# A dual Simplex-type algorithm for the smallest enclosing ball of balls and related problems

Marta Cavaleiro[*]        Farid Alizadeh[†]

Tuesday 4th December, 2018

## Abstract

We define the notion of infimum of a set of points with respect to the second order cone. This problem can be showed to be equivalent to the minimum ball containing a set of balls problem and to the maximum intersecting ball problem, as well as others. We present a dual algorithm which can be viewed as an extension of the simplex method to solve this problem.

**Keywords:** Smallest enclosing ball, simplex-type methods, second order cone programming, computational geometry

## 1   Introduction

Let $\mathcal{Q}$ denote the second-order cone, $\mathcal{Q} := \{x := (x_0; \overline{x}) \in \mathbb{R}^n : \|\overline{x}\|_2 \leq x_0\}$. Given a set of points $\mathcal{P} = \{p_1, ..., p_m\} \subset \mathbb{R}^n$, we consider the problem defined as follows

$$\text{Inf}_{\mathcal{Q}}(\mathcal{P}) := \max_{x} \quad \langle e_1, x \rangle \\ \text{s.t.} \quad x \preceq_{\mathcal{Q}} p_i, \ i = 1, ..., m \tag{P}$$

with $e_1 = (1, 0, ..., 0)$. We define this problem as the *infimum of $\mathcal{P}$ with respect to $\mathcal{Q}$* (given its resemblance with the problem of finding the minimum number of a set of numbers with a linear program). Throughout the article we shall refer to this problem as $\text{Inf}_{\mathcal{Q}}(\mathcal{P})$, or simply (P).

Denote by $x^*$ the optimal solution to (P). One possible geometric interpretation of the problem in question is to find $x^*$ as "high" as possible (when height is defined as the value of $x_0$) such that $x^* + \mathcal{Q}$ covers all points of set $\mathcal{P}$, in the sense that they either fall inside or on the boundary of $x^* + \mathcal{Q}$ (Figure 1). But more interestingly, it is possible to prove that (P) is in fact equivalent to several relevant problems in computational geometry, such as the smallest enclosing ball of balls or the largest intersecting ball.

---

[*]Rutgers University, MSIS Department & RUTCOR, 100 Rockafellar Rd, Piscataway, NJ 08854. `marta.cavaleiro@rutgers.edu`

[†]Rutgers University, MSIS Department & RUTCOR, 100 Rockafellar Rd, Piscataway, NJ 08854. `farid.alizadeh@rutgers.edu`
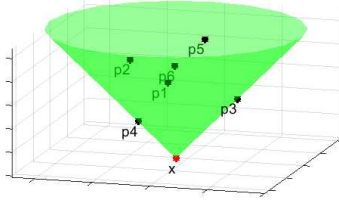
Figure 1: Geometric interpretation of the *infimum of with respect to* $\mathcal{Q}$.

Problem (P) is a Second-Order Cone Program (SOCP) and so it can be solved in polynomial time using interior point methods, such as the primal-dual path following algorithms [20, 21].These algorithms generate interior-points for the primal and dual problems that follow the so-called *central path*, which converges to a primal-dual optimal solution in the limit. Each iteration, in terms of computational work, basically consists on the solution of a linear system to compute the search direction resulting from the application of Newton's method to the KKT conditions of a modified (P) with a *logarithmic barrier function*. The matrix in question is positive definite (see e.g. [1]), and usually its Cholesky factorization is used. In terms of our problem, the matrix has size $\mathfrak{n} \times \mathfrak{n}$ (note that the standard form cone LP is the dual), and would take $\mathcal{O}(\mathfrak{m}\mathfrak{n}^2)$ basic arithmetic operations to be computed [25] resulting in a iteration with $\mathcal{O}(\mathfrak{m}\mathfrak{n}^2 + \mathfrak{n}^3)$ computational complexity. Therefore, computing a solution with an error $\epsilon$ with an interior-point method would have an overall complexity of $\mathcal{O}((\mathfrak{m} + \mathfrak{n})\mathfrak{n}^2\sqrt{\mathfrak{m}}\log \frac{1}{\epsilon}))$, though in practice it has been observed that the number of iterations is often very small and independent on $\mathfrak{m}$. Interior point methods have been used to solve (P) in particular in [14, 25]. Specifically, in [25] an algorithm that computes a $(1 + \epsilon)$-approximation in $\mathcal{O}(\frac{\mathfrak{m}\mathfrak{n}}{\epsilon} + \frac{1}{\epsilon^{4.5}}\log \frac{1}{\epsilon})$ time is presented.

