $x = ②$, and dual optimality, too (take the same cube $[U, V]$ and x, and set $I = \emptyset$).

On the other hand, primal and dual optimality are *not* stronger than LP-type locality in general, as problem SEBP shows: $(T, \leq, \Omega_{\text{MB}}, \text{MB})$ is a primal weak LP-type problem (Lemma 3.21), and here, the induced problems $(U, \geq, w^U)$, $U \subseteq T$, *need not* be LP-type. In Fig. 2.9, for instance, the four points $U = T$ give an induced problem where locality is not fulfilled. If we drop only one point from the second argument of $\text{MB}(U, U)$ then the ball does not change, but dropping $\{p_2, p_4\}$ does!—In fact, it is precisely the *feature* of LP-type problems that they do not require such locality. In contrast, the induced problems of the *strong LP-type problems* we will encounter in the next section are *always* LP-type, and therefore only instances of SEBP in 'general position' can be formulate as strong LP-type problems.

2.5.1 Reducibility

In the above formulation of SEBP, all faces $[U, \emptyset]$, $U \subseteq T$, have a weak basis that is a *vertex*, i.e., zero-dimensional subcube. As a matter of fact, *every* face (including *infeasible* faces, i.e., faces with value $\bowtie$) has a vertex as a weak basis (Lemma 3.11 in the next chapter). This, however, need not be the case in general (Fig. 2.8), but if it is, we speak of *reducible* weak LP-type problems.

Definition 2.16. *A function* $w : F(C^{[T,\emptyset]}) \to \Omega$ *is called* reducible *if for all* $[U, V] \in 2^T$ *and every* $x \in U \setminus V$ *we have*

$$w(U, V) \in \{w(U \setminus \{x\}, V), w(U, V \cup \{x\})\}.$$

A weak LP-type problem (T, w) *is called* reducible *if* w *is reducible.*

Reducibility has a nice interpretation in terms of cubes and subcubes. The cube $[U, V]$ from the definition can be 'divided' along the 'direction' $x \in U \setminus V$, leaving us with the facet $[U \setminus \{x\}, V]$ (all whose vertices do not contain x) and the facet $[U, V \cup \{x\}]$ (whose vertices contain x). All edges in the cube $[U, V]$ containing x are between these two subcubes (see Fig. 2.10). Reducibility now says that the value of a subcube is attained by at least one of its facets, regardless along which label you divide.

Reducibility implies that all weak bases in (T, w) are vertices. This can very easily be seen using induction: given a face $[X, Y]$ of $[U, V] \subseteq$

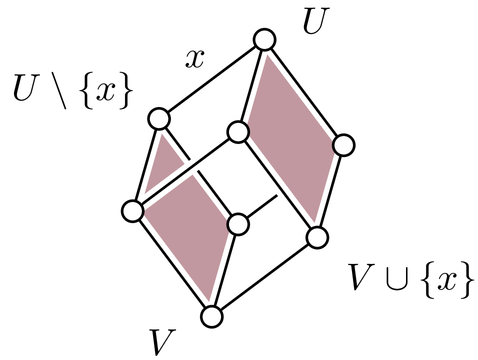


Figure 2.10. *Reducibility in* (T, w) *means that the value of any subcube* $[U, V]$ *of* $C^{[T,\emptyset]}$ *is attained by one of the subcube's facets, regardless of the label* $x \in U \setminus V$ *you choose in order to spilt the subcube into facets.*

2^T, apply reducibility $|X \setminus Y|$ times to it. Each invocation reduces the dimension $\dim([X, Y]) = |X \setminus Y|$ of the current face by one (as we jump from a face to one of its facets) while spanning the same value, so that eventually we arrive at a vertex (i.e., $X = Y$).

Welzl's algorithm. We now turn to algorithm welzl from Fig. 2.7. Let us first outline why it computes a *weak* basis; a detailed correctness proof showing that it returns a *strong* basis is given below. In particular, this will settle that strong bases exist.

In computing a weak basis J of $[U, V]$, we assume that J is a *small* set. The chances are then high that a random $x \in U \setminus V$ is not contained in J (more precisely, that x is not contained in all weak basis of $[U, V]$). Thus, we first drop x, recursively computing a weak basis I of the facet $[U \setminus \{x\}, V]$. Subsequently, we check whether x is *infeasible for* I, that is, whether

$$w(I \cup \{x\}, I) > w(I, I); \tag{2.2}$$

this is the condition the routine infeasible(x, I) from Fig. 2.7 tests. If (2.2) is not true, we use dual nondegeneracy (as shown in the proof below) to deduce $w(U, V) = w(I, I)$, which proves I to be a weak basis of $[U, V]$. If on the other hand (2.2) holds, we cannot have $w(U, V) = w(I, I)$ and we therefore apply reducibility to $w(U, V) > w(I, I) = w(U \setminus \{x\}, V)$, yielding

$$w(U, V) = w(U, V \cup \{x\}).$$

Thus, all we have to do if (2.2) holds, is to recursively compute a weak basis of $w(U, V \cup \{x\})$ by a call to welzl$(U, V \cup \{x\})$. It follows from these

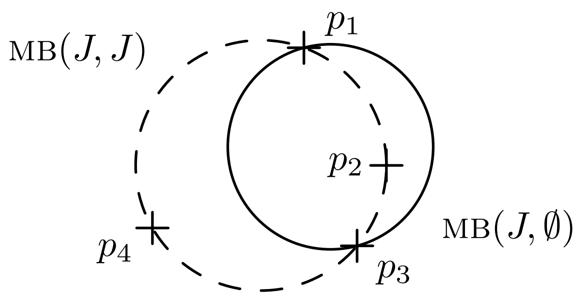


Figure 2.11. *The set* $J = \{p_1, p_2, p_3\}$ *is a weak basis of* (T, MB) *for* $T = J \cup \{p_4\}$*, but the points* J *do* not *span the miniball of* T*.*

observations that algorithm $\mathsf{welzl}(U, V)$ computes a weak basis of $[U, V]$; termination of the algorithm is obvious as the dimension of the face $[U, V]$ passed to the algorithm drops by one in every recursive subcall.

In case of SEBP, the fact that Welzl's algorithm produces a *weak* basis means that $\mathsf{welzl}(T, \emptyset)$ might produce either of $\{p_1, p_2, p_3\}$, $\{p_1, p_2, p_4\}$, $\{p_1, p_3, p_4\}$, or T for the four points T depicted in Fig. 2.11. However, the first of these weak bases does (as a set of points) *not* span the miniball $\text{MB}(T)$ (but the smaller ball drawn in solid instead). Clearly, we would prefer $\{p_1, p_2, p_3\}$, which fulfills the additional property of being inclusion-minimal with $w(J, J) = w(J, \emptyset) = w(T, J)$. And, yes, algorithm welzl auto-magically computes a weak basis with this property!

Lemma 2.17. *In a reducible primal weak problem* (T, w)*,* $\mathsf{welzl}(U, V)$ *finds an inclusion-minimal strong basis of any feasible face* $[U, V] \subseteq 2^T$*.*

Proof. We prove the claim by induction on $m := |U \setminus V|$, which is the dimension of the face $[U, V]$. If $U = V$, the algorithm returns $U = V$, which is clearly an inclusion-minimal strong basis of $[U, V]$.

If $m > 0$, the algorithm calls itself on the facet $[U \setminus \{x\}, V]$ for some $x \in U \setminus V$. As this face has dimension smaller than m and $w(U \setminus \{x\}, V) \leq w(U, V) < \bowtie$, the induction hypothesis applies and the subcall returns an inclusion-minimal $I \in [U \setminus \{x\}, V]$ with

$$w(I, I) = w(U \setminus \{x\}, V) = w(I, V) = w(U \setminus \{x\}, I). \qquad (2.3)$$

Two cases may occur now, depending on whether the feasibility test reports a violation or not. We claim that $w(I \cup \{x\}, I) = w(I, I)$ if and only if $w(U, V) = w(U \setminus \{x\}, V)$. To see the implication ($\Rightarrow$) of this, we use dual nondegeneracy, applied to the faces $[I \cup \{x\}, I]$ and $[U \setminus \{x\}, I]$ (which share the same value), yielding $w(I, I) = w(U, I)$. Then, however,

$$w(U, I) \geq w(U, V) \geq w(U \setminus \{x\}, V) = w(I, I) = w(U, I),$$

which using uniqueness shows $w(U, V) = w(U \setminus \{x\}, V)$. For the other direction ($\Leftarrow$), we invoke dual nondegeneracy on the faces $[U, V]$ and $[I, I]$ whose values agree under $w(U, V) = w(U \setminus \{x\}, V)$. This gives $w(U, I) = w(I, I)$ from which $w(I \cup \{x\}, I) = w(I, I)$ follows via monotonicity and uniqueness. We conclude that $w(I \cup \{x\}, I) = w(I, I)$ if and only if $w(U, V) = w(U \setminus \{x\}, V)$, and as a byproduct we obtain that the former condition also implies $w(I, I) = w(U, I)$ (see proof of ($\Rightarrow$) above).

Consider the case when the infeasibility test reports no violation. Then $w(U, V) = w(U \setminus \{x\}, V)$, which together with (2.3) and $w(I, I) = w(U, I)$ establishes

$$w(U, V) = w(I, I) = w(I, V) = w(U \setminus \{x\}, I) = w(U, I),$$

so I as a strong basis of $[U, V]$. Inclusion-minimality of I is obvious.

If the feasibility test yields $w(I \cup \{x\}, I) > w(I, I)$, we must have $w(U, V) > w(U \setminus \{x\}, V)$, so reducibility yields $\bowtie > w(U, V) = w(U, V \cup \{x\})$. The algorithm now invokes itself on $[U, V \setminus \{x\}]$, and since this face is feasible and has dimension smaller than m, the result is an inclusion-minimal $J \in [U, V \cup \{x\}]$ with

$$w(U, V \cup \{x\}) = w(J, J) = w(J, V \cup \{x\}) = w(U, J).$$

This shows that J is a strong basis of $[U, V]$, provided we can demonstrate $w(J, J) = w(J, V)$. The latter equality, however, follows from primal optimality applied to J and $w(U, V) > w(U \setminus \{x\}, V)$.

Finally, suppose there is a strong basis J' of $[U, V]$ with $J' \subset J$. As J is by induction an inclusion-minimal strong basis of $[U, V \cup \{x\}]$, the set J' cannot contain x; if it did we would have

$$w(U, V \cup \{x\}) = w(U, V) = w(J', J') = w(U, J') = w(J', V),$$

and since $w(J', V) = w(U, V \cup \{x\}) \geq w(J', V \cup \{x\}) \geq w(J', V)$, the above equation proves J' to be a smaller strong basis of $[U, V \cup \{x\}]$, contradiction. So $x \notin J'$ and therefore

$$w(U, V) > w(U \setminus \{x\}, V) \geq w(J', V) = w(U, V)$$

which uniqueness exposes as a contradiction. □

```
procedure welzl-dual(U, V)
{ Computes a strong basis of [U, V] }
{ Precondition: V ⊆ U, w(U, V) }
begin
    if U = V then
        return V
    else
        choose x ∈ U \ V uniformly at random
        J:= welzl(U, V ∪ {x})
        if loose(x, J) then
            return welzl(U \ {x}, V)
        else
            return J
end welzl
```

Figure 2.12. *Welzl's dual algorithm for solving a reducible dual weak problem (T, w). The solution is obtained by calling* welzl$(T, \emptyset)$.

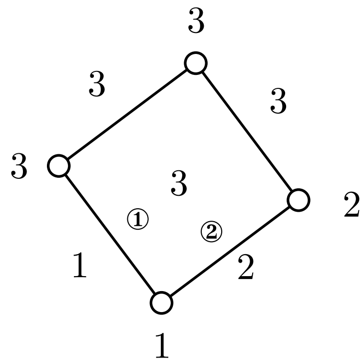
In order to solve a reducible *dual* weak LP-type problem,, we can employ a *dual* version of Welzl's algorithm, see Fig. 2.12. It uses the primitive loose(x, J) which for a given strong basis J and $x \notin J$ returns 'yes' if and only if

$$w(J, J \setminus \{x\}) < w(J, J),$$

and 'no' otherwise. Given this, the above proof (if 'dualized' appropriately) can be reused to show the following

Lemma 2.18. *In a reducible dual weak problem (T, w),* welzl-dual(U, V) *finds an inclusion-maximal strong basis of any bounded face $[U, V] \subseteq 2^T$.*

We note that if it is known in advance that a strong basis of $[T, \emptyset]$ contains many elements then welzl-dual is preferable to algorithm welzl. It is also possible to follow a mixed strategy that throws a coin and depending on the result first visits the upper or lower facet, see for instance algorithm ForceOrNot in [40]. (Algorithm welzl-dual does not work in general for SEBP since primal nondegeneracy need not hold: in Fig. 2.9 we have $\text{MB}(V, V) = \text{MB}(V', V')$ for $V = \{p_1, p_2, p_3\}$ and $V' = \{p_2, p_3, p_4\}$, but $\text{MB}(V \cap V', V \cap V')$ is a smaller ball.)



U vs. V	$\emptyset$	$\{①\}$	$\{②\}$	$\{①, ②\}$
$\emptyset$	1			
$\{①\}$	1	3		
$\{②\}$	2		2	
$\{①, ②\}$	3	3	3	3

Figure 2.13. *The pair (T, w) with $T = \{①, ②\}$ and w as in the table fulfills all requirements of a reducible primal weak problem except primal optimality. Welzl's algorithm fails to produce a strong basis of $[T, \emptyset]$.*

Reducibility and optimality. As we know from the correctness proof of algorithm welzl, primal optimality ensures that the second recursive call (if taken at all) returns a weak basis that is *strong*. If we drop primal optimality from the definition of a primal weak problem, it may indeed happen that algorithm welzl fails in this respect; Fig. 2.13 attests this.

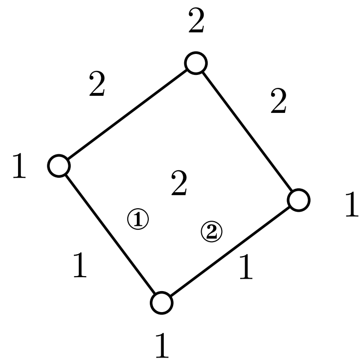
The configuration in the figure shows a set of values w for the faces of the 2-cube $C^{[T,\emptyset]}$, where $T = \{①, ②\}$. It is a simple matter to check that (T, w) satisfies all conditions of a primal weak problem except primal optimality, and that also reducibility applies. Nonetheless, Welzl's algorithm does not compute a strong basis of $[T, \emptyset]$ as we can easily convince ourselves. Assume that in the initial call welzl$(T, \emptyset)$, the algorithm decides to drop constraint ① and recursively computes $w(\{②\}, \emptyset)$, which turns out to be smaller than $w(T, \emptyset)$ according to the table. Therefore, reducibility implies

$$w(\{①, ②\}, \emptyset) = w(\{①, ②\}, \{①\}),$$

and welzl computes a strong basis of $[\{①, ②\}, \{①\}]$, namely $J = \{①\}$ which satisfies $w(\{①, ②\}, \{①\}) = w(J, J) = w(\{①\}, \{①\})$. Back in the call welzl$(T, \emptyset)$, the algorithm returns J as the solution of the whole problem. However, we can read off the table in Fig. 2.13 that

$$w(T, \emptyset) = w(J, J) = 3 > 1 = w(J, J \setminus \{①\}) = w(J, \emptyset),$$

showing that J is *not* a strong basis of $[T, \emptyset]$ (and that dual optimality and hence the last part of the proof of Lemma 2.17 fails). Thus, welzl



U vs. V	$\emptyset$	$\{①\}$	$\{②\}$	$\{①,②\}$
$\emptyset$	1			
$\{①\}$	1	1		
$\{②\}$	1		1	
$\{①,②\}$	2	2	2	2

Figure 2.14. *The pair* (T, w) *with* $T = \{①, ②\}$ *and* w *as in the table is a weak LP-type problem except that dual nondegeneracy does not hold. Algorithm* welzl$(T, \emptyset)$ *fails to compute a strong basis.*

need not produce a strong basis if the problem does not exhibit dual optimality (although it *does* produce a weak basis also in this case).

Given this example, we see that the primal algorithm 'requires' primal optimality in order to work. Recall however, that welzl does not at all rely on primal nondegeneracy. But in fact, the required primal optimality is, under reducibility, just a special case of primal nondegeneracy (and similarly, dual optimality is a consequence of dual nondegeneracy and reducibility).

Lemma 2.19. *Under reducibility, primal (dual, respectively) nondegeneracy implies primal (dual, respectively) optimality.*

Proof. We prove that reducibility and primal nondegeneracy imply primal optimality; the other case is proved similarly. If $J \in [U, V \cup \{x\}]$ is a strong basis of the face $[U, V \cup \{x\}]$ for some $x \in U \setminus V$, then in particular $w(U, V \cup \{x\}) = w(J, J)$. If in addition $w(U, V) > w(U \setminus \{x\}, V)$ holds, reducibility yields $w(U, V) = w(J, J)$. By applying primal nondegeneracy to these two faces we obtain $w(J, J) = w(J, V)$ as needed. □

Considering this lemma, it seems natural to look at problems where *both* variants of optimality hold. We will do this in the next section when we study (reducible) strong LP-type problems.

Remarks. Are *all* requirements in the definition of a primal weak problem necessary in order for algorithm welzl to return an inclusion-minimal strong basis? Reducibility cannot be circumvented as strong bases need not exist otherwise. Also, primal optimality cannot be dropped as the example in Fig. 2.13 shows, and neither can dual nondegeneracy: the instance in Fig. 2.14 is a weak problem except for dual nondegeneracy. If the algorithm drops ① in the initial call welzl$(T, \emptyset)$, it recursively finds the strong basis $I = \emptyset$ of $[\{②\}, \emptyset]$. At this point the feasibility test reports $w(I \cup \{①\}, I) = w(I, I)$, which from the point of view of the algorithm is a 'lie' because $w(T, \emptyset) > w(I, I)$. So the result of the whole run is I, which is not even a weak basis of $[T, \emptyset]$.

Thus, the requirements in the definition of a weak problem are indeed all needed. However, there might exist more appealing properties under which welzl computes strong bases. We do not think that the presented class of problems represents the ultimate answer to the question 'what (nice properties) does algorithm welzl need in order compute strong bases?'

2.6 Strong LP-type problems

We finally turn to the already mentioned link between LP-type problems and unique sink orientations. For this, we consider a special subclass of weak LP-type problems, so-called *strong (LP-type) problems* which were introduced by Gärtner [36].

Definition 2.20. *A tuple* $(T, \leq, \Omega, w)$ *is a* strong problem *if* T *is a finite set,* $\leq$ *is a quasiorder on* Ω*, and* $w : F(C^{[T,\emptyset]}) \to \Omega$ *satisfies the following conditions for all* $[U', V'], [U, V] \subseteq 2^H$*.*

(i) $w(U', V') \leq w(U, V)$ *for all* $U' \subseteq U$*,* $V' \subseteq V$ (monotonicity),

(ii) If $U' \subseteq U \subseteq T$ *and* $V \subseteq T$ *then* $w(U, V) \leq w(U', V)$ *implies* $w(U, V) = w(U', V)$ (upper uniqueness).

(iii) If $V' \subseteq V \subseteq T$ *and* $U \subseteq T$ *then* $w(U, V) \leq w(U, V')$ *implies* $w(U, V) = w(U, V')$ (lower uniqueness).

(iv) $w(U', V') = w(U, V)$ *iff* $w(U' \cap U, V' \cap V) = w(U' \cup U, V' \cup V)$ (strong locality).

The goal of a strong LP-type problem is to find a strong basis.

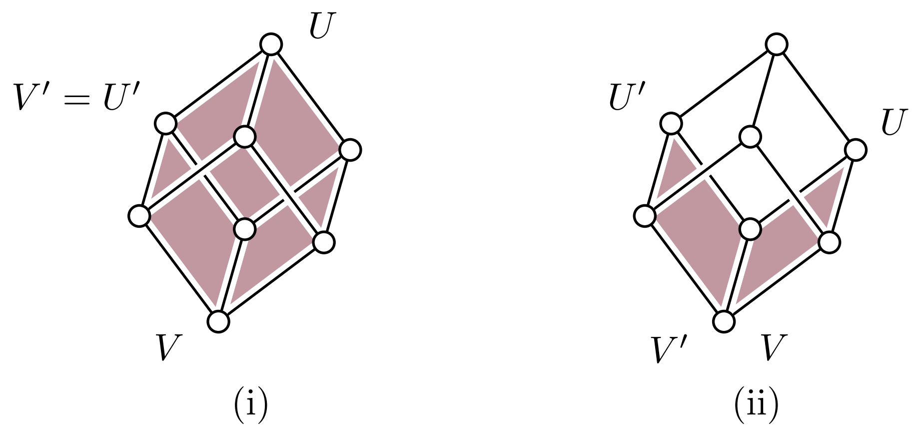


Figure 2.15. *The two interesting cases of strong locality among* $[U', V']$ *and* $[U, V]$*: (i) one face is a subface of the other, (ii) the faces intersect.*

Observe in the definition of strong locality that the direction ($\Leftarrow$) already follows from monotonicity. For, by applying it four times we obtain

$$w(U' \cap U, V' \cap V) \leq w(U', V'), w(U, V) \leq w(U' \cup U, V' \cup V),$$

and if the outer values agree, all of them must, which can be seen using upper and lower uniqueness as follows: from

$$w(U' \cap U, V' \cap V) \leq w(U', V' \cap V) \leq w(U' \cup U, V' \cup V),$$

upper uniqueness, and the fact that the outer values are identical we conclude that $w(U', V' \cap V)$ equals $w(U' \cap U, V' \cap V)$. Given this, we use lower uniqueness in

$$w(U', V' \cap V) \leq w(U', V') \leq w(U' \cup U, V' \cup V)$$

(where again the outer values agree) to obtain $w(U', V') = w(U', V' \cap V) = w(U' \cap U, V' \cap V)$. In a similar fashion, one can prove that the remaining inequalities are indeed equalities.

We can observe that if $U' \subseteq U$ and $V' \subseteq V$, strong locality does not yield anything at all, and the same holds by symmetry if $U' \supseteq U$ and $V' \supseteq V$. The interesting cases are when one among the two faces is a subface of the other or when the two faces have nonempty symmetric difference. A particular instance of the former case is shown in Fig. 2.15(i). Here, the face $[U', V']$ is a vertex $J = U' = V'$ and $[U, V]$

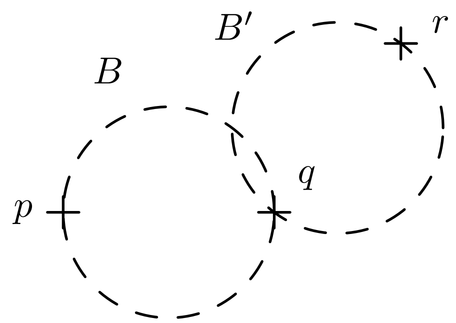


Figure 2.16. *(T, w) with $w(U, V) := \rho_{\mathrm{MB}(U,V)}$ and $T := \{p, q, r\}$ is not a strong LP-type problem: we have $\rho_{\mathrm{MB}(\{p,q\},\emptyset)} = \rho_{\mathrm{MB}(\{q,r\},\emptyset)}$, but this does not imply $\rho_{\mathrm{MB}(\{p,q,r\},\emptyset)} = \rho_{\mathrm{MB}(\{q\},\emptyset)}$.*

is a cube containing it. If these two faces share the same value, strong locality tells us that the outer values of

$$w(J, V) \leq w(J, J), w(U, V) \leq w(U, J)$$

coincide, and thus (as we have seen above in general) all four values are equal. Thus, in the particular case that a cube spans the same value as one of its vertices, all three cubes in the 'chain' $[J, V]$, $[J, J]$, $[U, J]$ span this value. An example of the latter case is shown in Fig. 2.15(ii), where the two-dimensional faces $[U', V']$ and $[U, V] = [U, V']$ intersect in a one-dimensional face $F = [U \cap U', V]$. According to strong locality, F and the whole cube span the same value.

It is crucial here that we work with a *quasiorder* $\leq$ (and not with a total order as in [36]): in case of SEBP, for instance, the pair (T, w) with

$$w(U, V) := \rho_{\mathrm{MB}(U,V)}, \qquad V \subseteq U \subseteq T, \tag{2.4}$$

is *not* a strong LP-type problem as the example in Fig. 2.16 shows. Here, two faces $[\{p, q\}, \emptyset]$ and $[\{q, r\}, \emptyset]$ share the same value, but the *underlying balls are different*. If $w(U, V)$ is defined to be the *ball* MB(U, V), the points from the figure fulfill strong locality (we will prove this in Lemma 3.21), with the definition from (2.4) however, they fail it. Observe that the groundset of this problem is not degenerate (e.g., affinely dependent).

Lemma 2.21. *A reducible strong problem is primal and dual weak.*

Notice that there are weak problems that are *not* strong (SEBP, for instance) and that there are strong problems that are neither reducible nor weak (and example for the latter is the cube $C^{[T,\emptyset]}$ for $T = \{①, ②\}$ whose faces all have value 1 except $w(\{②\}, \{②\}) = 2$, $w(\{①, ②\}, \{②\}) = 3$, and $w(\{①, ②\}, \{①, ②\}) = 4$).

Proof. Weak monotonicity and strong monotonicity are identical. Also, strong locality implies primal and dual nondegeneracy, which in turn yield primal and dual optimality via reducibility (Lemma 2.19). □

Induced quasiorder problems. We have seen on page 35 that every value function w on the faces of a cube with satisfies monotonicity comes with two *induced quasiorder problems*. If w is the value function of a strong problem (T, w), these quasiorder problems are in fact LP-type problems. For the problem $(U \setminus V, \leq, w_V)$, $[U, V] \subseteq 2^T$, this can be seen as follows: monotonicity is inherited from strong monotonicity and if $w_V(X) = w_V(X')$ for $X' \subseteq X \subseteq U \setminus V$ then $w_V(X' \cup \{x\}) = w_V(X')$ implies, using strong locality, $w_V(X \cup \{x\}) = w_V(X')$ as needed. (The proof for the other problems proceeds along the same lines.)

Link to USOs. Finally, here is the link between strong problems and the unique sink orientations from Sec. 2.4.

Theorem 2.22 (Gärtner)**.** *Let (T, w) be reducible strong LP-type problem. For $J \in 2^T$, $x \in T \setminus J$ orient the edge $\{J, J \cup \{x\}\}$ of $C^{[T,\emptyset]}$ via*

$$J \xrightarrow{\phi} J \cup \{x\} \qquad \Leftrightarrow \qquad w(J, J) < w(J \cup \{x\}, J).$$

Then ϕ is a unique sink orientation, and its global sink J is inclusion-minimal with $w(J, J) = w(T, \emptyset)$.

Proof. Consider a face $[U, V]$ of the cube $C^{[T,\emptyset]}$. Below we will show the a vertex $J \in [U, V]$ is a sink in $[U, V]$ if and only if J is inclusion-minimal with $w(J, J) = w(U, V)$. From this it follows that each face $[U, V]$ *has* a sink; just take some inclusion-minimal basis. Also, there cannot be more than one sink in $[U, V]$, for if $w(J, J) = w(J', J')$ then $w(J, J) = w(J \cap J', J \cap J')$ by strong locality, which implies $J = J'$ as both were inclusion-minimal with this value.

The vertex J is a sink in $[U, V]$ if and only if all edges of ϕ are incoming, which in turn is equivalent to

(i) $w(J, J) = w(J, J) = w(J \cup \{x\}, J)$ for all $x \in U \setminus B$, and

(ii) $w(J \setminus \{x\}, J \setminus \{x\}) \neq w(J, J \setminus \{x\})$ for all $x \in J \setminus V$.

Using reducibility, (ii) implies

(ii') $w(J, J \setminus \{x\}) = w(J, J)$ for all $x \in J \setminus V$,

and it follows from strong locality of the functions w_J and w^J that (i) and (ii') are equivalent to

(a) $w(J, J) = w(U, J)$,

(b) $w(J, J) = w(J, V)$.

Invoking strong locality of w, the latter two conditions are equivalent to $w(J, J) = w(U, V)$. Also, $[J\setminus\{x\}, J\setminus\{x\}]$ does not span the value $w(J, J)$ as we see from (ii) combined with (ii'). Thus, J is inclusion-minimal as needed.

Conversely, if J is inclusion-minimal with $w(J, J) = w(U, V)$ then

$$w(J \setminus \{x\}, J \setminus \{x\}]) \neq w(J, J) \tag{2.5}$$

for all $x \in J \setminus V$. From $w(J, J) = w(U, V)$ and monotonicity it follows that (a) and (b) hold which, as we have shown, are equivalent to (i) and (ii'). Now (ii') implies (ii), for if (ii) held with equality, it and (ii') would contradict (2.5). Thus, (i) and (ii) hold showing that J is indeed a sink in $[U, V]$. □

In case of SEBP, the orientation from the above theorem has the following interpretation (which we already encountered in the introduction, see Fig. 1.2). Sitting at a vertex $J \subseteq T$, we orient the edge $\{J, J \cup \{x\}\}$ towards the vertex $J \cup \{x\}$ if and only if the ball $\text{MB}(J, J)$ does not contain the point x (and thus $[J, J]$ cannot be a basis of T).

We remark that the converse question, whether a given unique sink orientation comes from a reducible strong LP-type problem, has been addressed by Schurr [73].

Chapter 3

Properties of the smallest enclosing ball

In this chapter we introduce the problem SEBB of finding the smallest enclosing ball—the *miniball*—of a set of balls. We prove some basic properties of the miniball which will help us in the following chapters when we consider the problem of actually computing it. In particular, we show that SEBB fits into the LP-type framework from the previous chapter. Also, we briefly address a variant of SEBB in which the goal is to find the smallest 'superorthogonal' ball.

Throughout this chapter we will stick to balls with *nonnegative radius*. Later on, we will generalize the properties of SEBB from this chapter also to negative balls; please refer to Chap. 5 for more information.

3.1 The problem

A d-dimensional ball with center $c \in \mathbb{R}^d$ and nonnegative radius $\rho \in \mathbb{R}$ is the point set $B(c,\rho) = \{x \in \mathbb{R}^d \mid \|x-c\|^2 \leq \rho^2\}$, and we write c_B and ρ_B to denote the center and radius, respectively, of a given ball B. We say that a ball is *proper* if its radius is nonzero.

Ball $B' = B(c',\rho')$ is contained in ball $B = B(c,\rho)$ if and only if

$$\|c-c'\| \leq \rho-\rho', \tag{3.1}$$

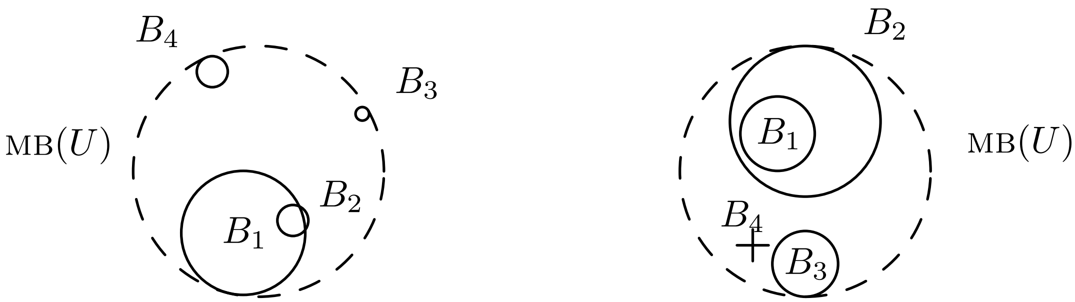


Figure 3.1. *Two examples in the plane* $\mathbb{R}^2$ *of the miniball* MB(U) *for a set* $U = \{B_1, \ldots, B_4\}$ *of four balls.*

with equality if and only if B' is *internally tangent* to B.

We define the *miniball* MB(U) of a finite set U of balls in $\mathbb{R}^d$ to be the unique ball of smallest radius which contains all balls in U (Fig. 3.1). We also set MB($\emptyset$) $= \emptyset$ (note that MB($\emptyset$) is not a ball). The next lemma shows that MB(U) is well-defined.

Lemma 3.1. *For a finite nonempty set* U *of balls, there is a unique ball of smallest radius that contains all balls of* U.

For the proof of this, we make use of *convex combinations of balls* [6, 86, 21], a concept we will also need later on: a proper ball $B = B(c, \rho)$ can be written as the set of points $x \in \mathbb{R}^d$ satisfying $f_B(x) \leq 1$ for $f_B(x) = \|x - c\|^2/\rho^2$. For any $\lambda \in [0, 1]$, the *convex combination* B_λ of two intersecting balls B, B' is the set of points x fulfilling

$$f_{B_\lambda}(x) = (1 - \lambda) f_B(x) + \lambda f_{B'}(x) \leq 1;$$

it has the following properties.

Lemma 3.2. *Let* $B_0, B_1 \subseteq \mathbb{R}^d$ *be two different intersecting balls. Then for any* $\lambda \in [0, 1]$ *the convex combination* B_λ *of* B_0 *and* B_1 *satisfies:*

(i) B_λ *is a ball.*

(ii) $B_0 \cap B_1 \subseteq B_\lambda$, *and* $\partial B_0 \cap \partial B_1 \subseteq \partial B_\lambda$.

(iii) For $\lambda \in (0, 1)$ *the radius of* B_λ *is smaller than* $\max\{\rho_{B_0}, \rho_{B_1}\}$.

Here, '∂B' denotes the boundary of a ball B. Please refer to Fig. 3.2 for an illustration of the lemma.

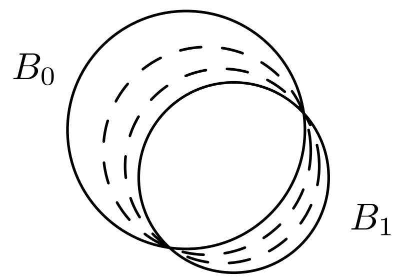


Figure 3.2. *Two convex combinations B_λ (dashed) of the balls B_0 and B_1 (solid), for $\lambda \in \{1/3, 2/3\}$.*

Proof. Consider the defining functions $f_{B_0} = \|x - c_0\|^2/\rho_{B_0}$ and $f_{B_1} = \|x - c_1\|^2/\rho_{B_1}$ of the balls B_0 and B_1. Expanding $f_{B_\lambda} \le 1$ we obtain

$$f_{B_\lambda} = x^T x \left(\frac{1-\lambda}{\rho_{B_0}} + \frac{\lambda}{\rho_{B_1}}\right) - 2x^T \left(\frac{1-\lambda}{\rho_{B_0}}\, c_{B_0} + \frac{\lambda}{\rho_{B_1}}\, c_{B_1}\right) + \alpha \le 1,$$

for some $\alpha \in \mathbb{R}$. This we can write in the form $\|x - c\|^2/\gamma \le 1$ by setting

$$c = \left(\frac{1-\lambda}{\rho_0}\, c_{B_0} + \frac{\lambda}{\rho_{B_1}}\, c_{B_1}\right) \Big/ \left(\frac{1-\lambda}{\rho_{B_0}} + \frac{\lambda}{\rho_{B_1}}\right). \tag{3.2}$$

Since $B_0 \cap B_1 \ne \emptyset$, there exists at least one real point y for which both $f_{B_0}(y) \le 1$ and $f_{B_1}(y) \le 1$ hold. It follows that

$$f_{B_\lambda}(y) = (1-\lambda)\, f_{B_0}(y) + \lambda f_{B_1}(y) \le 1. \tag{3.3}$$

So $\|x - c\|^2/\gamma \le 1$ has a real solution and we see from this that $\gamma \ge 0$, which in particular proves (i). Property (ii) is obvious from (3.3).

(iii) We distinguish two cases: if $\partial B_0 \cap \partial B_1$ is empty, then $B_0 \cap B_1 \ne \emptyset$ implies that one ball, B_0, w.l.o.g., is contained in the interior of B_1. So $f_{B_0}(y) \le 1$ implies $f_{B_1}(y) < 1$ for all $y \in \mathbb{R}^d$. It follows from this that whenever $f_{B_\lambda}(y) \le 1$ for $y \in \mathbb{R}^d$ and $\lambda \in [0, 1]$ (which by (3.3) implies $f_{B_0}(y) \le 1$ or $f_{B_1}(y) \le 1$) then also $f_{B_1}(y) \le 1$, and moreover that whenever $f_{B_\lambda}(y) = 1$ for $\lambda \in (0, 1)$ (which again implies $f_{B_0}(y) \le 1$ or $f_{B_1}(y) \le 1$) then $f_{B_1}(y) < 1$. So $B_\lambda \subset B_1$ for all $\lambda \in (0, 1)$; in particular, the radius of B_λ must be smaller than ρ_{B_1}.

If the intersection of the boundaries is nonempty, we read off from (3.2) that the center c_{B_λ} of B_λ is a convex combination of the centers c_{B_0} and c_{B_1}. That is, as λ varies from 0 to 1, the center c_{B_λ} travels on a line from c_{B_0} to c_{B_1}. Notice now that the radius of B_λ is simply the distance from c_{B_λ} to a point $p \in \partial B_0 \cap \partial B_1$, because by (ii) the point p lies on the boundary of B_λ for any $\lambda \in [0,1]$. The claim now follows from the fact that the distance from p to a point c_{B_λ} moving on a line is a strictly convex function. □

Proof of Lemma 3.1. A standard compactness argument shows that some enclosing ball of smallest radius exists. If this radius is zero, the lemma easily follows. Otherwise, we use Lemma 3.2: assuming there are two distinct smallest enclosing balls, a proper convex combination of them is still enclosing, but has smaller radius, a contradiction. □

We denote by SEBB the problem of computing the center and radius of the ball $\text{MB}(T)$ for a given set T of balls. By SEBP we denote the more specific problem of computing $\text{MB}(T)$ when all balls in T are *points* (radius zero).

3.2 Properties

Optimality criterion. The following optimality criterion generalizes a statement for points due to Seidel [75]. Recall that a point $q \in \mathbb{R}^d$ lies in the convex hull $\text{conv}(P)$ of a finite point set $P \subseteq \mathbb{R}^d$ if and only if $\min_{p \in P} (p-q)^T u \leq 0$ for all unit vectors u, equivalently, if and only if q cannot be separated from $\text{conv}(P)$ by a hyperplane.

Lemma 3.3. *Let V be a nonempty set of balls, all internally tangent to some ball D. Then $D = \text{MB}(V)$ iff $c_D \in \text{conv}(\{c_B \mid B \in V\})$.*

Proof. For direction ($\Leftarrow$), assume $D \neq \text{MB}(V)$, i.e., there exists an enclosing ball D' with radius $\rho_{D'} < \rho_D$. Write its center (which must be different from c_D by the internal tangency assumption) as $c_{D'} = c_D + \lambda u$ for some unit vector u and $\lambda > 0$. Then the distance from $c_{D'}$ to the farthest point in a ball $B \in V$ is

$$\begin{aligned}
\delta_B &= \|c_{D'} - c_B\| + \rho_B \\
&= \sqrt{(c_D + \lambda u - c_B)^T (c_D + \lambda u - c_B)} + \rho_B \\
&= \sqrt{\|c_D - c_B\|^2 + \lambda^2 u^T u - 2\lambda\,(c_B - c_D)^T u} + \rho_B \\
&= \sqrt{(\rho_D - \rho_B)^2 + \lambda^2 - 2\lambda\,(c_B - c_D)^T u} + \rho_B, \qquad (3.4)
\end{aligned}$$

because (3.1) holds with equality by our tangency assumption. Since D' is enclosing, we must have

$$\rho_{D'} \geq \max_{B \in V} \delta_B. \tag{3.5}$$

Furthermore, the observation preceding the lemma yields the existence of $B' \in V$ such that $(c_{B'} - c_D)^T u \leq 0$, for c_D lies in the convex hull of the centers of V. Consequently,

$$\delta_{B'} > \sqrt{(\rho_D - \rho_{B'})^2} + \rho_{B'} = \rho_D > \rho_{D'}$$

by equation (3.4), a contradiction to (3.5).

For direction ($\Rightarrow$), suppose that c_D does not lie in the convex hull of the centers of V. By the observation preceding the lemma, there exists a vector u of unit length with $(c_B - c_D)^T u > 0$ for all $B \in V$. Consider the point $c_{D'} := c_D + \lambda u$, for some strictly positive $\lambda < 2 \min_{B \in V} (c_B - c_D)^T u$. According to (3.4), $\delta_B < (\rho_D - \rho_B) + \rho_B = \rho_D$ for all B, and consequently, the ball D' with center $c_{D'}$ and radius $\max_B \delta_B < \rho_D$ is enclosing, contradiction. □

If we write the center of $\textsc{mb}(V)$ as a convex combination (such a combination exists by the lemma), the involved coefficients fulfill the following simple property.

Corollary 3.4. *Let U be a finite set of balls and let $D = \textsc{mb}(U)$ be a ball of positive radius. If $B' \in U$ is a point (radius zero) internally tangent to D and we write*

$$c_D = \sum_{B \in U} \lambda_B c_B, \qquad \sum_{B \in U} \lambda_B = 1 \tag{3.6}$$

for nonnegative coefficients λ_B, $B \in U$, then $\lambda_{B'} \leq 1/2$.

Proof. Notice first that through the previous lemma, we can indeed write the center c_D of D in the form (3.6). Moreover, we may assume w.l.o.g. that the center of B lies at the origin. Fix $B' \in U$. From (3.6) we obtain $0 = \|\lambda_{B'} c_{B'} + \sum_{B \neq B'} \lambda_B c_B\|$ which using the triangle inequality yields

$$0 \geq \lambda_{B'} \|c_{B'}\| - \| \sum_{B \neq B'} \lambda_B c_B \| \geq \lambda_{B'} \|c_{B'}\| - \sum_{B \neq B'} \lambda_B \|c_B\|.$$

Since $\|c_{B'}\| = \rho_D$ and $\|c_B\| \leq \rho_D$ for all $B \neq B'$, it follows $0 \geq \rho_D\,(\lambda_{B'} - \sum_{B\neq B'} \lambda_B)$. Dividing by $\rho_D > 0$ and plugging in $\sum_{B\neq B'} \lambda_B = 1 - \lambda_{B'}$, we obtain $\lambda_{B'} \leq 1/2$. □

A statement for points going into a similar direction is the following (which we will use in Chap. 4).

Lemma 3.5. *Let $D = B(c, \rho)$ be a ball of positive radius through some finite pointset $V \subset \mathbb{R}^d$. If*

$$c = \sum_{p\in V} \lambda_p p, \qquad \sum_{p\in V} \lambda_p = 1,$$

for real coefficients λ_p then at least two coefficients λ_p are positive.

Proof. W.l.o.g. we may assume that the ball D is centered at the origin and has unit radius, i.e., $\mathbf{0} = c = \sum_{p\in V} \lambda_p p$ and $\|p\| = 1$ for all $p \in V$. Clearly, at least one of the coefficients λ_p, $p \in V$, must be positive. So all we need to show is that some fixed $q \in V$ cannot be the only point with a positive coefficient.

If $\lambda_q \geq 0$ and $\lambda_p < 0$ for all $q \neq p \in V$ then taking the norm on both sides of $(1 - \sum_{p\neq q} \lambda_p)\, q = -\sum_{p\neq q} \lambda_p p$ yields $1 - \sum_{p\neq q} \lambda_p \leq -\sum_{p\neq q} \lambda_p$, a contradiction. □

Another property we will use for our algorithms in Sec. 5.1 is the following intuitive statement which has been proved by Welzl for points [86].

Lemma 3.6. *If a ball $B \in U$ is properly contained in the miniball $\mathrm{MB}(U)$ (that is, not internally tangent to it) then*

$$\mathrm{MB}(U) = \mathrm{MB}(U \setminus \{B\}),$$

equivalently, $B \subseteq \mathrm{MB}(U \setminus \{B\})$.

Proof. Consider the convex combination D_λ of the balls $D = \mathrm{MB}(U)$ and $D' = \mathrm{MB}(U \setminus \{B\})$; it continuously transforms D into D' as λ ranges from 0 to 1 and contains all balls in $U \setminus \{B\}$. Since B is not tangent to $\mathrm{MB}(U)$, there is a $\lambda' > 0$ such that $D_{\lambda'}$ still encloses all balls from U. But if D and D' do not coincide, $D_{\lambda'}$ has smaller radius than D, a contradiction to the minimality of $D = \mathrm{MB}(U)$. □

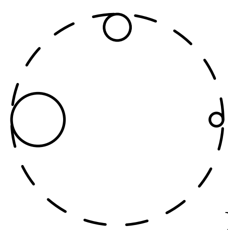


Figure 3.3. *$U = \{B_1, B_2, B_3\}$ is a support set (but not a basis) of U; $V = \{B_1, B_3\}$ is a basis.*

Motivated by this observation, we call a set $U' \subseteq U$ a *support set of U* if all balls in U' are internally tangent to $\textsc{mb}(U)$ and $\textsc{mb}(U') = \textsc{mb}(U)$. An inclusion-minimal support set of U is called *basis of U* (see Fig. 3.3), and we call ball set V a *basis* if it is a basis of itself. (Notice that this is in accordance with the definition of a 'basis' on page 17!) A standard argument based on Helly's Theorem reveals that the miniball is determined by a support set of size at most $d + 1$.