In this paper we introduce a dual algorithm for (P) that mechanically is analogous to the Simplex method for Linear Programming. Each (major) iteration of our algorithm starts with a primal solution corresponding to a *dual feasible basic solution*, but then, instead of a line search, our algorithm will perform a sequence of *exact curve searches* until it arrives to a new *dual feasible basic solution* with a better objective function value. We will define the notion of *basis* as well as *dual feasible basic solution* applied to our problem, which we will rename as *support set* and *dual feasible S-pair*, respectively. Mechanically speaking, our algorithm shares similarities with the dual active set algorithm for strictly convex QPs by Goldfarb and Idnani in [10], and with the work of Dearing and Zeck who proposed a dual simplex method for the minimum enclosing ball of points in [5].

Besides providing an exact solution, one advantage that a dual simplex algorithm has over interior point methods is that, after solving the problem, if it suffers small changes (e.g. adding an extra constraint), the dual simplex method will usually require a small number of iterations to calculate the new solution when it starts with the original solution. Another advantage of simplex-type algorithms in general is that they generate *basic* solutions. For instance in the case of the minimum enclosing ball of balls (and in particular of points too), that will be able to tell us which input balls (points) determine the smallest enclosing ball. On the other hand, we do not have any overall polynomial complexity guarantees for the algorithm.

The paper is organized as follows. In section 2 we show that problem (P) is equivalent to relevant problems in computational geometry involving hyperspheres.

Section 3 presents important theoretical background to the algorithm, such as duality results and the definition of *support set*. The dual simplex algorithm for (P) is then introduced in section 4. We then briefly explain the implementation details in section 5 and show some computational results in section 6.

## 2    Equivalent geometric problems

We now show that problem (P) is equivalent to several relevant problems in computational geometry such as the smallest enclosing ball of balls. For what follows, denote by $B(c, r)$ a ball with the Euclidean norm with center at $c \in \mathbb{R}^{n-1}$ and radius $r \geq 0$.

**The smallest enclosing ball problem.** The classical and well studied problem of enclosing a set of Euclidean balls with an Euclidean ball of smallest radius can be reduced to (P) as a consequence of the fact that $B(c_1, r_1) \cap B(c_2, r_2) \neq \emptyset$ if and only if $\|c_2 - c_1\| \leq r_2 + r_1$. Thus, the problem of enclosing a set $\mathcal{B}$ of balls $B(c_i, r_i)$, $i = 1, ..., m$ with a ball $B(c, r)$ of minimum radius can be solved by (P) considering $\mathcal{P} = \{(-r_i; c_i), i = 1, ..., m\}$. The optimal ball will then be given by the center $c = \overline{x}^*$ and radius $r = -x_0^*$. When $r_i = 0$ for all $i$, the problem reduces to the smallest enclosing ball of points. Figure 2 illustrates this equivalence: the minimum enclosing ball of set $\mathcal{B}$ is the intersection of the cone $x^* + \mathcal{Q}$ with the plane $x_0 = 0$.

The smallest enclosing ball of balls, and in particular of points, is a classical problem in computational geometry. This problem has been extensively studied, specially from a combinatorial point of view, in particular in the *LP-type* framework, see e.g. [6, 8, 16, 17, 23]. In particular, the minimum enclosing ball of points can easily be converted in a Quadratic Program (QP) and solved using off-the-shelf QP solvers. Gärtner and Schönherr [9] developed a generalization of the simplex method for QP with the goal of targeting geometric QPs, with one of the main applications being the MB problem, while later, Fischer and Gärtner [8] proposed an algorithm with a pivoting scheme resembling the simplex method for LP based on previous ideas from [12]. Using related ideas, Dearing and Zeck [5] developed a dual algorithm for the MB problem. This algorithm was further improved in [4]. Several approximation algorithms have also been developed focusing on finding an $\epsilon$-*core set*, [3], that is a subset of $\mathcal{S} \subset \mathcal{P}$ that has the property that the smallest ball containing $\mathcal{S}$ once expanded by $1 + \epsilon$ covers $\mathcal{P}$. A surprising fact is the existence of an $\epsilon$-core set of size at most $\lceil \frac{1}{\epsilon} \rceil$, independent of the dimension $n$, for any point set $\mathcal{P} \subset \mathbb{R}^n$, [14, 2]. Several algorithms focused on finding $\epsilon$-core sets have been proposed [2, 3, 14, 15, 22, 24].



(a) *Cone view*

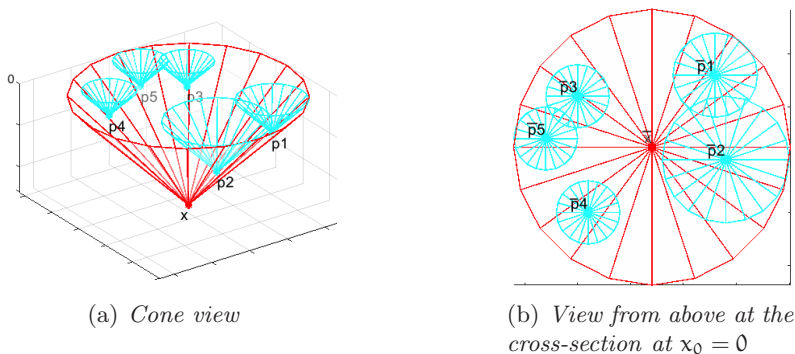(b) *View from above at the cross-section at $x_0 = 0$*

Figure 2: The smallest enclosing ball of a set of balls.

**The smallest intersecting ball and the largest enclosed ball problems.** Consider now the *smallest intersecting ball problem* of finding the Euclidean ball with smallest radius that intersects all balls $B(c_i, r_i)$, $i = 1, ..., m$. This problem also reduces to (P) given that $B(c_1, r_1) \cap B(c_2, r_2) \neq \emptyset$ if and only if $\|c_2 - c_1\| \leq r_2 + r_1$. Thus, considering $\mathcal{P} = \{(r_i; c_i), i = 1, ..., m\}$ in (P), the smallest ball that intersects all balls has center $c = \overline{x}^*$ and radius $r = -x_0^*$. Figure 3 illustrates the equivalence. Balls $B(c_i, r_i)$, $i = 1, ..., m$, may all intersect, and so the smallest intersecting ball could be considered any point in the intersection. In such case, the smallest intersecting ball problem then looses its interest. In fact, when all balls intersect, the solution is such that $x_0^* > 0$. This solution however is not meaningless, in fact, it is not difficult to see that, when $\bigcap B(c_i, r_i) \neq \emptyset$, ball $B(\overline{x}, x_0^*)$ is the largest radius ball that is enclosed in the intersection of the balls. We shall refer to this problem as the *largest enclosed ball problem* (see Figure 4). For previous work on this problem see [18, 19] and the references therein.
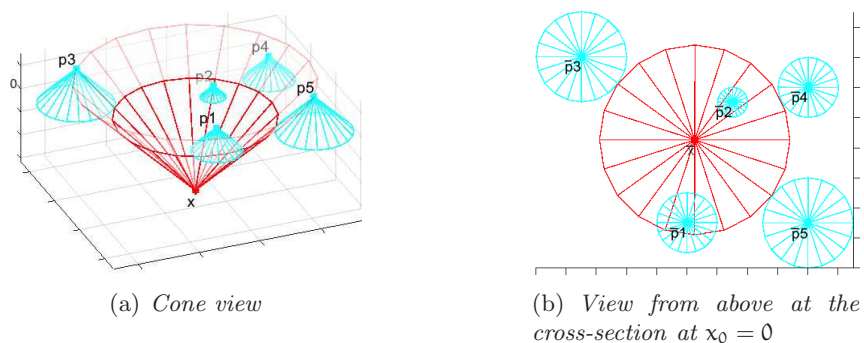


(a) *Cone view*  (b) *View from above at the cross-section at $x_0 = 0$*

Figure 3: The smallest intersecting ball of a set of balls.