Lemma 3.7. *Let U be a set of at least $d + 1$ balls in $\mathbb{R}^d$. Then there exists a subset $U' \subseteq U$ of $d + 1$ balls such that $\textsc{mb}(U) = \textsc{mb}(U')$.*

Proof. Let $D = \textsc{mb}(U)$ and consider the set $I = \bigcap_{B \in U} B(c_B, \rho_D - \rho_B)$. Observe that $B(c_B, \rho_D - \rho_B)$ is the set of all centers which admit a ball of radius ρ_D that encloses B. By the existence and uniqueness of $\textsc{mb}(U)$, I thus contains exactly one point, namely c_D. It follows that $\bigcap_{B \in U} \operatorname{int} B(c_B, \rho_D - \rho_B) = \emptyset$, where $\operatorname{int} B'$ denotes the interior of ball B'. Helly's Theorem[1] yields a set $U' \subseteq U$ of $d + 1$ elements such that $\bigcap_{B \in U'} \operatorname{int} B(c_B, \rho_D - \rho_B) = \emptyset$. Consequently, no ball of radius $< \rho_D$ encloses the balls U', and thus $\textsc{mb}(U)$ and $\textsc{mb}(U')$ have the same radius. This however implies $\textsc{mb}(U) = \textsc{mb}(U')$, since we would have found two different miniballs of U' otherwise. □

Lemma 3.8. *The centers of a basis V of U are affinely independent.*

Proof. The claim is obvious for $V = \emptyset$. Otherwise, by Lemma 3.3, the center c_D of the miniball $D = \textsc{mb}(V) = \textsc{mb}(U)$ can be written

[1] *Helly's Theorem* [23] states that if $C_1, \ldots, C_m \subset \mathbb{R}^d$ are $m \geq d + 1$ convex sets such that any $d + 1$ of them have a common point then also $\bigcap_{i=1}^m C_i$ is nonempty.

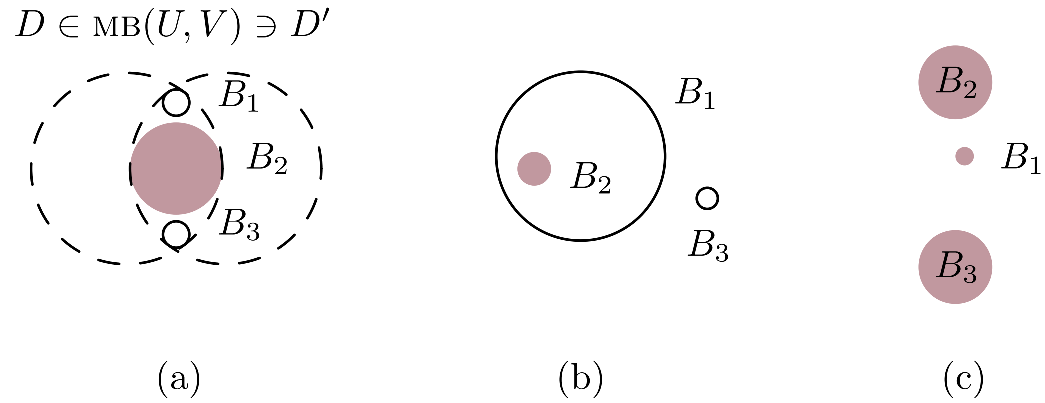


Figure 3.4. $\text{MB}(U,V)$ *may contain several balls (a) or none (b): set* $U = \{B_1, B_2, B_3\}$*,* $V = \{B_2\}$*. (c) shows another example where the set* $\text{MB}(U,U)$ *is empty; here, no ball is contained in another.*

as $c_D = \sum_{B \in V} \lambda_B c_B$ for some coefficients $\lambda_B \geq 0$ summing up to 1. Observe that $\lambda_B > 0$, $B \in V$, by minimality of V. Suppose that the centers $\{c_B \mid B \in V\}$ are affinely dependent, or, equivalently, that there exist coefficients μ_B, not all zero, such that $\sum_{B \in V} \mu_B c_B = \mathbf{0}$ and $\sum \mu_B = 0$. Consequently,

$$c_D = \sum_{B \in V} (\lambda_B + \alpha \mu_B)\, c_B \qquad \text{for any } \alpha \in \mathbb{R}. \tag{3.7}$$

Change α continuously, starting from 0, until $\lambda_{B'} + \alpha \mu_{B'} = 0$ for some B'. At this moment all nonzero coefficients $\lambda'_B = \lambda_B + \alpha \mu_B$ of the combination (3.7) are strictly positive, sum up to 1, but $\lambda'_{B'} = 0$, a contradiction to the minimality of V. □

A generalization. We proceed with some basic properties of '$\text{MB}(U,V)$' which is the following generalization of $\text{MB}(U)$. For sets $U \supseteq V$ of balls, we denote by $\text{B}(U,V)$ the set of balls B that contain the balls U and to which (at least) the balls in V are internally tangent (we set $\text{B}(\emptyset,\emptyset) = \{\emptyset\}$). Based on this, we define $\text{MB}(U,V)$ to be the set of smallest balls in $\text{B}(U,V)$; in case $\text{MB}(U,V)$ contains exactly one ball D, we abuse notation and refer to D as $\text{MB}(U,V)$. Observe that $\text{MB}(U) = \text{MB}(U,\emptyset)$ and hence any algorithm for computing $\text{MB}(U,V)$ solves the SEBB problem. However, several intuitive properties of $\text{MB}(U)$ do not carry over to $\text{MB}(U,V)$: the set $\text{MB}(U,V)$ can be empty, or there can be *several* smallest balls in $\text{B}(U,V)$, see Fig. 3.4. Furthermore, properly contained balls cannot

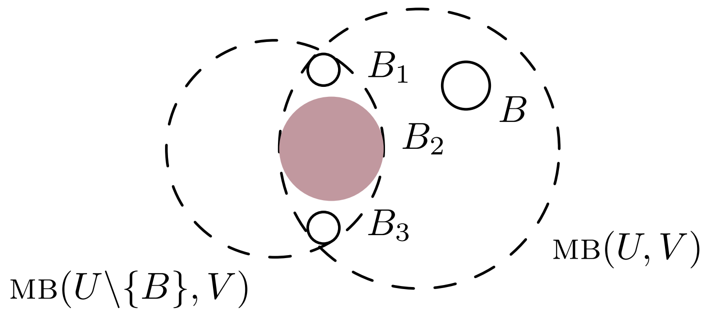


Figure 3.5. *Ball B cannot be dropped although it is properly contained in* MB(U, V)*: set $U = \{B, B_1, B_2, B_3\}$ and $V = \{B_2\}$.*

be dropped as in the case of MB(U) (Lemma 3.6): for a counterexample refer to Fig. 3.5, where MB$(U, V) \neq$ MB$(U\backslash\{B\}, V)$ for $V = \{B_2\}$ and $U = \{B_1, B_2, B_3, B\}$, although B is properly contained in MB(U, V).

In the sequel we will also deal with

$$\text{MB}_p(U) := \text{MB}(U \cup \{p\}, \{p\}), \tag{3.8}$$

where $p \in \mathbb{R}^d$ is some point and U as usual is a set of balls. (In writing $U \cup \{p\}$ we abuse notation and identify the ball $B(p, 0)$ with the point p.) Again the set MB$_p(U)$ may be empty (place p in the interior of the convex hull $\text{conv}(U) := \text{conv}(\bigcup_{B \in U} B)$), but in the nonempty case it contains a unique ball. This follows from

Lemma 3.9. *Let $U \supseteq V$ be two sets of balls, V being a set of points (balls of radius zero). Then* MB(U, V) *consists of at most one ball.*

Proof. If $D, D' \in$ MB(U, V), their convex combination D_λ contains U and in addition has the points V on the boundary. Thus, $D_\lambda \in$ B(U, V) for any $\lambda \in [0, 1]$. If D and D' were distinct, a proper convex combination would have smaller radius than D' or D, a contradiction to the minimality of D, D'. □

Combining a compactness argument as in the proof of Lemma 3.1 with the reasoning from the previous lemma, we can also show the following.

Lemma 3.10. *Let U be a set of balls and $p \in \mathbb{R}^d$ such that no ball in U contains p. Then* $\text{MB}_p(U) = \emptyset$ *iff* $p \in \text{conv}(U) := \text{conv}(\bigcup_{B \in U} B)$.

Without the assumption on U and p, it may happen that $\text{MB}_p(U) \neq \emptyset$ although $p \in \text{conv}(U)$ (take a single ball, $U = \{B\}$, and a point p on its boundary).

In case both sets $U \supseteq V$ in $\text{MB}(U, V)$ are actually pointsets, one can prove some sort of 'reducibility' for the function MB (the proof is taken from the paper [86] introducing Welzl's algorithm).

Lemma 3.11 (Welzl)**.** *Let $V \subseteq U \subseteq T \subset \mathbb{R}^d$ with T finite.*

(i) If there is a ball through V containing U then $|\,\text{MB}(U, V)| = 1$.

(ii) If $D \in \text{MB}(U, V)$ and $x \notin D$ for $x \in U \setminus V$ then $D \in \text{MB}(U, V \cup \{x\})$.

Proof. (i) follows from a standard compactness argument and Lemma 3.9. (ii) Suppose the ball D does not go through x. By (i), there exists $D' \in \text{MB}(U \setminus \{x\}, V)$, and by assumption D' does not contain x. So consider the convex combination D_λ of D and D'. For some $\lambda^* \in (0, 1)$ the ball D_{λ^*} has x on its boundary, goes through V, contains U, and has a smaller radius than D, a contradiction. □

Circumball. An important notion for our method in Chap. 4 is the *circumball* $\text{CB}(T)$ of a nonempty affinely independent set T, which is the unique sphere with center in the affine hull $\text{aff}(T)$ that goes through the points in T. The following lemma shows that $\text{CB}(T)$ is indeed well-defined.

Lemma 3.12. *Given a nonempty affinely independent pointset $T \subset \mathbb{R}^d$, there exists exactly one ball through T whose center lies in* $\text{aff}(T)$.

In the proof of this we use the simple fact that a matrix of the form $A = Q^T Q$ is regular provided the columns of Q are linearly independent. (If $Ax = \mathbf{0}$ then $\mathbf{0} = x^T Q^T Q x = \|Qx\|^2$, and hence $Qx = \mathbf{0}$, a contradiction to the linear independence of the columns of Q.)

Proof. Denote by c the center and by ρ the radius of a ball through T with center in $\text{aff}(T)$. As $c \in \text{aff}(T)$, we can write c in the form

$c = \sum_{t \in T} \lambda_t t$ for real coefficients λ_t, $t \in T$, summing up to 1. We need to show that the system of equations

$$\|c - t\|^2 = \rho^2, \qquad t \in T, \tag{3.9}$$

$$\textstyle\sum_{t \in T} \lambda_t t = c, \tag{3.10}$$

$$\textstyle\sum_{t \in T} \lambda_t = 1, \tag{3.11}$$

has exactly one solution (ρ, λ). To see this, we assume w.l.o.g., that one of the points in T coincides with the origin, i.e., $T = T' \cup \{\mathbf{0}\}$; this can always be achieved via a suitable translation.

By subtracting Eq. (3.9) for $t = \mathbf{0}$ from the remaining Eqs. (3.9), we see that a solution to system (3.9)–(3.11) satisfies

$$t'^T t' - 2c^T t' = 0, \qquad t' \in T', \tag{3.12}$$

$$\textstyle\sum_{t \in T} \lambda_t t = c. \tag{3.13}$$

By plugging the latter of these $|T'| + 1$ equations into the former, we obtain $1/2\, t'^T t' = c^T t' = \sum_{t \in T} \lambda_t t^T t'$ for all $t' \in T'$. In matrix notation, this is equivalent to $b = A\lambda$ where b contains the entries $1/2\, t'^T t'$, $t' \in T'$, and $A_{t't} = t^T t'$. Affine independence of the points T together with $\mathbf{0} \in T$ implies that the points in T are linearly independent. It follows from this that the matrix A is regular, and consequently, there is precisely one solution to $A\lambda = b$. Hence also the system (3.12)–(3.13) has exactly one solution, and by setting $\rho := \|c\|$, any solution of the latter can be turned in a solution of the original system (3.9)–(3.11). □

Linked to this is the following observation (originating from the lecture notes [37]) which allows us to *drop* affinely independent points from the boundary. Observe in the statement that since the sets V and J are pointsets (and not balls), the sets $\text{MB}(J, J)$ and $\text{MB}(V, V)$ contain at most one element each (Lemma 3.9).

Lemma 3.13. *Let $J \subseteq \mathbb{R}^d$ be finite and V be an inclusion-maximal subset of J that is affinely independent. If $D \in \text{MB}(J, J)$ then $D \in \text{MB}(V, V)$.*

Proof. Observe first that as $\text{MB}(J, J) \subseteq \text{MB}(V, V)$ by definition, the latter set is nonempty. So consider $D \in \text{MB}(J, J)$ and $D' \in \text{MB}(V, V)$, and

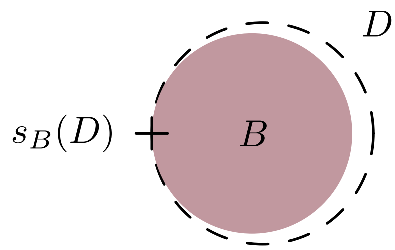


Figure 3.6. *The support point $s_D(B)$ of ball B (filled) w.r.t. a larger ball $D \in \mathrm{B}(\emptyset, \{B\})$ (dashed) is the single point in the set $\partial B \cap \partial D$.*

suppose that $q \in J \setminus V$ is not contained in $\partial D'$. Then

$$\begin{aligned} \rho_D &= \|p - c_D\| = p^T p - 2c_D^T p + c_D^T c_D, & p \in J, \quad (3.14) \\ \rho_{D'} &= \|p - c_{D'}\| = p^T p - 2c_{D'}^T p + c_{D'}^T c_{D'}, & p \in V, \quad (3.15) \\ \rho_{D'} \neq \delta &:= \|q - c_{D'}\| = q^T q - 2c_{D'}^T q + c_{D'}^T c_{D'}, & (3.16) \end{aligned}$$

Now $V' := V \cup \{q\}$ is affinely dependent, so there exist real numbers λ_p, $p \in V'$, not all zero, such that

$$\sum_{p \in V'} \lambda_p p = \mathbf{0}, \qquad \sum_{p \in V'} \lambda_p = 0. \tag{3.17}$$

We must have $\lambda_q \neq 0$ (otherwise V would be affinely dependent), and w.l.o.g. we can assume $\lambda_q > 0$ (scale the equations in (3.17) if necessary). By multiplying (3.14) with λ_p and summing over all $p \in V'$ we now obtain $0 = \sum_{p \in V'} \lambda_p \rho_D = \sum_{p \in V'} \lambda_p (p^T p - 2c_D^T p + c_D^T c_D) = \sum_{p \in V'} \lambda_p p^T p$. On the other hand, (3.17) together with (3.15), (3.16), and $\lambda_q > 0$ gives

$$\sum_{p \in V'} \lambda_p p^T p = \sum_{p \in V'} \lambda_p \|p - c_{D'}\| = \lambda_q \delta + \sum_{p \in V} \lambda_q \rho_{D'} = \lambda_q(\delta - \rho_{D'}),$$

which is nonzero, a contradiction. If follows $\mathrm{MB}(J, J) = \mathrm{MB}(V, V)$. □

Support points. If a ball B is internally tangent to some ball D, we call the points $\partial B \cap \partial D$ the *support points* of B w.r.t. D. Most of the time, we will find ourselves in situations where the ball D is *strictly larger* than B, see Fig. 3.6. In this case it is easy to verify that B has precisely one support point w.r.t. D, namely the point

$$s_D(B) := c_D + \frac{\rho_D}{\rho_D - \rho_B}(c_B - c_D). \tag{3.18}$$

We define $\mathrm{supp}_D(T)$ to be the set of support points of the balls $B \in T$ w.r.t. some ball $D \in \mathrm{B}(T,T)$. In case D is larger than every ball in T we have $\mathrm{supp}_D(T) = \{s_D(B) \mid B \in T\}$.

It does not come as a surprise that the miniball D of a set of balls is determined by the support points of the balls w.r.t. D.

Lemma 3.14. *Let D be a ball enclosing a set U of balls and suppose $D \notin U$. Then $D = \mathrm{MB}(U)$ if and only if $D = \mathrm{MB}(\mathrm{supp}_D(U))$.*

Proof. The assumption $D \notin U$ guarantees that ρ_D is greater than the radius of any ball $B \in U$. In particular, this implies that $\mathrm{supp}_D(U)$ is a finite set and hence $\mathrm{MB}(\mathrm{supp}_D(U))$ is well-defined.

For the direction ($\Rightarrow$) we assume $D = \mathrm{MB}(U)$ and consider some basis $V \subseteq U$ of U. By Lemma 3.3, we can write c_D as a convex combination $c_D = \sum_{B \in V} \lambda_B c_B$ for nonnegative real coefficients λ_B that add up to one. Consequently, we have

$$0 = \sum_{B \in V} \lambda_B \frac{\rho_D - \rho_B}{\rho_D} \frac{\rho_D}{\rho_D - \rho_B}(c_B - c_D) =: \sum_{B \in V} \mu_B c'_B, \tag{3.19}$$

where $c'_B = \rho_D/(\rho_D - \rho_B)(c_B - c_D)$, $B \in V$. By Eq. (3.18), the points $S := \{c_D + c'_B \mid B \in V\}$ constitute a subset of the support points of D, i.e., $S \subseteq \mathrm{supp}_D(U)$, and we claim that $D = \mathrm{MB}(S)$. From this the claim follows as D encloses the union of all balls U, in particular $\mathrm{supp}_D(U)$.

Notice next that the number $\gamma := \sum_{B \in V} \mu_B$ is strictly positive, for $\gamma = 1 - \sum_{B \in V} \lambda_B \rho_B/\rho_D > 1 - \sum_{B \in V} \lambda_B \rho_D/\rho_D = 0$. Thus, adding the term $c_D\gamma$ to both sides of (3.19) and solving for c_D results in

$$c_D = \sum_{B \in V} \frac{\mu_B}{\gamma}(c_D + c'_B) =: \sum_{B \in V} \nu_B (c_D + c'_B), \qquad \sum_{B \in V} \nu_B = 1.$$

This together with $\nu_B \geq 0$, $B \in V$, proves c_D to be a convex combination of the points S. By invoking Lemma 3.3 again, applied to the points S this time, we conclude that $D = \mathrm{MB}(S)$ as needed.

The direction ($\Leftarrow$) is easy: by assumption the ball $D = \mathrm{MB}(\mathrm{supp}_D(U))$ is an enclosing ball. So if there existed a smaller enclosing ball D' than D, this ball must enclose the points $\mathrm{supp}_D(U) \subseteq \bigcup_{B \in U} B$, which would result in a contradiction to the minimality of $D = \mathrm{MB}(\mathrm{supp}_D(U))$. $\square$

3.3 Properties of $\text{MB}(U, V)$

In this section we develop *optimality criteria* for the ball $\text{MB}(U, V)$. In particular, we will see that new effects pop up when we go from enclosed points to enclosed balls.

The point case. In the rest of this section we assume that whenever $V \subseteq U$ are *pointsets* then $\text{MB}(U, V)$ is not a *set* of balls (recall the definition) but the *unique* smallest ball in $\text{B}(U, V)$, provided it exists (Lemma 3.9 shows the uniqueness). If no smallest ball exists, which by Lemma 3.11(i) means that $\text{B}(U, V) = \emptyset$, we set $\text{MB}(U, V)$ to the *infeasible ball* (see page 15), and we say that the 'ball $\text{MB}(U, V)$ does not exist.'

We start with a *mathematical program* which will allow us to compute the ball $\text{MB}(U, V)$; please see page viii for some notation in connection with mathematical programs. So suppose we are given a finite pointset $U \subset \mathbb{R}^d$ and some nonnegative real numbers ρ_B, $B \in U$. (For the purpose of this section you can neglect the numbers ρ_B, i.e., assume $\rho_B = 0$, $B \in U$; we will need them later in Sec. 3.5 for a related geometric problem.) Arrange the Euclidean points $p \in U$ as columns to a $(d \times |U|)$-matrix C and consider the following convex mathematical program in the variables x_p, $p \in U$.

$$
\begin{array}{lll}
\mathcal{Q}(U,V) & \text{minimize} & x^T C^T C x + \sum_{p \in U} x_p \, (\rho_p^2 - p^T p) \\
 & \text{subject to} & \sum_{p \in U} x_p = 1, \\
 & & x_p \geq 0, \qquad p \in U \setminus V.
\end{array}
$$

Lemma 3.15. *Let $V \subseteq U$ be two finite pointsets in $\mathbb{R}^d$, each point coming with a positive real number ρ_p.*

(i) If $\tilde{x}$ is an optimal solution to $\mathcal{Q}(U, V)$ then its objective value is of the form $-\tilde{\rho}^2$ and there exist real number μ_p, $p \in U$, such that

$$
\begin{aligned}
\|\tilde{c} - p\|^2 - \rho_p^2 + \mu_p &= \tilde{\rho}^2, & & (3.20)\\
\tilde{x}_p \mu_p &= 0, \qquad p \in U \setminus V, & & (3.21)\\
\mu_p &\geq 0, \qquad p \in U \setminus V, & & (3.22)\\
\mu_p &= 0, \qquad p \in V, & & (3.23)
\end{aligned}
$$

holds for $\tilde{c} = C\tilde{x}$. Moreover, there is no other solution $(\tilde{c}', \tilde{\rho}')$ of the system (3.20), (3.22)–(3.23) *(in the variables $\tilde{c}, \tilde{\rho}$) with $\tilde{\rho}' \leq \tilde{\rho}$.*

(ii) If $\tilde{x}$ is feasible for $\mathcal{Q}(U, V)$ and Eqs. (3.20)–(3.23) *hold for some real $\tilde{\rho}$ are real μ_p, $p \in U$, with $\tilde{c} = C\tilde{x}$ then $\tilde{x}$ is optimal to $\mathcal{Q}(U, V)$.*

(iii) If (3.20), (3.22)–(3.23) *hold for some real $\tilde{\rho}$, some real vector $\tilde{c}$, and for real values μ_p, $p \in U$, then $\mathcal{Q}(U, V)$ has an optimal solution.*

The proof is follows an argument by Gärtner [34] and the second part is based on an idea by Seidel [75] (just like in the proof of Lemma 3.3).

Proof. As the objective function f of program $\mathcal{Q}(U, V)$ is convex, we can apply the *Karush-Kuhn-Tucker Theorem for Convex Programming* [5], which we use in the variant stated in Theorem 5.16. According to this, a feasible solution $\tilde{x}$ is optimal to $\mathcal{Q}(U, V)$ if and only if there exist real numbers μ_B, $B \in U$, and a real τ such that

$$2p^T C\tilde{x} + \rho_p^2 - p^T p + \tau - \mu_p = 0, \qquad p \in U, \tag{3.24}$$

and $\mu_p \geq 0$, $p \in U \setminus V$, hold with $\mu_p = 0$, $p \in V$, and $\tilde{x}_p \mu_p = 0$, $p \in U \setminus V$.

Set $\tilde{c} := C\tilde{x}$, multiply (3.24) by $\tilde{x}_p$, and sum over all $p \in U$. Using $\sum_{p \in U} \tilde{x}_p = 1$ and $\tilde{x}_p \mu_p = 0$, $p \in U \setminus V$, this yields

$$2\tilde{c}^T \tilde{c} + \sum_{p \in U} \tilde{x}_p \, (\rho_p^2 - p^T p) + \tau = 0,$$

from which we see that $f(\tilde{x}) = -\tilde{c}^T \tilde{c} - \tau$. Given this, we can negate (3.24) and add $\tilde{c}^T \tilde{c}$ on both sides in order to obtain

$$\|\tilde{c} - p\|^2 - \rho_p^2 + \mu_p = \tilde{\rho}^2, \qquad p \in U, \tag{3.25}$$

for $\tilde{\rho}^2 := -f(\tilde{x})$. This shows the first part of claim (i).

To show the second part of (i), suppose there exists a different solution $(\tilde{c}', \tilde{\rho}')$ with $\tilde{\rho}' \leq \tilde{\rho}$ that fulfills (3.20), (3.22), and (3.23). Write $\tilde{c}'$ in the unique form $\tilde{c}' = \tilde{c} + \lambda u$, where u is a unit vector and $\lambda \geq 0$. As $(\tilde{c}', \tilde{\rho}')$ does not coincide with $(\tilde{c}, \tilde{\rho})$, we must have $\lambda > 0$ or $\tilde{\rho}' < \tilde{\rho}$.

Set $F := \{p \in U \mid \tilde{x}_p > 0\}$ (which by the equality constraint of the program is a nonempty set) and recall from the above optimality conditions $\tilde{x}_p \mu_p = 0$ and (3.25) that every point $p \in F$ is fulfills $\|p - \tilde{c}\|^2 = \tilde{\rho}^2 + \tilde{\rho}_p^2$. Consequently,

$$\begin{aligned} \|p - \tilde{c}'\|^2 - \rho_p^2 &= \|p - \tilde{c} - \lambda u\|^2 - \rho_p^2 \\ &= \|p - \tilde{c}\|^2 - \rho_p^2 + \lambda^2 u^T u - 2\lambda u^T (p - \tilde{c}) \\ &= \tilde{\rho}^2 + \lambda^2 - 2\lambda u^T (p - \tilde{c}). \end{aligned}$$

Thus, in order for $(\tilde{c}', \tilde{\rho}')$ to fulfill the system (3.20), (3.22)–(3.23) with some set of real coefficients μ_p, $p \in U$, in such a way that $\tilde{\rho}' < \tilde{\rho}$ or $\lambda > 0$ holds, the number λ would have to be strictly positive and there would have to be a constant $\gamma > 0$ such that

$$\begin{aligned} u^T(p - \tilde{c}) &= \gamma, \quad i \in V, & (3.26) \\ u^T(p - \tilde{c}) &\geq \gamma, \quad i \in F. & (3.27) \end{aligned}$$

Using $\tilde{x}_p \geq 0$, $p \in U \setminus V$, and $\sum_{p \in U} \tilde{x}_p = 1$, we then get

$$\sum_{p \in U} \tilde{x}_p u^T(p - \tilde{c}) \geq \sum_{p \in V} \tilde{x}_p \gamma + \sum_{p \in U \setminus V} \tilde{x}_p \gamma = \gamma > 0.$$

On the other hand, $\sum_{p \in U} \tilde{x}_p u^T(p - \tilde{c}) = u^T(\sum_{p \in U} \tilde{x}_p p - \sum_{p \in U} \tilde{x}_p \tilde{c}) = u^T(\tilde{c} - \tilde{c}) = 0$, a contradiction. This settles (i).

(ii) If $\tilde{x}$ is feasible for $\mathcal{Q}(U, V)$ with numbers μ_p, $p \in U$, and $\tilde{\rho}$ fulfilling the conditions (3.20)–(3.23), we can subtract $\tilde{c}^T \tilde{c}$ from both sides of (3.20) and negate the result in order to arrive at (3.24) for $\tau := \tilde{\rho}^2 - \tilde{c}^T \tilde{c}$. Applied to this, the Karush-Kuhn-Tucker optimality criterion proves $\tilde{x}$ to be an optimal solution to $\mathcal{Q}(U, V)$.

(iii) We show that under the given assumptions, the program $\mathcal{Q}(U, V)$ is bounded; convexity then implies that an optimal solution $\tilde{x}$ exists [5]. It suffices to show that $\sum_{p \in U} x_p \, (\rho_p^2 - p^T p)$ is bounded from below. As (3.20), (3.22), and (3.23) hold for real numbers μ_p, $p \in U$, and a real vector $\tilde{c}$, we have $\|\tilde{c} - p\|^2 \leq \tilde{\rho}^2 + \rho_p^2$ for all $p \in U$, with equality for $p \in V$. It is easily verified that the objective function value $f(x)$ does not change for a feasible solution x if we replace 'p' by '$p - \tilde{c}$', so we may assume w.l.o.g. $\tilde{c} = \mathbf{0}$. Then the above equations simplify to $p^T p \leq \tilde{\rho}^2 + \rho_p^2$, $p \in U$, again with equality for $p \in V$. It follows for any feasible solution x of $\mathcal{Q}(U, V)$ that

$$\begin{aligned} \sum_{p \in U} x_p \, (\rho_p^2 - p^T p) &= \sum_{p \in V} x_p \, (\rho_p^2 - p^T p) + \sum_{p \in U \setminus V} x_p \, (\rho_p^2 - p^T p) \\ &\geq -\sum_{p \in U} x_p \tilde{\rho}^2 = -\tilde{\rho}^2 \end{aligned}$$

So $f(x) \geq x^T C^T C x - \tilde{\rho}^2 \geq -\tilde{\rho}^2$ for all feasible solutions x. □

In particular, the lemma shows that an optimal solution to $\mathcal{Q}(U, V)$ 'encodes' the ball $\text{MB}(U, V)$, and that if $\text{MB}(U, V)$ exists, $\mathcal{Q}(U, V)$ has an optimal solution:

Corollary 3.16. *Let* $V \subseteq U \subset \mathbb{R}^d$ *be two finite pointsets.*

(i) If $\tilde{x}$ *is an optimal solution to* $\mathcal{Q}(U, V)$ *with objective value* $-\tilde{\rho}^2$ *then* $B(C\tilde{x}, \tilde{\rho}) = \mathrm{MB}(U, V)$.

(ii) If the ball $\mathrm{MB}(U, V)$ *exists then* $\mathcal{Q}(U, V)$ *has an optimal solution (which encodes* $\mathrm{MB}(U, V)$ *by (i)).*

Proof. (i) follows from Lemma 3.15(i) by setting $\rho_p := 0$, $p \in U$: the Eqs. (3.20), (3.22), and (3.23) show that the ball $D := B(\tilde{c}, \tilde{\rho})$ encloses U and goes through V, and by the second part of (i), there does not exist any ball $B(\tilde{c}', \tilde{\rho}') \in \mathrm{B}(U, V)$ with a smaller radius $\tilde{\rho}' < \tilde{\rho}$, so $D = \mathrm{MB}(U, V)$.

(ii) If $D = \mathrm{MB}(U, V)$, the Eqs. (3.20), (3.22)–(3.23) hold with $\tilde{\rho} := \rho_D$, $\tilde{c} := c_D$, $\rho_p := 0$, $p \in U$, and appropriate μ_p, $p \in U$. Lemma 3.15(iii) then guarantees that $\mathcal{Q}(U, V)$ has an optimal solution. □

More generally, we can use program $\mathcal{Q}(U, V)$ to derive optimality conditions for a ball $D \in \mathrm{B}(U, V)$ to coincide with the ball $\mathrm{MB}(U, V)$.

Lemma 3.17. *Let* $V \subseteq U$ *be two pointsets in* $\mathbb{R}^d$ *and let* $D \in \mathrm{B}(U, V)$. *Then* $D = \mathrm{MB}(U, V)$ *iff there exist real coefficients* λ_p, $p \in U$, *such that*

$$c_D = \sum_{p \in U} \lambda_p p, \qquad \sum_{p \in U} \lambda_p = 1 \tag{3.28}$$

holds and for all $p \in U \setminus V$ *either* $\lambda_p = 0$, *or* $\lambda_p > 0$ *and* p *is tangent to* D.

In other words, the lemma's condition on $p \in U \setminus V$—which we call a *complementarity condition*—requires $\lambda_p \geq 0$ and that λ_p cannot be strictly positive when p is actually contained in the *interior* of D.

Proof. (⇐) If $D \in \mathrm{B}(U, V)$ comes with coefficients λ_p, $p \in U$, that satisfy (3.28) and the complementarity conditions in the lemma then the Eqs. (3.20)–(3.23) hold with $\rho_p := 0$, $p \in U$, $\tilde{\rho} := \rho_D$, $\tilde{x}_p := \lambda_p$, $\tilde{c} = C\tilde{x}$, and with appropriate numbers $\mu_p \geq 0$, $p \in U$. Applied to this, Lemma 3.15(ii) shows that $\tilde{x}$ is an optimal solution to $\mathcal{Q}(U, V)$. This in turn implies via the above corollary that $D = \mathrm{MB}(U, V)$.

(⇒) If $D = \mathrm{MB}(U, V)$ then the above corollary shows that program $\mathcal{Q}(U, V)$ has an optimal solution, $\tilde{x}$, say, with $D = B(C\tilde{x}, \tilde{\rho})$ where $-\tilde{\rho}^2$ is the solution's objective value. In particular, the center c_D of D can be written in the form (3.28), and (3.21) provides the complementarity conditions. □

In particular, the lemma shows that the circumball $D = \textsc{cb}(V)$ of an affinely independent pointset V coincides with $\textsc{mb}(V, V)$ (for we have $c_D \in \mathrm{aff}(V)$, which provides the numbers λ_p required by the lemma).

The balls case. Let us go a step further and consider $\textsc{mb}(U, V)$ for sets $V \subseteq U$ of *balls*. What is then the counterpart to the above Lemma 3.17 for points? A first observation is that if we can write the center of $\textsc{mb}(U, V)$ as both an affine combination of the centers of U and an affine combination of the support points $\mathrm{supp}_{\textsc{mb}(U,V)}(U)$ then the respective coefficients are closely related.

Lemma 3.18. *Let V be a set of balls in $\mathbb{R}^d$, centers affinely independent, that are internally tangent to some larger ball $D = B(c, \rho)$, and denote by s_B, $B \in V$, the support point of B w.r.t. D. If*

$$c = \sum_{B \in V} \mu_B s_B \quad \textit{with} \quad \sum_{B \in V} \mu_B = 1 \tag{3.29}$$

then the center $c = c_D$ can be written in the form

$$c = \sum_{B \in V} \lambda_B c_B \quad \textit{with} \quad \sum_{B \in V} \lambda_B = 1, \tag{3.30}$$

for unique real coefficients λ_B. If moreover the s_B, $B \in V$, are affinely independent then $\mathrm{sgn}(\lambda_B) = \mathrm{sgn}(\gamma)\,\mathrm{sgn}(\mu_B)$ for all $B \in V$, where

$$\gamma := \rho \sum_{B \in V} \mu_B/(\rho - \rho_B) = 1 + \sum_{B \in V} \mu_B \rho_B/(\rho - \rho_B).$$

Moreover, $\gamma\delta = 1$ for $\delta := 1 - \sum_{B \in V} \lambda_B \rho_B/\rho$.

Observe that if the centers of V are not affinely independent then the center c of $\textsc{mb}(V, V)$ need not lie in the affine hull of the centers of V (see Fig. 3.4(a) for example). Also, the last equality in the equation defining γ follows from the identity $x/(x - y) = 1 + y/(x - y)$.

Proof. Using (3.18), Eq. (3.29) yields

$$\mathbf{0} = \sum_{B \in V} \mu_B \frac{\rho}{\rho - \rho_B}\,(c_B - c) =: \sum_{B \in V} \mu'_B\,(c_B - c), \tag{3.31}$$

with the μ'_B summing up to γ. The number γ cannot be zero because then $\mathbf{0} = \gamma c = \sum \mu'_B c_B$ with $\sum \mu'_B = 0$, which contradicts the affine independence of the centers c_B. Dividing (3.31) by γ yields $c = \sum \mu'_B/\gamma\, c_B$ and hence

$$\mu'_B/\gamma = \frac{\mu_B}{\gamma}\frac{\rho}{\rho - \rho_B} = \lambda_B, \qquad B \in V, \tag{3.32}$$

by uniqueness of the numbers λ_B. This shows the first part of the lemma.

Let us argue the other way around. By equation (3.30) we have

$$\mathbf{0} = \sum_{B \in V} \lambda_B\,(c_B - c) = \sum_{B \in V} \lambda'_B \frac{\rho}{\rho - \rho_B}\,(c_B - c),$$

for coefficients $\lambda'_B := \lambda_B\,(\rho - \rho_B)/\rho$ summing up to $1 - \sum \lambda_B \rho_B/\rho = \delta$. Adding δc to both sides yields

$$\begin{aligned} \delta c &= \sum_{B \in V} \lambda'_B \frac{\rho}{\rho - \rho_B}\,(c_B - c) + \delta c \\ &= \sum_{B \in V} \lambda'_B \left(\frac{\rho}{\rho - \rho_B}\,(c_B - c) + c\right) = \sum_{B \in V} \lambda'_B s_B \end{aligned} \tag{3.33}$$

By the affine independence of the s_B, the number δ is nonzero. Dividing by δ and comparing with the unique representation (3.29) we deduce $\delta\mu_B = \lambda'_B$ for all $B \in V$. And since there is at least one strictly positive λ_B, $B \in V$, we get

$$\delta\lambda_B = \frac{\delta\mu'_B}{\gamma} = \frac{\rho}{\rho - \rho_B}\frac{\delta\mu_B}{\gamma} = \frac{\rho}{\rho - \rho_B}\frac{\lambda'_B}{\gamma} = \frac{\rho}{\rho - \rho_B}\frac{\rho - \rho_B}{\rho}\frac{\lambda_B}{\gamma}.$$

From this, we derive $\gamma\delta\lambda_B = \lambda_B > 0$ which in turn shows $\gamma\delta = 1$. □

We point out that both cases, $\gamma > 0$ and $\gamma < 0$, may occur. When the radii of the input balls are zero, the representations (3.30) and (3.29) coincide and thus $\gamma = 1$. Figure 3.7 on the other hand shows a configuration where the signs are swapped: the points c and c_{B_1} lie on different sides of the halfspace through c_{B_2} and c_{B_3} (which shows $\lambda_{B_1} < 0$) while the points c and s_{B_1} lie on the same side of the line through s_{B_2} and s_{B_3} (showing $\mu_{B_1} > 0$); thus, $\operatorname{sgn}(\lambda_B) = -\operatorname{sgn}(\mu_B)$ for $B = B_1$ and via a similar geometrical argument, you can verify this also for $B \in \{B_2, B_3\}$.

We can state optimality conditions for $\text{MB}(U, V)$ in the ball case. For this, we define $\text{tang}_D(T)$ for a set T of balls to be the set of those balls in T that are internally tangent to a given ball D.

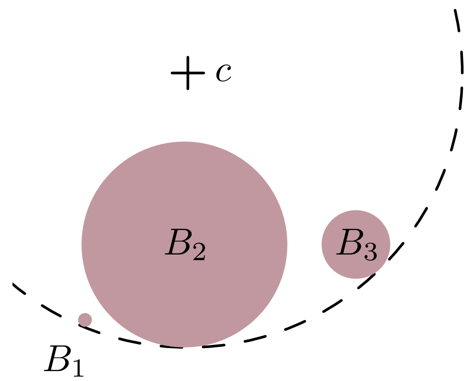


Figure 3.7. *A configuration where the signs of the coefficients λ_B and μ_B from Lemma 3.18 are swapped.*

Lemma 3.19. *Let $V \subseteq U$ be two sets of ball in $\mathbb{R}^d$, let $D \in \mathrm{B}(U,V)$, and suppose $\mathrm{supp}_D(U)$ and $\{c_B \mid B \in \mathrm{tang}_D(U)\}$ are affinely independent.*

(i) If there exist real coefficients λ_B, $B \in U$, such that

$$c_D = \sum_{B \in U} \lambda_B c_B, \qquad \sum_{B \in U} \lambda_B = 1 \tag{3.34}$$

and for all $B \in U \setminus V$ either $\lambda_B = 0$, or $\mathrm{sgn}(\delta)\lambda_B > 0$ and B is tangent to D then $D \in \mathrm{MB}(U,V)$. Here, $\delta = 1 - \sum_{B \in U} \lambda_B \rho_B / \rho_D$ is the (nonzero) number from the previous lemma.

(ii) Conversely, if $D \in \mathrm{MB}(U,V)$ then there exist real λ_B, $B \in U$, that fulfill the conditions in (i).

The condition on $B \in U \setminus V$ requires $\mathrm{sgn}(\delta)\lambda_B \geq 0$ for every $B \in U \setminus V$ and that λ_B cannot be nonzero when B is actually contained in the *interior* of D.

Proof. Consider the following mathematical program in the $d+1$ free variables $x \in \mathbb{R}^d$ and $\rho \in \mathbb{R}$.

$$\begin{array}{lll} \mathcal{Q}(U,V) & \text{minimize} & \rho \\ & \text{subject to} & \|x - c_B\| - (\rho - \rho_B) \leq 0, \qquad B \in U \setminus V, \\ & & \|x - c_B\| - (\rho - \rho_B) = 0, \qquad B \in V. \end{array}$$

Clearly, an optimal solution $(\tilde{x}, \tilde{\rho})$ to $\mathcal{Q}(U,V)$ (if such a solution exists) represents the center and radius of a ball in $\mathrm{MB}(U,V)$. Denote by $f(x,\rho) = \rho$ the program's objective function and write $g_B(x,\rho) = \|x - c_B\| - (\rho - \rho_B)$ for $B \in U$. As f and all the g_B are convex functions, $\mathcal{Q}(U,V)$ is a convex program, and we can therefore apply the *Karush-Kuhn-Tucker Theorem for Convex Programming*, which we invoke in the version of Theorem 4.3.8 and 5.3.1 from the book by Bazaraa, Sherali & Shetty [5]. According to this, a feasible solution $(\tilde{x}, \tilde{\rho})$ is optimal to $\mathcal{Q}(U,V)$ if and only if there exist real numbers τ_B, $B \in U$, such that $\nabla f + \sum_{B \in U} \tau_B \nabla g_B = \mathbf{0}$ holds at the point $(\tilde{x}, \tilde{\rho})$ and such that the latter fulfills the conditions

$$
\begin{aligned}
\tau_B &\geq 0, \qquad B \in U \setminus V, \qquad (3.35)\\
\tau_B \left(\|x - c_B\| - (\rho - \rho_B) \right) &= 0, \qquad B \in U \setminus V. \qquad (3.36)
\end{aligned}
$$

However, the direction ($\Rightarrow$) of this statement only holds if a so-called *constraint qualification* applies. We choose the *Linear Independence Constraint Qualification* (also described in the above book) which requires the following for the subset $I \subseteq U$ of balls that are internally tangent to D: the functions g_B, $B \in I$, must be continuously differentiable at $(\tilde{x}, \tilde{\rho})$ and their gradients $\nabla g_B(\tilde{x}, \tilde{\rho})$, $B \in I$, need to be linearly independent. Using this, we can proof the lemma as follows.

Firstly, we can assume in both statements (i) and (ii) that the center c_D of the ball D does not coincide with the center of a ball in V: if it did, tangency would imply that D actually coincides with one of the balls in I, in which case (i) and (ii) are trivial. Therefore, the gradients

$$
\nabla g_B(x,\rho) = \begin{pmatrix} \frac{x - c_B}{\|x - c_B\|} \\ -1 \end{pmatrix}, \qquad B \in I, \qquad (3.37)
$$

are continuous at $(x,\rho) = (c_D, \rho_D)$.

Under the assumptions of (i), Eq. (3.34) implies

$$
\mathbf{0} = \sum_{B \in U} \lambda_B \, (c_D - c_B) = \sum_{B \in U} \|c_D - c_B\| \lambda_B \, \frac{c_D - c_B}{\|c_D - c_B\|}, \qquad (3.38)
$$

where the coefficients $\lambda'_B := \|c_D - c_B\| \lambda_B$, $B \in U$, have the same signs as the original numbers λ_B. As they sum up to

$$
\alpha := \sum_{B \in U} \lambda'_B = \sum_{B \in U} (\rho_D - \rho_B) \lambda_B = \rho_D - \sum_{B \in U} \rho_B \lambda_B = \rho_D \delta
$$

and as $\rho_D > 0$, the number α is nonzero and has the same sign as δ. Dividing (3.38) by δ we see that $\nabla f + \sum_{B\in U} \tau_B \nabla g_B = \mathbf{0}$ holds at the point (c_D, ρ_D) for the coefficients $\tau_B := \lambda'_B/\alpha$. Thus, the above optimality conditions prove (c_D, ρ_D) to be optimal to $\mathcal{Q}(U,V)$, which shows $D \in \mathrm{MB}(U,V)$.