(a) *Cone view*  (b) *View from below at the cross-section at $x_0 = 0$*
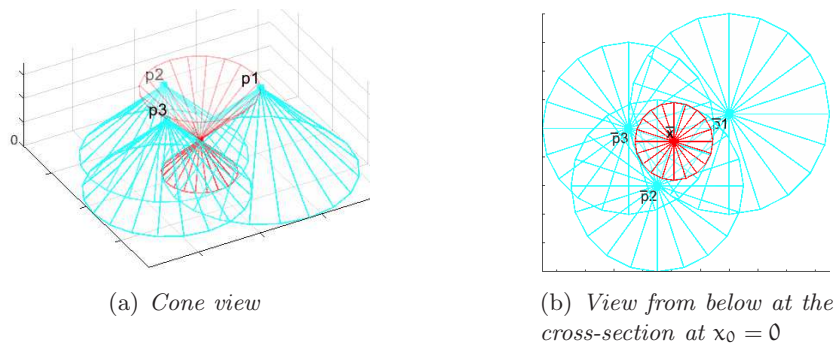
Figure 4: The largest enclosed ball in a set of balls.

**The smallest intersecting and enclosed ball problem.** From the previous two sections, we conclude that the problem of finding a ball with smallest radius that simultaneously encloses balls $B(c_i, r_i)$, $i = 1, ..., m_1$, and intersects balls $B(c_j, r_j)$, $j = 1, ..., m_2$, can be solved by considering $\mathcal{P} = \{(-r_i; c_i), i = 1, ..., m_1\} \cup \{(r_j; c_j), j = 1, ..., m_2\}$. The optimal ball will then have center $\overline{x}^*$ and radius $-x_0^*$.

4

# 3  Preliminaries

Before we proceed, let us introduce/review some notation.

| | |
|---|---|
| $\mathcal{P}$ | $\{p_1, ..., p_m\}$ |
| $p_i = (p_{i0}; \overline{p}_i)$ | a point from $\mathcal{P}$ |
| $\overline{\mathcal{P}}$ | set $\{\overline{p}_1, ..., \overline{p}_m\}$ |
| $x = (x_0; \overline{x})$ | primal variables |
| $y_i = (y_{i0}; \overline{y}_i)$ | dual variables, $i = 1, ..., m$ |

## 3.1  Duality and optimality conditions

It is easy to see that the solution to (P) always exists and that it is unique. Moreover, it is possible to prove the following:

**Lemma 1.** *The solution to* $\mathrm{Inf}_Q(\mathcal{P})$ *is* $p_k \in \mathcal{P}$, *for some* $k = 1, ..., m$, *if and only if* $p_k \preceq_Q p_i$ *for all* $i = 1, ..., m$ *and* $p_{k0} = \min\{p_{i0}, i = 1, ..., m\}$.

The dual problem of $\mathrm{Inf}_Q(\mathcal{P})$ is

$$
\begin{aligned}
\min_y \quad & \sum_{i=1}^{m} \langle p_i, y_i \rangle \\
\text{s.t.} \quad & \sum_{i=1}^{m} y_{i0} = 1 \\
& \sum_{i=1}^{m} \overline{y}_i = 0 \\
& y_i \succeq_Q 0, \quad i = 1, ..., m.
\end{aligned}
\tag{D}
$$

The solution to (D) may not be unique.

**Lemma 2.** *Both primal problem (P) and dual problem (D) are strictly feasible; i.e. there exists a primal-feasible vector* $x$ *such that* $x \prec_Q p_i$ *for all* $i = 1, ..., m$, *and there exist dual-feasible* $y_1, ..., y_m$ *such that* $y_i \succ_Q 0$ *for all* $i = 1, ..., m$.

Since both primal and dual problems are strictly feasible, the duality gap is zero, that is, strong duality holds and the Karush-Kuhn-Tucker conditions are also sufficient [1, p. 25], as Thoerem 3 states.

**Theorem 3** (Optimality conditions)**.** *Let* $x^*$ *and* $y_i^*$, $i = 1, ..., m$ *be any points in* $\mathbb{R}^n$. *The pair* $(x^*, \{y_i^*\}_{i=1,...,m})$ *is primal-dual optimal if and only if*

- *primal feasibility:* $x^* \preceq_Q p_i$, $i = 1, ..., m$;
- *dual feasibility:* $\sum_{i=1}^{m} y_i^* = e_1$ *and* $y_i^* \succeq_Q 0$, $i = 1, ..., m$;
- *complementary slackness:* $\langle p_i - x^*, y_i^* \rangle = 0$, $i = 1, ..., m$.

The complementary slackness conditions imply

- if $y_i^* \succ_Q 0$ then $x^* = p_i$ (which can happen for a single $i$, and in that case $y_i^* = e_1$ and $y_j^* = 0$ for all $j \neq i$);
- if $x^* \succ_Q p_i$ then $y_i^* = 0$;

- if $p_i - x^* \in \partial \mathcal{Q}$ and $y_i^* \in \partial \mathcal{Q}$ then

$$\overline{y}_i^* = \frac{y_{i0}^*}{p_{i0} - x_0^*} \left( \overline{x}^* - \overline{p}_i \right) = y_{i0}^* \frac{\overline{x}^* - \overline{p}_i}{\|\overline{p}_i - \overline{x}^*\|}. \tag{1}$$

A consequence of Theorem 3 is the following characterization of optimality:

**Theorem 4.** $x^*$ *is the optimal solution to problem (P) if and only if* $x^* \preceq_Q p_i$ *for all* $i = 1, ..., m$, *and*

$$\overline{x}^* \in \operatorname{conv} \left( \{ \overline{p}_i : \|\overline{p}_i - \overline{x}^*\| = p_{i0} - x_0^* \} \right). \tag{2}$$

*Proof.* The theorem follows trivially from the optimality conditions in the case when $x^* = p_k$ for some $p_k \in \mathcal{P}$. Consider then, that is not the case.

First, consider $x^*$ is optimal. Let $\mathcal{I} = \{i : \|\overline{p}_i - \overline{x}^*\| = p_{i0} - x_0^*\}$, and $y_i^*$, $i = 1, ..., m$ be an optimal dual solution. From complementary slackness and dual feasibility we conclude

$$0 = \sum_{i \in \mathcal{I}} \overline{y}_i^* = \sum_{i \in \mathcal{I}} \frac{y_{i0}^*}{p_{i0} - x_0^*} \left( \overline{x}^* - \overline{p}_i \right)$$

since $y_i^* = 0$ for $i \notin \mathcal{I}$. From the previous equation we have

$$\overline{x}^* = \sum_{i \in \mathcal{I}} \alpha_i \overline{p}_i \quad \text{with} \quad \alpha_i = \frac{y_{i0}^*/p_{i0} - x_0^*}{\sum_{i \in \mathcal{I}} y_{i0}^*/p_{i0} - x_0^*}.$$

Since $\sum_i y_{i0}^* = 1$ there must be at least one $\alpha_i \neq 0$, and $y_i^* \succeq_Q 0$ implies $\alpha_i \geq 0$ for all $i$. Because $\sum_{i=1}^m \alpha_i = 1$ we conclude that (2) holds.

Conversely, suppose we have $x^*$ that is primal feasible and such that $\overline{x}^*$ satisfies (2). Then

$$\overline{x}^* = \sum_{i \in \mathcal{I}} \alpha_i \overline{p}_i, \quad \text{with} \quad \alpha_i \geq 0, \ i \in \mathcal{I}, \quad \text{and} \quad \sum_{i \in \mathcal{I}} \alpha_i = 1.$$

We will now build a dual solution that is feasible and together with $x^*$ satisfy complementary slackness. Consider $y_{i0}$, $i = 1, ..., m$, such that

$$\alpha_i = \frac{y_{i0}/p_{i0} - x_0^*}{\sum_j y_{i0}/p_{i0} - x_0^*}, \tag{3}$$