(ii) Suppose $D \in \mathrm{MB}(U,V)$ and denote by $I \subseteq U$ the balls that are internally tangent to D (equivalently, the constraints that are fulfilled with equality). Clearly, $\mathrm{supp}_D(U) = \mathrm{supp}_D(I)$. Now suppose that the vectors $\nabla g_B(c_D, \rho_D)$, $B \in I$, are linearly dependent. This implies $\sum_{B\in I} \tau_B = 0$ and

$$\mathbf{0} = \sum_{B\in I} \tau_B \frac{c_D - c_B}{\|c_D - c_B\|}$$

for real coefficients τ_B, $B \in I$. Using $\|c_D - c_B\| = \rho_D - \rho_B$, we get

$$\mathbf{0} = \sum_{B\in I} \tau_B \rho_D \frac{c_D - c_B}{\rho_D - \rho_B} = \sum_{B\in I} \tau_B \left(c_D + \frac{\rho_D}{\rho_D - \rho_B}(c_D - c_B)\right).$$

Hence $\sum_{B\in I} \tau_B s_D(B) = \mathbf{0}$, meaning that the support points $\mathrm{supp}_D(I)$ are affinely dependent, a case we excluded. Consequently, the above Karush-Kuhn-Tucker optimality conditions apply, yielding coefficients τ_B, $B \in U$, that satisfy (3.35), (3.36),

$$\mathbf{0} = \sum_{B\in U} \frac{\tau_B}{\|c_D - c_B\|}(c_D - c_B) \tag{3.39}$$

and $\sum_{B\in U} \tau_B = 1$. The coefficients $\tau'_B := \tau_B/\|c_D - c_B\|$ add up to a nonzero number β, and we cannot have $\beta = 0$ because (3.39) would show the c_B to be affinely dependent. Dividing (3.39) by β, Eq. (3.34) holds for coefficients $\lambda_B := \tau'_B/\beta$, and since $\rho_D > 0$ in

$$\beta\delta\rho_D = \beta\,(\rho - \sum_{B\in U} \lambda_B \rho_B) = \sum_{B\in U} \tau'_B\,(\rho - \rho_B) = \sum_{B\in U} \tau_B = 1,$$

the numbers β and δ have the same sign. Thus $\mathrm{sgn}(\lambda_B) = \mathrm{sgn}(\delta)\,\mathrm{sgn}(\tau_B)$, and from this, the claim follows. □

3.4 LP-type formulations

Miniball of points. It is well-known that SEBP is LP-type of combinatorial dimension at most $d+1$ (and we prove this below for the more general problem SEBB). Using the optimality criterion for $\text{MB}(U, V)$ developed in the previous section we can moreover formulate problem SEBP as a reducible weak LP-type problem (see Sec. 2.5). For this, recall from Chap. 2 that Ω_{MB} is the set of all d-dimensional balls, including the empty ball $\emptyset$ of radius $-\infty$ and the infeasible ball $\bowtie$ of radius ∞, and that $\leq$ is the quasiorder on Ω_{MB} that orders the balls according to their radii.

Lemma 3.20. *Let $T \subset \mathbb{R}^d$ be a finite pointset. Then $(T, \leq, \Omega_{\text{MB}}, \text{MB})$ is a reducible primal weak LP-type problem.*

Proof. Monotonicity of MB is obvious, dual nondegeneracy follows from the uniqueness of $\text{MB}(U, V)$ (Lemma 3.9), and Lemma 3.11 establishes reducibility. It remains to show primal optimality.

So assume that J is an inclusion-minimal strong basis of $[U, V \cup \{x\}]$ for $[U, V] \subseteq 2^T$ with $w(U, V) < \bowtie$ and $x \in U \setminus V$. Given that reducibility holds, it suffices to show that

$$\text{MB}(U, V \cup \{x\}) = \text{MB}(U, V) > \text{MB}(U \setminus \{x\}, V) \qquad (3.40)$$

implies $\text{MB}(J, J) = \text{MB}(J, J \setminus \{x\})$. Let the λ_p, $p \in J$, be a set of coefficients as asserted by Lemma 3.17 for the ball $D := \text{MB}(J, J) < \bowtie$. It suffices to show $\lambda_x \geq 0$, which via the lemma shows $D = \text{MB}(J, J \setminus \{x\})$.

We first show that $\lambda_p > 0$ for all $p \in J \setminus V'$, with $V' = V \cup \{x\}$. To see this, we consider the coefficients λ'_p, $p \in J$, obtained from Lemma 3.17 for the ball $D' := \text{MB}(J, V \cup \{x\})$; we have $\lambda'_p \geq 0$ for $x \in J \setminus V'$. From $D = D'$ (recall for this that J is a strong basis of $[U, V \cup \{x\}]$) it follows that $c_D = (1 - \tau)c_{D'} + \tau c_D$ for any real number τ, that is,

$$c_D = \sum_{p \in J} ((1 - \tau)\lambda'_p + \tau \lambda_p)p =: \sum_{p \in J} \mu'_p(\tau)p.$$

Set $N := \{p \in J \setminus V' \mid \lambda_p < 0\}$ and suppose that this set is nonempty. We increase τ from 0 and stop as soon as $\mu'_q(\tau) = 0$ for some $q \in N$; such a τ exists because $\lambda_p \leq 0$ and $\lambda'_p \geq 0$ for all $p \in N$. At this moment, all points $p \in J \setminus V'$ still have $\mu'_p(\tau) \geq 0$, and $c_D = \sum_{p \in J} \mu'_p(\tau)$ holds with the coefficients $\mu'_p(\tau)$ summing up to 1. Thus, Lemma 3.17 yields $D = \text{MB}(J \setminus \{q\}, J \setminus \{q\})$ and $D = \text{MB}(J \setminus \{q\}, V')$. Moreover, as $\text{MB}(J \setminus \{q\}, J \setminus \{q\})$ equals $D = \text{MB}(U, V')$, we have $D = \text{MB}(U, J \setminus \{q\})$ by dual nondegeneracy, and therefore $J \setminus \{q\}$ is a strong basis of $[U, V']$, a contradiction to the inclusion-minimality of J.

Next, we consider the coefficients λ''_p, $p \in U$, that the Lemma 3.17 guarantees for $D = \text{MB}(U, V)$. These numbers satisfy $\lambda''_p \geq 0$, for $p \in U \setminus V$, in particular $\lambda''_x \geq 0$. We claim that also $\lambda_x > 0$, which one can see as follows. We rewrite c_D as

$$c_D = \sum_{p \in J} ((1-\tau)\lambda''_p + \tau \lambda_p) p =: \sum_{p \in J} \mu''_p(\tau) p,$$

for which we introduce $\lambda_p := 0$, $p \in U \setminus J$. With this, the equation holds for any real τ. If $\lambda_x \leq 0$ then there exists a value $\tau^* \in (0, 1]$ such that $\mu''_x(\tau^*) = 0$. As $\lambda''_p \geq 0$ and $\lambda_p \geq 0$ for all $p \in U \setminus V'$, we have $\mu''_p(\tau^*) \geq 0$ for $p \in U \setminus V$. Plugging this together with $\mu''_x(\tau^*) = 0$ into Lemma 3.17, we obtain $D = \text{MB}(U \setminus \{x\}, V)$, a contradiction to (3.40). Thus, $\lambda_x \geq 0$, which proves the claim. □

In particular, this proves that the call welzl$(T, \emptyset)$ to Welzl's algorithm returns a pointset V that satisfies (not only $\text{MB}(T, \emptyset) = \text{MB}(V, V)$ but also) $\text{MB}(V, V) = \text{MB}(V, \emptyset)$, a fact that (nobody doubted but that) was not settled so far.

Finally, we show that SEBP over affinely independent pointsets induces unique sink orientations, as was advertised in the introduction and the preceding chapter. This is a consequence of the following lemma.

Lemma 3.21. *Let $T \subset \mathbb{R}^d$ be an affinely independent pointset. Then the tuple $(T, \leq, \Omega_{\text{MB}}, \text{MB})$ is a reducible strong LP-type problem.*

Proof. We first show that the balls $\text{MB}(U, V)$, $V \subseteq U \subseteq T$, exist (i.e., not equal to the infeasible ball): on the one hand, $\text{MB}(F, F)$ exists for every $F \subseteq T$ because affine independence and Lemma 3.12 ensure the existence of a ball $D \in \text{B}(F, F)$ with center in $\text{aff}(F)$, and Lemma 3.17 proves D to coincide with $\text{MB}(F, F)$. On the other hand, reducibility of MB (which was proved in Lemma 3.20) implies that every ball $\text{MB}(U, V)$ equals $\text{MB}(F, F)$ for some $F \in [U, V]$.

Given this, we can establish primal nondegeneracy, which together with Lemma 3.20 proves that (T, MB) is a reducible strong problem. So suppose $\text{MB}(U', V') = \text{MB}(U, V) =: D$ for sets $V' \subseteq U'$ and $V \subseteq U$. Clearly, $D \in \text{B}(U' \cap U, V' \cap V)$, and it remains to show that D is the smallest ball in the latter set. Consider the coefficients λ_p, $p \in U$, that Lemma 3.17 produces for the ball $\text{MB}(U, V)$ and the coefficients λ'_p, $p \in U'$, that it yields for $\text{MB}(U', V')$. (Notice that these coefficients exists because the two balls are not the infeasible ball by the above discussion.) These numbers satisfy $\lambda_p \geq 0$, $p \in U \setminus V$, and $\lambda'_p \geq 0$, $p \in U' \setminus V'$. As $\text{MB}(U,V)$ and $\text{MB}(U',V')$ have the same center c_D, we have

$$c_D = \sum_{p \in U} \lambda_p p = \sum_{p \in U'} \lambda'_p p. \tag{3.41}$$

By setting $\lambda_p := 0$, $p \in T \setminus U$ (and $\lambda'_p := 0$, $p \in T \setminus U'$), we can extend the coefficients λ_p of U (and likewise the coefficients λ'_p of U') to coefficients on T, and this does not change (3.41). Affine independence of the involved points then yields $\lambda_p = \lambda'_p$, $p \in U' \cup U$, in particular, $\lambda_p = \lambda'_p = 0$ for $p \in U' \cup U \setminus (U' \cap U)$. It follows $c_D = \sum_{p \in U' \cap U} \lambda_p p$ with $\lambda_p \geq 0$ for all $p \in U' \cap U \setminus (V' \cap V)$, which in turn implies $D = \text{MB}(U' \cap U, V' \cap V)$ via Lemma 3.17 again. □

Miniball of balls. With the material we have developed so far, it is now a simple matter to show that SEBB is an LP-type problem.

Lemma 3.22. *Let T be a finite set of balls in $\mathbb{R}^d$. Then $(T, \leq, \Omega_{\text{MB}}, \text{MB})$ is an LP-type problem of combinatorial dimension at most $d+1$.*

Proof. Monotonicity and locality are proved along the same lines as for problem SEBP in Chap. 2. Lemma 3.7 provides the bound on the problem's combinatorial dimension. □

As we will see in Chap. 5, the function $\text{MB}(\cdot,\cdot)$ is not reducible,[2] so Welzl's algorithm welzl need not (and, as an example will show, does not) solve SEBB. However, if the centers of the balls are affinely independent, the situation changes, and we will show in Theorem 5.21 that a variant of SEBB admits a formulation as a reducible strong (and hence also reducible weak) problem.

3.5 Smallest superorthogonal ball

We conclude this chapter with an excursion on a variation of problem SEBB. Instead of searching for a ball *enclosing* a given ball set U, this section's focus lies on a ball that either covers the input ball $B \in U$ or

[2] Actually, $\text{MB}(U,V)$ is a set; we mean here the function that assigns (U,V) to some ball in $\text{MB}(U,V)$. Even if this ball is unique, the latter function need not be reducible.

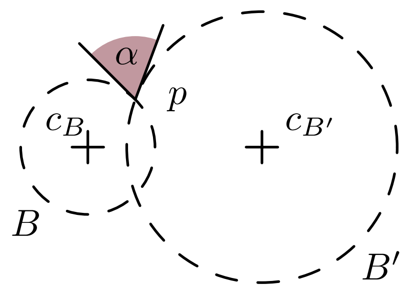


Figure 3.8. *The angle $\angle(B, B') = \pi - \angle(c_B, p, c_{B'})$ between two intersecting spheres B, B'; the spheres are superorthogonal if $\angle(B, B') \geq \pi/2$.*

intersects it in such a way that the tangent planes at every boundary intersection point span an outer angle of at least 90 degrees, see Fig. 3.8.

We begin by reviewing the notion of *orthogonality* [13] and 'superorthogonality' between balls and subsequently derive the aforementioned geometric interpretation in terms of the dihedral angle. We will then introduce the ball '$\textsc{mob}(U)$,' which is the smallest ball that is superorthogonal to all balls in the set U, and present a quadratic program that computes it.

Let $B \subset \mathbb{R}^d$ be a ball and $x \in \mathbb{R}^d$. We call the number $\mathrm{pw}_B(x) = \|c_B - x\|^2 - \rho_B^2$ the *power of x w.r.t. the ball B.*

Definition 3.23. *Two balls $B, B' \subset \mathbb{R}^d$ are* superorthogonal *to each other if $\mathrm{pw}_B(c_{B'}) \leq \rho_{B'}^2$ (equivalently, $\mathrm{pw}_{B'}(c_B) \leq \rho_B^2$). If equality holds, the balls are said to be* orthogonal.

We can rewrite $\mathrm{pw}_B(c_{B'}) \leq \rho_{B'}^2$ as

$$0 \leq 2c_B^T c_{B'} - (c_B^T c_B + c_{B'}^T c_{B'} - \rho_B^2 - \rho_{B'}^2). \tag{3.42}$$

Superorthogonality and orthogonality have the following geometric interpretation in terms of the *angle between spheres* which we define as follows. Given two balls B, B' and an intersection point $p \in \partial B \cap \partial B'$, we define $\angle(B, B') := \pi - \angle(c_B, p, c_{B'})$ where $\angle(c_B, p, c_{B'})$ is the angle between the line segments $s := \mathrm{conv}(\{c_B, p\})$ and $s' := \mathrm{conv}(\{p, c_{B'}\})$, see Fig. 3.8. By congruence of the triangles $\mathrm{conv}(\{c_B, p, c_{B'}\})$, $p \in \partial B \cap \partial B'$, the angle $\angle(B, B')$ is independent of the choice of the point $p \in \partial B \cap \partial B'$. In case any of the line segments s, s' is a point (equivalently, one of the balls B, B' is a point), we define $\angle(B, B') := \pi/2$. Also, if the balls' boundaries do not intersect and one ball is contained in the other we set $\angle(B, B') := \infty$, and $\angle(B, B') := -\infty$ if the balls are completely disjoint.

Lemma 3.24. *Let $B, B' \subset \mathbb{R}^d$ be two balls. Then B and B' are superorthogonal iff B and B' intersect and $\angle(B, B') \geq \pi/2$.*

Observe here that the intersection $\partial B \cap \partial B'$ might be empty, in which case the lemma states that B and B' are superorthogonal if and only if one ball contains the other.

Proof. ($\Rightarrow$) The balls B and B' intersect because

$$\|c_B - c_{B'}\|^2 = \mathrm{pw}_B(c_{B'}) + \rho_B^2 \leq \rho_B^2 + \rho_{B'}^2 \leq \rho_B^2 + 2\rho_B\rho_{B'} + \rho_{B'}^2,$$

and hence $\|c_B - c_{B'}\| \leq \rho_B + \rho_{B'}$.

In case the balls' boundaries do not intersect and also in case one of the balls is a point, $\angle(B, B') \geq \pi/2$ by definition; otherwise, fix any $p \in \partial B \cap \partial B'$. We can w.l.o.g. assume that $p = \mathbf{0}$ (translate the balls appropriately to achieve this) so that c_B is a normal vector of the tangent plane to ∂B in p, and similarly, $c_{B'}$ is a normal of the tangent plane to $\partial B'$ in p, see Fig. 3.8. Their angle α fulfills

$$\cos\alpha = \frac{c_B^T c_{B'}}{\rho_B \rho_{B'}} \geq \frac{c_B^T c_B + c_{B'}^T c_{B'} - \rho_B^2 - \rho_{B'}^2}{2\rho_B\rho_{B'}} = 0,$$

where we used (3.42) to obtain the inequality. From our choice of normals $c_B, c_{B'}$ it follows $\angle(c_B, p, c_{B'}) \leq \pi/2$, as needed.

($\Leftarrow$) If B, B' intersect, we distinguish two cases. If one ball is contained in the other, i.e., $B \subseteq B'$ w.l.o.g., then $\|c_B - c_{B'}\| \leq \rho_{B'} - \rho_B$ and by squaring this, it follows $\mathrm{pw}_B(c_{B'}) = \|c_B - c_{B'}\| - \rho_B^2 \leq \rho_{B'}^2 - 2\rho_B\rho_{B'} \leq \rho_{B'}^2$, as needed. Otherwise the balls' boundaries intersect and both balls have a strictly positive radius. In this case, fix any $p \in \partial B \cap \partial B'$, and assume w.l.o.g. that p coincides with origin. Then the angle $\beta := \angle(B, B')$ satisfies $c_B^T c_{B'}/(\rho_B\rho_{B'}) = \cos\beta \geq 0$, implying (3.42). □

Let U be a finite set of balls in $\mathbb{R}^d$. *Some* ball superorthogonal to all balls in U exists: take any ball that encloses all balls U; by the above lemma it is superorthogonal to every $B \in U$. Using this, a simple compactness argument establishes that a *smallest* ball superorthogonal to all balls in U exists. In fact, the following lemma establishes as a side result that there is only one such ball, and therefore we already now denote by

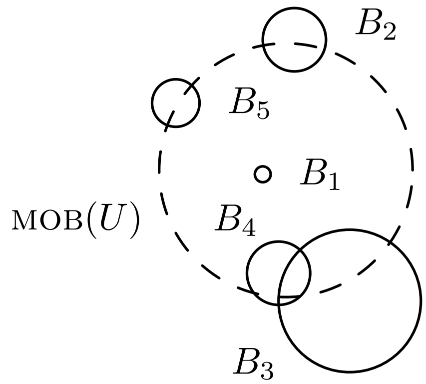


Figure 3.9. *An example of* MOB(U) *for a set of five balls in* $\mathbb{R}^2$.

MOB(U) the unique ball of smallest radius that is superorthogonal to all balls in U, see Fig. 3.9.

Similar to the notions '$\mathrm{B}(U,V)$' and '$\mathrm{MB}(U,V)$,' we define $\mathrm{OB}(U,V)$ to be the set of all balls which are orthogonal to the balls in V and superorthogonal to the balls in U; $\mathrm{MOB}(U,V)$ is then the *smallest* ball in $\mathrm{OB}(U,V)$ (again, the following lemma shows that there is only one smallest ball) and we have $\mathrm{MOB}(U,\emptyset) = \mathrm{MOB}(U)$ by definition.

Lemma 3.25. *Let $V \subseteq U$ be two finite sets of balls, let C be the matrix whose columns are the centers of the balls U and set $\rho_{c_B} := \rho_B$, $B \in U$.*

(i) The set $\mathrm{OB}(U,V)$ *contains a unique smallest ball.*

(ii) If $\tilde{x}$ is an optimal solution to $\mathcal{Q}(U,V)$ with objective value $-\tilde{\rho}^2$ then $B(C\tilde{x},\tilde{\rho}) = \mathrm{MOB}(U,V)$.

(iii) If the ball $\mathrm{MOB}(U,V)$ *exists then* $\mathcal{Q}(U,V)$ *has an optimal solution (which encodes* $\mathrm{MOB}(U,V)$ *by (i)).*

Proof. The proof of (ii) and (iii) is completely analogous to the proof of Corollary 3.16(i)–(ii), which is based on Lemma 3.15. The uniqueness of the smallest ball in $\mathrm{OB}(U,V)$ follows from Lemma 3.15(i). □

Using this and Lemma 3.15 we can also give optimality conditions for a ball $D \in \mathrm{OB}(U,V)$ to be the ball $\mathrm{MOB}(U,V)$.

Lemma 3.26. *Let $V \subseteq U$ be two sets of ball in $\mathbb{R}^d$ and let $D \in \textsc{ob}(U, V)$. Then $D = \textsc{mob}(U, V)$ iff there exist real numbers λ_B, $B \in U$, such that*

$$c_D = \sum_{B \in U} \lambda_B c_B, \qquad \sum_{B \in U} \lambda_B = 1, \tag{3.43}$$

and for all $B \in U \setminus V$ either $\lambda_B = 0$, or $\lambda_B > 0$ and B is tangent to D.

Here, the complementarity conditions on the balls $B \in U \setminus V$ means that for all $B \in U \setminus V$ the number λ_B is nonzero and that it cannot be strictly positive when the ball B is (only superorthogonal but) not orthogonal to D.—Again, the proof is completely analogous to the proof of Lemma 3.17.

We remark that along the lines of Lemmata 3.20 and 3.21 one can show that problem MOB is a reducible primal weak problem and, under affine independence, even a reducible strong problem. We also mention that the problem is related to *power diagrams* [13].

Chapter 4

Smallest enclosing balls of points

In this chapter we present a simple combinatorial algorithm, a joint work with Bernd Gärtner and Martin Kutz [29], for solving the miniball problem in the special case when the input consists of *points* only. The algorithm resembles the *simplex method* for linear programming (LP); it comes with a Bland-type rule to avoid cycling in presence of degeneracies and it typically requires very few iterations.

In contrast to Welzl's algorithm whose applicability for SEBP is limited in practice to dimensions $d \leq 30$, the method from this chapter behaves nicely in (mildly) high dimensions: a floating-point implementation solves instances in dimensions up to 10,000 within hours, and with a suitable stopping-criterion (to compensate for rounding-errors), all degeneracies we have tested so far are handled without problems.

4.1 Sketch of the algorithm

The idea behind the algorithm is simple: start with a balloon strictly containing all the points and then deflate it until it cannot shrink anymore without losing a point. (A variant of this idea was proposed by Hopp & Reeve [46] in 1996, but only as a heuristic for $d = 3$ without proof of correctness and termination.) In this section we sketch the main ingredients necessary for implementing this, postponing the details to Sec. 4.2.

An important notion for our method is the circumball $\text{CB}(T)$ of a nonempty affinely independent set T, which is the unique sphere with center in the affine hull $\text{aff}(T)$ that goes through the points in T (see Lemma 3.12). In the following we call the center of this ball the *circumcenter of* T, denoted by $\text{CC}(T)$. Moreover, a nonempty affinely independent subset T of the set S of given points will be called a *support set*.[1] Also, we introduce the notation $B(c, T)$ for a pointset T and a point $c \in \mathbb{R}^d$, by which we mean the ball $B(c, \max_{p \in T} \|p - c\|)$, i.e., the smallest ball with given center c that encloses the points T.

Our algorithm steps through a sequence of pairs (T, c), maintaining the invariant that T is a support set and c is the center of a ball B containing S and having T on its boundary. Lemma 3.3 tells us that we have found the smallest enclosing ball when $c = \text{CC}(T)$ and $c \in \text{conv}(T)$. Until this criterion is fulfilled, the algorithm performs an iteration (a so-called *pivot step*) consisting of a *walking phase* which is preceded by a *dropping phase* in case $c \in \text{aff}(T)$.

Dropping. If $c \in \text{aff}(T)$, the invariant and Lemma 3.12 guarantee that $c = \text{CC}(T)$. Because $c \notin \text{conv}(T)$, there is at least one point $s \in T$ whose coefficient in the affine combination of T forming c is negative. We drop such an s and enter the walking phase with the pair $(T \setminus \{s\}, c)$, see left of Fig. 4.1.

Walking. If $c \notin \text{aff}(T)$, we move our center on a straight line towards $\text{CC}(T)$. Lemma 4.1 below establishes that the moving center is always the center of a (progressively smaller) ball with T on its boundary. To maintain the algorithm's invariant, we must stop walking as soon as a new point $s' \in S$ hits the boundary of the shrinking ball. In that case we enter the next iteration with the pair $(T \cup \{s'\}, c')$, where c' is the stopped center; see Fig. 4.1. If no point stops the walk, the center reaches $\text{aff}(T)$ and we enter the next iteration with $(T, \text{CC}(T))$.

[1] We note that this definition of a 'support set' differs from the one given in Chap. 3 (which we do not use here).

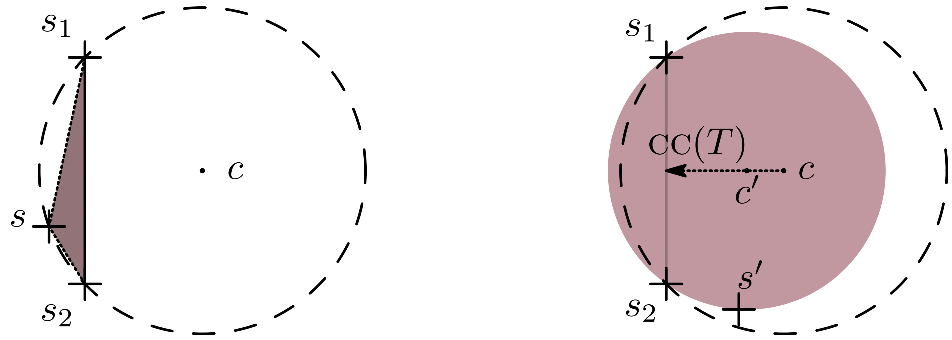


Figure 4.1. *Dropping the point s from $T = \{s, s_1, s_2\}$ (left) and walking towards the center $\mathrm{CC}(T)$ of the circumball of $T = \{s_1, s_2\}$ until s' stops us (right).*

4.2 The algorithm in detail

Let us start with some basic facts about the walking direction from the current center c towards the circumcenter of the current boundary points T.

Lemma 4.1. *Let T be a nonempty affinely independent pointset on the boundary of some ball $B(c, \rho)$, i.e., $T \subseteq \partial B(c, \rho) = \partial B(c, T)$. Then*

(i) the line segment $[c, \mathrm{CC}(T)]$ is orthogonal to $\mathrm{aff}(T)$,

(ii) $T \subseteq \partial B(c', T)$ for each $c' \in [c, \mathrm{CC}(T)]$,

(iii) $\rho_{B(\cdot,T))}$, i.e., the radius of $B(\cdot, T)$, is a strictly monotone decreasing function on $[c, \mathrm{CC}(T)]$, with minimum attained at $\mathrm{CC}(T)$.

Note that part (i) of this lemma implies that the circumcenter of T coincides with the orthogonal projection of c onto $\mathrm{aff}(T)$, a fact that is important for the actual implementation of the method.

When moving the center of our ball along $[c, \mathrm{CC}(T)]$, we have to check for new points to hit the shrinking boundary. The subsequent lemma tells us that all points 'behind' $\mathrm{aff}(T)$ are uncritical in this respect, i.e., they cannot hit the boundary and thus cannot stop the movement of the center. Hence, we may ignore these points during the walking phase. In Fig. 4.1 (right), for instance, $\mathrm{aff}(T)$ is the line though the points $\{s_1, s_2\}$ and the halfspace that is bounded by $\mathrm{aff}(T)$ and does not contain c is the regions of all points that lie 'behind' $\mathrm{aff}(T)$: any point therein cannot stop the movement of the center.

Lemma 4.2. *Let T and c as in Lemma 4.1 above and let $q \in B(c,T)$ lie behind* $\text{aff}(T)$*, precisely,*

$$(q-c)^T(\text{CC}(T)-c) \geq (\text{CC}(T)-c)^T(\text{CC}(T)-c). \tag{4.1}$$

Then q is contained in $B(c',T)$ for any $c' \in [c, \text{CC}(T)]$.

Proof of Lemmata 4.1 and 4.2. Let w be the vector from c to the orthogonal projection of c onto $\text{aff}(T)$. By definition, w satisfies

$$w^T(p-q)=0, \qquad p,q \in \text{aff}(T). \tag{4.2}$$

Consider any point $c_\mu = c + \mu w$ on the real ray $L = \{c + \mu w \mid \mu \geq 0\}$. For any $p \in T$ we have

$$\begin{aligned} \|c_\mu - p\|^2 &= \|c-p\|^2 + \mu^2 w^T w + 2\mu w^T(c-p) & (4.3)\\ &= \|c-p\|^2 + \mu^2 w^T w + 2\mu w^T(c-\text{CC}(T)), & (4.4)\end{aligned}$$

where the second equality follows from $c - p = (c - \text{CC}(T)) - (p - \text{CC}(T))$, with the second vector being orthogonal to w by (4.2). As the numbers $\|c-p\|^2$ are identical for all $p \in T$, the distance from c_μ to p is independent of the chosen point $p \in T$, and hence $T \subseteq \partial B(c_\mu, T)$. In particular, the point $c+w$, i.e., the intersection of L with $\text{aff}(T)$, has identical distance to the points in T and thus coincides with the unique circumcenter $\text{CC}(T)$. We conclude $w = \text{CC}(T) - c$ from which (i) and (ii) follow.

To show (iii), we use $w = \text{CC}(T) - c$ to write the squared radius (4.4) of the ball $B(c_\mu, T)$ as

$$\rho^2_{B(c_\mu,T)} = \|c-t_0\|^2 + \mu\,(\mu-2)\,w^T w. \tag{4.5}$$

This is a strictly convex function in μ, with the minimum attained at $\mu = 1$. Therefore, the radius strictly decreases on the interval $[c_0, c_1] = [c, \text{CC}(T)]$, achieving its minimum at $c_1 = \text{CC}(T)$.

In order to settle Lemma 4.2, we show that q is contained in $B(c_\mu, T)$ for all $\mu \geq 0$. Denoting by p any arbitrary element from T,

$$\begin{aligned} \|c_\mu - q\|^2 &= \|c-q\|^2 + \mu^2 w^T w + 2\mu w^T(c-q)\\ &\leq \|c-p\|^2 + \mu^2 w^T w + 2\mu w^T(c-q)\\ &\leq \|c-p\|^2 + \mu^2 w^T w - 2\mu w^T w = \rho^2_{B(c_\mu,T)},\end{aligned}$$

where we have used $q \in B(c,T)$ for the first inequality, $\mu \geq 0$ and (4.1) for the second one, and (4.5) for the final equality. □

```
procedure bubble(S)
{ Computes mb(S) }
{ Precondition: S ≠ ∅ }
begin
    c := any point of S
    T := {p}, for a point p of S at maximal distance from c
    while c ∉ conv(T) do
        { Invariant: B(c,T) ⊇ S, ∂B(c,T) ⊇ T, T aff. indep. }
        if c ∈ aff(T) then
            drop a point q from T with λ_q < 0 in (4.6)
        { Here, c ∉ aff(T). }
        among the points in S \ T that do not satisfy (4.1)
            find one, p, say, that restricts movement of c
            towards cc(T) most, if one exists
        move c as far as possible towards cc(T)
        if walk has been stopped then
            T := T ∪ {p}
    return B(c,T)
end bubble
```

Figure 4.2. *The algorithm to compute* MB(S).

It remains to identify which point of the boundary set T should be dropped in case that $c \in \operatorname{aff}(T)$ but $c \notin \operatorname{conv}(T)$. Here are the suitable candidates.

Lemma 4.3. *Let T and c be as in Lemma 4.1 above and assume that $c \in \operatorname{aff}(T)$. Let*

$$c = \sum_{p \in T} \lambda_p p, \quad \sum_{p \in T} \lambda_p = 1 \tag{4.6}$$

be the affine representation of c with respect to T. If $c \notin \operatorname{conv}(T)$ then $\lambda_q < 0$ for at least one $q \in T$ and any such q satisfies inequality (4.1) with T replaced by the reduced set $T \setminus \{q\}$ there.

Combining Lemmata 4.2 and 4.3, we see that if we drop a point with negative coefficient in (4.6), this point will not stop us in the subsequent walking step.

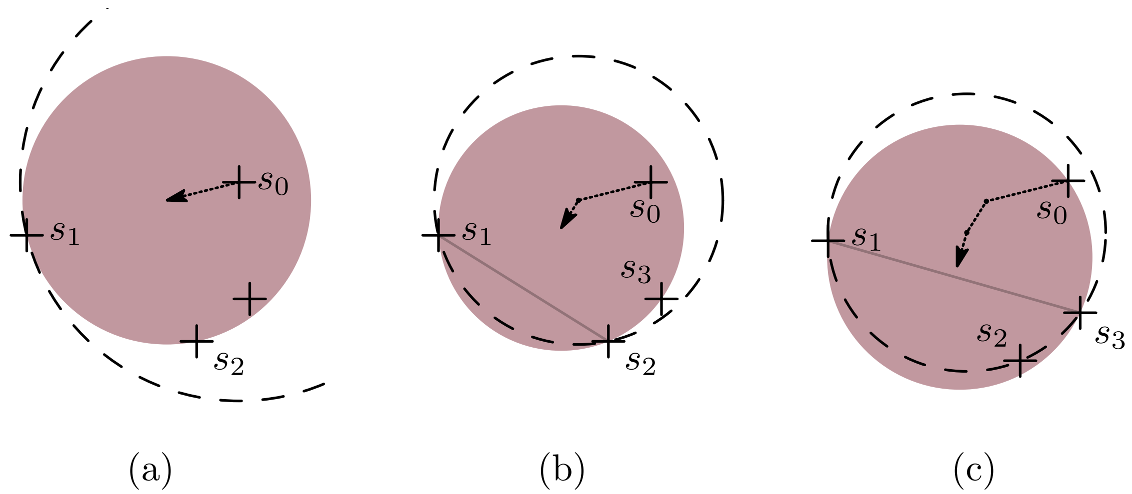


Figure 4.3. *A full run of the algorithm in 2D.*

Proof. We set $T' := T \setminus \{p\}$ and $w = \text{CC}(T') - c$. By Lemma 4.1(i) we then have $w^T(p-q) = 0$ for any two points $p, q \in \text{aff}(T')$. Using this and (4.6), we deduce

$$\begin{aligned} 0 < w^T w &= w^T(\text{CC}(T') - c) \\ &= \sum_{p \in T} \lambda_p w^T(\text{CC}(T') - p) \\ &= \lambda_q w^T(\text{CC}(T') - q). \end{aligned}$$

Consequently, $\lambda_q < 0$ implies $w^T \text{CC}(T') < w^T q$, from which we conclude $w^T(q - c) > w^T(\text{CC}(T') - c) = w^T w$ as needed. □

The algorithm in detail. Fig. 4.2 gives a formal description of our algorithm. The correctness follows easily from the previous considerations and we will address the issue of termination in a minute. Before doing so, let us consider an example in the plane. Figure 4.3, (a)–(c), depicts all three iterations of our algorithm on a four-point set. Each picture shows the current ball $B(c, T)$ just before (dashed) and right after (filled) the walking phase.

After the initialization $c = s_0$, $T = \{s_1\}$, we move towards the singleton T until s_2 hits the boundary (step (a)). The subsequent motion towards the circumcenter of two points is stopped by the point s_3, yielding a 3-element support (step (b)). Before the next walking we drop the point s_2 from T. The last movement (c) is eventually stopped by s_0 and then the center lies in the convex hull of $T = \{s_0, s_1, s_3\}$.

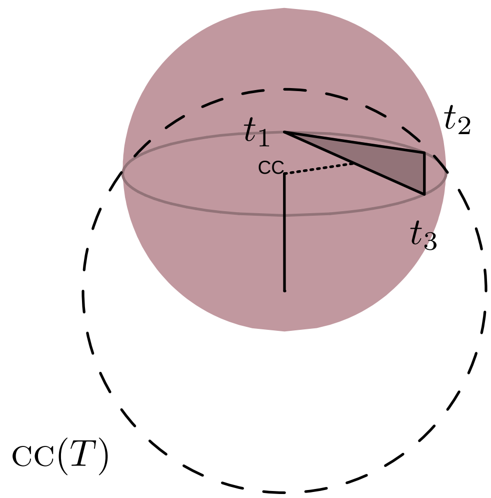


Figure 4.4. *Two consecutive steps of the algorithm in 3D.*

Observe that the 2-dimensional case obscures the fact that in higher dimensions, the target $\text{cc}(T)$ of a walk need not lie in the convex hull of the support set T. In Fig. 4.4, the current center c first moves to $\text{cc}(T) \notin \text{conv}(T)$, where $T = \{t_1, t_2, t_3\}$. Then, t_2 is dropped and the walk continues towards $\text{aff}(T \setminus \{t_2\})$.

Termination. It is not clear whether the algorithm as stated in Fig. 4.2 always terminates. Although the radius of the ball clearly decreases whenever the center moves, it might happen that a stopper already lies on the current ball and thus no real movement is possible. In principle, this might happen repeatedly from some point on, i.e., we might run in an infinite cycle, perpetually collecting and dropping points without ever moving the center at all. However, for points in sufficiently general position such infinite loops cannot occur.

Lemma 4.4. *If for all affinely independent subsets $T \subseteq S$, no point of $S \setminus T$ lies on the circumball of T then algorithm* $\mathsf{bubble}(S)$ *terminates.*

Proof. Right after a dropping phase, the dropped point cannot be reinserted (Lemmata 4.2 and 4.3) and by assumption no other point lies on the current boundary. Thus, the sequence of radii measured right before the dropping steps is strictly decreasing; and since at least one out of d consecutive iterations demands a drop, it would have to take infinitely many values if the algorithm did not terminate. But this is impossible because before a drop, the center c coincides with the circumcenter $\mathrm{CC}(T)$ of one out of finitely many subsets T of S. □

The degenerate case. In order to achieve termination for arbitrary instances, we equip the procedure $\mathsf{bubble}(S)$ with the following simple rule, resembling Bland's pivoting rule for the simplex algorithm [19] (for simplicity, we will actually call it *Bland's rule* in the sequel):

> *Fix an arbitrary order on the set S. When dropping a point with negative coefficient in* (4.6)*, choose the one of smallest rank in the order. Also, pick the smallest-rank point for inclusion in T when the algorithm is simultaneously stopped by more than one point during the walking phase.*

As it turns out, this rule prevents the algorithm from 'cycling', i.e., it guarantees that the center of the current ball cannot stay at its position for an infinite number of iterations.

Theorem 4.5. *Using Bland's rule,* $\mathsf{bubble}(S)$ *terminates.*

Proof. Assume for a contradiction that the algorithm cycles, i.e., there is a sequence of iterations where the first support set equals the last and the center does not move. We assume w.l.o.g. that the center coincides with the origin. Let $C \subseteq S$ denote the set of all points that enter and leave the support during the cycle and let among these be m the one of maximal rank.

The key idea is to consider a slightly modified instance X of the SEBP problem. Choose a support set $D \not\ni m$ right after dropping m and let $X := D \cup \{-m\}$, mirroring the point m at 0. There is a unique affine representation of the center 0 by the points in $D \cup \{m\}$, where by Bland's rule, the coefficients of points in D are all nonnegative while m's is negative. This gives us a *convex* representation of 0 by the points in X and we may write

$$0 = (\sum_{p \in X} \lambda_p p)^T \,\mathrm{CC}(I) = \sum_{p \in D} \lambda_p p^T \,\mathrm{CC}(I) - \lambda_{-m} m^T \,\mathrm{CC}(I). \qquad (4.7)$$

We have introduced the scalar products because of their close relation to criterion (4.1) of the algorithm. We bound these by considering a support set $I \not\ni m$ just before insertion of the point m. We have $m^T \text{cc}(I) < \text{cc}(I)^T \text{cc}(I)$ and by Bland's rule and the maximality of m, there cannot be any other points of C in front of $\text{aff}(I)$; further, all points of D that do not lie in C must, by definition, also lie in I. Hence, we get $p^T \text{cc}(I) \geq \text{cc}(I)^T \text{cc}(I)$ for all $p \in I$. Plugging these inequalities into (4.7) we obtain

$$0 > \Big(\sum_{p \in D} \lambda_p - \lambda_{-m} \Big) \text{cc}(I)^T \text{cc}(I) = (1 - 2\lambda_{-m}) \text{cc}(I)^T \text{cc}(I),$$

which implies $\lambda_{-m} > 1/2$, a contradiction to Corollary 3.4. □

Implementation and results. We have programmed algorithm bubble in C++ using floating point arithmetic. In order to represent intermediate solutions (i.e., the current support set with its circumcenter) we use a QR-factorization technique, allowing fast and robust updates under insertion and deletion of a single point into and from the current support set. Instead of Bland's rule (which is slow in practice and because of rounding errors difficult to implement), we resort to a different heuristic. The resulting code shows very stable behavior even with highly degenerate input instances (points sampled from the surface of a sphere). An a 480Mhz Sun Ultra 4 workstation, pointsets in dimensions up to $d = 2{,}000$ can be handled efficiently; within hours, we were even able to compute the miniball of pointsets of 10,000 points in dimensions up to 10,000. Please refer to [29] for more details on the implementation and test results.

4.3 Remarks

To the best of our knowledge, algorithm bubble is the first combinatorial algorithm (i.e., an *exact* method in the RAM model) that is efficient in practice in (mildly) high dimensions. Although the quadratic programming (QP) approach of Gärtner and Schönherr [38] is in practice polynomial in d, it critically requires arbitrary-precision linear algebra to avoid robustness issues, limiting the tractable dimensions to $d \leq 300$, see [38]. Also, codes based on Welzl's algorithm from Chap. 2 cannot reasonably handle pointsets beyond dimension $d = 30$ [33].

The resulting code is in most cases faster (sometimes significantly) than recent dedicated methods that only deliver approximate results, and it beats off-the-shelf solutions, based e.g. on quadratic programming solvers. For the details, please refer to [29].

The code can efficiently handle point sets in dimensions up to 2,000, and it solves instances of dimension 10,000 within hours. In low dimensions, the algorithm can keep up with the fastest computational geometry codes that are available.

Chapter 5

Smallest enclosing balls of balls

In this chapter we investigate the problem of computing the smallest enclosing ball of a set of balls. We start off with an example showing that Welzl's algorithm sebb from Chap. 2 does *not* generalize from points to balls. Given this, we turn to Matoušek, Sharir & Welzl's algorithm msw and describe how one can implement the primitives needed for this. The resulting algorithm and heuristical variant of it which works well in practice have been implemented in CGAL 3.0, the *computational geometry algorithms library* [16].

As far as small dimensions (up to 10, say) are concerned, codes based on algorithm msw are already the best we can offer from a practical point of view. In higher dimensions however, these methods become inefficient, and therefore we focus on this setting in the second part of this chapter. As a first step, we generalize problem SEBB to *signed* balls, allowing negative radii. This will reveal the fact that the combinatorial structure of an SEBB instance only depends on the ball centers and the pairwise *differences* in radii. An important consequence is that we may assume one of the input balls to be a point—even that this point is the origin, and that it lies on the boundary of the miniball, if the SEBB instance arises during the basis computation of algorithm msw.

Building on these insights, Sec. 5.4 linearizes the problem, using the geometric *inversion* transform. Under inversion, balls through the origin map to halfspaces, so that we get the equivalent problem of finding a halfspace that is optimal under suitable criteria. This halfspace turns out to be the solution to an 'almost linear' mathematical program. As a byproduct, the formulation provides us with a method for computing the distance of a point to the convex hull of a union of balls.

Section 5.5 further investigates the mathematical programming approach in case the input ball centers are affinely independent. We establish a program that is well-behaved in the sense that it has a unique solution, characterized by *Karush-Kuhn-Tucker* optimality conditions. This also holds if some of the input balls are required to be tangent to the miniball, entailing the possible nonexistence of a true miniball. In the latter case, the solution to the program has an interpretation in terms of a 'generalized' miniball.

This generalization lets us fit the problem into the framework of *reducible strong LP-type problems* (as introduced in Chap. 2) once we assume the centers of the input balls to be affinely independent (which an embedding in sufficiently high-dimensional space and a subsequent perturbation always achieves). As a concrete consequence of this, Welzl's algorithm *does* work, affine independence assumed. Also, we can use the material from Chap. 2 to reduce SEBB to the problem of finding the sink in a unique sink orientation. With this, we can improve the trivial bound of $\Omega(2^d)$ on the (expected) combinatorial complexity of solving small instances of the SEBB problem; through the general LP-type techniques (Lemma 2.11) from Chap. 2, this will also provide improved bounds for large instances. On the practical side, the unique sink approach allows for algorithms (like RandomEdge or Murty's rule) that might not be worst-case efficient but have the potential to perform very well in practice.

5.1 Welzl's algorithm

In Sec. 3.4 we have already seen that SEBP can be formulated as a reducible primal weak problem. Consequently, Welzl's algorithm sebb from Fig. 2.7 solves SEBP and the question remains whether SEBB, too, can be solved with it.

The answer to this is 'no.' In general, Welzl's algorithm (which Fig. 5.1 shows again in its specialization to SEBB) does not work anymore

```
procedure sebb(U, V)
{ Intended to compute MB(U, V) but does not work }
{ Precondition: U ⊇ V, |MB(U, V)| = 1 }
begin
    if U = V then
        return any ball from the set MB(V, V)
    else
        choose B ∈ U\V uniformly at random
        D:= sebb(U \ {B}, V)
        if B ⊈ D then
            return sebb(U, V ∪ {B})
        else
            return D
end sebb
```

Figure 5.1. *Algorithm* welzl *from Fig. 2.7 specialized to problem* SEBB.

when balls are input. The reason for this is that problem SEBB is not a reducible problem. (If you have read Welzl's original paper this means that Welzl's Lemma [86], underlying the algorithm's correctness proof in the point case, fails for balls.) Reducibility would read as follows in the context of SEBB.

Dilemma 5.1. *Let $U \supseteq V$ be sets of balls such that $\mathrm{MB}(U, V)$ and $\mathrm{MB}(U \setminus \{B\}, V)$ contain unique balls each. If*

$$B \not\subseteq \mathrm{MB}(U \setminus \{B\}, V)$$

for some $B \in U \setminus V$ then B is tangent to $\mathrm{MB}(U, V)$, so $\mathrm{MB}(U, V) = \mathrm{MB}(U, V \cup \{B\})$.

A counterexample to this is depicted in Fig. 5.2: the point B_5 is not contained in the ball $D = \mathrm{MB}(\{B_1, B_3, B_4\}, \{B_1, B_3, B_4\})$, but B_5 is not tangent to

$$D' = \mathrm{MB}(\{B_1, B_3, B_4, B_5\}, \{B_1, B_3, B_4\}).$$

As a matter of fact, feeding the procedure sebb with the five balls from Fig. 5.2 produces incorrect results from time to time, depending on the

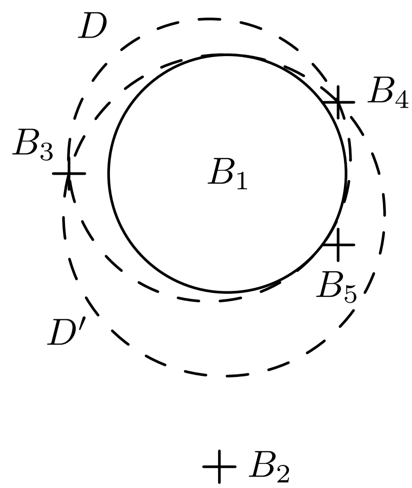


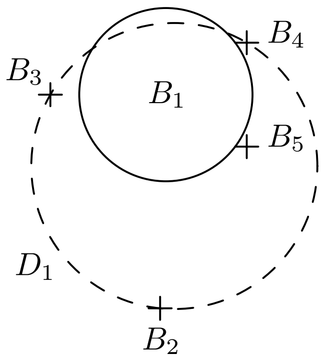
Figure 5.2. *Five circles $\{B_1, \ldots, B_5\}$ for which procedure* sebb *may fail.*

outcomes of the internal random choices in the algorithm.[1] If in each call, B is chosen to be the ball of lowest index in $U \setminus V$, the algorithm eventually gets stuck when it tries to find the ball $\text{MB}(\{B_1, B_3, B_4, B_5\}, \{B_1, B_3, B_4, B_5\})$, which does not exist.

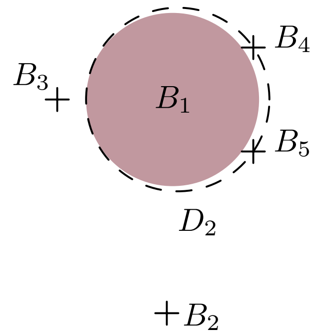
This is detailed in Fig. 5.3 which depicts a possible sequence of recursive calls (in order of their execution) to procedure sebb, triggered by the 'master' call sebb$(\{B_1, \ldots, B_5\}, \emptyset)$. Each of the six subfigures concentrates on the point in time where a recursive call of type 'sebb$(U \setminus \{B\}, V)$' has just delivered a ball D failing to contain the ball B (upper line of subfigure caption), so that another recursive call to sebb$(U, V \cup \{B\})$ has to be launched (lower line of subfigure caption). The latter call in turn triggers the first call of the next subfigure, after descending a suitable number of recursive levels. Observe that this counterexample is free of degeneracies, and that no set $\text{MB}(U, V)$ contains more than one ball.

We remark here that as we will see in Sec. 5.5, Welzl's algorithm *does* work if the centers of the input balls are affinely independent. (In this case, the computation of the ball $\text{MB}(V, V)$ in the base case of procedure sebb from Fig. 5.1 can be done using Lemma 5.2 from the next section.)

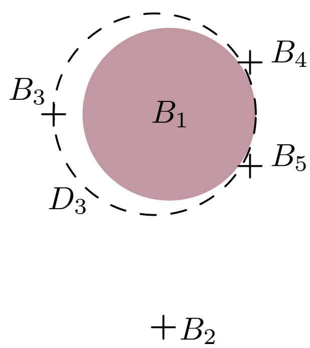
[1]The balls in Fig. 3.5 already constitute a counterexample to Dilemma 5.1 but cannot be used to fool Welzl's algorithm, as the complete enumeration of all possible runs (each being the result of different random choices '$B \in U \setminus V$' within the algorithm) shows.



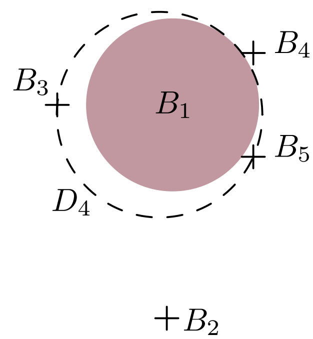
(i) $B_1 \not\subseteq D_1 = \mathsf{sebb}(\{B_2,\ldots,B_5\},\emptyset)$
$\rightsquigarrow \mathsf{sebb}(\{B_1,\ldots,B_5\},\{B_1\})$



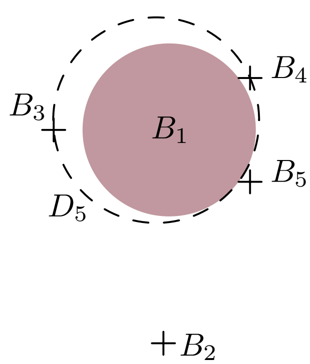
(ii) $B_3 \not\subseteq D_2 = \mathsf{sebb}(\{B_1,B_4,B_5\},\{B_1\})$
$\rightsquigarrow \mathsf{sebb}(\{B_1,B_3,B_4,B_5\},\{B_1,B_3\})$



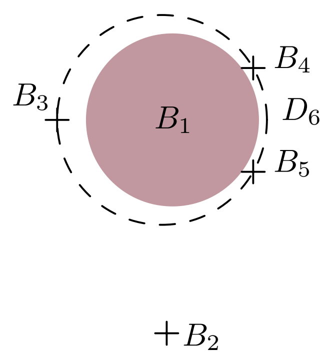
(iii) $B_5 \not\subseteq D_3 = \mathsf{sebb}(\{B_1,B_3\},\{B_1,B_3\})$
$\rightsquigarrow \mathsf{sebb}(\{B_1,B_3,B_5\},\{B_1,B_3,B_5\})$



(iv) $B_4 \not\subseteq D_4 = \mathsf{sebb}(\{B_1,B_3,B_5\},\{B_1,B_3\})$
$\rightsquigarrow \mathsf{sebb}(V \cup \{B_5\},V)$, for $V = \{B_1,B_3,B_4\}$



(v) $B_5 \not\subseteq D_5 = \mathsf{sebb}(V,V)$
$\rightsquigarrow \mathsf{sebb}(W,W)$, for $W = \{B_1,B_3,B_4,B_5\}$



(vi) $\textsc{mb}(W,W)=\emptyset$, as B_1 is not tangent to
$\textsc{b}(\{B_3,B_4,B_5\},\{B_3,B_4,B_5\}) = \{D_6\}$.

Figure 5.3. *A failing run* $\mathsf{sebb}(U,\emptyset)$ *on the circles* U *from Fig. 5.2.*

5.2 Algorithm msw

Having seen that Welzl's algorithm does not work for SEBB, we turn to algorithm msw from Chap. 2, which clearly solves the problem as it is LP-type (Lemma 3.22). In order to realize the two primitives violates and basis needed by procedure msw, we use the following lemma which allows for the calculation of the 'base case' $\mathrm{MB}(V,V)$.

Lemma 5.2. *Let V be a basis of U. Then $\mathrm{MB}(V,V)=\mathrm{MB}(U)$, and this ball can be computed in time $\mathcal{O}(d^3)$.*

Proof. For $V=\emptyset$, the claim is trivial, so assume $V\neq\emptyset$. As a basis of U, V satisfies $\mathrm{MB}(V)=\mathrm{MB}(U)$. Since the balls in V must be tangent to $\mathrm{MB}(U)$ (Lemma 3.6), we have $\mathrm{MB}(V)\in\mathrm{MB}(V,V)$. But then *any* ball in $\mathrm{MB}(V,V)$ is a smallest enclosing ball of V, so Lemma 3.1 guarantees that $\mathrm{MB}(V,V)$ is a singleton.

Let $V=\{B_1,\dots,B_m\}$, $m\leq d{+}1$, and observe that $B(c,\rho)\in\mathrm{B}(V,V)$ if and only if $\rho\geq\rho_{B_i}$ and $\|c-c_{B_i}\|^2=(\rho-\rho_{B_i})^2$ for all i. Defining $z_{B_i}=c_{B_i}-c_{B_1}$ for $1<i\leq m$ and $z=c-c_{B_1}$, these conditions are equivalent to $\rho\geq\max_i\rho_{B_i}$ and

$$\begin{aligned} z^Tz &= (\rho-\rho_{B_1})^2, \\ (z_{B_i}-z)^T(z_{B_i}-z) &= (\rho-\rho_{B_i})^2, \qquad 1<i\leq m. \end{aligned} \tag{5.1}$$

Subtracting the latter from the former yields the $m-1$ linear equations

$$2z_{B_i}^Tz-z_{B_i}^Tz_{B_i}=2\rho\,(\rho_{B_i}-\rho_{B_1})+\rho_{B_1}^2-\rho_{B_i}^2, \qquad 1<i\leq m.$$

If $B(c,\rho)=\mathrm{MB}(V,V)$ then $c\in\mathrm{conv}(\{c_{B_1},\dots,c_{B_m}\})$ by Lemma 3.3. Thus we get $c=\sum_{i=1}^m\lambda_ic_{B_i}$ with the λ_i summing up to 1. Then, $z=\sum_{i=2}^m\lambda_i(c_{B_i}-c_{B_1})=Q\lambda$, where $Q=(z_{B_2},\dots,z_{B_m})$ and $\lambda=(\lambda_2,\dots,\lambda_m)^T$. Substituting this into our linear equations results in

$$2z_{B_i}^TQ\lambda=z_{B_i}^Tz_{B_i}+\rho_{B_1}^2-\rho_{B_i}^2+2\rho\,(\rho_{B_i}-\rho_{B_1}), \qquad 1<i\leq m. \tag{5.2}$$

This is a linear system of the form $A\lambda=e+f\rho$, with $A=2Q^TQ$. So $B(c,\rho)=\mathrm{MB}(V,V)$ satisfies $c-c_{B_1}=z=Q\lambda$ with (λ,ρ) being a solution of (5.1), (5.2) and $\rho\geq\max_i\rho_{B_i}$. Moreover, the columns of Q are linearly independent as a consequence of Lemma 3.8, which implies that A is in fact regular (see the discussion after Lemma 3.12).

Hence we can in time $\mathcal{O}(d^3)$ find the solution space of the linear system (which is one-dimensional, parameterized by ρ) and substitute this into the quadratic equation (5.1). From the possible solutions (λ, ρ) we select one such that $\rho \geq \max_i \rho_{B_i}$, $\lambda \geq 0$ and $\lambda_1 = 1 - \sum_{i=2}^{m} \lambda_i \geq 0$; by $\text{MB}(V) = \text{MB}(V, V)$ and Lemma 3.3 such a pair (λ, ρ) exists, and in fact, there is only one such pair because the ball determined by any (λ, ρ) with the above properties is tangent, enclosing and by Lemma 3.3 equal to $\text{MB}(V)$. □

We note that the existing, robust formulas for computing $\text{MB}(V, V)$ in the point case [33] can be generalized to balls (and are employed in our code); please refer to the implementation documentation [28].

The primitives. The violation test violates(B, V) needs to check whether $B \not\subseteq \text{MB}(V)$; as V is a basis, Lemma 5.2 can be used to compute $\text{MB}(V) = \text{MB}(V, V)$ and therefore the test is easy. In the basis computation we are given a basis V and a violating ball B (i.e., $B \not\subseteq \text{MB}(V)$), and we are to produce a basis of $V \cup \{B\}$. By Lemma 3.6, the ball B is internally tangent to $\text{MB}(V \cup \{B\})$. A basis of $V \cup \{B\}$ can then be computed in a brute-force manner[2] by using Lemma 5.2 as follows.

We generate all subsets V', $B \in V' \subseteq V \cup \{B\}$, in increasing order of size. For each V' we test whether it is a support set of $V \cup \{B\}$. From our enumeration order it follows that the first set V' which passes this test constitutes a basis of $V \cup \{B\}$.

We claim that V' is a support set of $V \cup \{B\}$ if and only if the computations from Lemma 5.2 go through and produce a ball that in addition encloses the balls in $V \cup \{B\}$: if V' is a support set of $V \cup \{B\}$ then it is, by our enumeration order, a basis and hence the lemma applies. Conversely, a successful computation yields a ball $D \in \text{B}(V', V')$ (enclosing $V \cup \{B\}$) whose center is a convex combination of the centers of V'; by Lemma 3.3, $D = \text{MB}(V') = \text{MB}(V \cup \{B\})$.

Plugging these primitives into algorithm msw yields an expected $\mathcal{O}(d^3 2^{2d} n)$-algorithm for computing the miniball $\text{MB}(U)$ of any set of n balls in d-space (Lemma 2.7). Moreover, it is possible to do all computations in rational arithmetic (provided the input balls have rational coordinates and radii): although the center and the radius of the miniball

[2] We will improve on this in Sec. 5.5. Also, Welzl's algorithm could be used here, by lifting and subsequently perturbing the centers, but this will not be better than the brute-force approach, in the worst case.

may have irrational coordinates, the proof of Lemma 5.2 show that they actually are of the form $\alpha_i + \beta_i\sqrt{\gamma}$, where $\alpha_i, \beta_i, \gamma \in \mathbb{Q}$ and where $\gamma \geq 0$ is the discriminant of the quadratic equation (5.1). Therefore, we can represent the coordinates and the radius by pairs $(\alpha_i, \beta_i) \in \mathbb{Q}^2$, together with the number γ. Since the only required predicate is the containment test, which boils down to determining the sign of an algebraic number of degree 2, all computations can be done in $\mathbb{Q}$.

We have implemented the algorithm in C++, and the resulting package has been released with CGAL 3.0. The code follows the generic programming paradigm. In particular, it is parameterized with a type F which specifies the number type to be used in the computation: choosing F to be a type realizing rational numbers of arbitrary precision, no roundoff errors occur and the computed ball is the exact smallest enclosing ball of the input balls. Efficient implementations of such types are available (see for instance the CORE [51], LEDA [64], and GNU MP [43] libraries); some of them even use *filtering techniques* which take advantage of the floating-point hardware and resort to expensive multiple-precision arithmetic only if needed in order to guarantee exact results.

Under a naive floating-point implementation, numerical problems may arise when balls are 'almost' tangent to the current miniball. In order to overcome these issues, we also provide a (deterministic) variant of algorithm msw. In this heuristic—it comes without any theoretical guarantee on the running time—we maintain a basis V (initially consisting of a single input ball) and repeatedly add to it, by an invocation of the basis computation, a ball farthest away from the basis, that is, a ball B' satisfying

$$\|c - c_{B'}\| + \rho_{B'} = \max_{B \in U} (\|c - c_B\| + \rho_B) =: \chi_V,$$

with c being the center of $\text{MB}(V)$. The algorithm stops as soon as χ_V is smaller or equal to the radius of $\text{MB}(V)$, i.e., when all balls are contained in $\text{MB}(V)$. This method, together with a suitable adaptation [28] of efficient and robust methods for the point case [33], handles degeneracies in a satisfactory manner: numerical problems tend to occur only towards the very end of the computation, when the ball $\text{MB}(V)$ is already near-optimal; a suitable numerical stopping criterion avoids cycling in such situations and ensures that we actually output a correct basis in almost all cases. An extensive testsuite containing various degenerate configurations of balls is passed without problems.

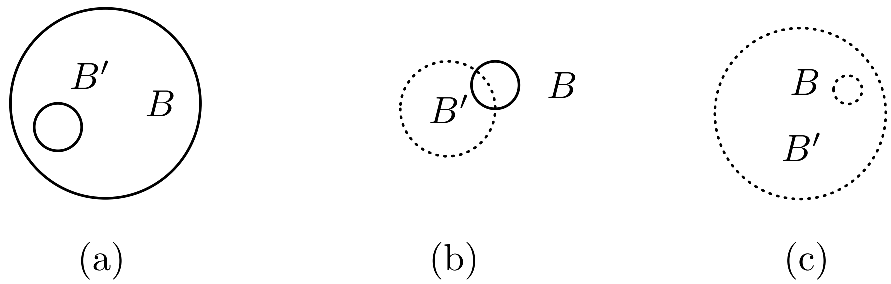


Figure 5.4. *B dominates B' (a) if both balls are positive and $B \supseteq B'$, (b) if B is positive and B' negative and the two intersect, or (c) if both are negative and $B \subseteq B'$. (Negative balls are drawn dotted.)*

5.3 Signed balls and shrinking

In this section we show that under a suitable generalization of SEBB, one of the input balls can be assumed to be a point, and that SEBB can be reduced to the problem of finding the miniball with some point *fixed on the boundary.* With this, we prepare the ground for the more sophisticated material of Secs. 5.4 and 5.5.

Recall that a ball $B = B(c, \rho)$ encloses a ball $B' = B(c', \rho')$ if and only if relation (3.1) holds. Now we are going to use this relation for *signed* balls. A signed ball is of the form $B(c, \rho)$, where—unlike before—ρ can be *any* real number, possibly negative. $B(c, \rho)$ and $B(c, -\rho)$ represent the same ball $\{x \in \mathbb{R}^d \mid \|x-c\|^2 \leq \rho^2\}$, meaning that a signed ball can be interpreted as a regular ball with a sign attached to it; we simply encode the sign into the radius. If $\rho \geq 0$, we call the ball *positive*, otherwise *negative.*

Definition 5.3. *Let $B = B(c, \rho)$ and $B' = B(c', \rho')$ be signed balls. B* dominates *B' if and only if*

$$\|c - c'\| \leq \rho - \rho'. \tag{5.3}$$

B marginally dominates *B' if and only if (5.3) holds with equality.*

Figure 5.4 depicts three examples of the dominance relation. Furthermore, marginal dominance has the following geometric interpretation: if both B, B' are positive, B' is internally tangent to B; if B is positive and B' is negative then B and B' are *externally tangent* to each other, and finally, if both B, B' are negative then B is internally tangent to B'.

We generalize SEBB to the problem of finding the ball of smallest signed radius that dominates a given set of signed balls. For two sets $U \supseteq V$ of signed balls, we denote by $\text{B}(U, V)$ the set of signed balls that dominate the balls in U and that marginally dominate the balls in V. We call a signed ball B *smaller* than another signed ball B' if $\rho_B < \rho_{B'}$. Then, $\text{MB}(U, V)$ is the set of smallest signed balls in $\text{B}(U, V)$. Again, we set $\text{B}(\emptyset, \emptyset) = \{\emptyset\}$ and $\text{MB}(\emptyset, \emptyset) = \{\emptyset\}$, and abuse notation in writing $\text{MB}(U, V)$ for the ball D in case $\text{MB}(U, V)$ is a singleton $\{D\}$.

Figure 5.5 depicts some examples of $\text{MB}(U) := \text{MB}(U, \emptyset)$. In particular, Fig. 5.5(c) illustrates that this generalization of SEBB covers the problem of computing a ball of largest volume (equivalently, smallest negative radius) contained in the intersection $I = \bigcap_{B \in U} B$ of a set U of balls: for this, simply encode the members of U as negative balls.

At this stage, it is not yet clear that $\text{MB}(U)$ is always nonempty and contains a unique ball. With the following argument, we can easily show this. Fix any ball O and define $s_O : B \mapsto B(c_B, \rho_B - \rho_O)$ to be the map which 'shrinks' a ball's radius by ρ_O while keeping its center unchanged. (Actually, s_O only depends on one real number, but in our application this number will always be the radius of an input ball.) We set $s_O(\emptyset) := \emptyset$ and extend s_O to sets T of signed balls by means of $s_O(T) = \{s_O(B) \mid B \in T\}$. From Eq. (5.3) it follows that dominance and marginal dominance are invariant under shrinking and we get the following

Lemma 5.4. *Let $U \supseteq V$ be two sets of signed balls, O any signed ball. Then $B \in \text{B}(U, V)$ iff $s_O(B) \in \text{B}(s_O(U), s_O(V))$, for any ball B.*

Obviously, the 'smaller' relation between signed balls is invariant under shrinking, from which we obtain

Corollary 5.5. $\text{MB}(s_O(U), s_O(V)) = s_O(\text{MB}(U, V))$ *for any two sets $U \supseteq V$ of signed balls, O any signed ball.*

This leads to the important consequence that an instance of SEBB defined by a set of signed balls U has the same combinatorial structure as the instance defined by the balls $s_O(U)$: Most obviously, Corollary 5.5 shows that both instances have the same number of miniballs, the ones in $\text{MB}(s_O(U))$ being shrunken copies of the ones in $\text{MB}(U)$. In fact, replacing

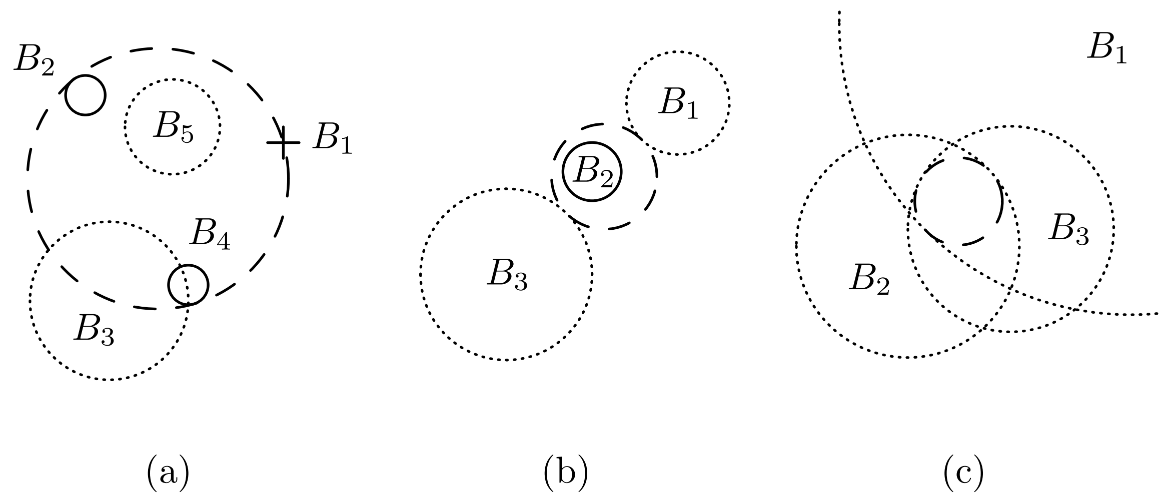


Figure 5.5. *The ball* MB(U) *(dashed) for three sets* U *of signed balls: (a)* MB(U) *is determined by three positive balls, (b)* MB(U) *is determined by two negative balls, (c) the miniball of intersecting, negative balls is the ball of largest volume contained in* $\bigcap_{B \in U} B$*; its radius is negative.*

the 'positive' concepts of containment and internal tangency with the 'signed' concepts of dominance and marginal dominance in Chap. 3, we can define support sets and bases for sets of signed balls. It then holds (Corollary 5.5) that U and $s_O(U)$ have the same support sets and bases, i.e., the combinatorial structure only depends on parameters which are invariant under shrinking: the ball centers and the differences in radii.

In particular, if the ball O in the corollary is a smallest ball in U then $s_O(U)$ is a set of positive balls, and the material we have developed for this special case in Chap. 3 carries over to the general case (most prominently, this shows that MB(U) for signed balls U consists of a single ball, and that SEBB over signed balls is of LP-type and thus solvable by algorithm msw). In this sense, any instance of SEBB over signed balls is combinatorially equivalent to an instance over positive balls, and from now on, we refer to SEBB as the problem of finding MB(U) for signed balls U.

Reconsidering the situation, it becomes clear that this extension to signed balls is not a real generalization; instead it shows that any instance comes with a 'slider' to simultaneously change all radii.

One very useful slider placement is obtained by shrinking w.r.t. some ball $O \in U$. In this case, we obtain a set $s_O(U)$ of balls where at least one ball is a *point*. Consequently, when we solve problem SEBB using algorithm msw of Fig. 2.3, we can also assume that the *violating* (which now means non-dominated) ball B entering the basis computation is actually a point. Moreover, using Lemma 3.6 with the obvious generalization to signed balls, we see that B is in fact marginally dominated by the ball $\mathrm{MB}(W \cup \{D\})$. We can therefore focus on the problem SEBB_p of finding the smallest ball that dominates a set U of signed balls, with an additional point p marginally dominated. More precisely, for given $p \in \mathbb{R}^d$ we define

$$\mathrm{B}_p(U, V) := \mathrm{B}(U \cup \{p\}, V \cup \{p\})$$

and denote the smallest balls in this set by $\mathrm{MB}_p(U, V)$. Then SEBB_p is the problem of finding $\mathrm{MB}_p(U) := \mathrm{MB}_p(U, \emptyset)$ for a given set U of signed balls and a point $p \in \mathbb{R}^d$. We note that all balls in $\mathrm{B}_p(U, V)$ are positive (they dominate the positive ball $\{p\}$) and that we can always reduce problem SEBB_p to $\mathrm{SEBB}_{\mathbf{0}}$ via a suitable translation.

In this way, we generalize the notion $\mathrm{MB}_p(U)$ of Eq. (3.8) from only positive balls to signed balls. In contrast to the case of positive balls (Lemma 3.9), the set $\mathrm{MB}_p(U)$ may contain more than one ball when U is a set of signed balls (to see this, shrink Fig. 3.4 (left) w.r.t. B_2).

Our main application of these findings is the solution of problem SEBB using algorithm msw from Chap. 2 (Lemma 2.11). As discussed above, the basis computation in this case amounts to solving an instance of $\mathrm{SEBB}_{\mathbf{0}}$ involving at most $d + 1$ balls, from which we obtain

Theorem 5.6. *Problem* SEBB *over a set of n signed balls can be reduced to problem* $\mathrm{SEBB}_{\mathbf{0}}$ *over a set of at most $d + 1$ signed balls: given an algorithm for the latter problem of (expected) runtime $f(d)$, we get an algorithm with expected runtime*

$$\mathcal{O}(d^2 n) + e^{\mathcal{O}(\sqrt{d \log d})} f(d)$$

for the former problem.

We note that all the sets $\mathrm{MB}_p(T)$ occurring in this reduction contain *exactly one ball* (and not more than one, as is possible in general) because we always have $\mathrm{MB}_p(T) \ni \mathrm{MB}(T \cup \{p\})$, where the latter balls is unique.

In the sequel (Secs. 5.4 and 5.5), we concentrate on methods for solving problem $\mathrm{SEBB}_{\mathbf{0}}$ with the goal of improving over the complete enumeration approach which has $f(d) = \Omega(2^d)$. From now on, all balls are assumed to be a signed balls, unless stated otherwise.

5.4 Inversion

In this section we present a 'dual' formulation of the $\textsc{sebb}_{\mathbf{0}}$ problem for (signed) balls. We derive this by employing the *inversion* transform to obtain a program that describes $\textsc{mb}_{\mathbf{0}}(U,V)$. This program is 'almost' linear (in contrast to the convex but far-from-linear programs obtained by Megiddo [63] and Dyer [22]) and will serve as the basis of our approach to small cases of problem $\textsc{sebb}_{\mathbf{0}}$ (Sec. 5.5).

As a by-product, this section links $\textsc{sebb}_{\mathbf{0}}$ to the problem of finding the distance from a point to the convex hull of a union of balls.

5.4.1 A dual formulation for $\textsc{sebb}_{\mathbf{0}}$

We use the *inversion transform* $x^* := x/\|x\|^2$, $x \neq \mathbf{0}$, to map a ball $B \in \textsc{b}_{\mathbf{0}}(U,V)$ to some linear object. To this end, we exploit the fact that under inversion, balls through the origin map to halfspaces while balls not containing the origin simply translate to balls again.

We start by briefly reviewing how balls and halfspaces transform under inversion. For this, we extend the inversion map to nonempty point sets via $P^* := \mathrm{cl}(\{p^* \mid p \in P \setminus \{\mathbf{0}\}\})$, where $\mathrm{cl}(Q)$ denotes the closure of set Q, and to sets S of balls or halfspaces by means of $S^* := \{P^* \mid P \in S\}$. (The use of the closure operator guarantees that if P is a ball or halfspace containing the origin, its image P^* is well-defined and has no 'holes;' we also set $\emptyset^* := \{\mathbf{0}\}$ to have $(P^*)^* = P$.)

Consider a halfspace $H \subset \mathbb{R}^d$; H can always be written in the form

$$H = \left\{x \mid v^T x + \alpha \geq 0\right\}, \qquad v^T v = 1. \tag{5.4}$$

In this case, the number $|\alpha|$ is the distance of the halfspace H to the origin. If H does not contain the origin (i.e., $\alpha < 0$) then H maps to the positive ball

$$H^* = B(-v/(2\alpha), -1/(2\alpha)). \tag{5.5}$$

Since $(P^*)^* = P$, if P is a ball or halfspace, the converse holds, too: a proper ball with the origin on its boundary transforms to a halfspace not containing the origin. On the other hand, a ball $B = B(c,\rho)$ not containing the origin maps to a ball again, namely to $B^* = B(d,\sigma)$ where

$$d = \frac{c}{c^T c - \rho^2} \quad \text{and} \quad \sigma = \frac{\rho}{c^T c - \rho^2}. \tag{5.6}$$

B^* again does not contain the origin, and B^* is positive if and only if B is positive. All these facts are easily verified [8].

The following lemma shows how the dominance relation in the 'primal' domain translates under inversion. For this, we say that a halfspace H of the form (5.4) *dominates* a ball $B = B(d, \sigma)$ if and only if

$$v^T d + \alpha \geq \sigma, \tag{5.7}$$

and we speak of *marginal dominance* in case of equality in (5.7).

As in the primal domain, the dominance relation has an interpretation in terms of containment and intersection: H dominates a positive ball B if and only if H contains B—we also say in this case that B *is internally tangent to* H—and H dominates a negative ball B if and only if H intersects B. In both cases, marginal dominance corresponds to B being tangent to the hyperplane underlying H, in addition.

Lemma 5.7. *Let D be a positive ball through $\mathbf{0}$ and B a signed ball not containing $\mathbf{0}$. Then D dominates B if and only if the halfspace D^* dominates the ball B^*.*

Proof. We first show that D dominates B if and only if

$$\|c_D - c_B\|^2 \leq (\rho_D - \rho_B)^2. \tag{5.8}$$

The direction ($\Rightarrow$) is clear from the definition of dominance, and so is ($\Leftarrow$) under the assumption that $\rho_D - \rho_B \geq 0$. So suppose (5.8) holds with $\rho_D - \rho_B < 0$. Then $0 \leq \|c_D - c_B\| \leq \rho_B - \rho_D$, from which we conclude that B is positive and dominates D. Thus, $\mathbf{0} \in D \subseteq B$, a contradiction to B not containing the origin.

It remains to show that Eq. (5.8) holds if and only if the halfspace D^* dominates the ball B^*. As $c_D^T c_D = \rho_D^2$, the former inequality is equivalent to

$$c_B^T c_B - \rho_B^2 \leq 2\,(c_D^T c_B - \rho_D \rho_B), \tag{5.9}$$

where the left hand side $\mu := c_B^T c_B - \rho_B^2$ is a strictly positive number, by the assumption on B. Write the halfspace D^* in the form (5.4) with $\alpha < 0$, and assume $B^* = B(d, \sigma)$. From (5.5) and (5.6) it follows that

$$c_D = -v/(2\alpha), \quad \rho_D = -1/(2\alpha), \quad d = c_B/\mu, \quad \sigma = \rho_B/\mu.$$

Using this, we obtain the equivalence of Eqs. (5.7) and (5.9) by multiplying (5.9) with the number $\alpha/\mu < 0$. □

For $U \supseteq V$ two sets of balls, we define $\mathrm{H}(U, V)$ to be the set of halfspaces not containing the origin that dominate the balls in U and marginally dominate the balls in V. The following is an immediate consequence of Lemma 5.7. Observe that any ball D satisfying $D \in \mathrm{B}_{\mathbf{0}}(U, V)$ or $D^* \in \mathrm{H}(U^*, V^*)$ is positive by definition.

Lemma 5.8. *Let $U \supseteq V$, $U \neq \emptyset$, be two sets of balls, no ball in U containing the origin. Then D is a ball in $\mathrm{B}_{\mathbf{0}}(U, V)$ if and only if D^* is a halfspace in $\mathrm{H}(U^*, V^*)$.*

We are interested in *smallest* balls in $\mathrm{B}_{\mathbf{0}}(U, V)$. In order to obtain an interpretation for these in the dual, we use the fact that under inversion, the radius of a ball $D \in \mathrm{B}_{\mathbf{0}}(U, V)$ is inversely proportional to the distance of the halfspace D^* to the origin, see (5.5). It follows that D is a smallest ball in $\mathrm{B}_{\mathbf{0}}(U, V)$, i.e., $D \in \mathrm{MB}_{\mathbf{0}}(U, V)$, if and only if the halfspace D^* has largest distance to the origin among all halfspaces in $\mathrm{H}(U^*, V^*)$. We call such a halfspace D^* a *farthest* halfspace in $\mathrm{H}(U^*, V^*)$.

Corollary 5.9. *Let $U \supseteq V$, $U \neq \emptyset$, be two sets of balls, no ball in U containing the origin. Then $D \in \mathrm{MB}_{\mathbf{0}}(U, V)$ if and only if D^* is a farthest halfspace in $\mathrm{H}(U^*, V^*)$.*

An example of four balls $U = \{B_1, \ldots, B_4\}$ is shown in Fig. 5.6(a), together with the dashed ball $D := \mathrm{MB}_{\mathbf{0}}(U, \{B_2\})$. Part (b) of the figure depicts the configuration after inversion w.r.t. the origin. The image D^* of D corresponds to the gray halfspace; it is the farthest among the halfspaces which avoid the origin, contain B_4^*, intersect B_1^* and B_3^*, and to which B_2^* is internally tangent.

The previous considerations imply that the following mathematical program searches for the halfspace(s) $\mathrm{MB}_{\mathbf{0}}(U, V)^*$ in the set $\mathrm{H}(U^*, V^*)$. (In this and the following mathematical programs we index the constraints by primal balls $B \in U$ for convenience; the constraints themselves involve the parameters d_B and σ_B of the inverted balls B^*.)

Corollary 5.10. *Let $U \supseteq V$, $U \neq \emptyset$, be two sets of balls, no ball in U containing the origin. Consider the program*

$$
\begin{array}{llrcll}
P_{\mathbf{0}}(U, V) & \textit{minimize} & \alpha & & & \\
& \textit{subject to} & v^T d_B + \alpha & \geq & \sigma_B, & B \in U \setminus V, \\
& & v^T d_B + \alpha & = & \sigma_B, & B \in V, \\
& & v^T v & = & 1, &
\end{array}
$$

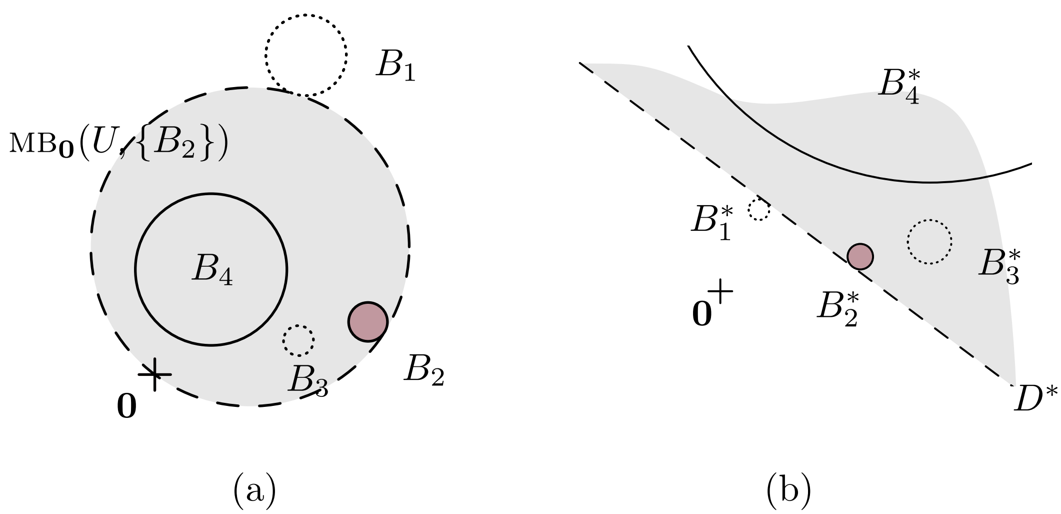


Figure 5.6. *(a) Four circles U and $D := \mathrm{MB}_{\mathbf{0}}(U, \{B_2\})$ (dashed). (b) The balls from (a) after inversion: dominance carries over in the sense of Lemma 5.7, so D^* must contain B_4^*, intersect B_1^* and B_3^*, and B_2^* must be internally tangent to it. (In addition to these requirements, D^* marginally dominates B_1^* in this example.)*

where the d_B and σ_B are the centers and radii of the inverted balls U^*, see Eq. (5.6). Then $D \in \mathrm{MB}_{\mathbf{0}}(U, V)$ if and only if

$$D^* = \{x \in \mathbb{R}^d \mid \tilde{v}^T x + \tilde{\alpha} \geq 0\}$$

for an optimal solution $(\tilde{v}, \tilde{\alpha})$ to the above program satisfying $\tilde{\alpha} < 0$.

The assumption $U \neq \emptyset$ guarantees $D \neq \mathbf{0}$; if U is empty, program $P_{\mathbf{0}}(U, V)$ consists of a quadratic constraint only and is thus unbounded.

5.4.2 The distance to the convex hull

With the material from the previous subsection at hand, we can easily relate problems SEBB and $\text{SEBB}_{\mathbf{0}}$ to the problem DHB of finding the point q in the convex hull $\mathrm{conv}(U) = \mathrm{conv}(\bigcup_{B \in U} B)$ of a given set U of positive balls that is nearest to some given point $p \in \mathbb{R}^d$ (Fig. 5.7). W.l.o.g. we may assume $p = \mathbf{0}$ in the following, in which case the problem amounts to finding the minimum-norm point in $\mathrm{conv}(U)$. Recall also from Lemma 3.9 that for a set U of positive balls, the set $\mathrm{MB}_{\mathbf{0}}(U)$ consists of at most one ball.

We claim that in the special case where all input balls are positive, the problems SEBB$_{\mathbf{0}}$ and DHB are equivalent in the sense that we can solve one with an algorithm for the other. To prepare this, observe that the respective problems are easy when the origin is contained in some input ball (which we can check in linear time): in case of DHB, we can then right away output '$q = \mathbf{0}$,' and for SEBB$_{\mathbf{0}}$ we can proceed as follows. If the origin is properly contained in some $B \in U$ then $\mathrm{MB}_{\mathbf{0}}(U) = \emptyset$, obviously. However, if B contains the origin on its boundary, the set $\mathrm{MB}_{\mathbf{0}}(U)$ might be nonempty (see the discussion following Lemma 3.10). In order to solve this case, we observe that any $D \in \mathrm{MB}_{\mathbf{0}}(U)$ must be tangent to B in the origin, and so its center c_D lies on the ray r through c_B, starting from $\mathbf{0}$. (We have $c_B \neq \mathbf{0}$ because we could remove $B = \mathbf{0}$ from the set U otherwise.) Thus, in order to determine $\mathrm{MB}_{\mathbf{0}}(U)$, we (conceptually) move a 'center' c on r in direction of c_B, starting from $\mathbf{0}$, and check how far we need to go until the ball $D_c := B(c, \|c\|)$ encloses all balls in U. Notice here that once a ball $B' \in U$ is contained in D_c, it will remain so when we continue moving c on r. Consequently, it suffices to compute for all $B' \in U$ a candidate center (which need not exist for $B' \neq B$) and finally select the candidate center c' that is farthest away from the origin: then $\mathrm{MB}_{\mathbf{0}}(U) = \{D_{c'}\}$ if $D_{c'}$ encloses all balls in U, or $\mathrm{MB}_{\mathbf{0}}(U) = \emptyset$ otherwise.

After this preprocessing step we are (in both problems) left with a set U of positive balls, none of which contains the origin. Our reduc-

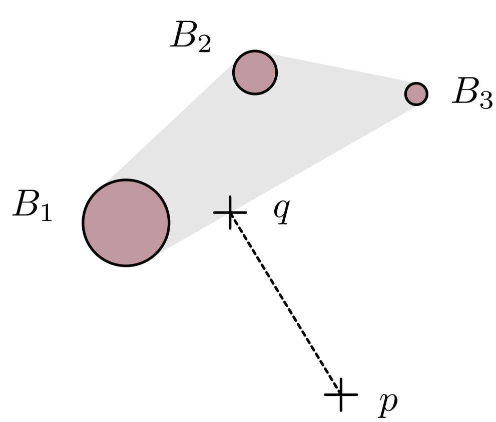


Figure 5.7. *The* DHB *problem: find the point q in the convex hull* conv(U) *(gray) of the positive balls U (solid) that lies closest to some given point $p \in \mathbb{R}^d$. In this example, $U = \{B_1, B_2, B_3\}$.*

tions between $\text{SEBB}_{\mathbf{0}}$ and DHB for such inputs are based on the following observation (which is easily proved using the material from page 160 in the book by Peressini, Sullivan & Uhl [67]).

Lemma 5.11. *Let $U \neq \emptyset$ be a set of positive balls. Then a point $q \neq \mathbf{0}$ (which we can always uniquely write as $q = -\alpha v$ with $v^T v = 1$ and $\alpha < 0$) is the minimum-norm point in $\operatorname{conv}(U)$ if and only if the halfspace (5.4) is the farthest halfspace in $\text{H}(U, \emptyset)$.*

In order to determine $\text{MB}_{\mathbf{0}}(U)$, we invoke an algorithm for problem DHB on U^*. If it delivers $q = \mathbf{0}$, we know $\text{MB}_{\mathbf{0}}(U) = \emptyset$ by Lemma 3.10. Otherwise, we write q as in the lemma with the result that the halfspace H from (5.4) is the farthest halfspace in $\text{H}(U, \emptyset)$, equivalently, that $H^* \in \text{MB}_{\mathbf{0}}(U)$ (Corollary 5.9). Conversely, in order to compute the minimum-norm point in $\operatorname{conv}(U)$, we run an algorithm for $\text{SEBB}_{\mathbf{0}}$ on U^*. If it outputs $\text{MB}_{\mathbf{0}}(U^*) = \emptyset$, we have $\mathbf{0} \in \operatorname{conv}(U^*)$ by Lemma 3.10, which is equivalent to $\mathbf{0} \in \operatorname{conv}(U)$. If on the other hand $D \in \text{MB}_{\mathbf{0}}(U^*)$ then D^* is a farthest halfspace in $\text{H}(U, \emptyset)$ (Corollary 5.9), which by the above lemma means that we can read the minimum-norm point $q \in \operatorname{conv}(U)$ off D^*.

We can also solve SEBB for signed balls with an algorithm $\mathcal{D}$ for DHB. For this, assume for the moment that we know the smallest (possibly negative) ball $O \in U$ that is marginally dominated by $\text{MB}(U)$. To obtain $\text{MB}(U)$, we find the balls $U' \subseteq U$ that shrink to positive balls under s_O, $U' := \{B \in U \mid s_O(B) \text{ positive}\}$. Since no ball in $U \setminus U'$ contributes to $\text{MB}(U)$ (Lemma 3.6), we have $\text{MB}(U) = \text{MB}(U')$, with $s_O(U')$ a set of positive balls. It follows that

$$s_O(\text{MB}(U)) = \text{MB}(s_O(U')) = \text{MB}_{c_O}(s_O(U' \setminus \{O\})) =: D_O,$$

and so it suffices to compute the latter ball using inversion (Corollary 5.10) and algorithm $\mathcal{D}$. From D_O, $\text{MB}(U)$ is easily reconstructed via Corollary 5.5.

As we do not know O in advance, we 'guess' it. For each possible guess $O' \in U$ the above procedure either results in a candidate ball $D_{O'}$, or $\mathcal{D}$ outputs that no such ball exists. Since $\text{MB}(U)$ itself appears as a candidate, it suffices to select the smallest candidate ball that encloses the input balls U in order to find $\text{MB}(U)$.

As long as U is a small set, the at most $|U|$ guesses introduce a negligible polynomial overhead. For large input sets however, a direct

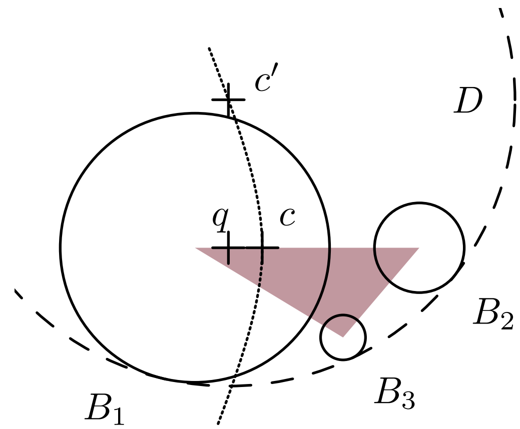


Figure 5.8. *The center c of the miniball of $U = \{B_1, B_2, B_3\}$ is not the point q in* $\mathrm{conv}(\{c_{B_1}, c_{B_2}, c_{B_3}\})$ *(gray) closest to the center c' of the circumball D of U; the dotted line constitutes the centers of all balls to which both B_1 and B_2 are internally tangent.*

application of the reduction leads to an unnecessarily slow algorithm. Thus, it pays off to run algorithm msw and use the reduction for the small cases only (where $|U| \leq d + 2$).