Equations (3) together with $\sum_i y_{i0} = 1$ give the following linear systems of equations

$$\begin{cases} \alpha_1 (z_1 + \ldots + z_m) = z_1 \\ \vdots \\ \alpha_m (z_1 + \ldots + z_m) = z_m \\ \sigma_1 z_1 + \ldots + \sigma_m z_m = 1 \end{cases}$$

with $z_i = y_{i0}/(p_{i0} - x_0^*)$ and $\sigma_j = p_{i0} - x_0^*$. The last equation implies

$$\sigma_1 \alpha_1 (z_1 + \ldots + z_m) + \ldots + \sigma_m \alpha_m (z_1 + \ldots + z_m) = 1$$

$$(z_1 + \ldots + z_m) = \frac{1}{\sigma_1 \alpha_1 + \ldots + \sigma_m \alpha_m}.$$

Therefore
$$z_i = \frac{\alpha_i}{\sigma_1 \alpha_1 + \ldots + \sigma_m \alpha_m},$$
that is,
$$y_{i0} = \frac{\alpha_i (p_{i0} - x_0^*)}{\sum_j \alpha_j (p_{j0} - x_0^*)}.$$

We have that $y_{i0} \geq 0$ for $i \in \mathcal{I}$. Setting $y_{i0} = 0$ for $i \notin \mathcal{I}$, and considering
$$\overline{y}_i = \frac{y_{i0}}{p_{i0} - x_0^*} (\overline{x} - \overline{p}_i),$$

for all $i = 1, \ldots, m$, we have that $(y_{i0}; \overline{y}_i)$ is feasible for the dual problem (D). Since $x^*$ is primal feasible and $\|\overline{p}_i - \overline{x}^*\| = p_{i0} - x_0^*$ for $i \in \mathcal{I}$, we have that, together with $y_i$, complementary slackness is satisfied, thus $x^*$ is optimal for (P). $\qquad\square$

We now present two insightful conclusions from the proof of Theorem 4.

**Observation 5.** The optimal solution $x^*$ to (P) is such that
$$\overline{x} = \mathrm{ri}\,\mathrm{conv}(\{\overline{p}_i : y_{i0}^* > 0\}). \tag{4}$$

Note that $y_{i0}^* > 0$ cannot be replaced by $y_i^* \in \partial \mathcal{Q}$ (when $\overline{x}^* = \overline{p}_k$ we have $y_k^* = e_1 \notin \partial \mathcal{Q}$).

Another conclusion from Theorem 4, which will be at the core of our algorithm, is the following corollary.

**Corollary 6.** *Consider* $x$, *not necessarily primal feasible, that satisfies*
$$\overline{x} \in \mathrm{aff}(\{\overline{p}_i : i \in \mathcal{I}\}), \quad \text{for } \mathcal{I} = \{i : \|\overline{p}_i - \overline{x}\| = p_{i0} - x_0\}. \tag{5}$$

*Let* $\alpha_1, \ldots, \alpha_m$ *be the coefficients of the affine combination. There exists a dual solution given by*
$$y_i = 0, \text{ for } i \notin \mathcal{I},$$
$$y_{i0} = \frac{\alpha_i (p_{i0} - x_0)}{\sum_j \alpha_j (p_{j0} - x_0)} \text{ and } \overline{y}_i = \frac{y_{i0}}{p_{i0} - x_0} (\overline{x} - \overline{p}_i), \text{ for } i \in \mathcal{I},$$

*that satisfies the dual constraint* $\sum_{i=1}^m y_i = e_1$, *and, together with* $x$, *the complementary slackness conditions.*

Note that, unless $\{\overline{p}_i : i \in \mathcal{I}\}$ is affinely independent, the coefficients of the affine combination are not unique and therefore $x$ may correspond to more than one such dual solution. If, additionally, $\overline{x} \in \mathrm{conv}(\{\overline{p}_i : i \in \mathcal{I}\})$, then $y_{i0} \geq 0$ for all $i$, and so $x$ corresponds to a dual feasible solution.

Finally, we can use the result from Theorem 4 to easily calculate algebraically the solution to $\mathrm{Inf}_\mathcal{Q}(\mathcal{P})$ when $\mathcal{P}$ only has two points.