Finally, we note that it is well-known [32, 70] that small instances of the SEBP problem can be reduced to the problem DHP of finding the distance from a given point to the convex hull of a pointset P, together with the point in $\mathrm{conv}(P)$ where this distance is attained. (Again, algorithm msw can be used to handle large instances of DHP, once small cases can be dealt with.) The reduction is based on the following fact [69], which holds for points but is not true in general for balls (see Fig. 5.8).

Lemma 5.12. *Let $P \subset \mathbb{R}^d$ be an affinely independent pointset with circumcenter c'. The center of the ball* $\mathrm{MB}(P)$ *is the point in* $\mathrm{conv}(P)$ *with minimal distance to c'.*

Proof. Let C be the $(d \times |P|)$-matrix holding as columns the Euclidean centers of P. By Corollary 3.16, the center c of $\mathrm{MB}(P)$ fulfills $c = C\tilde{x}$, where $\tilde{x}$ is an optimal solution to the program $\mathcal{Q}(P, \emptyset)$ from p. 62.

Translate all points such that the origin of the coordinate system coincides with c'. By definition of the circumcenter we then have $\sum_{p \in P} p^T p x_p = \rho'^2$, where ρ' is the radius of the circumball. Thus, the objective function simplifies to $x^T C^T C x - \rho'^2$, and from this the claim follows. □

The miniball of $d+2$ points (recall that this is what we need for algorithm msw) can thus be found by solving $d+2$ instances of DHP, one for every subset of $d+1$ points.—We point out that this reduction is entirely different from the reductions for the balls case.

5.5 Small cases

We have shown in Sec. 5.3 that problem SEBB can be reduced to the problem $\text{SEBB}_{\mathbf{0}}$ of computing $\text{MB}_{\mathbf{0}}(T)$, for T some set of signed balls, $|T| \leq d+1$. Using the fact that we now have the *origin* fixed on the boundary we can improve over the previous complete enumeration approach, by using inversion and the concept of *unique sink orientations* [85].

In the sequel, we assume that T is a set of signed balls with linearly independent centers,[3] no ball in T containing the origin. The latter assumption is satisfied in our application, where $\text{MB}_{\mathbf{0}}(T)$ is needed only during the basis computation of algorithm msw (Fig. 2.3). The linear independence assumption is no loss of generality, because we can embed the balls into $\mathbb{R}^{d+1}$ and symbolically perturb them; in fact, this is easy if T comes from the set V during the basis computation basis(V, B) of the algorithm msw (see Sec. 5.5.2)

Our method for finding $\text{MB}_{\mathbf{0}}(T)$ computes as intermediate steps balls of the form

$$\text{MB}_{\mathbf{0}}(U, V) = \text{MB}(U \cup \{\mathbf{0}\}, V \cup \{\mathbf{0}\}),$$

for $V \subseteq U \subseteq T$. One obstacle we have to overcome for this is the possible nonexistence of $\text{MB}_{\mathbf{0}}(U, V)$: take for instance a positive ball B not containing the origin, place a positive ball B' into $\text{conv}(B \cup \{\mathbf{0}\})$, and set $U = \{B, B'\}$ and $V = \{B'\}$. (Such a configuration may turn up in our application.) Our solution employs the inversion transform: it defines for all pairs (U, V) a 'generalized ball' $\text{GMB}_{\mathbf{0}}(U, V)$ which coincides with $\text{MB}_{\mathbf{0}}(U, V)$ if the latter exists.

Performing inversion as described in the previous section gives us $|T| \leq d$ balls T^* with centers d_B and radii σ_B, $B \in T$, as in (5.6). The latter equation also shows that the d_B are linearly independent. The following lemma is then an easy consequence of previous considerations.

[3] For this, we interpret the centers as vectors, which is quite natural because of the translation employed in the reduction from SEBB_p to $\text{SEBB}_{\mathbf{0}}$.

Lemma 5.13. *For given $V \subseteq U \subseteq T$ with $U \neq \emptyset$, consider the following (nonconvex) optimization problem in the variables $v \in \mathbb{R}^d$, $\alpha \in \mathbb{R}$.*

$$\begin{array}{llrcll} \mathcal{P}_{\mathbf{0}}(U,V) & \textit{lexmin} & (v^T v, \alpha), & & & \\ & \textit{subject to} & v^T d_B + \alpha & \geq & \sigma_B, & B \in U \setminus V, \\ & & v^T d_B + \alpha & = & \sigma_B, & B \in V, \\ & & v^T v & \geq & 1. & \end{array}$$

Then the following two statements hold.

(i) $\mathcal{P}_{\mathbf{0}}(U,V)$ has a unique optimal solution $(\tilde{v}, \tilde{\alpha})$.

(ii) Let $H = \{x \in \mathbb{R}^d \mid \tilde{v}^T x + \tilde{\alpha} \geq 0\}$ be the halfspace defined by the optimal solution $(\tilde{v}, \tilde{\alpha})$. Then $H^* \in \mathrm{MB}_{\mathbf{0}}(U,V)$ if and only if $\tilde{v}^T \tilde{v} = 1$ and $\tilde{\alpha} < 0$.

Proof. (i) If we can show that $\mathcal{P}_{\mathbf{0}}(U,V)$ has a feasible solution then it also has an optimal solution, again using a compactness argument (this requires $U \neq \emptyset$). To construct a feasible solution, we first observe that by linear independence of the d_B, the system of equations

$$v^T d_B + \alpha = \sigma_B, \quad B \in U,$$

has a solution v for any given α; moreover, if we choose α large enough, any corresponding v must satisfy $v^T v \geq 1$, in which case (v, α) is a feasible solution.

To prove the uniqueness of the optimal solution, we again invoke linear independence of the d_B and derive the existence of a vector w (which we call an *unbounded direction*) such that

$$w^T d_B = 1, \quad B \in U. \tag{5.10}$$

Now assume that $\mathcal{P}_{\mathbf{0}}(U,V)$ has two distinct optimal solutions $(\tilde{v}_1, \tilde{\alpha})$, $(\tilde{v}_2, \tilde{\alpha})$ with $\tilde{v}_1^T \tilde{v}_1 = \tilde{v}_2^T \tilde{v}_2 = \delta \geq 1$. Consider any proper convex combination v of $\tilde{v}_1$ and $\tilde{v}_2$; v satisfies $v^T v < \delta$. Then there is a suitable positive constant Θ such that $(v + \Theta w)^T (v + \Theta w) = \delta$, and hence the pair $(v + \Theta w, \tilde{\alpha} - \Theta)$ is a feasible solution for $\mathcal{P}_{\mathbf{0}}(U,V)$, a contradiction to lexicographic minimality of the initial solutions.

(ii) Under $\tilde{v}^T \tilde{v} = 1$, this is equivalent to the statement of Corollary 5.10. □

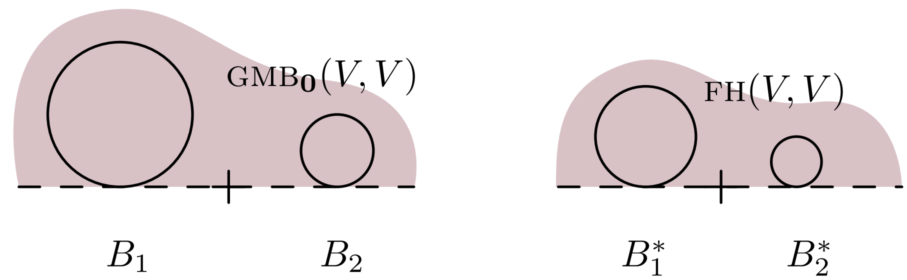


Figure 5.9. *Two balls $V = \{B_1, B_2\}$ (left) and their images under inversion (right). In this example, the value $(\tilde{v}, \tilde{\alpha})$ of (V,V) has $\tilde{\alpha} = 0$, in which case the 'generalized ball' $\textsc{gmb}_{\mathbf{0}}(V,V)$ is a halfspace.*

In particular, part (i) of the lemma implies that the set $\textsc{mb}_{\mathbf{0}}(U,V)$ contains *at most one ball* in our scenario, i.e., whenever the balls in U do not contain the origin and have linearly independent centers. Moreover, even if $\textsc{mb}_{\mathbf{0}}(U,V) = \emptyset$, program $\mathcal{P}_{\mathbf{0}}(U,V)$ has a unique optimal solution, and we call it the *value of* (U,V).

Definition 5.14. *For $U \supseteq V$ with $U \neq \emptyset$, the* value of (U,V), *denoted by* $\textsc{val}(U,V)$, *is the unique optimal solution $(\tilde{v}, \tilde{\alpha})$ of program $\mathcal{P}_{\mathbf{0}}(U,V)$, and we define $\textsc{val}(\emptyset,\emptyset) := (\mathbf{0}, -\infty)$. Moreover, we call the halfspace*

$$\textsc{fh}(U,V) := \{x \in \mathbb{R}^d \mid \tilde{v}^T x + \tilde{\alpha} \geq 0\},$$

the farthest (dual) halfspace *of (U,V). In particular, $\textsc{fh}(\emptyset,\emptyset) = \emptyset$.*

The farthest halfspace of (U,V) has a meaningful geometric interpretation even if $\textsc{mb}_{\mathbf{0}}(U,V) = \emptyset$. If the value $(\tilde{v}, \tilde{\alpha})$ of (U,V) satisfies $\tilde{v}^T\tilde{v} = 1$, we already know that $\textsc{fh}(U,V)$ dominates the balls in U^* and marginally dominates the balls in V^*, see Eq. (5.7). If on the other hand $\tilde{v}^T\tilde{v} > 1$, it is easy to see that the halfspace $\textsc{fh}(U,V)$ dominates the *scaled* balls

$$B(d_B, \sigma_B/\sqrt{\tau}) \qquad \text{with} \qquad \tau := \tilde{v}^T\tilde{v}, \tag{5.11}$$

for $B \in U$, and marginally dominates the scaled versions of the balls in V^* (divide the linear constraints of program $\mathcal{P}_{\mathbf{0}}(U,V)$ by $\sqrt{\tau}$ to see this). For an interpretation of $\textsc{fh}(U,V)$ in the primal, we associate to the pair (U,V) the 'generalized ball'

$$\textsc{gmb}_{\mathbf{0}}(U,V) := \textsc{fh}(U,V)^*,$$

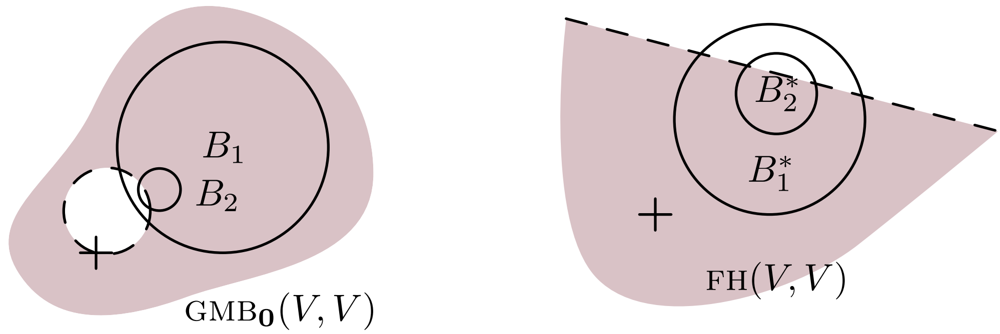


Figure 5.10. *Two positive balls $V = \{B_1, B_2\}$ (left) and their images under inversion (right). The value $(\tilde{v}, \tilde{\alpha})$ of (V,V) has $\tilde{v}^T\tilde{v} > 1$ and the 'generalized ball' $\text{GMB}_{\mathbf{0}}(V,V)$ is not tangent to the balls in V.*

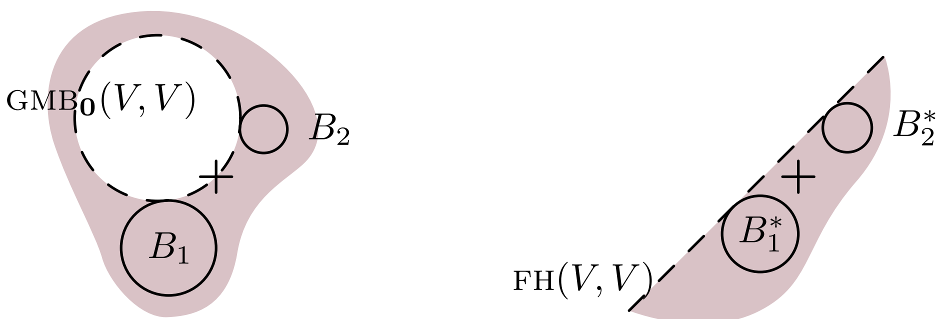


Figure 5.11. *Two positive balls $V = \{B_1, B_2\}$ (left) and their images under inversion (right). In this case, the value $(\tilde{v}, \tilde{\alpha})$ has $\tilde{v}^T\tilde{v} = 1$ but $\tilde{\alpha} > 0$, i.e., the balls do* not *admit a ball $\text{MB}_{\mathbf{0}}(V,V)$. Still, all balls V are 'internally' tangent to the 'generalized ball' $\text{GMB}_{\mathbf{0}}(V,V)$.*

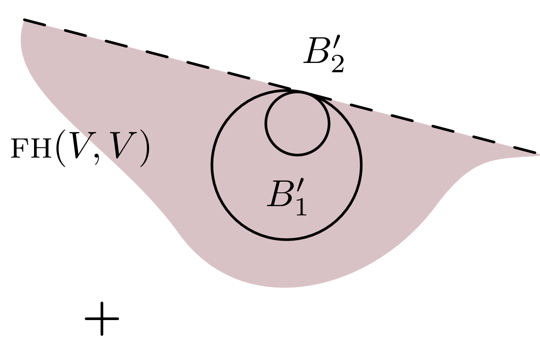


Figure 5.12. *The* scaled *balls $\{B_1', B_2'\}$, obtained from the balls $V^* = \{B_1^*, B_2^*\}$ in Fig. 5.10 by scaling their radii with $1/\sqrt{\tau}$, $\tau = \tilde{v}^T\tilde{v}$, are marginally dominated by $\text{FH}(V,V)$.*

which in general need *not* be a ball, as we will see. However, in the geometrically interesting case when the set $\mathrm{MB}_{\mathbf{0}}(U,V)$ is nonempty, it follows from Lemma 5.13(ii) that $\mathrm{GMB}_{\mathbf{0}}(U,V) = \mathrm{MB}_{\mathbf{0}}(U,V)$. Recall that this occurs precisely if the value $(\tilde{v},\tilde{\alpha})$ of the pair (U,V) fulfills $\tilde{v}^T\tilde{v} = 1$ and $\tilde{\alpha} < 0$.

In general, $\mathrm{GMB}_{\mathbf{0}}(U,V)$ can be a ball, the complement of an open ball, or a halfspace. In case $\tilde{\alpha} > 0$, the halfspace $\mathrm{FH}(U,V)$ contains the origin, and $\mathrm{GMB}_{\mathbf{0}}(U,V)$ hence is the complement of an open ball through the origin. If $\tilde{\alpha} = 0$ then $\mathrm{FH}(U,V)$ goes through the origin, and inversion does not provide us with a ball $\mathrm{GMB}_{\mathbf{0}}(U,V)$ but with a halfspace instead (Fig. 5.9). We remark that if $\tilde{v}^T\tilde{v} > 1$, $\mathrm{GMB}_{\mathbf{0}}(U,V)$ will not even be tangent to the proper balls in V (Fig. 5.10).

In Fig. 5.11, the inverted balls V^* do not admit a dominating halfspace that avoids the origin. Hence program $P_{\mathbf{0}}(V,V)$ has no solution, implying $\mathrm{MB}_{\mathbf{0}}(V,V) = \emptyset$. In order to obtain $\mathrm{GMB}_{\mathbf{0}}(V,V)$, we have to solve program $\mathcal{P}_{\mathbf{0}}(V,V)$. For this, we observe that the balls V^* admit two tangent hyperplanes, i.e., there are two halfspaces, parameterized by v and α, which satisfy the equality constraints of $\mathcal{P}_{\mathbf{0}}(V,V)$ with $v^Tv = 1$. Since the program in this case *minimizes* the distance to the halfspace, $\mathrm{FH}(V,V)$ is the enclosing halfspace corresponding to the 'upper' hyperplane in the figure (painted in gray). Since it contains the origin, $\mathrm{GMB}_{\mathbf{0}}(V,V)$ is the complement of a ball. Finally, Fig. 5.12 depicts the scaled versions (5.11) of the balls V^* from Fig. 5.10. Indeed, $\mathrm{FH}(V,V)$ marginally dominates these balls. (Since scaled balls do not invert to scaled balls in general—the centers may move—the situation is more complicated in the primal.)

We now investigate program $\mathcal{P}_{\mathbf{0}}(U,V)$ further. Although it is not a convex program, it turns out to be equivalent to one of two related convex programs. Program $\mathcal{C}'_{\mathbf{0}}(U,V)$ below finds the lowest point in a cylinder, subject to linear (in)equality constraints. In case it is infeasible (which will be the case if and only if $\mathrm{MB}_{\mathbf{0}}(U,V) = \emptyset$), the other program $\mathcal{C}_{\mathbf{0}}(U,V)$ applies in which case the cylinder is allowed to enlarge until the feasible region becomes nonempty.

Lemma 5.15. *Let $(\tilde{v},\tilde{\alpha})$ be the optimal solution to $\mathcal{P}_{\mathbf{0}}(U,V)$, for $U \neq \emptyset$, and let γ be the minimum value of the convex quadratic program*

$$\begin{array}{llrcll} \mathcal{C}_{\mathbf{0}}(U,V) & \textit{minimize} & v^Tv & & & \\ & \textit{subject to} & v^Td_B + \alpha & \geq & \sigma_B, & B \in U \setminus V, \\ & & v^Td_B + \alpha & = & \sigma_B, & B \in V. \end{array}$$

Then the following three statements hold.

(i) *Program* $\mathcal{C}_{\mathbf{0}}(U,V)$ *has a unique optimal solution, provided* $V \neq \emptyset$.

(ii) *If* $\gamma \geq 1$ *then* $(\tilde{v},\tilde{\alpha})$ *is the unique optimal solution to* $\mathcal{C}_{\mathbf{0}}(U,V)$.

(iii) *If* $\gamma \leq 1$ *then* $\tilde{v}^T\tilde{v}=1$ *and* $(\tilde{v},\tilde{\alpha})$ *is the unique optimal solution to the convex program*

$$\begin{array}{llcll} \mathcal{C}'_{\mathbf{0}}(U,V) & \textit{minimize} & \alpha & & \\ & \textit{subject to} & v^T d_B + \alpha & \geq & \sigma_B, \quad B \in U \setminus V, \\ & & v^T d_B + \alpha & = & \sigma_B, \quad B \in V, \\ & & v^T v & \leq & 1. \end{array}$$

Also, $\mathcal{C}_{\mathbf{0}}(U,V)$ *is strictly feasible (i.e., feasible values exist that satisfy all inequality constraints with strict inequality). If* $\gamma < 1$, $\mathcal{C}'_{\mathbf{0}}(U,V)$ *is strictly feasible, too.*

Proof. (i) A compactness argument shows that some optimal solution exists. Moreover, $\mathcal{C}_{\mathbf{0}}(U,V)$ has a unique optimal vector $\tilde{v}'$ because any proper convex combination of two different optimal vectors would still be feasible with smaller objective function value. The optimal $\tilde{v}'$ uniquely determines α because $\mathcal{C}_{\mathbf{0}}(U,V)$ has at least one equality constraint. (ii) Under $\gamma \geq 1$, $(\tilde{v},\tilde{\alpha})$ is an optimal solution to $\mathcal{C}_{\mathbf{0}}(U,V)$. By (i) it is the unique one because $\gamma \geq 1$ implies $V \neq \emptyset$. (iii) Under $\gamma \leq 1$, $\mathcal{C}'_{\mathbf{0}}(U,V)$ is feasible and a compactness argument shows that an optimal solution $(\tilde{v}',\tilde{\alpha}')$ exists. Using the unbounded direction (5.10) again, $\tilde{v}'^T\tilde{v}'=1$ and the uniqueness of the optimal solution can be established. Because $(\tilde{v}',\tilde{\alpha}')$ is feasible for $\mathcal{P}_{\mathbf{0}}(U,V)$, we have $\tilde{v}^T\tilde{v} = 1$, and from lexicographic minimality of $(\tilde{v},\tilde{\alpha})$, it follows that $(\tilde{v},\tilde{\alpha}) = (\tilde{v}',\tilde{\alpha}')$.

To see strict feasibility of $\mathcal{C}'_{\mathbf{0}}(U,V)$, first note that $\gamma < 1$ implies the existence of a feasible pair (v,α) for which $v^Tv < 1$. Linear independence of the d_B yields a vector w such that

$$w^T d_B = \begin{cases} 1, & B \in U \setminus V, \\ 0, & B \in V. \end{cases}$$

For sufficiently small $\Theta > 0$, the pair $(v + \Theta w, \alpha)$ is strictly feasible for $\mathcal{C}'_{\mathbf{0}}(U,V)$. Strict feasibility of $\mathcal{C}_{\mathbf{0}}(U,V)$ follows by an even simpler proof along these lines. □

This shows that given the minimum value γ of $\mathcal{C}_{\mathbf{0}}(U,V)$, the solution of $\mathcal{P}_{\mathbf{0}}(U,V)$ can be read off $\mathcal{C}_{\mathbf{0}}(U,V)$ (in case $\gamma \geq 1$) or $\mathcal{C}'_{\mathbf{0}}(U,V)$ (in case $\gamma \leq 1$). To characterize the optimal solutions of the latter programs we invoke the following version of the *Karush-Kuhn-Tucker Theorem* which is a specialization of Theorems 5.3.1 and 4.3.8 (with Slater's constraint qualification) in Bazaraa, Sherali & Shetty's book [5].

Theorem 5.16. *Let $f, g_1, \ldots, g_m$ be differentiable convex functions, let $a_1, \ldots, a_\ell \in \mathbb{R}^n$ be linearly independent vectors, and let $\beta_1, \ldots, \beta_\ell$ be real numbers. Consider the optimization problem*

$$\begin{array}{llcll} \textit{minimize} & f(x), & & & \\ \textit{subject to} & g_i(x) & \leq & 0, & i=1,\ldots,m, \\ & a_i^T x & = & \beta_i, & i=1,\ldots,\ell. \end{array} \tag{5.12}$$

(i) *If $\tilde{x}$ is an optimal solution to (5.12) and if there exists a vector $\tilde{y}$ such that*

$$\begin{array}{rcll} g_i(\tilde{y}) & < & 0, & i=1,\ldots,m, \\ a_i^T \tilde{y} & = & \beta_i, & i=1,\ldots,\ell, \end{array}$$

then there are real numbers $\mu_1, \ldots, \mu_m$ and $\lambda_1, \ldots, \lambda_\ell$ such that

$$\mu_i \geq 0, \quad i=1,\ldots,m, \tag{5.13}$$

$$\mu_i g_i(\tilde{x}) = 0, \quad i=1,\ldots,m, \tag{5.14}$$

$$\nabla f(\tilde{x}) + \sum_{i=1}^{m} \mu_i \nabla g_i(\tilde{x}) + \sum_{i=1}^{\ell} \lambda_i a_i \;=\; 0. \tag{5.15}$$

(ii) *Conversely, if $\tilde{x}$ is a feasible solution to program (5.12) such that numbers satisfying (5.13), (5.14) and (5.15) exist then $\tilde{x}$ is an optimal solution to (5.12).*

Applied to our two programs, we obtain the following optimality conditions.

Lemma 5.17. *Let $V \subseteq U \subseteq T$.*

(i) *A feasible solution $(\tilde{v}, \tilde{\alpha})$ for $\mathcal{C}_{\mathbf{0}}(U,V)$ is optimal if and only if there exist real numbers λ_B, $B \in U$, such that*

$$\begin{array}{rcll} \lambda_B & \geq & 0, & B \in U \setminus V \\ \lambda_B(\tilde{v}^T d_B + \tilde{\alpha} - \sigma_B) & = & 0, & B \in U \setminus V, \end{array}$$

$$\textstyle\sum_{B\in U} \lambda_B d_B \;=\; \tilde{v}, \tag{5.16}$$

$$\textstyle\sum_{B\in U} \lambda_B \;=\; 0. \tag{5.17}$$

(ii) A feasible solution $(\tilde{v}, \tilde{\alpha})$ *to* $\mathcal{C}'_{\mathbf{0}}(U, V)$ *satisfying* $\tilde{v}^T \tilde{v} = 1$ *is optimal if there exist real numbers* λ_B*,* $B \in U$*, such that*

$$\begin{aligned} \lambda_B &\geq 0, \quad B \in U \setminus V && (5.18)\\ \lambda_B(\tilde{v}^T d_B + \tilde{\alpha} - \sigma_B) &= 0, \quad B \in U \setminus V, && (5.19)\\ \textstyle\sum_{B \in U} \lambda_B d_B &= \tilde{v}, && (5.20)\\ \textstyle\sum_{B \in U} \lambda_B &> 0. && (5.21) \end{aligned}$$

Conversely, if $(\tilde{v}, \tilde{\alpha})$ *is an optimal solution to* $\mathcal{C}'_{\mathbf{0}}(U, V)$*, and if* $\mathcal{C}'_{\mathbf{0}}(U, V)$ *is strictly feasible (which in particular is the case if the minimum value* γ *of program* $\mathcal{C}_{\mathbf{0}}(U, V)$ *fulfills* $\gamma < 1$*) then there exist real numbers* λ_B*,* $B \in U$*, such that (5.18), (5.19), (5.20) and (5.21) hold.*

In both cases, the λ_B *are uniquely determined by* $\tilde{v}$ *via linear independence of the* d_B*.*

Unifying these characterizations, we obtain necessary and sufficient optimality conditions for the nonconvex program $\mathcal{P}_{\mathbf{0}}(U, V)$.

Theorem 5.18. *A feasible solution* $(\tilde{v}, \tilde{\alpha})$ *for program* $\mathcal{P}_{\mathbf{0}}(U, V)$ *is optimal if and only if there exist real numbers* λ_B*,* $B \in U$*, with* $\mu := \sum_{B \in U} \lambda_B$ *such that*

$$\begin{aligned} \lambda_B &\geq 0, \quad B \in U \setminus V,\\ \mu &\geq 0,\\ \lambda_B\,(\tilde{v}^T d_B + \tilde{\alpha} - \sigma_B) &= 0, \quad B \in U \setminus V,\\ \mu\,(\tilde{v}^T \tilde{v} - 1) &= 0, && (5.22)\\ \textstyle\sum_{B \in U} \lambda_B d_B &= \tilde{v}. \end{aligned}$$

Proof. The direction ($\Rightarrow$) follows through Lemmata 5.15 and 5.17, so it remains to settle ($\Leftarrow$). For this, we distinguish two cases, depending on the minimum value γ of program $\mathcal{C}_{\mathbf{0}}(U, V)$.

Consider the case $\gamma < 1$ first. If $\sum_{B \in U} \lambda_B = 0$ then Lemma 5.17(i) shows that $(\tilde{v}, \tilde{\alpha})$, which is clearly feasible for $\mathcal{C}_{\mathbf{0}}(U, V)$, is optimal to $\mathcal{C}_{\mathbf{0}}(U, V)$; hence $\gamma = \tilde{v}^T \tilde{v} \geq 1$, a contradiction. Thus $\sum_{B \in U} \lambda_B > 0$, which by (5.22) implies $\tilde{v}^T \tilde{v} = 1$. So $(\tilde{v}, \tilde{\alpha})$ is feasible and optimal to $\mathcal{C}'_{\mathbf{0}}(U, V)$ (Lemma 5.17(ii)), which together with Lemma 5.15(iii) establishes the claim.

In case $\gamma \geq 1$ the argument is as follows. If $\sum_{B \in U} \lambda_B = 0$ holds, the Lemmata 5.17(i) and 5.15(ii) certify that the solution $(\tilde{v}, \tilde{\alpha})$ is optimal to program $\mathcal{P}_{\mathbf{0}}(U,V)$. If on the other hand $\sum_{B \in U} \lambda_B > 0$ then $\tilde{v}^T \tilde{v} = 1$ by (5.22), and this shows $\gamma = 1$ because $(\tilde{v}, \tilde{\alpha})$ is feasible for program $\mathcal{C}_{\mathbf{0}}(U,V)$. Consequently, $(\tilde{v}, \tilde{\alpha})$ is feasible and optimal to $\mathcal{C}'_{\mathbf{0}}(U,V)$ (through Lemma 5.17(ii)) and hence optimal to program $\mathcal{P}_{\mathbf{0}}(U,V)$ by Lemma 5.15(iii) and $\gamma = 1$. □

As promised, we can state a version of Welzl's Lemma [86]. We prepare this by presenting the statement in the dual space, i.e., in terms of values of pairs (U,V) and associated halfspaces $\text{FH}(U,V)$.

Lemma 5.19. *Let $V \subseteq U \subseteq T$ and $B \in U \setminus V$. Denote by $(\tilde{v}, \tilde{\alpha})$ the value of the pair $(U \setminus \{B\}, V)$. Then*

$$\text{VAL}(U,V) = \begin{cases} \text{VAL}(U \setminus \{B\}, V), & \text{if } \tilde{v}^T d_B + \tilde{\alpha} \geq \sigma_B, \\ \text{VAL}(U, V \cup \{B\}), & \text{otherwise.} \end{cases}$$

As the value of a pair uniquely determines its associated farthest halfspace, the lemma holds also for farthest halfspaces (i.e., if we replace 'VAL' by 'FH' in the lemma). In this case, we obtain the following geometric interpretation. The halfspace $\text{FH}(U,V)$ coincides with the halfspace $\text{FH}(U \setminus \{B\}, V)$ if the latter dominates the scaled version (5.11) of ball B^*, and equals the halfspace $\text{FH}(U, V \cup \{B\})$ otherwise.

Proof. The case $U = \{B\}$ is easily checked directly, so assume $|U| > 1$. If $\tilde{v}^T d_B + \tilde{\alpha} \geq \sigma_B$ then $(\tilde{v}, \tilde{\alpha})$ is feasible and hence optimal to the more restricted problem $\mathcal{P}_{\mathbf{0}}(U,V)$, and $\text{VAL}(U,V) = \text{VAL}(U \setminus \{B\}, V)$ follows. Otherwise, the value $(\tilde{v}', \tilde{\alpha}')$ of (U,V) is different from $(\tilde{v}, \tilde{\alpha})$. Now consider the coefficient λ'_B resulting from the application of Theorem 5.18 to $(\tilde{v}', \tilde{\alpha}')$. We must have $\lambda'_B \neq 0$, because Theorem 5.18 would otherwise certify that $(\tilde{v}', \tilde{\alpha}') = \text{VAL}(U \setminus \{B\}, V)$. This, however, implies that

$$\tilde{v}'^T d_B + \tilde{\alpha}' = \sigma_B,$$

from which we conclude $\text{VAL}(U,V) = \text{VAL}(U, V \cup \{B\})$. □

Here is the fix for Dilemma 5.1 in the case when the input ball centers are affinely independent.

Lemma 5.20. *Let $V \subseteq U$, where U is any set of signed balls with affinely independent centers, and assume* $\textsc{mb}(U,V) \neq \emptyset$. *Then the sets* $\textsc{mb}(U,V)$ *and* $\textsc{mb}(U \setminus \{B\}, V)$ *are singletons, for any $B \in U \setminus V$. Moreover, if no ball in V is dominated by another ball in U, and if*

$$B \text{ is not dominated by } \textsc{mb}(U \setminus \{B\}, V), \tag{5.23}$$

for some $B \in U \setminus V$, then $\textsc{mb}(U,V) = \textsc{mb}(U, V \cup \{B\})$, *and B is not dominated by another ball in U, either.*

It easily follows by induction that Welzl's algorithm `sebb` (Fig. 5.1, with the test '$B \not\subseteq D$' replaced by 'B not dominated by D') computes $\textsc{mb}(U)$ for a set of signed balls, provided the centers of the input balls are affinely independent (a perturbed embedding into $\mathbb{R}^{|U|-1}$ always accomplishes this). No other preconditions are required; in particular, balls can overlap in an arbitrary fashion.

Proof. For $V = \emptyset$, this is Lemma 3.6, with the obvious generalization to signed balls (refer to the discussion after Corollary 5.5). For all V, transitivity of the dominance relation shows that if B is not dominated by $\textsc{mb}(U \setminus \{B\}, V)$, it cannot be dominated by a ball in $U \setminus \{B\}$, either.

In case $V \neq \emptyset$, we fix any ball $O \in V$ and may assume—after a suitable translation and a shrinking step—that $O = \mathbf{0}$; Eq. (5.23) is not affected by this. Moreover, we can assume that O does not dominate any other (negative) ball in $U \setminus V$: such a ball can be removed from consideration (and added back later), without affecting the miniball (here, we again use transitivity of dominance).

Then, no ball in U contains $O = \mathbf{0}$, and the centers of the balls

$$U' = U \setminus \{O\}$$

are *linearly independent*. Under (5.23), we have $B \in U'$. Therefore, we can apply our previous machinery. Setting

$$V' = V \setminus \{O\},$$

Lemma 5.13 yields that the two sets $\textsc{mb}(U,V) = \textsc{mb}_{\mathbf{0}}(U',V')$ and $\textsc{mb}(U \setminus \{B\}, V) = \textsc{mb}_{\mathbf{0}}(U' \setminus \{B\}, V')$ contain at most one ball each. Also, the assumption $\textsc{mb}(U,V) \neq \emptyset$ implies $\textsc{mb}(U \setminus \{B\}, V) \neq \emptyset$ (this is easily verified using the program in Lemma 5.13). Consequently, the ball sets are singletons.

Now let $(\tilde{v}, \tilde{\alpha})$ be the value of the pair $(U' \setminus \{B\}, V')$. As $\textsc{mb}_{\mathbf{0}}(U' \setminus \{B\}, V') \neq \emptyset$, we have $\tilde{v}^T\tilde{v} = 1$ (Lemma 5.13). Then, Lemma 5.7 shows that B is not dominated by the ball $\textsc{mb}_{\mathbf{0}}(U' \setminus \{B\}, V')$ if and only if $\tilde{v}^T d_B + \tilde{\alpha} < \sigma_B$ holds, for d_B, σ_B being center and radius of the inverted ball B^*. Lemma 5.19 in turn implies

$$\textsc{val}(U', V') = \textsc{val}(U', V' \cup \{B\}), \tag{5.24}$$

and from this, $\textsc{fh}(U', V') = \textsc{fh}(U', V' \cup \{B\})$ along with $\textsc{gmb}_{\mathbf{0}}(U', V') = \textsc{gmb}_{\mathbf{0}}(U', V' \cup \{B\})$ follows. By assumption, the former 'generalized ball' coincides with $\textsc{mb}_{\mathbf{0}}(U', V')$, from which it follows that the value $(\tilde{v}', \tilde{\alpha}')$ of (U', V') fulfills $\tilde{v}'^T\tilde{v}' = 1$ and $\tilde{\alpha}' < 0$ (Lemma 5.13). By (5.24), this shows that $\textsc{gmb}_{\mathbf{0}}(U', V' \cup \{B\}) = \textsc{mb}_{\mathbf{0}}(U', V' \cup \{B\})$, which establishes the lemma. □

5.5.1 The unique sink orientation

In this last part we want to use the results developed so far to reduce the problem of finding $\textsc{mb}_{\mathbf{0}}(T)$ to the problem of finding the sink in a unique sink orientation. To this end, we begin with a brief recapitulation of unique sink orientations and proceed with the presentation of our orientation.

As in the previous subsection, we consider a set T of $m \leq d$ balls such that the centers of T are linearly independent and such that no ball in T contains the origin. Consider the m-dimensional cube. Its vertices can be identified with the subsets $J \subseteq T$; faces of the cube then correspond to *intervals* $[V, U] := \{J \mid V \subseteq J \subseteq U\}$, where $V \subseteq U \subseteq T$. We consider the *cube graph*

$$\mathcal{G} = (2^T, \{\{J, J \oplus \{B\}\} \mid J \in 2^T, B \in T\}),$$

where $\oplus$ denotes symmetric difference. An orientation $\mathcal{O}$ of the edges of $\mathcal{G}$ is called a *unique sink orientation* (USO) if for any nonempty face $[V, U]$, the subgraph of $\mathcal{G}$ induced by the vertices of $[V, U]$ has a unique sink w.r.t. $\mathcal{O}$ [85].

As before, we write d_B and σ_B for the center and radius of the inverted balls $B^* \in T^*$, see (5.6). The following is the main result of this section.

Theorem 5.21. *Let f be the objective function of program $\mathcal{P}_{\mathbf{0}}(T, \emptyset)$. Then the pair $(T, \leq_{\text{lex}}, \text{VAL})$ with the total order*

$$\text{VAL}(U', V') \leq \text{VAL}(U, V) \;\Leftrightarrow\; f(\text{VAL}(U', V')) \leq_{\text{lex}} f(\text{VAL}(U, V)),$$

is a reducible strong LP-type problem.

Observe that the image of the function f is $\mathbb{R} \times \mathbb{R} \cup \{\infty\}$; the lexicographical order $\leq_{\text{lex}}$ is a total order on this set.

Proof. Monotonicity of w is clearly satisfied because dropping a constraint cannot degrade the objective value. Upper and lower uniqueness are implied by the fact that $\leq_{\text{lex}}$ is a total order on the image of w, and reducibility follows from Lemma 5.19. Thus, it remains to prove that w satisfies strong locality.

So suppose $(\tilde{v}, \tilde{\alpha}) := \text{VAL}(U', V') = \text{VAL}(U, V)$ for sets $V' \subseteq U'$ and $V \subseteq U$. The case $\tilde{\alpha} = \infty$ (which implies $U' = V' = U = V = \emptyset$) is easily checked directly, so we can assume $U' \neq \emptyset$ and $U \neq \emptyset$, which allows us to make use of Theorem 5.18. Observe first that $(\tilde{v}, \tilde{\alpha})$ is a feasible solution to the programs $\mathcal{P}_{\mathbf{0}}(U' \cap U, V' \cap V)$ and $\mathcal{P}_{\mathbf{0}}(U' \cup U, V' \cup V)$. Given this, we verify optimality by means of the unique Karush-Kuhn-Tucker multipliers λ_B, $B \in T$, that come with $(\tilde{v}, \tilde{\alpha})$ (Theorem 5.18). As $(\tilde{v}, \tilde{\alpha})$ optimally solves $\mathcal{P}_{\mathbf{0}}(U', V')$ and $\mathcal{P}_{\mathbf{0}}(U, V)$, we must have

$$\begin{aligned} \lambda_B &= 0, && B \notin U' \cap U, \\ \lambda_B &\geq 0, && B \in (U' \cup U) \setminus (V' \cap V), \\ \lambda_B\,(\tilde{v}^T d_B + \tilde{\alpha} - \sigma_B) &= 0, && B \in (U' \cup U) \setminus (V' \cap V), \end{aligned}$$

from which its follows via Theorem 5.18 again that $(\tilde{v}, \tilde{\lambda})$ optimally solves $\mathcal{P}_{\mathbf{0}}(U' \cap U, V' \cap V)$ and $\mathcal{P}_{\mathbf{0}}(U' \cup U, V' \cup V)$. Hence, $(\tilde{v}, \tilde{\lambda})$ equals $\text{VAL}(U' \cap U, V' \cap V) = \text{VAL}(U' \cup U, V' \cup V)$. □

As a strong LP-type problem, $(T, \leq_{\text{lex}}, \text{VAL})$ induces an orientation on the cube $C^{[T,\emptyset]}$ that satisfies the *unique sink property.* More precisely, Theorem 2.22 from Chap. 2 yields

Corollary 5.22. *Consider the orientation $\mathcal{O}$ of $C^{[T,\emptyset]}$ defined by*

$$J \to J \cup \{B\} \quad :\Leftrightarrow \quad \text{VAL}(J, J) \neq \text{VAL}(J \cup \{B\}, J). \tag{5.25}$$

Then $\mathcal{O}$ is a USO, and the sink S of $\mathcal{O}$ is a strong basis of T, meaning that S is inclusion-minimal with $\text{VAL}(S, S) = \text{VAL}(T, \emptyset)$.

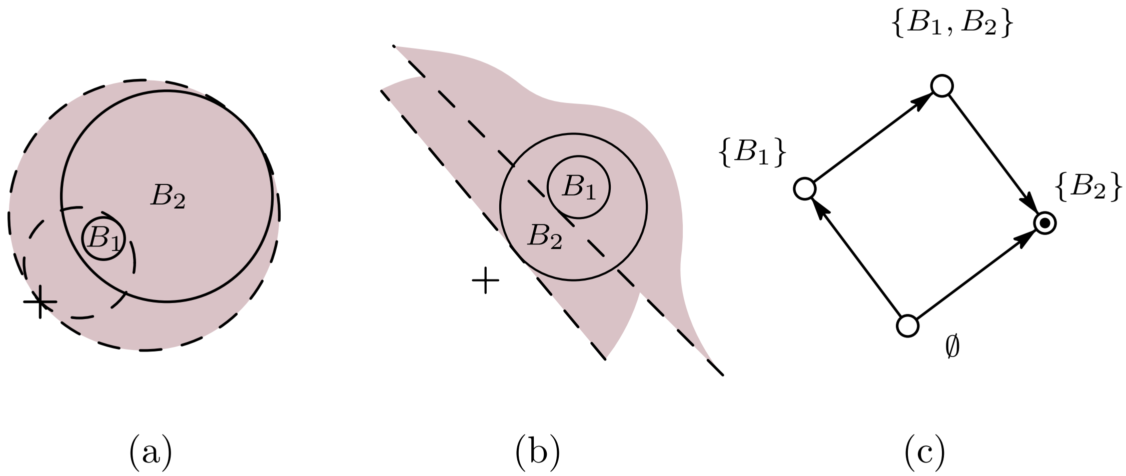


Figure 5.13. *The USO (c) from Corollary 5.22 for a set $T = \{B_1, B_2\}$ of two circles (a). A vertex $J \subseteq T$ of the cube corresponds to the solution* VAL(J, J) *of program $\mathcal{P}_{\mathbf{0}}(J, J)$ and represents a halfspace* FH(J, J) *in the dual (b) and the ball* GMB$_{\mathbf{0}}(J, J)$ *(gray) in the primal (a).*

Specialized to the case of points, this result is already known [40]; however, our proof removes the general position assumption.

In terms of halfspaces FH(U, V), we can interpret this geometrically as follows. The edge $\{J, J \cup \{B\}\}$ is directed towards the larger set if and only if the halfspace FH(J, J) does not dominate the scaled version (5.11) of ball B^*. Figure 5.13 illustrates the theorem for a set T of two circles. A vertex $J \subseteq T$ of the cube in part (c) of the figure corresponds to the solution VAL(J, J) of program $\mathcal{P}_{\mathbf{0}}(J, J)$ and represents a halfspace FH(J, J) in the dual (part (b) of the figure) and the ball GMB$_{\mathbf{0}}(J, J)$ shown in gray in the primal (part (a) of the figure). Every edge $\{J, J \cup \{B\}\}$ of the cube is oriented towards $J \cup \{B\}$ if and only if the halfspace FH(J, J) does not dominate the scaled version of B^* (which in this example is B^* itself). The global sink S in the resulting orientation corresponds to the inclusion-minimal subset S with VAL$(S, S) =$ VAL$(T, \emptyset)$. For the definition of the USO, the halfspace FH(T, T) is irrelevant, and since FH$(\emptyset, \emptyset)$ does not dominate any ball, all edges incident to $\emptyset$ are outgoing (therefore, the figure does not show these halfspaces).

Solution via USO-framework. In order to apply USO-algorithms [85] to find the sink of our orientation $\mathcal{O}$, we have to evaluate the orientation of an edge $\{J, J \cup \{B\}\}$, i.e., we must check

$$\text{VAL}(J, J) \neq \text{VAL}(J \cup \{B\}, J). \tag{5.26}$$

If $J = \emptyset$, this condition is always satisfied. Otherwise, we first solve program $\mathcal{C}_\mathbf{0}(J, J)$, which is easy: by the Karush-Kuhn-Tucker conditions from Lemma 5.17(i), it suffices to solve the linear system consisting of the Eqs. (5.16), (5.17), and the feasibility constraints $v^T d_B + \alpha = \sigma_B$, $B \in J$. We know that this system is regular because the optimal solution is unique and uniquely determines the Karush-Kuhn-Tucker multipliers.

If the solution $(\tilde{v}, \tilde{\alpha})$ satisfies $\tilde{v}^T \tilde{v} \geq 1$, we have already found the value $(\tilde{v}, \tilde{\alpha})$ of (J, J) (Lemma 5.15(i)), and we simply check whether

$$\tilde{v}^T d_B + \tilde{\alpha} < \sigma_B, \tag{5.27}$$

a condition equivalent to (5.26). If $\tilde{v}^T \tilde{v} < 1$, we solve $\mathcal{C}'_\mathbf{0}(J, J)$, which we can do by *reusing* the solution of $\mathcal{C}_\mathbf{0}(J, J)$ as the follow lemma, developed by Geigenfeind [41] in a semester thesis, shows.