**Theorem 7.** *The solution to* $\mathrm{Inf}_\mathcal{Q}(\{p_1, p_2\})$ *is given by*
$$x_0^* = \min\left(p_{10}, p_{20}, \frac{p_{10} + p_{20} - \|\overline{p}_1 - \overline{p}_2\|}{2}\right), \overline{x}^* = \frac{(p_{10} - x_0^*)\overline{p}_2 + (p_{20} - x_0^*)\overline{p}_1}{(p_{10} - x_0^*) + (p_{20} - x_0^*)}. \tag{6}$$

## 3.2 The definition of *support set* and *S-pair*

We now define *support set* for the $\mathrm{Inf}_Q(\mathcal{P})$ problem the same way a basis is defined for an LP-type problem [7].

**Definition 8** (Support set)**.** A subset $\mathcal{S}$ is called a *support set* if no proper subset $\mathcal{S}'$ of $\mathcal{S}$ is such that $\mathrm{Inf}_Q(\mathcal{S}') = \mathrm{Inf}_Q(\mathcal{S})$. A subset $\mathcal{S} \subseteq \mathcal{P}$ is said to be an *optimal support set* if $\mathcal{S}$ is a support set and $\mathrm{Inf}_Q(\mathcal{S}) = \mathrm{Inf}_Q(\mathcal{P})$.

Note that an optimal support set may not be unique.

**Definition 9** (Dual feasible S-pair)**.** Let $\mathcal{S} \subseteq \mathcal{P}$ and $x$ a vector. We say $(\mathcal{S}, x)$ is a *dual feasible S-pair* if

(a) the points of $\mathcal{S}$ lie on the boundary of $x + Q$, that is,

$$\|\overline{p}_i - \overline{x}\| = p_{i0} - x_0, \quad \forall\, p_i \in \mathcal{S};$$

(b) $\overline{x} \in \mathrm{ri\, conv}(\overline{\mathcal{S}})$;

(c) $\overline{\mathcal{S}}$ is affinely independent.

Comparing with LP, *support set* is analogous to the definition of *basis*, and *dual feasible S-pair* to the definition of *dual basic feasible solution*.

Using Theorem 4, it is now possible to prove the following equivalence.

**Theorem 10.** $(\mathcal{S}, x)$ *is a dual feasible S-pair iff $\mathcal{S}$ is a support set and $x$ is the solution to* $\mathrm{Inf}_Q(\mathcal{S})$.

As a consequence of the previous theorem, a support set has at least $1$ point and at most $n$ points. This is, in fact, a well known result: the minimum enclosing ball of balls in $\mathbb{R}^{n-1}$ is defined by at most $n$ balls. Another consequence, is the fact that if $(\mathcal{S}, x)$ is a dual feasible S-pair and $x$ is primal feasible, then $x$ solves $\mathrm{Inf}_Q(\mathcal{P})$.

Finally, Theorem 11 is a direct consequence of the definition of support set.

**Theorem 11.** *Let $\mathcal{S}$ be a support set. If a point $p^*$ is infeasible to* $\mathrm{Inf}_Q(\mathcal{S})$*, then $p^*$ belongs to an optimal support set for problem* $\mathrm{Inf}_Q(\mathcal{S} \cup \{p^*\})$.

## 4 Algorithm description

We now outline the basic framework of the algorithm:

**Initialization:** Suppose a dual feasible S-pair $(\mathcal{S}^0, x^0)$, $\mathcal{S}^0 \subseteq \mathcal{P}$, is given.

**Loop:** For $j = 0, 1, 2, \dots$, do:

(a) If $x^j$ is primal feasible, stop - $x^j$ is the optimal solution of (P).

(b) Else, get a $p^* \in \mathcal{P}$ corresponding to a chosen violated constraint.

(c) Obtain a new dual feasible S-pair $(\hat{\mathcal{S}} \cup \{p^*\}, \hat{x})$, for $\hat{\mathcal{S}} \subseteq \mathcal{S}^j$ and $\hat{x}_0 < x_0^j$;
    Set $(\mathcal{S}^{j+1}, x^{j+1}) \leftarrow (\hat{\mathcal{S}} \cup \{p^*\}, \hat{x})$.

Note that, as an initial solution, one can simply pick any point $p \in \mathcal{P}$, and consider the dual feasible S-pair $(\{p\}, p)$.