Lemma 5.23. *Let $(\tilde{v}, \tilde{\alpha})$ with $\tilde{v}^T \tilde{v} < 1$ be the optimal solution to $\mathcal{C}_\mathbf{0}(J, J)$, and let x be the (unique) solution to*

$$D^T D x = \mathbf{1},$$

where the matrix D contains the points d_B, $B \in J$, as its columns. Then the point $(\tilde{v}', \tilde{\alpha}') := (\tilde{v} + \Theta w, \tilde{\alpha} - \Theta)$ with

$$w = Dx \qquad \text{and} \qquad \Theta = \sqrt{\frac{1 - \tilde{v}^T \tilde{v}}{w^T w}} > 0$$

is the optimal solution to program $\mathcal{C}'_\mathbf{0}(J, J)$.

Proof. By Lemma 5.17(ii), it suffices to show that the point $(\tilde{v}', \tilde{\alpha}')$ is feasible (i.e., $D^T \tilde{v}' + \tilde{\alpha}' \mathbf{1} = \sigma$ and $\tilde{v}'^T \tilde{v}' = 1$) and that there are real numbers λ' such that $\mathbf{1}^T \lambda' > 0$ and $\tilde{v}' = D\lambda'$.

Something we need for both these parts is the identity $\tilde{v}^T w = 0$; so let us settle this first. As the optimal solution to $\mathcal{C}_\mathbf{0}(J, J)$, the pair $(\tilde{v}, \tilde{\alpha})$ satisfies $D\lambda = \tilde{v}$ and $\mathbf{1}^T \lambda = 0$ for some real vector λ (Lemma 5.17(i)). It follows that

$$\tilde{v}^T w = \lambda^T D^T D x = \lambda^T \mathbf{1} = 0.$$

With this at hand, the choice of Θ implies

$$\tilde{v}'^T\tilde{v}' = \tilde{v}^T\tilde{v} + \Theta^2 w^T w = \tilde{v}^T\tilde{v} + \frac{1-\tilde{v}^T\tilde{v}}{w^T w} w^T w = 1$$

and using $D^T w = \mathbf{1}$,

$$D^T\tilde{v}' + \tilde{\alpha}'\mathbf{1} = D^T\tilde{v} + \tilde{\alpha}\mathbf{1} + \Theta D^T w - \Theta\mathbf{1} = D^T\tilde{v} + \tilde{\alpha}\mathbf{1} = s;$$

these two equations together show feasibility. As to optimality, we take $\tilde{v}' = \tilde{v} + \Theta w = D\lambda + \Theta Dx$ as a motivation for setting $\lambda' := \lambda + \Theta x$. Indeed, this implies

$$\mathbf{1}^T\lambda' = \mathbf{1}^T\lambda + \Theta\mathbf{1}^T x = 0 + \Theta x^T D^T D x = \Theta w^T w > 0,$$

as desired. □

Equation (5.26) gives an easy way to evaluate the orientation of the upward edge $\{J, J \cup \{B\}\}$, given the value of (J, J). We note that the orientation of the downward edge $\{J, J \setminus \{B\}\}$ can be read off the Karush-Kuhn-Tucker multiplier λ_B associated with $\textsc{val}(J, J)$: orient from $J \setminus \{B\}$ towards J if and only if $\lambda_B > 0$.

Lemma 5.24. *Let $B \in J \subseteq T$. The multiplier λ_B of $\textsc{val}(J, J)$ is strictly positive if and only if*

$$\textsc{val}(J \setminus \{B\}, J \setminus \{B\}) \neq \textsc{val}(J, J \setminus \{B\}), \tag{5.28}$$

i.e., we orient the edge from $J \setminus \{i\}$ towards J if and only if $\lambda_B > 0$.

Proof. By Lemma 5.5, $\textsc{val}(J, J)$ equals $\textsc{val}(J, J \setminus \{i\})$ if and only if $\lambda_i \geq 0$. With this at hand, we distinguish three cases. If $\lambda_i < 0$, Lemma 5.19 guarantees that

$$\textsc{val}(J, J \setminus \{i\}) \in \{\textsc{val}(J \setminus \{i\}, J \setminus \{i\}), \textsc{val}(J, J)\}.$$

So if $\lambda_i < 0$, or, equivalently, $\textsc{val}(J, J) \neq \textsc{val}(J, J \setminus \{i\})$, then Eq. (5.28) must be false.

If $\lambda_i = 0$, we have $\textsc{val}(J, J) = \textsc{val}(J, J \setminus \{i\})$ and via $\lambda_i = 0$ and Theorem 5.18, the latter clearly equals $\textsc{val}(J \setminus \{i\}, J \setminus \{i\})$.

Finally, also in case $\lambda_i > 0$ we have $\textsc{val}(J, J) = \textsc{val}(J, J \setminus \{i\})$, but an invocation of Theorem 5.18 shows that $\textsc{val}(J, J \setminus \{i\})$ and $\textsc{val}(J \setminus \{i\}, J \setminus \{i\})$ cannot be equal. □

With the currently best known USO algorithm (Theorem 2.13) we can find the sink of an m-dimensional USO with an expected number of $\mathcal{O}(c^m)$ vertex evaluations, where $c \approx 1.438$. Since in our case a vertex evaluation (determine the orientations of all edges incident to some vertex) essentially requires to solve one system of linear equations, we obtain an expected running time of $\mathcal{O}(d^3 c^m)$ to solve problem $\textsc{sebb}_\mathbf{0}$ for a set of $m \leq d$ signed balls (and by invoking Theorem 5.6, we can also solve the general problem).

5.5.2 Symbolic perturbation

In this final section we show how the unique sink orientation from Corollary 5.22 can be implemented efficiently in practice. Until now, the theorem only applies under the assumption that the signed input balls have linearly independent centers and no ball contains the origin. While this can always be achieved in theory via an embedding into $\mathbb{R}^{|T|}$ and a subsequent symbolic perturbation à la [24], doing so in practice results in an unnecessarily complicated and inefficient procedure.—As we will argue, the unique sink orientation from Corollary 5.22 is in fact *nicely tailored* to our main application, the solution of the basis computation in algorithm msw.

Let us briefly recapitulate the situation here. Our goal is to implement the basis computation of algorithm msw for problem $\textsc{sebb}$. That is, we need to compute a basis of $V_{\rm in} \cup \{D\}$, where $V_{\rm in}$ is a set of signed balls in $\mathbb{R}^d$ forming a basis, and where D is a ball violating the basis (i.e., not dominated by $\textsc{mb}(V_{\rm in})$). Through Lemma 3.6 we already know that the ball D will be part of the new basis. Consequently, it suffices to find an inclusion-minimal subset $V' \subseteq V_{\rm in}$ such that

$$\textsc{mb}(V' \cup \{D\}, \{D\}) = \textsc{mb}(V_{\rm in} \cup \{D\}, \{D\}).$$

By Corollary 5.5, this is equivalent to $\textsc{mb}_{c_D}(s_D(V')) = \textsc{mb}_{c_D}(s_D(V_{\rm in}))$. Therefore, we translate all balls such that the center of D coincides with the origin and shrink them by the radius of D. (Notice that at this point the centers c_B, $B \in s_D(V_{\rm in})$, are still affinely independent by Lemma 3.8.) After inverting the shrunken balls $V := s_D(V_{\rm in})$, we end up with at most $d+1$ balls of centers d_B and radii σ_B, $B \in V$. (Notice that the shrunken D is not 'present' anymore after inversion, i.e., it is not among the balls V.)

Lemma 5.25. *Let V be a set of signed balls forming a basis, and denote by d_B the inverted center of ball $B \in V$, see Eq. (5.6). If no ball in V contains the origin then the points*

$$d'_B := (d_B, \epsilon/\gamma_B), \qquad B \in V, \tag{5.29}$$

where $\gamma_B := c_B^2 - \rho_B^2$, are linearly independent for all $\epsilon > 0$.

Proof. Write $V = \{B_1, \ldots, B_n\}$. Since the centers of the balls in V are affinely independent (Lemma 3.8), the matrix

$$\begin{bmatrix} c_1 & \cdots & c_n \\ 1 & \cdots & 1 \end{bmatrix}$$

has full rank, i.e., its columns are linearly independent. This does not change if we multiply some of the rows and columns of the matrix by nonzero constants. Therefore, the matrix

$$\begin{bmatrix} \frac{c_{B_1}}{c_{B_1}^2 - \rho_{B_1}^2} & \cdots & \frac{c_{B_n}}{c_{B_n}^2 - \rho_{B_n}^2} \\ \frac{\epsilon}{c_{B_1}^2 - \rho_{B_1}^2} & \cdots & \frac{\epsilon}{c_{B_n}^2 - \rho_{B_n}^2} \end{bmatrix}$$

has linearly independent columns, too. And as $d_{B_i} = c_{B_i}/(c_{B_i}^2 - \rho_{B_i}^2)$ for all i, these columns precisely coincide with the d'_{B_i} and the claim follows. □

The lemma suggest the following approach. Instead of directly taking the balls V^* to establish a unique sink orientation—which we cannot always do for possible lack of general position—we take the balls

$$T_\epsilon^* := \{B(d'_B, \sigma_B)) \mid B \in V\};$$

the lemma shows that for any $\epsilon > 0$ they fulfill the general position assumption of Corollary 5.22,[4] and thus the machinery from the previous section applies. In particular, we obtain a unique sink orientation for every $\epsilon > 0$, and for every such ϵ, a USO-algorithm delivers a basis of $\mathrm{MB}_{\mathbf{0}}((T_\epsilon^*)^*) = \mathrm{MB}_{\mathbf{0}}(T_\epsilon)$ to us. The next lemma proves that this basis does not change if we let (a small enough) ϵ go to zero. And as the points T_0 correspond to our initial pointset V (with the only difference that the former points are embedded in $\mathbb{R}^d$), a basis of T_ϵ is also a basis of V (and hence of V_{in}).

[4]Here, we use (again) the fact that a pointset P is linearly independent if and only if P^* is linearly independent.

Lemma 5.26. *Let $S_\epsilon \subseteq T_\epsilon$ be the sink of the USO* (5.25) *for the balls T_ϵ. Then there exists $\epsilon^* > 0$ such that S_ϵ is constant on* $(0, \epsilon^*)$.

Proof. In the discussion after Corollary 5.22 up to Lemma 5.23 we have seen that in order to determine the orientation (5.25) of an edge $(J, J \oplus \{B\})$, it suffices to solve a system of linear equations (into which the d'_B, $B \in J$, go) and then evaluate the sign of two expressions (one expression to test if $\mathcal{C}_{\mathbf{0}}(J,J)$ or $\mathcal{C}'_{\mathbf{0}}(J,J)$ applies and one for the actual test (5.27)). By (5.29), these expressions are polynomials of bounded degree in ϵ, and hence Cauchy's Theorem implies that the signs of these expressions do not change in a sufficiently small interval $(0, \epsilon^*)$. $\square$

In practice, it is not necessary to compute the number ϵ^*; the limits of the involved expressions can easily be derived analytically.

We point out that this tailored symbolic perturbation results in points d_B whose coordinates are *linear* in ϵ (and we thus only need degree-one polynomial arithmetic, essentially); a general perturbation scheme would set $d'_B := (d_B, \epsilon^i)$, resulting in high-degree polynomials.

Also, we remark that in an implementation targeted at rational input, it might be unfavorable to explicitly perform the inversion of the input balls: the division in (5.6) forces us to deal with *quotients* (imposing an additional restriction on the input (ring) number type) and might introduce an inadvisable growth in the size of intermediate results. These issues can be dealt with however, essentially by shifting the division from the points c_D to the coefficients λ_B of programs $\mathcal{C}_{\mathbf{0}}(J,J)$ and $\mathcal{C}'_{\mathbf{0}}(J,J)$, that is, by working with the numbers $\tau_B := \lambda_B/(c_B^T c_B - \rho_B^2)$ instead of the λ_B.

Chapter 6

More programs in subexponential time

In this final part of the thesis, we improve the *exponential* worst-case complexity $\mathcal{O}(d^3 1.438^d)$ obtained in the previous chapter for solving small instances of SEBB to a *subexponential* bound.

In order to achieve this, we first formulate SEBB—actually, the variant $\text{SEBB}_\mathbf{0}$ of it—as a convex *mathematical program* $\mathcal{P}$, that is, as the problem of minimizing a convex function over a convex feasibility domain. Next, we develop an *abstract optimization problem* (see Chap. 2) with the property that any of its best bases yields an optimal solution to $\mathcal{P}$. Thus, in order to solve SEBB we can run Gärtner's randomized, subexponential algorithm (Theorem 2.10) to solve the AOP and with it, the program $\mathcal{P}$.

Part of the AOP we devise is an *improving oracle* which Gärtner's algorithm repeatedly calls, its task being to deliver for a given AOP basis a better one (in a certain subset of the groundset), if possible. Our method for realizing this follows an improving path in the feasibility region of program $\mathcal{P}$. The way we construct this path is inspired by Gärtner & Schönherr's method [38, 72] for solving convex quadratic programs. (If both program in Lemma 5.15 were quadratic, we would use Gärtner & Schönherr's method, together with the AOP algorithm, directly.)

Although the main result of this chapter is the subexponential bound for SEBB, we keep the presentation abstract enough so that it applies to some other convex mathematical programs as well.

6.1 A mathematical program for SEBB$_\mathbf{0}$

In this section we show that given an instance T of problem SEBB$_\mathbf{0}$ with *linearly independent* ball centers, the desired ball MB$_\mathbf{0}(T)$ can be read off the following mathematical program $\mathcal{D}(T,\emptyset)$ (whose precise definition is given below).

$$\begin{array}{llll} \mathcal{D}(U,V) & \text{minimize} & \frac{1}{4\mu}x^TQ^TQx+\mu-\sum_{B\in T}x_B\sigma_B & \\ & \text{subject to} & \sum_{B\in T}x_B=1, & \\ & & x_B\geq 0, & B\in U\setminus V, \\ & & x_B=0, & B\in T\setminus V, \\ & & \mu\geq 0. & \end{array}$$

Using this result and an algorithm $\mathcal{A}$ to solve $\mathcal{D}(U,V)$, problem SEBB can then be solved as follows. From Theorem 5.6 we know already that algorithm msw reduces problem SEBB over a set of n signed d-dimensional balls to problem SEBB$_\mathbf{0}$ over a set of at most $d{+}1$ signed balls in $\mathbb{R}^d$ (each of the latter instances corresponds to the input of a basis computation of msw). Moreover, the findings in Sec. 5.5.2 allow us to *enforce linear independence* of the ball centers: the combined embedding and perturbation of Lemma 5.25 produces from the given instance T of SEBB$_\mathbf{0}$ in $\mathbb{R}^d$ an instance T' of SEBB$_\mathbf{0}$ in $\mathbb{R}^{d+1}$ whose at most d balls have linearly independent centers. Consequently, we are in a position to invoke algorithm $\mathcal{A}$ on program $\mathcal{D}(T',\emptyset)$, and from its solution we derive MB$_\mathbf{0}(T)$ using Lemma 5.26.

We remark that Lemma 5.25 produces an instance whose radii and center coordinates are polynomials from $\mathbb{R}[\epsilon]$. Therefore, algorithm $\mathcal{A}$ must (and will) be able to cope with such input.

The program. The program $\mathcal{D}(U,V)$ from above is defined for sets $V\subseteq U\subseteq T$ of input balls, and its variables are the numbers $x_B\in\mathbb{R}$, $B\in T$, and $\mu\in\mathbb{R}$. The symbol 'Q' denotes the $(d\times|T|)$-matrix holding the scaled centers $d_B=c_B/(c_B^Tc_B-\rho_B^2)$ from (5.6) as its columns, and the scalars $\sigma_B=\rho_B/(c_B^Tc_B-\rho_B^2)$ are the similarly scaled radii. For $\mu=0$ we define the value of objective function of $\mathcal{D}(U,V)$ to be ∞; clearly, the function is continuous in the interior of the feasibility domain.

As the following lemma shows, program $\mathcal{D}(T, \emptyset)$ is a *convex* mathematical program.

Lemma 6.1. *The objective function f of program $\mathcal{D}(U, V)$ is convex on the domain $\mathcal{F} := \{(x, \mu) \in \mathbb{R}^{n+1} \mid \mu > 0\}$. Moreover, f is strictly convex on $\mathcal{F}$ provided the matrix Q has full rank.*

Proof. We prove the claim for any f of the form $f(x, \mu) = x^T Q^t Q x/\mu + g(x, \mu)$, where g is a linear function; as convexity is invariant under scaling by a strictly positive constant, the claim then also holds for the objective function of program $\mathcal{D}(U, V)$.

To show convexity of f on $\mathcal{F}$, we need to verify that

$$f((1-\alpha)s + \alpha s') - ((1-\alpha)f(s) + \alpha f(s')) \leq 0 \tag{6.1}$$

holds for any $\alpha \in (0, 1)$ and any points $s = (x, \mu)$ and $s' = (x', \mu')$ from the domain $\mathcal{F}$. Denote the left-hand side of (6.1), multiplied by the number $\gamma := (1-\alpha)\mu + \alpha\mu' > 0$, by δ, and write $E := Q^T Q$. Using linearity of g, we obtain

$$\begin{aligned}
\delta &= ((1-\alpha)x + \alpha x')^T E((1-\alpha)x + \alpha x') - \\
&\quad \gamma\,((1-\alpha)/\mu x^T E x + \alpha/\mu' x'^T E x') \\
&= (1-\alpha)^2 x^T E x + 2\alpha\,(1-\alpha) x^T E x' + \alpha^2 x'^T E x' - \\
&\quad \gamma\,((1-\alpha)/\mu x^T E x + \alpha/\mu' x'^T E x') \\
&= (1-\alpha)((1-\alpha) - \gamma/\mu) x^T E x + \\
&\quad 2\alpha\,(1-\alpha) x^T E x' + \alpha(\alpha - \gamma/\mu') x'^T E x' \\
&= \alpha\,(\alpha - 1)\,(\mu'/\mu x^T E x - 2 x^T E x' + \mu/\mu' x'^T E x') \\
&= \alpha\,(\alpha - 1)\,\|\sqrt{\mu'/\mu} Q x - \sqrt{\mu/\mu'} Q x'\|^2 \leq 0.
\end{aligned} \tag{6.2}$$

This shows that f is convex. To see that f is strictly convex on $\mathcal{F}$ we verify that (6.2) is in fact fulfilled with strict inequality. Resorting to the assumption that E is invertible (recall linear independence of the columns of Q), equality in (6.2) yields $\sqrt{\mu'/\mu} x = \sqrt{\mu/\mu'} x'$. By multiplying this with $\mathbf{1}^T$ from the left, we finally arrive at

$$\sqrt{\mu'/\mu} = \mathbf{1}^T \sqrt{\mu'/\mu} x = \mathbf{1}^T \sqrt{\mu/\mu'} x' = \sqrt{\mu/\mu'}.$$

We conclude that $\mu' = \mu$ and hence $x' = x$. □

We first give KKT conditions for the program $\mathcal{D}(U,V)$ and then show that a minimizer to $\mathcal{D}(T,\emptyset)$ indeed provides us with $\mathrm{MB}_{\mathbf{0}}(T)$.

Lemma 6.2. *Let T be a set of balls, and let $V \subseteq U \subseteq T$ with U nonempty. Then a feasible point $(\tilde{x},\tilde{\mu}) \geq \mathbf{0}$ is optimal to $\mathcal{D}(U,V)$ iff $\tilde{\mu} > 0$ and there exists a real $\tilde{\alpha}$ such that for $\tilde{v} = Q_U\tilde{x}_U/(2\tilde{\mu})$ we have*

$$\begin{aligned} \tilde{v}^T d_B + \tilde{\alpha} &\geq \sigma_B, \quad B \in U \setminus V, \\ \tilde{v}^T d_B + \tilde{\alpha} &= \sigma_B, \quad B \in V, \\ \tilde{v}^T \tilde{v} &= 1, \\ \tilde{x}_B\,(\tilde{v}^T d_B + \tilde{\alpha} - \sigma_B) &= 0, \quad B \in U \setminus V. \end{aligned} \tag{6.3}$$

If these conditions are met, the objective value of $(\tilde{x},\tilde{\mu})$ equals $-\tilde{\alpha}$.

We note that the pair $(\tilde{v},\tilde{\alpha})$ from the lemma is *unique*. The vector $\tilde{v}$ is uniquely defined through $(\tilde{x},\tilde{\mu})$, and since the feasible point $(\tilde{x},\tilde{\mu})$ fulfills $\tilde{x}_D > 0$ for at least one $D \in U$ (recall the constraint $\sum_{B\in T}\tilde{x}_B = 1$), Eq. (6.3) implies $\tilde{v}^T d_D + \tilde{\alpha} - \sigma_D = 0$, which uniquely determines $\tilde{\alpha}$, too.

Proof. If $\tilde{\mu} = 0$ then $(\tilde{x},\tilde{\mu})$ cannot be optimal to $\mathcal{D}(U,V)$ because any solution with $\mu > 0$ has an objective value less than ∞ and is thus better. Hence, we can assume $\tilde{\mu} > 0$ for the rest of the proof.

As we have just seen, the objective function of program $\mathcal{D}(U,V)$ is convex over the domain $\mathcal{F} = \{(x,\mu) \in \mathbb{R}^{n+1}_+ \mid \mu > 0\}$, and thus we can invoke the Karush-Kuhn-Tucker Theorem for convex programming (Theorem 5.16). According to this, a feasible solution $(\tilde{x},\tilde{\mu}) \in \mathcal{F}$ is optimal to $\mathcal{D}(U,V)$ if and only if there exists a real number $\tilde{\alpha}$ and real numbers $\tilde{\delta}_B \geq 0$, $B \in U$, such that $\tilde{x}_B\tilde{\delta}_B = 0$ for all $B \in U$, $\tilde{\delta}_B = 0$ for all $B \in V$, and

$$\frac{1}{2\tilde{\mu}} d_B^T Q_U \tilde{x}_U - \sigma_B + \tilde{\alpha} - \tilde{\delta}_B = 0, \quad B \in U, \tag{6.4}$$

$$\frac{1}{4\tilde{\mu}^2} \tilde{x}_U^T Q_U^T Q_U \tilde{x}_U = 1, \tag{6.5}$$

which are precisely the conditions in the claim.

For the second claim of the lemma, we multiply (6.4) by $\tilde{x}_B$ and sum over all $B \in U$. Using $\sum_{B\in U}\tilde{x}_B = 1$, this gives

$$0 = \frac{1}{2\tilde{\mu}} \tilde{x}_U^T Q_U^T Q_U \tilde{x}_U - \sum_{B\in U} \tilde{x}_B\sigma_B + \tilde{\alpha} - \sum_{B\in U} \tilde{x}_B\tilde{\delta}_B,$$

where the last sum vanishes. Using $\tilde{\mu} = \tilde{x}_U^T Q_U^T Q_U \tilde{x}_U / (4\tilde{\mu})$ (which follows from (6.5)), we see that indeed the objective value of an optimal solution equals $-\tilde{\alpha}$. □

A transformation is now all it needs to obtain the center and radius of $\mathrm{MB}_{\mathbf{0}}(U, V)$ from an optimal solution of $\mathcal{D}(U, V)$.

Corollary 6.3. *Let T be a set of balls, centers linearly independent, and $V \subseteq U \subseteq T$ with U nonempty. If $(\tilde{x}, \tilde{\mu})$ is a minimizer to program $\mathcal{D}(U, V)$ with objective value $-\tilde{\alpha}$ then the ball $D = B(c, \rho)$ with*

$$c = -Q_U \tilde{x}_U / (4\tilde{\mu}\tilde{\alpha}), \qquad \rho = -1/(2\tilde{\alpha}),$$

lies in the set $\mathrm{MB}_{\mathbf{0}}(U, V)$.

Proof. The claim follows immediately from Theorem 5.18 by applying the transformation $x_B := \tilde{x}_B / (2\tilde{\mu})$, $B \in U$. □

6.2 Overview of the method

The method we are going to use to solve program $\mathcal{D}(T, \emptyset)$ (and hence the corresponding instance of $\mathrm{SEBB}_{\mathbf{0}}$) can be generalized to some extent to *other* convex programs. We therefore formulate it for the general mathematical program $\mathcal{P}(T, \emptyset)$ introduced below, and collect on the fly the properties of $\mathcal{P}(T, \emptyset)$ we require to hold.

So the remainder of this chapter is organized as follows. We first present the general setup of the method and state the abstract optimization problem which we actually solve. The following section then discusses how we realize the AOP's improving oracle. Afterwards, we will summarize the main result of this chapter (Theorem 6.16) and give some more applications.

The general setup. The mathematical program $\mathcal{P}(T, \emptyset)$ is defined as follows for an objective function of the form $f : \mathbb{R}^{n+m} \to \mathbb{R} \cup \{\infty\}$ and functions g_i from $\mathbb{R}^{n+m}$ to $\mathbb{R}$, with $n = |T|$ and $m \geq 0$.

$$\begin{array}{llrcll} \mathcal{P}(U,V) & \text{minimize} & f(x,y) & & & \\ & \text{subject to} & g_i(x,y) & = & 0, & i \in I_E, \\ & & g_i(x,y) & \leq & 0, & i \in I_I, \\ & & x_B & \geq & 0, & B \in U \setminus V, \\ & & x_B & = & 0, & B \in T \setminus U. \end{array} \qquad (6.6)$$

The variables of the program are the entries of the n-vector x and of the m-vector y. The two sets V and U, which must satisfy $V \subseteq U \subseteq T$, select which of the nonnegativity constraints $x_B \geq 0$, $B \in T$, are enabled and which variables x_B are set to zero. Each constraint $g_i \leqq 0$ may either be an equality (iff $i \in I_E$) or an inequality (iff $i \in I_I$), and the sets I_E, I_I index these constraints.

Given a vector $x \in \mathbb{R}^n$ and a subset U of T, we use 'x_U' to denote the vector of dimension $|U|$ whose Bth entry, $B \in T$, is x_B, $B \in U$. (In particular, 'x_B' denotes the Bth entry of vector x.)

Outline. Our strategy for solving $\mathcal{P}(T, \emptyset)$ is to rephrase the problem as an *abstract optimization problem* P (see Sec. 2.3), which we can then feed to Gärtner's algorithm from Theorem 2.10. Below, we will explain the AOP P in detail except that we do not yet state how we realize the AOP's improving oracle (which is the trickier part of the method). The latter will be the subject of Sec. 6.3.

The two main conditions we need to impose on program $\mathcal{P}(U, V)$ in order for our method to work are *uniqueness* of the optimal solution and the existence of a *violation test for* $\mathcal{P}(U, \emptyset)$.

Condition C1. *$\mathcal{P}(U, \emptyset)$ has a unique finite minimizer for all $U \subseteq T$.*

(Refer to page viii for definitions and notation in connection with mathematical programs.) As every global minimizer is a minimizer, the minimizer guaranteed by (C1) is the program's global minimizer.

To state the second condition we define $w_{\mathcal{P}}(U, V)$ to be the optimal solution of $\mathcal{P}(U, V)$; if the program is infeasible, we set $w_{\mathcal{P}}(U, V) := \bowtie$, if it is unbounded then $w_{\mathcal{P}}(U, V) := -\bowtie$. Also, we order the image of the function $w_{\mathcal{P}}$ by defining for $[U', V'], [U, V] \subseteq 2^T$ that

$$w_{\mathcal{P}}(U', V') \preceq w_{\mathcal{P}}(U, V) \quad \Leftrightarrow \quad f(w_{\mathcal{P}}(U', V')) \leq f(w_{\mathcal{P}}(U, V)),$$

where we set $f(\bowtie) := \infty$ and $f(-\bowtie) := -\infty$. Clearly, $\preceq$ is a quasiorder on $\operatorname{im}(w_{\mathcal{P}})$. We now impose the following kind of locality.

Condition C2. *$(T, \succeq, w_{\mathcal{P}}(\cdot, \emptyset))$ is an LP-type problem, and there is a routine* $\mathsf{violates}(B, V, \tilde{s})$ *that for the solution $\tilde{s}$ to $\mathcal{P}(V, \emptyset)$ returns 'yes' iff*

$$w_{\mathcal{P}}(V \cup \{B\}, \emptyset) \prec w_{\mathcal{P}}(V, \emptyset),$$

for $B \in T \supseteq V$. The routine is only called if $\tilde{s}$ minimizes $\mathcal{P}(V, V)$.

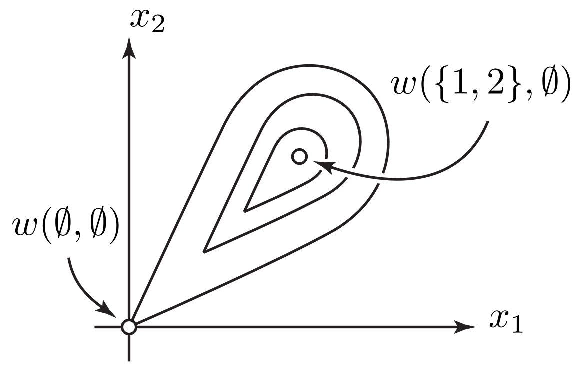


Figure 6.1. *The contour lines of a convex objective function $f : \mathbb{R}^2 \to \mathbb{R}$ for which program $\mathcal{P}(T, \emptyset)$ does not fulfill locality (C2).*

Observe that monotonicity and uniqueness hold, so the first part of the condition only asks for locality. That is, for all $U' \subseteq U \subseteq T$ and $B \in T$ with $w_{\mathcal{P}}(U', \emptyset) = w_{\mathcal{P}}(U, \emptyset)$, the condition $w_{\mathcal{P}}(U \cup \{B\}, \emptyset) \prec w_{\mathcal{P}}(U, \emptyset)$ implies $w_{\mathcal{P}}(U' \cup \{B\}, \emptyset) \prec w_{\mathcal{P}}(U', \emptyset)$. (If $w_{\mathcal{P}}(U', \emptyset) \preceq w_{\mathcal{P}}(U, \emptyset)$ holds for $U' \subseteq U$ then $w_{\mathcal{P}}(U', \emptyset)$ is not only a solution of the less restricted program $\mathcal{P}(U, \emptyset)$ but it attains also the latter's optimal objective value, so (C1) implies that both solutions coincide. This establishes uniqueness.)

For our main application, the solution of SEBB$_{\mathbf{0}}$ via program $\mathcal{D}(U, V)$, uniqueness (C1) follows from strict convexity (Lemma 6.1) and the fact that every optimal solution has a strictly positive value of μ. To verify (C2), we show that $(T, \leq, w_{\mathcal{D}}(\cdot, V))$ is an LP-type problem for any fixed $V \subseteq T$; the condition then follows by setting $V = \emptyset$. So suppose $w_{\mathcal{D}}(U', V) = w_{\mathcal{D}}(U, V)$ for $U' \subseteq U$ and $w_{\mathcal{D}}(U \cup \{B\}, V) \prec w_{\mathcal{D}}(U, V)$. Lemma 6.2 implies

$$\tilde{v}^T d_B + \tilde{\alpha} < \sigma_B, \qquad (6.7)$$

where $\tilde{v}$ and $\tilde{\alpha}$ are the *unique* numbers guaranteed by the lemma for the optimal solution $\tilde{s} := w_{\mathcal{D}}(U, V)$ of program $\mathcal{D}(U, V)$. Now if $w_{\mathcal{D}}(U' \cup \{B\}, V) = \tilde{s}$ then these multipliers must by Lemma 6.2 prove that $\tilde{s}$ is also optimal to $\mathcal{D}(U' \cup \{B\}, V)$, which (6.7) contradicts. Moreover, we can easily implement the routine violates for $\mathcal{D}(U, V)$: all we need to do is compute $\tilde{v}$ and $\tilde{\alpha}$ and check (6.7).

We remark that locality as demanded in (C2) need not hold in general. The program $\mathcal{P}(T, \emptyset)$ with $T = \{1, 2\}$ and $I_E = I_I = \emptyset$ and the function f from Fig. 6.1 as its objective fulfills $w_{\mathcal{P}}(\{1, 2\}, \emptyset) \prec w_{\mathcal{P}}(\emptyset, \emptyset)$. Yet, $w_{\mathcal{P}}(\{i\}, \emptyset)$ does not differ from $w_{\mathcal{P}}(\emptyset, \emptyset)$ for both $i \in \{1, 2\}$.

The AOP. We take T as the ground set of AOP P, and define a subset $F \subseteq T$ to be an AOP-basis of P if and only if it is a *basis* according to the following definition. For this, we say that a feasible point $\tilde{s} = (\tilde{x}, \tilde{y})$ of program $\mathcal{P}(U, V)$ is *proper* if all variables that are not forced to zero are strictly positive, i.e., if $\tilde{x}_B > 0$ for all $B \in U$ (equivalently, if $\tilde{x}_U > \mathbf{0}$). Similarly, we say that $\tilde{s}$ is *nonnegative* if all variables that are not forced to zero are nonnegative, i.e., $\tilde{x}_U \geq \mathbf{0}$. Notice that the notion of being proper (nonnegative, respectively) is *with respect to* some program; if we subsequently talk about 'the proper minimizer of $\mathcal{P}(U, V)$' (e.g., that the program has 'a proper minimizer') then we mean a minimizer that is proper w.r.t. $\mathcal{P}(U, V)$.

Definition 6.4. *A subset $F \subseteq T$ is called a* basis *if $\mathcal{P}(F, F)$ has a proper minimizer. We denote by $\tilde{s}^F = (\tilde{x}^F, \tilde{y}^F)$ the proper minimizer of a basis F and call it the* solution *of F.*

The most important property of a basis is that its (locally optimal) solution *globally* minimizes program $\mathcal{P}(F, \emptyset)$ (for $\tilde{s}^F$ is feasible to the more restricted problem $\mathcal{P}(F, \emptyset)$, which has only one minimizer). From this, it follows via (C1) that there exists at most one proper minimizer of program $\mathcal{P}(F, F)$, and hence $\tilde{s}^F$ is well-defined.

In view of the LP-type problem $(T, \succeq, w_{\mathcal{P}}(\cdot, \emptyset))$ from (C2), the name 'basis' is justified because a basis $F \subseteq T$ (in the sense of the above definition) is as a matter of fact also a basis in the LP-type sense: if there existed a proper subset F' of F such that the solution of $\mathcal{P}(F', \emptyset)$ equals $\tilde{s}^F$ then $(\tilde{x}^F)_B = 0$ for any $B \in F \setminus F'$, a contradiction.—In case of $\text{SEBB}_{\mathbf{0}}$, this observation amounts to the fact that bases $F \in T$ correspond to miniballs $\text{MB}_{\mathbf{0}}(F)$. We now have to define the order among these bases in such a way that the best bases correspond to $\text{MB}_{\mathbf{0}}(T)$.

The function $F \mapsto f(\tilde{s}^F)$ defines a quasiorder on the bases: for two bases $F', F \subseteq T$ we set $F' \preceq F$ if and only if $f(\tilde{s}^{F'}) \geq f(\tilde{s}^F)$. Solving AOP P then means, by definition of $\preceq$, to find a basis F with minimal objective value. That such a basis indeed provides us with the sought-for solution of problem $\mathcal{P}(T, \emptyset)$ is now an easy matter.

Lemma 6.5. *Let $F \subseteq G \subseteq T$ with F a basis. Then F is a largest basis in 2^G, i.e., $F \in \mathcal{B}(G)$, iff the solution of F minimizes $\mathcal{P}(G, \emptyset)$.*

By setting $G = T$ in the lemma, we see that the solution $\tilde{s}^F$ of some $\preceq$-largest basis $F \in \mathcal{B}$ minimizes program $\mathcal{P}(T, \emptyset)$ and thus by (C1) globally solves it.

Proof. Clearly, every solution coming from a basis in 2^G is feasible for $\mathcal{P}(G,\emptyset)$ (in fact, even feasible for $\mathcal{P}(T,\emptyset)$), with $\preceq$-larger bases yielding better objective values. Given this, all we have to do for direction ($\Rightarrow$) is to show that one of the bases in 2^G has a solution that is optimal to $\mathcal{P}(G,\emptyset)$. Let $\tilde{s} = (\tilde{x},\tilde{y})$ be feasible and optimal to the latter program; by (C1) such a minimizer exists. Set $F := \{B \in G \mid \tilde{x}_B > 0\}$. Since $\tilde{s}$ is a proper minimizer of $\mathcal{P}(F,F)$, the set F is a basis. Moreover, $\tilde{s}^F = \tilde{s}$ achieves the same objective value as the solution $\tilde{s}$ of $\mathcal{P}(G,\emptyset)$, so $\tilde{s}^F$ minimizes $\mathcal{P}(G,\emptyset)$.

For direction ($\Leftarrow$), assume that $F \in \mathcal{B}$ has a solution $\tilde{s}^F$ which optimally solves $\mathcal{P}(G,\emptyset)$. We know that any basis $F' \subseteq G$ has a feasible solution for $\mathcal{P}(G,\emptyset)$ and that $\preceq$-larger bases yield better objective values. So if F were not a $\preceq$-largest basis, a larger one, $F' \succ F$, say, would give a better solution $\tilde{s}^{F'}$ to $\mathcal{P}(G,\emptyset)$, contradiction. □

6.3 The oracle

Gärtner's algorithm requires a realization of the oracle Φ from the previous section in order to work. We are now going to show how this can be done. As input we receive $G \subseteq T$ and a basis $F \in \mathcal{B}$, $F \subseteq G$. The goal is to either assert that $F \in \mathcal{B}(G)$, or to compute a basis $F' \subseteq G$ with $F' \succ F$ otherwise. The algorithm we use for this task is sketched in Fig. 6.2; it works on the program $\mathcal{P}^{\epsilon}_D(F)$ introduced below and is similar in spirit to Gärtner & Schönherr's method [72] for solving convex quadratic programs. We first present the idea behind the algorithm and then proceed with the correctness proof.

The core of the oracle. As a first step, procedure update checks whether F is a $\preceq$-largest basis in G, which by Lemma 6.5 is equivalent to the solution of F minimizing program $\mathcal{P}(G,\emptyset)$. Resorting to (C2), this in turn holds if and only if all violation tests violates$(B,F,\tilde{s}^F)$, $B \in G \setminus F$, return 'no.' This being the case we have $\Phi(G,F) = F$ and are done (first if-clause in procedure update). Otherwise, we select any *violator* $D \in G \setminus F$ (i.e., an element D for which the test returns 'yes'); D remains fixed for the rest of the oracle call.

We observe now from locality (C2) that the solution of F does not minimize $\mathcal{P}(F',\emptyset)$, for $F' := F \cup \{D\}$. Thus, F' contains as a subset at least one basis which is $\preceq$-better than F (this is Lemma 6.5 again).

```
procedure update(G, F)
{ Implements the oracle Φ(G, F) of AOP P }
begin
  if violates(B, F, s̃^F) = no for all B ∈ G \ F then
    return F
  let D ∈ G \ F with violates(B, F, s̃^F) = yes
  (ε̃,s̃,finished):= (0,s̃^F,false)
  while not finished do
    { here, Invariant 6.6 holds }
    (ε̃_new, s̃):= prim(F,D,ε̃,s̃)
    { write s̃ as s̃ = (x̃, ỹ) }
    E:= {B ∈ F | x̃_B = 0}
    if E = ∅ then
      F:= F ∪ {D}
      finished:= true
    else
      F:= F \ {x̃_B = 0 | B ∈ F}
    ε̃:= ε̃_new
  return F
end update
```

Figure 6.2. *The oracle $\Phi(G, F)$ for the abstract optimization problem P: given a basis $F \subseteq G$ it returns F itself iff F is the $\preceq$-largest basis in G, whereas otherwise it computes a basis $F_* \subseteq G$ with $F_* \succ F$.*

Our strategy is to find one of these *improved bases*. The key observation for this to succeed is that the solution of any improved basis $F_* \subseteq F'$ has $(x^{F_*})_D > 0$, while the current basis F has $(x^F)_D = 0$; thus, the idea is to (conceptually) increase x_D continuously from 0 on, which will eventually lead us to a new basis.

To prove that every improved basis $F_* \subseteq F'$ has $(x^{F_*})_D > 0$, it suffices to show that $D \in F_*$: the basis F globally minimizes $\mathcal{P}(F, \emptyset)$ and hence is a best basis in 2^F by Lemma 6.5; so if we had $F_* \subseteq F$ then F_* cannot be a better basis than F, a contradiction.

Thus, we set the variable x_D to ϵ and (conceptually) increase ϵ from 0 on. Since our goal is to eventually reach the solution of an improved basis $F_* \subseteq F'$—which is a *best solution* of program $\mathcal{P}(F_*, F_*)$—we will

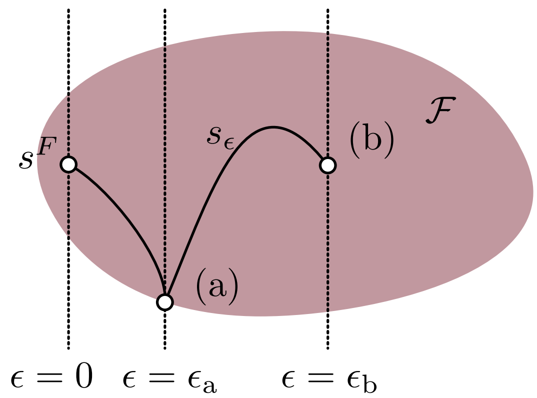


Figure 6.3. *By continuously increasing ϵ, procedure* update(G, F) *traces the proper minimizer s_ϵ of program $\mathcal{P}_D^\epsilon(F)$ until it either becomes non-proper (event (a)) or reaches the minimum of $\mathcal{P}(F \cup \{D\}, \emptyset)$.*

maintain for any value of ϵ we encounter the *best* among the solutions of program $\mathcal{P}(F', F')$ that satisfy $x_D = \epsilon$. That is, we keep track of the minimizer of program

$$\begin{array}{llllll} \mathcal{P}_D^\epsilon(F) & \text{minimize} & f(x,y) & & & \\ & \text{subject to} & g_i(x,y) & = & 0, & i \in I_E, \\ & & g_i(x,y) & \leq & 0, & i \in I_I, \\ & & x_B & = & 0, & B \in T \setminus F', \\ & & x_D & = & \epsilon. & \end{array} \tag{6.8}$$

We consider $\mathcal{P}_D^\epsilon(F)$ to be a program in the variables x_F (and not in $x_{F'}$, because $x_D = \epsilon$ is fixed); any feasible point $(\tilde{x}, \tilde{y})$ fulfills $\tilde{x}_D = \epsilon$ and is called *proper* if $\tilde{x}_F > 0$. At the moment it is not clear that $\mathcal{P}_D^\epsilon(F)$ indeed has a minimizer for every ϵ we encounter. We will have to confirm this later, see Lemma 6.12.

More precisely, procedure update maintains the following invariant from the beginning of the while-loop on.

Invariant 6.6. *In each iteration of the loop of procedure* update*, the procedure's variable $\tilde{s} = (\tilde{x}, \tilde{y})$ is a proper minimizer of $\mathcal{P}_D^{\tilde{\epsilon}}(F)$.*

Observe here that for $\tilde{\epsilon} = 0$, program $\mathcal{P}_D^{\tilde{\epsilon}}(F)$ coincides with program $\mathcal{P}(F, F)$; so when update enters the while-loop for the first time, $\tilde{s} = \tilde{s}^F$ indeed minimizes program $\mathcal{P}_D^0(F)$. So initially, the invariant holds.

Let $\tilde{s}' = (\tilde{x}', \tilde{y}')$ be the global minimizer of $\mathcal{P}(F', \emptyset)$, $F' := F \cup \{D\}$, where F is the current iteration's set 'F' from the procedure. As we prove below (Lemma 6.12), only one of two kinds of events may take place while we increase $\tilde{\epsilon}$; this is illustrated in Fig. 6.3 where the filled area represents the feasibility region of $\mathcal{P}(F',\emptyset)$.

(a) $\tilde{x}_B = 0$ for some $B \in F$: In this case, $\tilde{s}$ must be a minimizer of the more restricted program $\mathcal{P}_D^{\tilde{\epsilon}}(F^*)$ for $F^* := F \setminus \{B \mid \tilde{x}_B = 0\}$. Thus, we can set $F := F^*$, and the invariant is fulfilled again.

(b) $\tilde{x}_D = \tilde{x}'_D$: Here, we must have $\tilde{s} = \tilde{s}'$ because on the one hand, $\tilde{s}$ is feasible for $\mathcal{P}(F',\emptyset)$ and thus $f(\tilde{s}) \geq f(\tilde{s}')$, while on the other hand $\tilde{s}'$ is feasible to $\mathcal{P}_D^{\tilde{\epsilon}}(F)$ by $\tilde{x}_D = \tilde{x}'_D$, implying $f(\tilde{s}') \geq f(\tilde{s})$; so $\tilde{s} = \tilde{s}'$ by uniqueness (C1) of the minimizer of $\mathcal{P}(F',\emptyset)$. Consequently, $F' = F \cup \{D\}$ is a new basis.

In order to find out which case takes place, we require a *subroutine*, prim, to be available that decides which event happens first and returns its 'time' $\tilde{\epsilon}'$.

Condition C3. *There is an algorithm* prim$(F, D, \tilde{\epsilon}, \tilde{s}_{\tilde{\epsilon}})$ *which for a given proper minimizer* $\tilde{s}_{\tilde{\epsilon}}$ *to* $\mathcal{P}_D^{\tilde{\epsilon}}(F)$ *returns the pair* $(\tilde{\epsilon}', \tilde{s}_{\tilde{\epsilon}'})$*, where* $\tilde{\epsilon}' \geq \tilde{\epsilon}$ *is the time when the first of the following events* (a_B)*,* $B \in F$*, or (b) occurs if* ϵ *continuously increases from* $\tilde{\epsilon}$ *on.*

(a_B) $(\tilde{x}_\epsilon)_B = 0$ *for* $B \in F$*.*

(b) $\tilde{s}_\epsilon$ *is optimal to* $\mathcal{P}(F',\emptyset)$ *for* $F' := F \cup \{D\}$*.*

Here, $\tilde{s}_\epsilon = (\tilde{x}_\epsilon, \tilde{y}_\epsilon)$ *denotes the nonnegative minimizer of program* $\mathcal{P}_D^{\epsilon}(F)$*. Whenever procedure* prim *is called, it is guaranteed that* $\tilde{s}_\epsilon$ *exists and is finite on the interval* $[\tilde{\epsilon}, \tilde{\epsilon}']$*, and that* $\tilde{\epsilon}' < \infty$*.*

We emphasize that the procedure prim does not have to 'ensure' that s_ϵ exists. The routine will *only be called* if s_ϵ exists and is finite on the interval $[\tilde{\epsilon}, \tilde{\epsilon}']$ and $\tilde{\epsilon}' < \infty$. Notice also that the caller of the routine can detect which event took place; if the set $E := \{(\tilde{x}_{\tilde{\epsilon}'})_B = 0 \mid B \in F\}$ is empty then event (b) occurred, otherwise one or more of the events (a_B) stopped the motion.

From this condition it follows that the primitive prim decides which case takes place, and thus the invariant is fulfilled again after handling the respective case as described above. It remains to prove that (i) one of the above events always occurs, (ii) that $\mathcal{P}_D^{\epsilon}(F)$ has a minimizer for all values of ϵ we encounter, and (iii) that in case (b), F' is indeed a $\preceq$-better basis than the original basis F we started with. All these statements will be shown in the next section (see Lemmata 6.12 and 6.11), and we conclude from this that procedure update is correct. Termination follows from the fact that in each iteration (except the last one) the cardinality of F drops by one, yielding

Lemma 6.7. *The oracle* update(G, F) *of the abstract optimization problem P terminates after at most $|F|$ iterations of its* while*-loop.*

6.3.1 The primitive for $\mathrm{SEBB}_{\mathbf{0}}$

We close this section with the argument that shows how the primitive routine from condition (C3) can be realized for program $\mathcal{D}(T, \emptyset)$ (again, assuming that the centers of the input balls are linearly independent).

Lemma 6.8. *Under linear independence, the primitive* prim$(F, D, \tilde{\epsilon}, s_{\tilde{\epsilon}})$ *for program $\mathcal{D}(U, V)$ can be realized in time $\mathcal{O}(d^3)$.*

To show this, we derive Karush-Kuhn-Tucker optimality conditions for program $\mathcal{D}_D^{\epsilon}(F)$, the counterpart to $\mathcal{P}_D^{\epsilon}(F)$ for our program $\mathcal{D}(U, V)$. This program looks as follows, where F is some subset of the input balls T, and where $D \notin F$ is the iteration's violator; for convenience, we set $F' := F \cup \{D\}$ for the rest of this section.

$$
\begin{array}{lll}
\mathcal{D}_D^{\epsilon}(F) & \text{minimize} & \frac{1}{4\mu} x^T Q^T Q x + \mu - \sum_{B \in T} x_B \sigma_B \\
 & \text{subject to} & \sum_{B \in T} x_B = 1, \\
 & & x_B = 0, \qquad\qquad B \in T \setminus F', \\
 & & x_D = \epsilon, \\
 & & \mu \geq 0.
\end{array}
$$

As we will see, feasibility and optimality to $\mathcal{D}_D^{\epsilon}(F)$ of a point $s = (x, \mu)$ can be decided by plugging s into a linear system and one additional quadratic equation. The coefficient matrix of the linear system will turn out to be the following matrix M_F.

$$
M_F := \begin{pmatrix} -I_d & \mathbf{0} & Q_F \\ \mathbf{0}^T & 0 & \mathbf{1}^T \\ Q_F^T & \mathbf{1} & \mathbf{0} \end{pmatrix}
$$

(Here, 'Q_F' denotes the submatrix of Q consisting of the columns of Q that correspond to the balls $B \in F$.) This matrix is quadratic, symmetric, and nonsingular by the linear independence of the columns of Q_F:

Lemma 6.9. *Let T be a set of balls with linearly independent centers. Then M_F is regular for all $F \subseteq T$.*

Proof. Consider any vanishing linear combination, $M_F\,(u, \omega, v) = 0$, of the columns of M_F; we show that (u, ω, v) equals zero. From the definition of M_F we obtain

$$u = Q_F v, \qquad Q_F^T u = -\omega \mathbf{1}, \qquad \mathbf{1}^T v = 0. \tag{6.9}$$

It follows that $Q_F^T Q_F v = -\omega\mathbf{1}$. As Q_F has linearly independent columns, the matrix $Q_F^T Q_F$ is positive definite, in particular regular, and hence $v = -(Q_F^T Q_F)^{-1}\omega\mathbf{1}$. Using the identity $\mathbf{1}^T v = 0$ from (6.9) we can derive $0 = \omega\mathbf{1}^T v = -\omega\mathbf{1}^T (Q_F^T Q_F)^{-1}\omega\mathbf{1}$, and since the inverse of a positive definite matrix is positive definite again, we must have $\omega\mathbf{1} = \mathbf{0}$. It follows $\omega = 0$, and from $Q_F^T Q_F v = -\omega\mathbf{1} = \mathbf{0}$ we obtain $v = 0$. Hence, also u must be zero by (6.9). □

Letting b_F denote the vector containing the numbers σ_B, $B \in F$, we get the following characterization of feasibility and optimality for $\mathcal{D}_D^\epsilon(F)$.

Lemma 6.10. *A pair $(\tilde{x}, \tilde{\mu})$ is feasible and optimal to $\mathcal{D}_D^\epsilon(F)$ iff $\tilde{\mu} > 0$ and there exists a real vector w and a real number β such that*

$$M_F \begin{pmatrix} w \\ \beta \\ \tilde{x}_F \end{pmatrix} = \begin{pmatrix} -\epsilon d_D \\ 1-\epsilon \\ 2\tilde{\mu} b_F \end{pmatrix} \tag{6.10}$$

and $w^T w = 4\tilde{\mu}^2$ hold.

The argument is almost identical to the one employed in the proof of Lemma 6.2. We give the full proof for the sake of completeness.

Proof. Let $f(x, \mu)$ be the objective function of program $\mathcal{D}_D^\epsilon(F)$, and write $g(x, \mu) = \sum_{B \in F} x_B - (1-\epsilon)$ for the program's first constraint and $h(x, \mu) = -\mu$ for the program's second constraint. We have

$$f(x, \mu) = \frac{x_F^T Q_F^T (Q_F x_F + 2\epsilon d_D) + \epsilon^2 d_D^T d_D}{4\mu} + \mu - \sum_{B \in F} x_B \sigma_B - \epsilon \sigma_D,$$

where the last summand is a constant (i.e., depending neither on x_F nor on μ). Consequently,

$$\nabla f = \begin{pmatrix} \frac{1}{2\mu} Q_F^T (Q_F x_F + \epsilon d_D) - b_F \\ -\frac{1}{4\mu^2} x_{F'}^T Q_{F'}^T Q_{F'} x_{F'} + 1 \end{pmatrix},$$

$\nabla g = (\mathbf{1}, 0)$, and $\nabla h = (\mathbf{0}, -1)$. The Karush-Kuhn-Tucker Theorem for convex programming (Theorem 5.16) tells us that a feasible $(\tilde{x}, \tilde{\mu})$ is optimal if and only if there exist a real number α and a real positive number γ such that $\tilde{\mu}\gamma = 0$ and $\nabla f + \alpha \nabla g + \gamma \nabla h = \mathbf{0}$ holds for $(\tilde{x}, \tilde{\mu})$ plugged in. As $\tilde{\mu} > 0$, the latter criterion simplifies to the existence of a real number α with $\nabla f + \alpha \nabla g = \mathbf{0}$. Setting $w = Q_{F'} \tilde{x}_{F'}$, this condition reads

$$\begin{aligned} \frac{1}{2\tilde{\mu}} d_B^T w - \sigma_B + \alpha &= 0, \qquad B \in F, \\ \frac{1}{4\tilde{\mu}^2} w^T w = 1. \end{aligned}$$

Multiplying the former equation by $2\tilde{\mu} \neq 0$ and setting $\beta := 2\tilde{\mu}\alpha$, the upper and lower rows of equation (6.10) follow, and feasibility provides the remaining middle row. Conversely, if the conditions of the lemma hold then the point $(\tilde{x}, \tilde{\mu})$ is feasible. Moreover, the above two equations are fulfilled with $\alpha := \beta/(2\tilde{\mu})$, and the Karush-Kuhn-Tucker conditions prove $(\tilde{x}, \tilde{\mu})$ to be optimal to program $\mathcal{D}_D^\epsilon(F)$. □

Proof of Lemma 6.8. We can assume (see statement of (C3)) that there exists a bounded interval $I =: [\tilde{\epsilon}, \tilde{\epsilon}']$ such that for every $\epsilon \in I$ program $\mathcal{D}_D^\epsilon(F)$ has exactly one minimizer. With the preceding lemma, this means that for any such ϵ there exist two real numbers $\tilde{\mu}_\epsilon > 0$ and β_ϵ and a real vector u_ϵ such that (6.10) holds for these quantities, and $u_\epsilon^T u_\epsilon = 4\tilde{\mu}_\epsilon^2$. By multiplying (6.10) from the left by M_F^{-1}, we obtain the triple $(u_\epsilon, \beta_\epsilon, \tilde{x}_\epsilon)$, whose entries are linear polynomials from $\mathbb{R}[\epsilon, \tilde{\mu}_\epsilon]$. We can now solve $u_\epsilon^T u_\epsilon = 4\tilde{\mu}_\epsilon^2$ for $\tilde{\mu}_\epsilon$, yielding a polynomial from $\mathbb{R}[\epsilon]$, and plug this into the three polynomials $u_\epsilon, \beta_\epsilon, \tilde{x}_\epsilon$. This gives a formula for the triple $(u_\epsilon, \beta_\epsilon, \tilde{x}_\epsilon)$ that depends only on ϵ and has the property that $(\tilde{x}_\epsilon, \tilde{\mu}_\epsilon)$ minimizes $\mathcal{D}_D^\epsilon(F)$ provided $\epsilon \in I$.

In order to check for event (b), we use Lemma 6.2. Assuming $(\tilde{x}_\epsilon)_F > \mathbf{0}$ and $\tilde{\mu}_\epsilon > 0$, this states that $(\tilde{x}_\epsilon, \tilde{\mu}_\epsilon)$ is optimal to program $\mathcal{D}(F', \emptyset)$

if and only if there exists a real number α_ϵ such that $v_\epsilon^T v_\epsilon = 1$ and $v_\epsilon^T d_B + \alpha_\epsilon = \sigma_B$ for all $B \in F \cup \{D\}$, where

$$v_\epsilon = (Q_F(\tilde{x}_\epsilon)_F + \epsilon d_D)/(2\tilde{\mu}_\epsilon).$$

As $u = Q_F(\tilde{x}_\epsilon)_F + \epsilon d_D$ (evaluate (6.10) to see this), $v_\epsilon^T v_\epsilon = 1$ follows already from $u_\epsilon^T u_\epsilon = 4\tilde{\mu}_\epsilon^2$. Likewise, (6.10) implies $v_\epsilon^T d_B + \alpha_\epsilon = \sigma_B$, $B \in F$, if we set $\alpha_\epsilon := \beta_\epsilon/(2\tilde{\mu}_\epsilon)$. Thus, it suffices to solve $v_\epsilon^T d_B + \alpha_\epsilon = \sigma_B$ for ϵ, memorizing the smallest real value $\epsilon_{(\mathrm{b})}$ for which it holds.

On the other hand, an event (a_B), $B \in F$, takes place if and only if $(\tilde{x}_\epsilon)_B = 0$. Thus, we compute the smallest real solution of this equation and denote it by $\epsilon^{(\mathrm{a})}(B)$ for $B \in F$. (Observe that for both types of events, the equations we need to solve involve algebraic numbers of degree 2 at most and can thus be solved easily.)

Finally, we return the triple $(E, \epsilon^*, (\tilde{x}_{\epsilon^*}, \mu_{\epsilon^*}))$, where

$$\epsilon^* := \min\{\epsilon^{(\mathrm{b})}\} \cup \{\epsilon^{(\mathrm{a})}(B) \mid B \in F\},$$

and where $E = \{B \in F \mid \epsilon^{(\mathrm{a})}(B) = \epsilon^*\}$. We claim that this realizes the primitive: if $E \neq \emptyset$, the above argument for $\epsilon^{(\mathrm{a})}(B)$ shows that one or more events of type (a) occur first. If $E = \emptyset$ then $\epsilon^* = \epsilon^{(\mathrm{b})}$ by construction, so all events of type (a) occur strictly after time ϵ^*. This implies $(\tilde{x}_{\epsilon^*})_F > \mathbf{0}$ and $\tilde{\mu}_{\epsilon^*} > 0$, and as we have shown above, $\epsilon^{(\mathrm{b})}$ coincides under these conditions with the time when event (b) takes place. Finally, the costs of an invocation of the primitive are dominated by the computation of the inverse of the matrix M_F. Since this can be done in $\mathcal{O}(d^3)$, the lemma follows. □

We remark that we can take any singleton $F_* := \{B\} \subseteq T$ as an initial basis. By the constraint $\sum x_B = 1$ of program $\mathcal{D}(F_*, F_*)$ we must have $x_B^{F_*} > 0$, implying that F_* is indeed a basis.

6.4 The details

In this part we prove the leftovers from the previous section. To this end, we need to introduce some more requirements on program $\mathcal{P}(U, V)$; these are mostly technical and easily met for SEBB$_\mathbf{0}$.

Condition C4. *The objective function f is convex over the subset of the feasibility region of $\mathcal{P}(T, \emptyset)$ where f is finite, and it is continuous over the subset of the feasibility region of $\mathcal{P}(T, T)$ where f is finite.*

It follows that for any $U \subseteq T$, the function f is convex over the subset of the feasibility region of $\mathcal{P}(U, \emptyset)$ where f attains finite value.—In case of $\text{SEBB}_{\mathbf{0}}$, (C4), as well as the following condition, is satisfied because we are dealing with a convex program.

Condition C5. *The feasibility region of $\mathcal{P}(T, T)$ is convex.*

Then also the feasibility region of $\mathcal{P}(F, \emptyset)$ and $\mathcal{P}(F, F)$ is convex for all $F \subseteq T$.

Condition C6. *For any $F \subset T$, $D \in T \backslash F$, and any real $\epsilon \geq 0$, program $\mathcal{P}_D^\epsilon(F)$ has* at most *one minimizer that is both finite and nonnegative.*

For $\text{SEBB}_{\mathbf{0}}$, the condition follow from the strict convexity of program $\mathcal{D}(T, \emptyset)$ (see Lemma 6.1, observing that any minimizer has $\mu > 0$).

Condition C7. *The intersection of any compact set with the feasibility region of $\mathcal{P}(T, T)$ is compact again.*

Condition (C7) applies, for instance, to any program whose feasibility region is closed, as is the case with program $\mathcal{D}(T, T)$.

These are the assumptions we need for the method to work; using them we can now prove the remaining statements that show its correctness. We start by observing that a proper minimizer $\tilde{s}$ of $\mathcal{P}_D^\epsilon(F)$ with finite objective value is necessarily strict: if it were not, there existed for any $\delta > 0$ a feasible point $\tilde{s}_\delta \in \dot{U}_\delta(\tilde{s})$ with $f(\tilde{s}_\delta) = f(\tilde{s})$; by choosing δ small enough the point $\tilde{s}_\delta$ is also proper, meaning that the optimal solution is not unique, a contradiction to (C6).

Given this, we are ready to show that *if* we can increase the current ϵ to some larger value, $\overline{\epsilon}$, say, such that $\mathcal{P}_D^{\overline{\epsilon}}(F)$ has again a finite proper minimizer then this minimizer is better than the previous one.

Lemma 6.11. *Let $F' = F \cup \{D\} \subseteq T$, $D \notin F$, let $\tilde{s}' = (\tilde{x}', \tilde{y}')$ be the minimizer to $\mathcal{P}(F', \emptyset)$ and suppose $\tilde{s}_{\text{old}} = (\tilde{x}_{\text{old}}, \tilde{y}_{\text{old}})$ is a finite minimizer to $\mathcal{P}_D^\epsilon(F)$ with*

$$(\tilde{x}_{\text{old}})_F \geq \mathbf{0}. \tag{6.11}$$

If $0 \leq \underline{\epsilon} < \overline{\epsilon} \leq \tilde{x}'_D$ then a finite minimizer $\tilde{s}_{\text{new}} = (\tilde{x}_{\text{new}}, \tilde{y}_{\text{new}})$ of program $\mathcal{P}_D^{\overline{\epsilon}}(F)$ with

$$(\tilde{x}_{\text{new}})_F \geq \mathbf{0} \tag{6.12}$$

has a better objective function value than $\tilde{s}_{\text{old}}$.

(As a side remark, this proof of lemma is the only place where convexity of f is required; for all other statements, quasi-convexity suffices.)

Proof. By (6.11) and (6.12), all involved points, i.e., $\tilde{s}'$, $\tilde{s}_{\mathrm{old}}$, and $\tilde{s}_{\mathrm{new}}$, are feasible to program $\mathcal{P}(F', \emptyset)$. As $\tilde{s}'$ is the best among the feasible solutions of $\mathcal{P}(F', \emptyset)$, we clearly have

$$f(\tilde{s}') \leq f(\tilde{s}_{\mathrm{old}}). \tag{6.13}$$

Moreover, since $\underline{\epsilon} = (\tilde{x}_{\mathrm{old}})_D < (\tilde{x}_{\mathrm{new}})_D = \overline{\epsilon} \leq \tilde{x}'_D$, there exists a convex combination $\tilde{s} = (\tilde{x}, \tilde{y})$ of $\tilde{s}_{\mathrm{old}}$ and $\tilde{s}'$ with $\tilde{x}_D = \overline{\epsilon}$. By convexity (C5), $\tilde{s}$ is feasible to $\mathcal{P}(F', \emptyset)$, and by convexity (C4) and (6.13), $f(\tilde{s}) \leq f(\tilde{s}_{\mathrm{old}})$. Also, (C6) and convexity (C4) ensures that $f(\tilde{s}_{\mathrm{new}}) \leq f(\tilde{s})$, and thus

$$f(\tilde{s}_{\mathrm{new}}) \leq f(\tilde{s}) \leq f(\tilde{s}_{\mathrm{old}}).$$

Now if $f(\tilde{s}_{\mathrm{new}}) = f(\tilde{s}_{\mathrm{old}})$ then in particular $f(\tilde{s}_{\mathrm{old}}) = f(\tilde{s})$, which via convexity (C4) yields $f(\tilde{s}') = f(\tilde{s}_{\mathrm{old}})$ with $\tilde{s}' \neq \tilde{s}_{\mathrm{old}}$. This, however, is impossible because $\mathcal{P}(F', \emptyset)$ has only a single optimal solution. We conclude $f(\tilde{s}_{\mathrm{new}}) < f(\tilde{s}_{\mathrm{old}})$. □

We now turn to proving that whenever the procedure prim from (C3) is invoked, the respective program $\mathcal{P}_D^{\epsilon}(F)$ has a nonnegative minimizer up to the point when the first of the events (a_B), $B \in F$, or (b) occurs.

Lemma 6.12. *Let $F' = F \cup \{D\} \subseteq T$, $D \notin F$, and $\tilde{s}' = (\tilde{x}', \tilde{y}')$ be the minimizer of $\mathcal{P}(F', \emptyset)$. Suppose $\mathcal{P}_D^{\underline{\epsilon}}(F)$ has a proper minimizer for some $0 \leq \underline{\epsilon} \leq \tilde{x}'_D$, and set*

$$\overline{\epsilon} = \sup\{\epsilon' \in [\underline{\epsilon}, \tilde{x}'_D] \mid \mathcal{P}_D^{\epsilon}(F) \textit{ has a proper minimizer } \forall \epsilon \in [\underline{\epsilon}, \epsilon']\}.$$

Then $\overline{\epsilon} = \tilde{x}'_D$ or $\mathcal{P}_D^{\overline{\epsilon}}(F)$ has a nonnegative, nonproper minimizer.

If $\mathcal{P}_D^{\overline{\epsilon}}(F)$ has a nonnegative minimizer that is not proper then one or several of the events (a_B), $B \in F$, occur. Otherwise the lemma ensures $\overline{\epsilon} = \tilde{x}'_D$ in which case the minimizer of program $\mathcal{P}_D^{\overline{\epsilon}}(F)$ is proper and event (b) takes place. In total, the lemma implies that algorithm $\mathsf{update}(G, F)$ from Fig. 6.2 is correct.

In order to prove the lemma, we proceed in three steps. In all of them, we denote the proper minimizer of program $\mathcal{P}_D^{\epsilon}(F)$ for $\epsilon \in [\underline{\epsilon}, \overline{\epsilon})$ (which exists by definition of $\overline{\epsilon}$) by $\tilde{s}_\epsilon$ and write '$\mathcal{F}^\epsilon$' for the feasibility

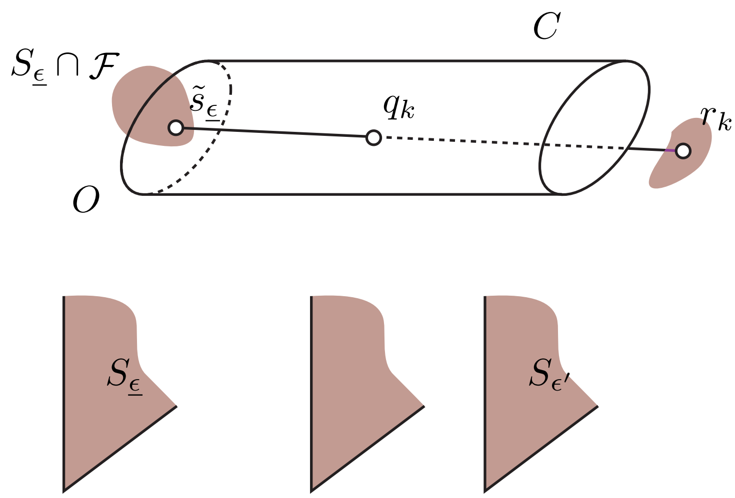


Figure 6.4. *The cylinder C from the proof of Lemma 6.13.*

region of $\mathcal{P}^{\epsilon}_{D}(F)$ and '$\mathcal{F}$' for the feasibility region of program $\mathcal{P}(F', \emptyset)$. Also, we introduce the set

$$S_\epsilon := \{(x, y) \in \mathbb{R}^{n+m} \mid x_D = \epsilon,\ x \geq \mathbf{0}\};$$

every feasible point (x, y) of program $\mathcal{P}(F', \emptyset)$ with $x_D = \epsilon$ is contained in S_ϵ, and every nonnegative feasible point of $\mathcal{P}^{\epsilon}_{D}(F)$ is a member of S_ϵ.

Lemma 6.13. *In the context of the previous lemma, with $\epsilon' \in [\underline{\epsilon}, \overline{\epsilon}]$, there cannot exist a sequence r_k, $k \in \mathbb{N}$, of points in $\mathcal{F} \cap S_{\epsilon'}$ with*

$$f(r_k) \leq \phi := f(\tilde{s}_{\underline{\epsilon}})$$

and $\|r_k\| \geq k$.

Proof. Suppose for a contradiction that there is a sequence $r_k \in \mathcal{F} \cap S_{\epsilon'}$, $k \in \mathbb{N}$, as excluded by the lemma. Consider the boundary C of the cylinder of radius $\beta > 0$ (to be defined later) and axis $\{\tilde{s}_{\underline{\epsilon}} + \gamma e_D \mid \gamma \in \mathbb{R}\}$, with e_D denoting the Dth unit vector, see Fig. 6.4. The set

$$O := C \cap S_{\underline{\epsilon}}$$

is compact and thus $O \cap \mathcal{F}$ is compact, too, by (C7). Below, we show that (for every value of $\beta > 0$) there exists a point $q^* \in O \cap \mathcal{F}$ with $f(q^*) \leq \phi$.

However, as $\tilde{s}_{\underline{\epsilon}}$ is in fact a strict minimizer (recall the discussion before Lemma 6.11 for this), we can choose the number $\beta > 0$ such that $f(t) > \phi$ for all $t \in O \cap \mathcal{F}$, a contradiction.

In order to construct the point q^*, we consider a sequence q_k, $k \in \mathbb{N}$, of points in $\mathbb{R}^{n+m}$ that for large enough k lie in the compact set

$$X := C \cap \bigcup_{\epsilon \in [\underline{\epsilon}, \epsilon']} S_\epsilon.$$

Namely, we take as q_k the intersection point of the cylinder boundary C with the line segment L_k between $\tilde{s}_{\underline{\epsilon}}$ and r_k; from the unboundedness of the sequence $z_k := \|r_k\|$, $k \in \mathbb{N}$, it easily follows that for any k larger than some $k^* \in \mathbb{N}$, this intersection point exists (and thus q_k, $k > k^*$, is well-defined). From convexity (C5) and convexity (C4) we obtain $q_k \in \mathcal{F}$ and $f(q_k) \leq \phi$. Moreover, using the unboundedness of the sequence z_k again, we can see that to any given $\delta' > 0$ there exists a $k' > k^*$ such that the points q_k, $k > k'$, have distance at most δ' to $X \cap S_{\underline{\epsilon}} = O$.

Now consider the limit point q^* of some convergent subsequence of the sequence q_k; such a subsequence and limit point exist by compactness of $X \subset \mathbb{R}^{n+m}$. Since the points q_k approach O arbitrarily close, we have $q^* \in O$. In fact, $q^* \in O \cap \mathcal{F}$, which one can see as follows. By (C7) the set $O \cap \mathcal{F}$ is compact, and so if $q^* \notin O \cap \mathcal{F}$, there exists a neighborhood of q^* that does not intersect $O \cap \mathcal{F}$; this together with the fact that the points $q_k \in \mathcal{F}$ approach q^* to any arbitrarily small distance yields a contradiction to q^* being the limit point of the subsequence of the q_k. Thus, the $q_k \in \mathcal{F}$ converge to the point $q^* \in O \cap \mathcal{F}$. Finally, $f(q_k) \leq \phi$, $k > k^*$, and continuity (C4) of f imply $f(q^*) \leq \phi$ as needed. □

Lemma 6.14. *In the context of Lemma 6.12, the sequence* $t_k := \tilde{s}_{\overline{\epsilon}-1/k}$, *with* $k > k^*$ *for some* k^*, *is bounded, i.e., contained in some ball in* $\mathbb{R}^{n+m}$.

Proof. Suppose the points t_k, $k > k^*$, are not bounded. Fix any $\epsilon' \in (\underline{\epsilon}, \overline{\epsilon})$. Clearly, there exists an integer $k' > k^*$ such that for all $k > k'$ the Dth coordinate of t_k lies in $[\epsilon', \overline{\epsilon}]$. Now define r_k, $k > k'$ to be the convex combination of $\tilde{s}_{\underline{\epsilon}}$ and t_k whose Dth coordinate is ϵ'. Again, one can easily see that also the r_k, $k > k'$, are unbounded. Moreover, convexity (C5) yields $r_k \in \mathcal{F} \cap S_{\epsilon'}$, and as Lemma 6.11 guarantees $f(\tilde{s}_\epsilon) \leq \phi$ for $\phi := f(\tilde{s}_{\underline{\epsilon}})$ and $\epsilon \in [\underline{\epsilon}, \overline{\epsilon})$, convexity implies $f(r_k) \leq \phi$ as well. Thus, we have found a sequence r_k that matches the specification of the previous lemma and therefore cannot exist. □

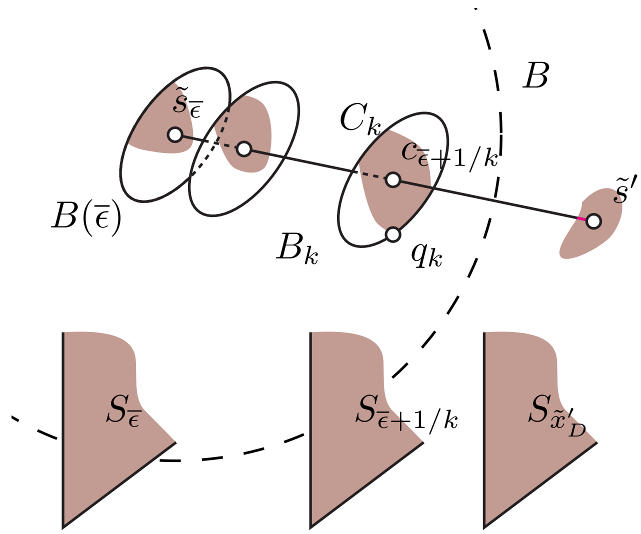


Figure 6.5. *The points q_k, $k \in \mathbb{N}$, from the proof of Lemma 6.15.*

Lemma 6.15. *In the context of Lemma 6.12, $\overline{\epsilon} < \tilde{x}'_D$ implies that $\mathcal{P}^{\overline{\epsilon}}_D(F)$ does not have a proper minimizer.*

Thus, the supremum $\overline{\epsilon}$ in Lemma 6.12 cannot be a maximum.

Proof. Assume for a contradiction that $\mathcal{P}^{\overline{\epsilon}}_D(F)$ does have a proper minimizer, $\tilde{s}_{\overline{\epsilon}}$, say, for $\overline{\epsilon} < \tilde{x}'_D$. Using an argument along the lines of the proof of Lemma 6.13 we can show that there exists then a proper interval $I^* := (\overline{\epsilon}, \overline{\epsilon}^*)$ such that all $\tilde{s}_\epsilon$ does not have a nonnegative minimizer for any ϵ in this interval. Furthermore, as $\tilde{s}_{\overline{\epsilon}}$ is proper, there exists a closed ball $B \subset \mathbb{R}^{n+m}$ of strictly positive radius centered at $\tilde{s}_{\overline{\epsilon}}$ that contains only proper points. Denote by $\tilde{s}' = (\tilde{x}', \tilde{y}')$ the minimizer of program $\mathcal{P}(F', \emptyset)$ and let c_ϵ, $\epsilon \in I^*$, be the convex combination of $\tilde{s}_{\overline{\epsilon}}$ and $\tilde{s}'$ whose Dth coordinate equals ϵ. From convexity (C5) and (C4) applied to $\phi := f(\tilde{s}_{\overline{\epsilon}}) \geq f(\tilde{s}')$ we obtain $c_\epsilon \in \mathcal{F}$ and $f(c_\epsilon) \leq \phi$ for $\epsilon \in I^*$.

Consider any radius ρ strictly smaller than the radius of B, and set $B(\epsilon) := B(c_\epsilon, \rho) \cap S_\epsilon$ for $\epsilon \in I^*$; $B(\epsilon)$ is the (lower-dimensional) ball of center c_ϵ and radius ρ in the hyperplane S_ϵ, see Fig. 6.5. From the fact that ρ is strictly smaller than the radius of B, it easily follows that $B(\epsilon + \delta)$ is contained in B for all numbers $\delta \geq 0$ that are smaller than some fixed $\delta^* > 0$.

In a similar fashion as in the proof of Lemma 6.13, we construct below (for any fixed radius $\rho < \rho_B$) a point $q^* \in \partial B(\overline{\epsilon}) \cap \mathcal{F}^{\overline{\epsilon}}$ with $f(q^*) \leq \phi$.

However, since $\tilde{s}_{\overline{\epsilon}}$ is in fact a strict minimizer, we can choose $\rho > 0$ such that $f(t) > \phi$ for all $t \in \partial B(\overline{\epsilon}) \cap \mathcal{F}^{\overline{\epsilon}}$, a contradiction to continuity of f.

In order to construct q^*, we consider the following sequence q_k, $k \in \mathbb{N}$, of points. For given k, we define q_k to be the minimizer of the set $C_k := B_k \cap \mathcal{F}_k$, where

$$B_k := B(\overline{\epsilon} + 1/k), \qquad \mathcal{F}_k := \mathcal{F}^{\overline{\epsilon}+1/k};$$

C_k is the intersection of the feasibility region of program $\mathcal{P}_D^{\epsilon_k}(F)$ and ball $B(\epsilon_k)$ for $\epsilon_k := \overline{\epsilon} + 1/k$. From (C7) we see that C_k is compact, and hence q_k is well-defined. Also, the above discussion shows that the sets C_k are for k larger than some k^* all contained in the ball B, and we can even choose k^* such that $\epsilon_k \in I^*$ for $k > k^*$. We claim now that for all such k, q_k actually lies *on the boundary* of B: if it did not, a ball of sufficiently small radius centered at q_k would witness q_k to be a minimizer of the feasibility region of $\mathcal{P}_D^{\epsilon_k}(F)$, a contradiction to the above observation that $\tilde{s}_{\epsilon_k}$ does not have a nonnegative minimizer. So $q_k \in \partial B_k \cap \mathcal{F}_k$, and $f(q_k) \leq \phi$ by the fact that q_k minimizes f over C_k and $c_{\overline{\epsilon}+1/k} \in C_k$ with $f(c_{\overline{\epsilon}+1/k}) \leq \phi$.

Finally, consider the limit point q^* of a convergent subsequence of the x_k. Continuity (C4) of the function f implies $f(q^*) \leq \phi$, and it is easily verified that $q^* \in \partial B(\overline{\epsilon})$. In fact, we also have $q^* \in \mathcal{F}^{\overline{\epsilon}}$: if the compact set $\mathcal{F}' := \bigcup_{\epsilon \in I} \mathcal{F}^{\epsilon} \cap \partial B(\epsilon)$ did not contain q^*, a neighborhood of q^* does not intersect the set, which contradicts the fact that the points q_k, $k \in \mathbb{N}$, which all lie in $\mathcal{F}'$, approach q^* to any arbitrarily small distance. □

Proof of Lemma 6.12. We assume $\overline{\epsilon} < \tilde{x}'_D$ and show that $\mathcal{P}_D^{\overline{\epsilon}}(F)$ has a nonnegative, nonproper minimizer. The previous lemma tells us that program $\mathcal{P}_D^{\overline{\epsilon}}(F)$ does not have a proper minimizer, which in particular shows that $\overline{\epsilon} > \underline{\epsilon}$. Thus, the interval $I := [\underline{\epsilon}, \overline{\epsilon})$ is nonempty, and we can consider the points

$$t_k := \tilde{s}_{\overline{\epsilon}-1/k}, \qquad k \in \mathbb{N},\, k \geq k^*,$$

where k^* is such that $\overline{\epsilon} - 1/k^* > \overline{\epsilon}$. By Lemma 6.14 the points t_k, $k \geq k^*$, are all contained in a closed ball, B, say, and therefore there exists a convergent subsequence of the t_k. Denote by t^* its limit point. Since t_k is proper and contained in the compact set $\mathcal{F} \cap B$ for all $k \geq k^*$, the point t^* must be nonnegative and contained in $\mathcal{F} \cap B$, hence in $\mathcal{F}$. If we can now show that t^* is in addition a minimizer of $\mathcal{P}_D^{\overline{\epsilon}}(F)$ then it must be a nonnegative and nonproper minimizer (Lemma 6.15), which will prove the claim.

We need another ingredient to show this. Denoting by s' the minimizer of $\mathcal{P}(F', \emptyset)$, we can see that $g(\epsilon) := f(\tilde{s}_\epsilon)$ is a monotonly decreasing function on the interval I, bounded from below by $f(\tilde{s}')$. Given this, basic analysis [11, Lemma 4.3] shows that the limit $\lim_{\epsilon \uparrow \overline{\epsilon}} f(\tilde{s}_\epsilon)$ exists and fulfills

$$\lim_{\epsilon \uparrow \overline{\epsilon}} f(\tilde{s}_\epsilon) = \inf\{f(\tilde{s}_\epsilon) \mid \epsilon \in I\} =: \phi.$$

And since $\tilde{s}_\epsilon$ is the minimizer of $\mathcal{P}_D^\epsilon(F)$, it follows that any point $t \in \mathcal{F}^\epsilon$, $\epsilon \in I$, satisfies $f(t) \geq \phi = f(t^*)$.

So suppose t^* is not a minimizer of $\mathcal{P}_D^{\overline{\epsilon}}(F)$. Then there exists a point $t' \in \mathcal{F}^{\overline{\epsilon}}$ with $f(t') < f(t^*)$. As both t' and t_{k^*} are feasible for program $\mathcal{P}(F', F')$, any convex combination of them is feasible to the program as well, by convexity (C5). Moreover, all such convex combinations except t' itself have their Dth x-coordinate in the set I, and thus f at such a point has value at least ϕ by the above observation. However, $f(t') < f(t^*) = \phi$, a contradiction to the continuity of f. □

6.5 The main result

We can summarize the findings of the preceding sections as follows.

Theorem 6.16. *Given (C1)–(C7) and any basis (together with its solution) to start with, the mathematical program $\mathcal{P}(T, \emptyset)$ in the n variables x and the m variables y can be solved in expected time*

$$(t_{\text{prim}} + t_{\text{viol}}) \cdot e^{\mathcal{O}(\sqrt{n})},$$

where t_{prim} is the (expected) running time of primitive prim *from (C3) and t_{viol} is the (expected) running time of primitive* violates *from (C2).*

Proof. Given an initial basis, we can run Gärtner's algorithm. It calls our AOP's oracle at most $\exp(\mathcal{O}(\sqrt{n}))$ times. This together with the fact that our oracle calls the primitives prim and violates each at most $|T| \leq n$ times (Lemma 6.7), shows the claim. □

We remark that for a convex program $\mathcal{P}(T, \emptyset)$ of the form (6.6), the objective function f is always continuous in the interior of the function's domain [5, Lemma 3.1.3]. If in addition f is continuous on the boundary of its domain, (C4) and (C5) are thus automatically satisfied. Also, (C1) and (C6) can usually be achieved via some sort of perturbation (allowing LP to be modeled, for instance). Moreover, the *Karush-Kuhn-Tucker Theorem for Convex Programming* with an appropriate constraint qualification might be a good starting point for obtaining (C2), see for instance [5]. We also mention that finding an initial basis is in general not an easy task.

In case of $\text{SEBB}_{\mathbf{0}}$, the material from the previous sections together with the above Theorem yields

Corollary 6.17. *Problem* SEBB *over a set of n signed balls in $\mathbb{R}^d$ can be solved in expected time*

$$\mathcal{O}(d^2 n) + e^{\mathcal{O}(\sqrt{d \log d})}.$$

Proof. The primitive can be realized in $\mathcal{O}(d^3)$ (Lemma 6.8), yielding an expected $d^4 \exp(\mathcal{O}(\sqrt{d}))$ algorithm for solving the mathematical program $\mathcal{D}(T, \emptyset)$, where T is any subset of the input balls as it arises in the basis computation of algorithm msw-subexp. The result then follows from Theorem 5.6. □

6.6 Further applications

6.6.1 Smallest enclosing ellipsoid

As our second application we consider the problem MEL of finding the smallest enclosing ellipsoid $\text{MEL}(U)$ of a n-element pointset $U \subset \mathbb{R}^d$. We show that the problem fits into our framework, but since we do not know how to realize the primitive (C3) in *subexponential* time, the resulting algorithm will have a running time that depends *exponentially* on the dimension. This is not a new result but marks an improvement over Amenta's method which does not apply to MEL.

Like SEBB, problem MEL falls into the class of LP-type problems (see [61]), and one can therefore solve it in expected time

$$\mathcal{O}(t_{\text{small}} \cdot (\delta n + e^{\mathcal{O}(\sqrt{\delta \log \delta})})), \tag{6.14}$$

where $\delta = d\,(d+3)/2$ and where t_{small} is the (expected) time required to compute $\text{MEL}(T)$ for sets T of at most $\delta+1$ points in $\mathbb{R}^d$ (see Lemma 2.11).

Instead of computing $\textsc{mel}(T)$ directly, we use the following trick by Khachiyan and Todd [54]; by embedding our input points T into $\mathbb{R}^{d+1}$, it allows us to focus on (smallest enclosing) ellipsoids with the center *fixed at the origin*. For this, we denote by $\textsc{cmel}(P)$ the smallest ellipsoid that contains the points $P \subset \mathbb{R}^d$ and has its center at the origin.

Lemma 6.18 (Khachiyan & Todd)**.** *Let $P \subset \mathbb{R}^d$ be a finite pointset with* $\mathrm{aff}(P) = \mathbb{R}^d$*. Set*

$$P' := \{(s,1) \mid s \in S\} \quad \textit{and} \quad \Pi = \{(x,\psi) \in \mathbb{R}^{d+1} \mid \psi = 1\},$$

and denote by π the projection $\pi(x,y) = x$ from $\mathbb{R}^{d+1}$ to $\mathbb{R}^d$. Then $\textsc{mel}(P) = \pi(\textsc{cmel}(P') \cap \Pi)$.

(In fact, Khachiyan & Todd prove a stronger statement about $(1+\epsilon)$-approximations of $\textsc{mel}(P)$; the above is simply the case $\epsilon = 0$.) In the following we will assume that P affinely spans $\mathbb{R}^d$, that is, $\mathrm{aff}(P) = \mathbb{R}^d$; if this is not the case, the smallest enclosing ellipsoid of P lives in a proper affine subspace $A \subset \mathbb{R}^d$ and can be found by identifying A (using linear algebra techniques) and doing the computation in there.

It is well-known [53, 26, 78, 79] that $\textsc{cmel}(T)$ can be read off the global minimizer of the convex mathematical program $\mathcal{E}(T,\emptyset)$, which is defined as follows.

$$\begin{array}{lll} \mathcal{E}(U,V) & \text{minimize} & -\log\det(\sum_{p\in T} x_p pp^T) \\ & \text{subject to} & \sum_{p\in T} x_p = 1, \\ & & x_p \geq 0, \qquad p \in U \setminus V, \\ & & x_p = 0, \qquad p \in T \setminus U. \end{array}$$

(We take $\log(\alpha) = -\infty$ for $\alpha \leq 0$) Namely, if $\tilde{x}$ optimally solves $\mathcal{E}(T,\emptyset)$, the matrix $M(\tilde{x}) := \sum_{p\in T} \tilde{x}_p pp^T$ defines the ellipsoid $\textsc{cmel}(P)$ via

$$\textsc{cmel}(P) = \{x \in \mathbb{R}^d \mid x^T M(\tilde{x})^{-1} x \leq d\}.$$

Here are the optimality conditions for program $\mathcal{E}(U,V)$ that we will use.

Lemma 6.19. *A finite feasible $\tilde{x} \geq \mathbf{0}$ minimizes program $\mathcal{E}(U,V)$ iff*

$$\begin{array}{rcll} p^T M(\tilde{x})^{-1} p & \leq & d, & p \in U \setminus V, \qquad (6.15) \\ p^T M(\tilde{x})^{-1} p & = & d, & p \in V, \qquad (6.16) \\ \tilde{x}_B\,(p^T M(\tilde{x})^{-1} p - d) & = & 0. & p \in U \setminus V, \qquad (6.17) \end{array}$$

In [53] and other papers, the special case $V = \emptyset$ is proved. Our version can be proved along the same lines as follows.—Note in the statement that the regularity of the matrix $M(\tilde{x})$ follows from the finiteness of the solution $\tilde{x}$.

Proof. It is well-known [47] that $\log\det(X)$ is concave over the cone of positive semidefinite matrices X; as $M := M(x)$ is positive semidefinite, the objective function of program $\mathcal{E}(U, V)$ is thus convex over the positive orthant.

We again use the Karush-Kuhn-Tucker Theorem for convex programming (Theorem 5.16). A little calculation (using *Jacobi's formula* $\frac{\mathrm{d}}{\mathrm{d}\alpha}\det(A) = \mathrm{Tr}(\mathrm{Adj}(A)\frac{\mathrm{d}}{\mathrm{d}\alpha}A$ and the identity $x^T Ax = \mathrm{Tr}(Axx^T)$ for a quadratic matrix A) shows that

$$\frac{\partial f}{\partial x_p} = -\,\mathrm{Tr}(M^{-1}pp^T) = -p^T M^{-1} p, \qquad p \in F. \tag{6.18}$$

Using this, the theorem states that a feasible $\tilde{x} \geq \mathbf{0}$ is locally optimal to $\mathcal{E}(U, V)$ if and only if there exists a real τ and real numbers $\mu_p \geq 0$, $p \in U$, such that $p^T M^{-1} p + \mu_p = \tau$ and $\tilde{x}_p \mu_p = 0$ for all $p \in U$.

Multiplying the latter equation by x_p and summing over all $p \in F'$ yields $\sum_{p\in U} x_p p^T M^{-1} p = \tau$, where we have used $\tilde{x}_p \mu_p = 0$. On the other hand, we have

$$\begin{aligned} \sum_{p\in U} x_p p^T M^{-1} p &= \sum_{p\in U} x_p \,\mathrm{Tr}(M^{-1}pp^T) \\ &= \mathrm{Tr}(M^{-1} \sum_{p\in U} x_p pp^T) \\ &= \mathrm{Tr}(M^{-1}M) = d. \end{aligned} \tag{6.19}$$

Combining these two equations we obtain $\tau = d$, and the claim follows. □

The primitive. From the above lemma it is clear that program $\mathcal{E}(U, V)$ satisfies condition (C2) from our framework; the resulting violation test $\mathsf{violates}(q, V, \tilde{x})$ computes $M(\tilde{x})^{-1}$ in time $\mathcal{O}(d^3)$ and returns whether $q^T M(\tilde{x})^{-1} q > d$ holds. Uniqueness (C1) follows from the fact that CMEL(T) is unique, and (C4), (C5), and (C7) are trivially satisfied. It remains to show that program $\mathcal{E}_q^{\epsilon}(F)$—the counterpart to the abstract program $\mathcal{P}_D^\epsilon(F)$—has at most one proper minimizer, and that the primitive (C3) can be realized.

Program $\mathcal{E}_q^\epsilon(F)$ is of the form

$$\begin{array}{lll} \mathcal{E}_q^\epsilon(F) & \text{minimize} & -\log\det(M(x)) \\ & \text{subject to} & \sum_{p\in F} x_p = 1-\epsilon, \\ & & x_p = 0, \qquad p \in T\setminus F', \end{array}$$

where $x_q := \epsilon$ is a constant and $F' := F \cup \{q\} \subseteq T$ with $q \notin F$.

In order to establish (C6) we use the well-known fact [47, Th. 7.6.7] that $\log\det(X)$ is *strictly* convex over the set of positive definite matrices X: consequently, if there were two proper minimizers of $\mathcal{E}_q^\epsilon(F)$ with equal, finite objective value, any proper convex combination of them yields a better solution, contradiction.

Lemma 6.20. *Let $F' = F \cup \{q\} \subseteq T$, $q \notin F$, and $\epsilon \geq 0$. Then a finite feasible $x \geq \mathbf{0}$ is optimal to program $\mathcal{E}_q^\epsilon(F)$ iff*

$$p^T M(\tilde{x})^{-1} p = \frac{d - \epsilon q^T M(\tilde{x})^{-1} q}{1-\epsilon}, \qquad p \in F. \tag{6.20}$$

Proof. Set $M := M(\tilde{x})$ and note from finiteness that M is regular. Using (6.18) and the Karush-Kuhn-Tucker Theorem 5.16 we see that a feasible $x \geq \mathbf{0}$ is optimal to $\mathcal{E}_q^\epsilon(F)$ if and only if there exists a real τ such that $p^T M^{-1} p = \tau$ holds for all $p \in F$. Multiplying this by x_p and summing over all $p \in F'$ yields $\sum_{p\in F'} x_p p^T M^{-1} p = \epsilon q^T M^{-1} q + (1-\epsilon)\tau$. On the other hand, we have $\sum_{p\in F'} x_p p^T M^{-1} p = d$, the proof of which is like in (6.19). By combining these two equations and solving for τ, the claim follows. □

Using the equations (6.20) and decision algorithms for the *existential theory of the reals* [10], procedure prim can be implemented in exponential time in the bit-complexity model.

6.6.2 Distance between convex hulls of balls

Let S be a finite set of closed balls in $\mathbb{R}^d$. We define the *convex hull* (or *hull* for short) of S to be the pointset $\text{conv}(S) := \text{conv}(\bigcup_{B\in S} B)$. (The set $\text{conv}(S)$ is also called a *spherically extended polytope*, or an *s-tope* in

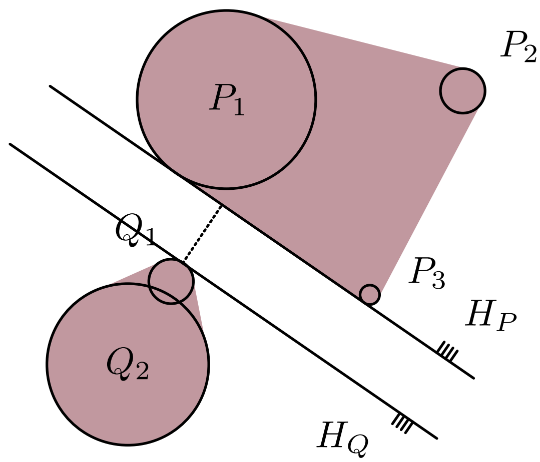


Figure 6.6. *Problem* PDS *for two sets P, Q of balls: the minimal distance is attained between p and q. (The meaning of the halfspaces H_P and H_Q is described in Lemma 6.24.)*

the literature, see [45, 44].) For two given ball sets P, Q in $\mathbb{R}^d$, we denote by

$$\mathrm{dist}(P, Q) := \min\{\|p - q\| \mid p \in \mathrm{conv}(P),\, q \in \mathrm{conv}(Q)\}.$$

the *distance* between P and Q. Observe that $\mathrm{dist}(P, Q) = 0$ if and only if $\mathrm{conv}(P)$ and $\mathrm{conv}(Q)$ have nonempty intersection. We denote by PDS the problem of computing the distance $\mathrm{dist}(P, Q)$ (together with the points $p \in \mathrm{conv}(P)$ and $q \in \mathrm{conv}(Q)$ for which the distance is attained) between two hulls of ball sets P and Q. Figure 6.6 shows an example instance of PDS.

We start with a simple observation (which is implicitly assumed but not proven in [45, 44]). To state it, we define for a set S of balls $\mathrm{sp}(S)$ to be the set of all balls $B(c, \rho)$ whose center c can be written as $c = \sum_{B \in S} x_B c_B$ for real, nonnegative coefficients x_B, $B \in S$ with sum 1 while at the same time the radius ρ fulfills $\rho = \sum_{B \in S} x_B \rho_B$. Figure 6.7 shows (some subsets of) the balls in $\mathrm{sp}(S)$ for some set S of three circles in the plane.

Lemma 6.21. $\mathrm{conv}(S) = \bigcup_{B \in \mathrm{sp}(S)} B$ *for any finite set S of balls in $\mathbb{R}^d$.*

We note that while all balls $B \in \mathrm{sp}(S)$ are contained in the pointset $\mathrm{conv}(S)$, not every ball from $\mathrm{conv}(S)$ is necessarily contained in $\mathrm{sp}(S)$.

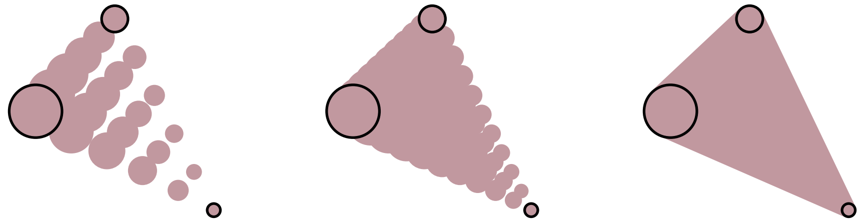


Figure 6.7. *The balls (filled) from the set* sp(S) *for three circles S (solid): (a) shows a few of them, (b) some more, and (c) all of them.*

Proof. We first show that every point $p \in \operatorname{conv}(S)$ is contained in some ball, B_p, say, from the set $\operatorname{sp}(S)$. As p is a convex combination of the points in S, Carathéodory's Theorem [71, Corollary 17.1.1] allows us to express p as the convex combination of at most $d+1$ points, each belonging to a different ball $B \in S$. That is, there exist nonnegative real coefficients x_B, $B \in S$, and points $p_B \in B$, $B \in S$, such that

$$p = \sum_{B\in S} x_B p_B, \qquad \sum_{B \in S} x_B = 1.$$

We claim that the ball $B_p := B(c,\rho)$ with center $c = \sum_{B\in S} x_B c_B$ and radius $\rho = \sum_{B\in S} x_B \rho_B$ contains the point p. This is easily verified as $\|c-p\|$ equals

$$\|\sum_{B\in S} x_B\,(c_B - p_B)\| \leq \sum_{B\in S} x_B \|c_B - p_B\| \leq \sum_{B\in S} x_B \rho_B = \rho.$$

For the converse inclusion '$\supseteq$' we fix some ball $D \in \operatorname{sp}(S)$ and denote by x_B, $B \in S$, the corresponding coefficients (which sum up to 1 and satisfy $c_D = \sum_{B\in S} x_B c_B$ and $\rho_D = \sum_{B\in S} x_B \rho_B$). We show that every point $p \in D$ is contained in $\operatorname{conv}(S)$. To see this, we write p as $p = c_D + \alpha\rho_D u$ for some unit vector u and some positive real number α (observe that $\alpha \leq 1$). Clearly, the point $p_B := c_B + \alpha\rho_B u$ is contained in ball B, and since these points fulfill

$$\sum_{B\in S} x_B p_B = \sum_{B\in S} x_B c_B + \alpha \sum_{B\in S} x_B \rho_B u = c_D + \alpha\rho_D u = p,$$

the claim follows. □

The above lemma allows us to derive a convex mathematical programming formulation of problem PDS. Let $P \cup Q =: T$ be two sets of balls, and define C to be the matrix $((c_B)_{B\in P}, (-c_B)_{B\in Q})$; C contains the centers of the balls in Q as its columns in a first block, followed by a second block with the negated centers of the balls from Q. To facilitate notation, we denote by 'C_S', where S is some subset of T, the submatrix of C that contains the columns with the (possibly negated) centers of precisely the balls from S. Furthermore, we write $r = (\rho_B)_{B\in P\cup Q}$ for the vector containing the radii of the balls $P \cup Q$ (in the same order as the columns of C) and again use 'r_S' for the subvector containing the radii of the balls S.

Consider now for $V \subseteq U \subseteq T$ the following mathematical program.

$$\begin{array}{lll} \mathcal{S}(U,V) & \text{minimize} & \sqrt{x^T C^T C x} - r^T x \\ & \text{subject to} & \sum_{B\in P} x_B = 1, \\ & & \sum_{B\in Q} x_B = 1, \\ & & x_B \geq 0, \qquad B \in U \setminus V, \\ & & x_B = 0, \qquad B \in T \setminus U. \end{array}$$

The next lemma shows that a solution to $\mathcal{S}(T, \emptyset)$ provides us with two balls that attain the *same* distance as P and Q.

Lemma 6.22. *Let $P \cup Q$ be two sets of balls and x^* a minimizer to program $\mathcal{S}(T, \emptyset)$ with γ^* its objective value. Then*

$$\mathrm{dist}(P, Q) = \max\{0, \gamma^*\}.$$

More precisely, $\mathrm{dist}(\{B_P\}, \{B_Q\}) = \mathrm{dist}(P, Q)$*, where*

$$\begin{array}{lll} B_P & = & B(C_P x_P^*, r_P^T x_P^*) \subseteq \mathrm{conv}(P), \\ B_Q & = & B(C_Q x_Q^*, r_Q^T x_Q^*) \subseteq \mathrm{conv}(Q). \end{array}$$

Proof. From Lemma 6.21 it is straightforward that

$$\mathrm{dist}(P, Q) = \min\{\mathrm{dist}(\{B\}, \{B'\}) \mid B \in \mathrm{sp}(P),\ B' \in \mathrm{sp}(Q)\}. \quad (6.21)$$

Moreover, the distance $\mathrm{dist}(\{B\}, \{B'\})$ between two balls is easily seen to be $\max\{0, \|c_B - c_{B'}\| - \rho_B - \rho_{B'}\}$ for any two balls $B, B' \subset \mathbb{R}^d$.

By definition of $\mathrm{sp}(P)$, the matrix C, and the vector r, a ball $B(c, \rho)$ lies in $\mathrm{sp}(P)$ if and only if $c = C_P x_P$ and $\rho = r_P^T x_P$ for some nonnegative vector x_P whose entries add up to 1. (And, of course, the very same statement holds if you replace 'P' by 'Q.') It follows that $x \in \mathbb{R}^{|T|}$ is feasible to program $\mathcal{S}(T)$ if and only if the ball $B(C_P x_P, r_P^T x_P)$ lies in $\mathrm{sp}(P)$ and the ball $B(C_Q x_Q, r_Q^T x_Q)$ lies in $\mathrm{sp}(Q)$. By the above formula, the distance between these balls is

$$\|C_P x_P - C_Q x_Q\| - \rho_P^T x_P - \rho_Q^T x_Q = \sqrt{x^T C^T C x} - r^T x,$$

if this number is positive and zero otherwise. From (6.21) we conclude that minimizing this number over all balls in $\mathrm{sp}(P)$ and $\mathrm{sp}(Q)$ yields $\mathrm{dist}(P, Q)$ (in case it is positive) or shows that $\mathrm{conv}(P)$ and $\mathrm{conv}(Q)$ intersect (in case it is nonpositive). □

Denoting by f the objective function of program $\mathcal{S}(U, V)$, the triangle inequality yields

$$\begin{aligned} f((1-\alpha)x + \alpha x') &\leq (1-\alpha)\|Cx\| + \alpha\|Cx'\| \\ &= (1-\alpha)f(x) + \alpha f(x'), \end{aligned}$$

which shows that $\mathcal{S}(U, V)$ is a *convex* program. We note that if the input centers are assumed to be linearly independent (equivalently, C has full rank), the program's objective function is even *strictly* convex: for arbitrary vectors $a, b \in \mathbb{R}^d$ we have $\|a + b\| = \|a\| + \|b\|$ if and only if $a = \gamma b$ for some scalar $\gamma \geq 0$. Using this, the above inequality is fulfilled with equality if and only if $Cx = Cx'$, equivalently, if and only if $x' = \gamma x$ for some γ. Then, however, we must have $\gamma = 1$ because otherwise not both points x and $x' = \gamma x$ can be feasible (recall that their entries add up to 1).

Lemma 6.23. *A feasible solution $\tilde{x} \geq \mathbf{0}$ with $\tilde{x}^T C^T C \tilde{x} \neq 0$ is optimal to $\mathcal{S}(U, V)$ iff there are real numbers τ_P, τ_Q such that*

$$\begin{aligned} \mu_B &= 0, & B &\in V, \\ \mu_B &\geq 0, & B &\in U \setminus V, \\ \mu_B \tilde{x}_B &= 0, & B &\in U \setminus V \end{aligned}$$

for $\mu_B := c_B^T C \tilde{x} / \sqrt{\tilde{x}^T C^T C \tilde{x}} - \rho_B + \tau_{[B]}$. Here, $\tau_{[B]} = \tau_P$ if $B \in P$ and $\tau_{[B]} = \tau_Q$ else. In this case, the objective value of $\tilde{x}$ equals $-(\tau_P + \tau_Q)$.

Notice here that the numbers τ_P, τ_Q are in fact *unique*: feasibility of $\tilde{x}$ ensures $\tilde{x}_B > 0$ for some $B \in P$ and some $B \in Q$, and therefore $\mu_B = 0$ for both these B, implying that τ_P and τ_Q are uniquely determined.

Let us see what happens if the assumption $\tilde{x}^T C^T C \tilde{x} \neq 0$ from the lemma is not fulfilled. Then the points $\tilde{p} := C_P \tilde{x}_P$ and $\tilde{q} := -C_Q \tilde{x_Q}$ have Euclidean distance zero. And as $\tilde{p}$ lies in the convex hull of the centers of the balls P, and likewise $\tilde{q}$ lies in the convex hull of the centers of Q, we see that $\text{conv}(P)$ intersects $\text{conv}(Q)$. Thus, if $\tilde{x}^T C^T C \tilde{x} = 0$, we can immediately output '$\text{dist}(P, Q) = 0$' (and the points $\tilde{p}$ and $\tilde{q}$ serve as witnesses for this).

Proof. Since $\mathcal{S}(U, V)$ is convex, the Karush-Kuhn-Tucker Theorem (Theorem 5.16) applies, yielding that a feasible $\tilde{x}$ is optimal if and only if there exist two real numbers τ_P, τ_Q and real numbers μ_B, $B \in T$, such that

$$\frac{1}{\sqrt{\tilde{x}^T C^T C \tilde{x}}} c_B^T C \tilde{x} - \rho_B + \tau_{[B]} - \mu_B = 0, \tag{6.22}$$

and $\tilde{x}_B \mu_B = 0$, $B \in T$, and such that $\mu_B \geq 0$ for all $B \in U \setminus V$ and $\mu_B = 0$, $B \in V$. From this, the first part of the claim follows.

Multiplying (6.22) by x_B and summing over all $B \in T$, we obtain

$$f(\tilde{x}) = \frac{1}{\sqrt{\tilde{x}^T C^T C \tilde{x}}} \tilde{x} C^T C \tilde{x} - r^T \tilde{x} = -(\tau_P + \tau_Q),$$

where f denotes the program's objective function and where we have used $\tilde{x}_B \mu_B = 0$ and $\sum_{B \in R} \tilde{x}_B = 1$, $R \in \{P, Q\}$. □

From the proof we can also extract a geometric interpretation of optimality. Let us focus for this on the case $V = \emptyset$. By multiplying (6.22) by x_B and summing over all $B \in P$ ($B \in Q$, respectively), we get

$$\begin{aligned} \tilde{p}^T u - r_P^T \tilde{x}_P + \tau_P &= 0, \\ \tilde{q}^T u + r_Q^T \tilde{x}_Q - \tau_Q &= 0, \end{aligned}$$

where we introduced the unit vector $u := C\tilde{x} / \sqrt{\tilde{x} C^T C \tilde{x}}$ (and $\tilde{p}$ and $\tilde{q}$ are defined as after the lemma). That is, the positive ball $B(\tilde{p}, r_P^T \tilde{x}_P)$ is *internally tangent*[1] to the halfspace $H_P := \{x \mid u^T x + \tau_P \geq 0\}$, and likewise the positive ball $B(\tilde{q}, r_Q^T \tilde{x}_Q)$ is internally tangent to the halfspace $H_Q := \{x \mid u^T x - \tau_Q \leq 0\}$. In addition, the conditions $\mu_B \geq 0$, $B \in U$, from the lemma show that all balls $B \in P$ ($B \in Q$, respectively) are contained in the halfspace H_P (H_Q, respectively). By finally observing that u is the vector $C\tilde{x} = \tilde{p} - \tilde{q}$ scaled to unit length, we arrive at

[1]Refer to page 102 for a precise definition of *internal tangency.*

Lemma 6.24. *Let $p \in \operatorname{conv}(P)$ and $q \in \operatorname{conv}(Q)$ be two points attaining minimal distance between the hulls of the ball sets P and Q. Then there exists a pair of halfspaces H_P and H_Q at distance $\|p-q\|$ such that* $\operatorname{conv}(P) \subset H_P$ *and* $\operatorname{conv}(Q) \subset H_Q$.

An example illustrating this is given in Fig. 6.6.

Lemma 6.25. PDS *can be formulated as a LP-type problem of combinatorial dimension at most $d+2$.*

For the proof of this we will use the following fact. Given a linear system $Ax = b$ of $k \leq l$ equalities in $x \in \mathbb{R}^l$, there exists a solution $\tilde{x}$ that has at most k nonzero entries. (The rank of the matrix A is at most k and therefore the kernel $\operatorname{kern}(A)$ of A has dimension at least $l-k$. So if $\tilde{x}$ has more than k nonzero entries, there must exists an element $\tilde{y} \in \operatorname{kern}(A)$ such that both $\tilde{x}$ and $\tilde{y}$ have nonzero ith entry for some i. Now $\tilde{x} + \lambda\tilde{y}$ is a solution of the system $Ax = b$ as well; in particular, setting $\lambda := -\tilde{x}_i$, we see that there exists a solution with one nonzero entry *less*. Using induction, this shows the claim.)

Proof. Let $T = P \cup Q$ be an instance of PDS, where we assume that the balls in P are different from the balls in Q—as a matter of fact, it suffices for what follows that the balls are *labeled* differently. Given $U \subseteq T$, we define $w(T)$ as follows. If U encodes a proper PDS subinstance, i.e., if both $P(U) := P \cap U$ and $Q(U) := Q \cap U$ are nonempty sets, we take the two halfspaces $H_{P(U)}$ and $H_{Q(U)}$ from the above lemma and define $w(U) := (H_{P(U)}, H_{Q(U)})$. In case one of the sets $P(U), Q(U)$ is empty, we set $w(U)$ to the special symbol $-\bowtie$. Furthermore, we define $w(U') \preceq w(U)$ for $U', U \in T$ if and only if $w(U') = -\bowtie$ or the distance between the halfspaces $w(U)$ is smaller or equal to the distance between the halfspaces $w(U')$. In this way, we obtain a quasiorder whose minimal element is $-\bowtie$.

Monotonicity of $(T, \preceq, w)$ is easily verified. To prove locality, assume $-\bowtie < w(U') = w(U)$ for $U' \subseteq U \subseteq T$. If $w(U' \cup \{B\}) \succ w(U')$ for some $B \in P$—the case $B \in Q$ is handled along the same lines—then the previous lemma shows that B is not contained in the halfspace $H_{P(U')}$. As the latter halfspace equals $H_{P(U)}$, we see that B is neither contained in $H_{P(U)}$, which in turn implies $w(U \cup \{B\}) \succ w(U)$ via the lemma.

To establish the bound on the combinatorial dimension, we show that program $\mathcal{S}(T, \emptyset)$ has an optimal solution $\tilde{x}^*$ such that $|F_{\tilde{x}^*}| \leq d+2$ for

$F_x := \{B \in T \mid x_B > 0\}$. Lemma 6.23 then shows that $\tilde{x}^*$ also solves $\mathcal{S}(F_{\tilde{x}^*}, \emptyset)$ optimally, so $w(F_{\tilde{x}^*}) = w(T)$, proving $\dim(T, w) = |F_{\tilde{x}^*}| \leq d+2$. So consider an optimal solution $\tilde{x}$ to $\mathcal{S}(T, \emptyset)$ and suppose $|F_{\tilde{x}}| > d+2$. Clearly, $\tilde{x}$ is a solution to the system

$$\begin{aligned} C\tilde{x} &= Cx, \\ \textstyle\sum_{B\in P} x_B &= 1, \\ \textstyle\sum_{B\in Q} x_B &= 1, \end{aligned}$$

consisting of $d+2$ linear equations; in it, x is a $|T|$-vector with $x_B = 0$, $B \in T \setminus F$, i.e., the variables of the system are the x_B, $B \in F$. The remark preceding the lemma yields a solution $\tilde{x}'$ to the system with at most $d+2$ nonzero entries; since $\tilde{x}'_B = 0$ for $B \in T \setminus F$, these entries are among the variables x_B, $B \in F$.

Now consider the convex combination $\tilde{y}(\tau) := (1-\tau)\tilde{x} + \tau\tilde{x}'$, which fulfills the two linear constraints of $\mathcal{S}(T, \emptyset)$ for all real τ. Increase τ continuously, starting from 0 on, and stop as soon as the first of the entries $\tilde{y}(\tau^*)_B$, $B \in F$, drops to zero. At this point we have $\tilde{y}(\tau^*)_B = 0$, $B \in T \setminus F$, and $\tilde{y}(\tau^*)_B \geq 0$, $B \in F$, so $\tilde{y}(\tau^*)$ is feasible to $\mathcal{S}(T, \emptyset)$. We claim that the objective function $f(x)$ of $\mathcal{S}(T, \emptyset)$ fulfills

$$f(\tilde{x}^*) = f(\tilde{y}(\tau^*)).$$

As $C\tilde{x} = C\tilde{y}(\tau^*)$, it suffices to show $r^T\tilde{x} = r^T\tilde{y}(\tau^*)$ in order to establish this. But $g(\tau) := r^T\tilde{y}(\tau)$ is a linear function in τ, so if $g(\tau)$ were not constant, it would increase for $\tau > 0$ and decrease for $\tau < 0$ (or the other way around), and we would thus obtain a solution $\tilde{y}(\tau)$ with *better* objective value, a contradiction to the optimality of $\tilde{x}$. Consequently, $\tilde{y}(\tau^*)$ is a feasible solution to $\mathcal{S}(T, \emptyset)$ with the same objective value as the optimal solution $\tilde{x}$, but it has one less nonzero coefficient. By induction, this shows that a solution $\tilde{x}^*$ with at most $d+2$ nonzero entries exists. □

A subexponential algorithm. As we are going to show now, program $\mathcal{S}(U, V)$, $V \subseteq U \subseteq T$, falls into framework from Theorem 6.16. In this case, Lemma 6.23, together with the fact that the numbers τ_P, τ_Q are unique, proves (C2); the argument parallels the respective proof in case of SEBB$_{\mathbf{0}}$ and MEL above. By embedding the input balls T into sufficiently high-dimensional space and perturbing, we can always achieve that the columns of the matrix C are linearly independent, in which case program $\mathcal{S}(U, V)$ is strictly convex, as we have seen. Using strictness it is also straightforward to verify that (C4)–(C7) apply, so that it remains to develop the primitive (C3). (We remark that the precondition '$\tilde{x}^T C^T C \tilde{x} \neq 0$ from Lemma 6.23 does not cause any problems because as soon as we—inside the primitive—detect a violation to it, we can immediately exit and output that the distance is zero.)

We realize the primitive in a similar fashion as in case of program $\mathcal{D}(U, V)$ solving $\textsc{sebb}_{\mathbf{0}}$. Denoting by $I_F(P, Q)$ the $(2 \times |F|)$-matrix whose first row is $(\mathbf{1}_F, \mathbf{0})$ and whose second row is $(\mathbf{0}_F, \mathbf{1})$, we introduce

$$M_F := \begin{pmatrix} -I_d & \mathbf{0} & C_F \\ \mathbf{0}^T & \mathbf{0} & I_F(P,Q) \\ C_F^T & I_F(P,Q)^T & \mathbf{0} \end{pmatrix}$$

Along the same lines as in the proof of Lemma 6.9, the matrix M_F is seen to be regular (this makes use of linear independence of C, which we assume for the rest of this section). The counterpart to Lemma 6.10 for $\mathcal{D}_D^\epsilon(F)$ is then

Lemma 6.26. *A real x is feasible and optimal to $\mathcal{S}_D^\epsilon(F)$ iff there exists a real vector u and two real numbers ν_P, ν_Q such that*

$$M_F \begin{pmatrix} u \\ \nu_P \\ \nu_Q \\ \tilde{x}_F \end{pmatrix} = \begin{pmatrix} -\epsilon d_D \\ 1 - [D \in P]\epsilon \\ 1 - [D \in Q]\epsilon \\ \mathbf{0} \end{pmatrix}. \qquad (6.23)$$

Here, '$[x \in X]$' equals 1 iff $x \in X$ and 0 otherwise.

Proof. An invocation of the Karush-Kuhn-Tucker Theorem similar to the one in the proof of Lemma 6.10 shows that a feasible x is optimal to $\mathcal{S}_D^\epsilon(F)$ if and only if there exist two real numbers τ_F and τ_G such that

$$c_B^T \frac{C_{F'} x_{F'}}{\sqrt{\tilde{x}^T C^T \tilde{x}}} + \tau_{[B]} = 0, \qquad B \in F.$$

Multiply this by $\zeta := \sqrt{\tilde{x}^T C^T \tilde{x}}$, and set $\nu_P := \zeta \tau_P$ and $\nu_Q := \zeta \tau_Q$; with this the system (6.23) encodes feasibility and optimality. □

Given this, it is now an easy matter to realize the primitive for $\mathcal{S}_D^\epsilon(F)$: the regularity of M_F allows us to solve (6.23) for the tuple (u, ν_P, ν_Q, x_F), yielding expressions for these entries that are linear in the unknown ϵ. Solving $x_B = 0$, $B \in F$, for ϵ allows us to recover the times of the events (a_B), and using Lemma 6.23 we can also calculate the arrival time of event (b). (Please refer to the proof of Lemma 6.8 for the almost identical details.)

Corollary 6.27. *Problem* PDS *over a set of n positive balls $T = P \cup Q$ in $\mathbb{R}^d$ can be solved in expected time*

$$\mathcal{O}(d^5 n) + d^4 e^{\mathcal{O}(\sqrt{d \log d})}.$$

Proof. As the problem is LP-type, we employ the algorithm behind Lemma 2.11 to solve it and use the machinery we have just developed only to realize the algorithm's basis computation. With the combinatorial dimension being bounded by $d+2$, we thus obtain a maximal expected running time of at most

$$t_{\mathrm{b}} \cdot \mathcal{O}(dn + e^{\mathcal{O}(\sqrt{d \log d})}), \tag{6.24}$$

where t_{b} denotes the (expected) time to perform a basis computation.

In order to implement the basis computation $\mathsf{basis}(W, B)$ for a given subset $W \cup \{B\}$ of the input balls with W a basis and B a violator, we can assume W to be of size at most $d+1$. (If W has already size $d+2$, the distance between the input hulls is zero and we are finished.) In order to apply Theorem 6.16, we perform a suitable embedding into $\mathbb{R}^{d+2}$ and a symbolic perturbation à la Lemma 5.25 that ensures linear independence of the centers of $W \cup \{B\}$. Then we select any two balls $F_* := \{B', B''\}$ from $W \cup \{B\}$, one from P and one from Q. By the constraints $\sum_{B \in P} x_B = 1$ and $\sum_{B \in Q} x_B = 1$ of program $\mathcal{S}(F_*, F_*)$, we see that $x_{B'} > 0$ and $x_{B''} > 0$, which proves F_* to be a basis. Theorem 6.16 together with the observation that the primitive prim can be realized in $\mathcal{O}(d^3)$ now yields an expected $d^4 \exp(\mathcal{O})(\sqrt{d}))$ algorithm for solving the mathematical program $\mathcal{S}(W \cup \{B\}, \emptyset)$. Plugging this into (6.24) proves the claim. □

We remark that the formulation of PDS as the mathematical program $\mathcal{S}(U, V)$ falls into Amenta's framework [2] (see the remarks at the end of this chapter). Thus, the subexponential bound we obtain for it is not a new result (although our method might be more suitable for an implementation).

6.7 Remarks

Based on Gärtner's algorithm for AOPs, Amenta [2] devised an expected subexponential-time algorithm for what she calls *convex linear programming*, that is, for the minimization of a smooth, strictly convex function over the intersection of a finite family of halfspaces: the goal is to solve the mathematical program $\mathcal{A}(T,\emptyset)$ defined via

$$\begin{array}{llrcll} \mathcal{A}(U,V) & \text{minimize} & f(x) & & & \\ & \text{subject to} & g_B(x) & \leq & 0, & B \in U \setminus V, \\ & & g_B(x) & = & 0, & B \in V, \end{array}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is strictly convex, and $V \subseteq U \subseteq T$ are sets indexing the given *linear* (in)equality constraints $g_B \leqq 0$. In order for her algorithm to work, the caller has to provide a polynomial-time subroutine $\mathsf{prim}_{\mathcal{A}}(F)$ that solves program $\mathcal{A}(F,F)$ for given $F \subseteq T$.

Amenta's framework is slightly more limited than ours. On the one hand, there are problems which almost fit into the above form '$\mathcal{A}(U,V)$,' yet not entirely. For instance, SEBB$_{\mathbf{0}}$ (in the formulation we developed in this chapter) involves a single additional nonlinear constraint only and thus fails Amenta's framework. On the other hand, there are problems like MEL, the problem of computing the smallest ellipsoid enclosing a d-dimensional set of points, that admit a formulation in the form '$\mathcal{A}(T,\emptyset)$' above, but for which a realization of the subroutine $\mathsf{prim}_{\mathcal{A}}(F)$ seems out of reach: MEL's mathematical programming formulation involves a convex objective function, subject to one equality constraint, and one nonnegativity constraint $x_p \geq 0$ per input point only. Yet, none of the inequality constraints can be dropped—as is needed for the subroutine $\mathsf{prim}_{\mathcal{A}}(F)$—because the objective function's convexity is lost if we do so (and with it, the KKT optimality conditions).

In contrast to Amenta's solver, the main advantage of our method is that we do not need to 'artificially' extend the program's feasibility domain for the realization of our computational primitive. (In Amenta's framework, the feasibility domain of program $\mathcal{A}(F,F)$, which the subroutine needs to solve, is in general larger than the one of the original program $\mathcal{A}(T,\emptyset)$.) Our algorithm *always stays in the interior* of the program's (original) feasibility domain, and thus we do not require the objective function to be convex everywhere. (Besides this, we only need (mere and not strict) convexity and can handle nonlinear convex constraints.) In particular, we do not require 'optimality conditions,' i.e., a violation test, for solutions outside the (original) feasibility domain. This is essential because only for the (originally) feasible points does the Karush-Kuhn-Tucker Theorem imply necessary and sufficient conditions (provided it applies at all). Our applications for SEBB$_\mathbf{0}$ and MEL heavily rely on this feature.

To the best of our knowledge, our algorithm for SEBB from Corollary 6.17 is the first one to achieve a *subexponential* time complexity. We have implemented a variant of the resulting algorithm for SEBB as a prototype in Maple [17]. Instead of running Gärtner's subexponential algorithm, we employ the deterministic procedure AOP-DET from [32], which simply iterates the computational primitive until the optimal solution has been found (requiring exponential time in the worst case). With this, we can solve instances of up to 300 balls in $\mathbb{R}^{300}$ within two hours (using exact arithmetic without filtering).—Of course, this does not say anything about the running time of Gärtner's subexponential algorithm.

Bibliography

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via core sets. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and computational geometry*. Mathematical Sciences Research Institute Publications, 2005 (to appear).

[2] N. Amenta. *Helly theorems and Generalized Linear Programming*. PhD thesis, University of California, Berkley, 1993.

[3] M. Bădoiu and K. L. Clarkson. Smaller core-sets for balls. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 801–802. Society for Industrial and Applied Mathematics, 2003.

[4] L. J. Bass and S. R. Schubert. On finding the disc of minimum radius containing a given set of points. *Mathematics of Computation*, 21(100):712–714, 1967.

[5] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and applications*. Wiley, 1979.

[6] F. Behrend. Über die kleinste umbeschriebene und die größte einbeschriebene Ellipse eines konvexen Bereichs. *Mathematische Annalen*, 115:379–411, 1938.

[7] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[8] M. Berger. *Geometry (vols. 1–2)*. Springer-Verlag, 1987.

[9] B. K. Bhattacharya and G. T. Toussaint. On geometric algorithms that use the furthest-point Voronoi diagram. In G. T. Toussaint, editor, *Computational Geometry*, pages 43–61. North-Holland, Amsterdam, Netherlands, 1985.

[10] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler. *Oriented Matroids*. Cambridge University Press, 2nd edition, 1999.

[11] C. Blatter. *Analysis I*. Springer Verlag, Berlin, 1991. ISBN 3-540-54239-6.

[12] L. M. Blumenthal and G. E. Wahlin. On the spherical surface of smallest radius enclosing a bounded subset of n-dimensional Euclidean space. *American Mathematical Society Bulletin*, 47:771–777, 1941.

[13] J.-D. Boissonnat and M. Yvinec. *Algorithmic geometry*. Cambridge University Press, New York, 1998.

[14] O. Bousquet, O. Chapelle, S. Mukherjee, and V. Vapnik. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002.

[15] Y. Bulatov, S. Jambawalikar, P. Kumar, and S. Sethia. Hand recognition using geometric classifiers, 2002. Abstract of presentation for the DIMACS Workshop on Computational Geometry (Rutgers University).

[16] The CGAL reference manual, August 2001. Release 2.3.

[17] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, B. B. Monagan, and S. M. Watt. *Maple V Library Reference Manual*. Springer-Verlag, 1991.

[18] G. Chrystal. On the problem to construct the minimum circle enclosing n given points in the plane. *Proceedings of the Edinburgh Mathematical Society, Third Meeting*, 1:30ff, 1885.

[19] V. Chvátal. *Linear Programming*. W. H. Freeman, New York, NY, 1983.

[20] K. L. Clarkson. Las Vegas algorithms for linear and integer programming. *J. ACM*, 42:488–499, 1995.

[21] L. Danzer, D. Laugwitz, and H. Lenz. Über das Löwnersche Ellipsoid und sein Analogon unter den einem Eikörper eingeschriebenen Ellipsoiden. *Archiv der Mathematik*, 8:214–219, 1957.

[22] M. E. Dyer. A class of convex programs with applications to computational geometry. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 9–15, 1992.

[23] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, West Germany, 1987.

[24] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degeneratcases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.

[25] D. Jack Elzinga and D. W. Hearn. The minimum covering sphere problem. *Management Science*, 19(1):94–104, September 1972.

[26] V. Fedorov. *Theory of optimal experiments*. Academic Press, Inc., 1972.

[27] K. Fischer. Approximate smallest enclosing ellipsoid—an implementation for CGAL. Technical report, ETH Zürich, 2003.

[28] K. Fischer. Fast and robust miniball of balls. Technical report, ETH Zürich, 2003.

[29] K. Fischer, B. Gärtner, and M. Kutz. Fast smallest-enclosing-ball computation in high dimensions. In *Proc. 11th Annu. European Sympos. Algorithms*, volume 2832 of *Lecture Notes Comput. Sci.*, pages 630–641. Springer-Verlag, 2003.

[30] K. Fischer and M. Kutz. Fast and robust miniball computation in high dimensions. Technical report, ETH Zürich, 2005, to appear.

[31] B. Gärtner. *Randomized optimization by Simplex-type methods*. PhD thesis, Freie Universität Berlin, 1995.

[32] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM J. Comput.*, 24(5):1018–1035, 1995.

[33] B. Gärtner. Fast and robust smallest enclosing balls. In *Proc. 7th Annual European Symposium on Algorithms (ESA)*, volume 1643 of *Lecture Notes Comput. Sci.*, pages 325–338. Springer-Verlag, 1999.

[34] B. Gärtner. Combinatorial structure in convex programs. Manuscript, see http://www.inf.ethz.ch/personal/gaertner/, 2001.

[35] B. Gärtner. The random facet simplex algorithm on combinatorial cubes. *Random Struct. Algorithms*, 20(3):353–381, 2002.

[36] B. Gärtner. Strong LP-type problems. Manuscript, see http://www.inf.ethz.ch/personal/gaertner/, September 2002.

[37] B. Gärtner. Randomized algorithms—an introduction through unique sink orientations. lecture notes, 2004.

[38] B. Gärtner and S. Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 110–118, 2000.

[39] B. Gärtner and E. Welzl. Linear programming—randomization and abstract frameworks. In *Proc. 13th Sympos. Theoret. Aspects Comput. Sci.*, volume 1046 of *Lecture Notes Comput. Sci.*, pages 669–687. Springer-Verlag, 1996.

[40] B. Gärtner and E. Welzl. Explicit and implicit enforcing—randomized optimization. In *Computational Discrete Mathematics*, volume 2122 of *LNCS*, pages 25–46. Springer-Verlag, 2001.

[41] I. Geigenfeind. Implementation of Miniball-of-Balls using USO-algorithms. Semester thesis, ETH Zürich, 2004.

[42] A. Goel, P. Indyk, and K. Varadarajan. Reductions among high dimensional proximity problems. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 769–778. Society for Industrial and Applied Mathematics, 2001.

[43] T. Granlund. *GMP, the GNU Multiple Precision Arithmetic Library*, 2.0.2 edition, 1996. http://www.swox.com/gmp/.

[44] G.J. Hamlin, R.B. Kelley, and J. Tornero. Spherical-object representation and fast distance computation for robotic applications. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1602–1608, 1991.

[45] G.J. Hamlin, R.B. Kelley, and J. Tornero. Efficient distance calculation using the spherically-extended polytope (s-tope) model. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 2502–2507, 1992.

[46] T. H. Hopp and C. P. Reeve. An algorithm for computing the minimum covering sphere in any dimension. Technical Report NISTIR 5831, National Institute of Standards and Technology, 1996.

[47] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

[48] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, 15(3):179–210, 1996.

[49] S. K. Jacobsen. An algorithm for the minimax weber problem. *European Journal of Operational Research*, 6(144-148), 1981.

[50] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.

[51] V. Karamcheti, C. Li, I. Pechtchanski, and C. Yap. A core library for robust numeric and geometric computation. In *Proceedings of the fifteenth annual symposium on Computational geometry*, pages 351–359. ACM Press, 1999.

[52] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[53] L. Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21(2):307–320, 1996.

[54] L. Khachiyan and M. J. Todd. On the complexity of approximating the maximum inscribed ellipsoid for a polytope. *Math. Program.*, 61:137–159, 1993.

[55] L. G. Khachiyan. Polynomial algorithm in linear programming. *U.S.S.R. Comput. Math. and Math. Phys.*, 20:53–72, 1980.

[56] P. Kumar, J. S. B. Mitchell, and E. A. Yıldırım. Approximate minimum enclosing balls in high dimensions using core-sets. *J. Exp. Algorithmics*, 8:1.1, 2003.

[57] R. Kurniawati, J. S. Jin, and J. A. Shepherd. The SS^+-tree: an improved index structure for similarity searches in a high-dimensional feature space. In *Proc. 5th Storage and Retrieval for Image and Video Databases SPIE*, volume 3022, pages 110–120, 1997.

[58] C. L. Lawson. The smallest covering cone or sphere. *SIAM Review*, 7(3):415–416, July 1965.

[59] J. Matoušek. Lower bounds for a subexponential optimization algorithm. *Random Struct. Algorithms*, 5(4):591–608, 1994.

[60] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 1–8, 1992.

[61] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.

[62] N. Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983.

[63] N. Megiddo. On the ball spanned by balls. *Discrete Comput. Geom.*, 4:605–610, 1989.

[64] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, UK, 2000.

[65] Y. Menguy. *Optimisation quadratique et géométrique de problèmes de dosimétrie inverse*. PhD thesis, Université Joseph Fourier Grenoble, 1996.

[66] K. G. Murty. Note on Bard-type scheme for solving the complementarity problem. In *Operations Research*, volume 11, pages 123–130, 1974.

[67] A. L. Peressini, F. E. Sullivan, and J. J. Uhl. *The Mathematics of Nonlinear Programming*. Undergraduate Texts in Mathematics. Springer-Verlag, 1988.

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

[69] V. T. Rajan. Optimality of the Delaunay triangulation in R^d. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 357–363, 1991.

[70] V. T. Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. *Discrete Comput. Geom.*, 12:189–202, 1994.

[71] R.T. Rockafellar. *Convex Analysis.* Princeton University press, 1970.

[72] S. Schönherr. *Quadratic programming in geometric optimization: theory, implementation, and applications.* PhD thesis, ETH Zürich, 2002.

[73] I. Schurr. *Unique sink orientations of cubes.* PhD thesis, ETH Zürich, 2004.

[74] R. Seidel. Linear programming and convex hulls made easy. In *SCG '90: Proceedings of the sixth annual symposium on Computational geometry*, pages 211–215, New York, NY, USA, 1990. ACM Press.

[75] R. Seidel. Personal communication, 1997.

[76] M. I. Shamos and D. Hoey. Closest-point problems. In *Proc. 16th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 151–162, 1975.

[77] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *STACS '92: Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, pages 569–579. Springer-Verlag, 1992.

[78] S. D. Silvey. Discussion of a paper by H. P. Wynn. *Journal of the Royal Statistical Society, Series B*, 34:174–175, 1972.

[79] S. D. Silvey and D. M. Titterington. A geometric approach to optimal design theory. *Biometrika*, 60(1):21–32, April 1973.

[80] M. Sipser. *Introduction to the theory of computation.* PWS Publishing Company, 20 Park Plaza, Boston, MA 02116, 2001.

[81] S. Skyum. A simple algorithm for computing the smallest enclosing circle˙ *Inform. Process. Lett.*, 37:121–125, 1991.

[82] A. Stickney and L. Watson. Digraph models of Bard-type algorithms for the linear complementary problem. *Mathematics of Operations Research*, 3:322–333, 1978.

[83] J. J. Sylvester. A question on the geometry of situation. *Quarterly Journal of Pure and Applied Mathematics (formerly published as Mesenger of Mathematics)*, 1:79, 1857.

[84] J. J. Sylvester. On Poncelet's approximate linear valuation of surd forms. *Philosophical Magazine*, 20 (fourth series):203–222, 1860.

[85] T. Szabó and E. Welzl. Unique sink orientations of cubes. In *Proc. 42nd annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 547–555, 2001.

[86] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes Comput. Sci.*, pages 359–370. Springer-Verlag, 1991.

[87] H.-M. Will. *Computation of additively weighted Voronoi cells for applications in molecular biology.* PhD thesis, ETH Zürich, 1998.

[88] S. Xu, R.M. Freund, and J. Sun. Solution methodologies for the smallest enclosing circle problem. *Computational Optimization and Applications*, 25:283–292, 2003.

[89] G. Zhou, K.-C. Toh, and J. Sun. Efficient algorithms for the smallest enclosing ball problem. To appear in *Computational Optimization and Applications*, see also http://www.bschool.nus.edu.sg/depart/ds/sunjiehomepage/, accepted in 2004.

Curriculum Vitae

Kaspar Fischer
born on May 23, 1975
Basel, Switzerland

1986–1994
High school
Holbein-Gymnasium, Basel
Maturität Typus D (Sprachen)

1995–2001
Studies at ETH Zürich, Switzerland
Master of Computer Science

2001–2002
Pre-doc studies at ETH Zürich, Switzerland
Graduate program *Combinatorics, Geometry, and Computation*

2002–2005
Ph.D. student at ETH Zürich, Switzerland
Theoretical Computer Science