The core of the algorithm is step (c) which will consist on a sequence of *curve searches* that keep dual feasibility. Assume, for now, that $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely independent. Since, at each iteration $j$, we start with a dual feasible S-pair $(\mathcal{S}^j, x^j)$, the set of

constraints associated with $\mathcal{S}^j$ are active at $x^j$. As we shall see, the set of points $x$ that correspond to dual feasible solutions and that keep those constraints active define a curve (see section 4.1). A *curve search* basically consists on the following problem: starting at $x^j$, we "move" on the curve in the direction of decrease of $x_0$ until either dual feasibility is lost or $p^*$ becomes feasible, whichever happens first. As we will see, we will be able to calculate exactly the points where dual feasibility is lost and where $p^*$ becomes feasible, so the *curve search* will be *exact*. This then boils down to calculating what we will define as the *partial step* and the *full step*:

***Partial step***: the maximum step on the curve without violating dual feasibility. The point corresponding to that step, which we will denote by $x^{partial}$, is the first point on the curve where one of the dual variables, which vary non-linearly as we move on the curve, becomes zero.

***Full step***: the minimum step on the curve such that the constraint corresponding to $p^*$ is feasible, that is, when it becomes active. When it exists, we denote that point by $x^{full}$.

If the full step happens before the partial step, then $x^{full}$ corresponds to a dual feasible variable and so it is the optimal solution to the subproblem $\mathrm{Inf}_Q(\mathcal{S}^j \cup \{p^*\})$. In this case, a major iteration is complete, and we go back to Step (a) after setting $(\mathcal{S}^{j+1}, x^{j+1}) \leftarrow (\mathcal{S}^j \cup \{p^*\}, x^{full})$. Otherwise, if the partial step happens first, then the point (say $p_k \in \mathcal{S}^j$) corresponding to the dual variable that was about to become infeasible is dropped from $\mathcal{S}^j$. A new curve search starting at $x^{partial}$ is then performed. Eventually, after at most $|\mathcal{S}^j|$ curve searches where the value of $x_0$ either decreases or stays the same, the optimal solution of $\mathrm{Inf}_Q(\mathcal{S}' \cup \{p^*\})$, for some $\hat{\mathcal{S}} \subseteq \mathcal{S}^j$, with a strictly smaller objective function is found.

A curve search is only performed whenever $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ affinely independent. When that is not the case, a point is immediately dropped from $\mathcal{S}^j$ before a curve search is done (this step can actually be seen as a partial step with zero length). Section 4.1 defines the curve and section 4.2 explains the details of the curve search.

## 4.1   The curve

Consider iteration $j$. Consider

- $(\mathcal{S}^j, x^j)$, a dual feasible $\mathcal{S}$-pair for $\mathrm{Inf}_Q(\mathcal{S}^j)$, with $\mathcal{S}^j \subseteq \mathcal{P}$;
- $p^* \in \mathcal{P}$ corresponding to an infeasible constraint at $x^j$;
- $\overline{\mathcal{S}}^j := \{\overline{p} : p \in \mathcal{S}^j\}$ and $s = |\mathcal{S}^j|$.

The algorithm restricts the search for the next iterate to the set of primal solutions $x$ where dual feasibility and complementary slackness are maintained, that is

$$\left\| \overline{p}_{i_j} - \overline{x} \right\| = p_{j_i 0} - x_0, \quad \forall p_{i_j} \in \mathcal{S}^j, \tag{7}$$

$$\overline{x} \in \mathrm{conv}(\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}). \tag{8}$$

while it decreases the objective function value $x_0$.

In this section we shall see that in general the set of points that satisfy (7) and

$$\overline{x} \in \mathrm{aff}(\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}) \tag{9}$$

constitute a manifold of dimension 1, that is, a curve.

By squaring equations (7) it is easy to prove the following Lemma:

**Lemma 12.** *If $|\mathcal{S}^j| > 1$, define*

$$M = \left[\ \overline{p}_{j_2} - \overline{p}_{j_1} \quad \overline{p}_{j_3} - \overline{p}_{j_1} \quad \cdots \quad \overline{p}_{j_s} - \overline{p}_{j_1}\ \right],$$

$$c = \begin{pmatrix} p_{j_2 0} - p_{j_1 0} \\ \vdots \\ p_{j_s 0} - p_{j_1 0} \end{pmatrix}, \ and\ b = \frac{1}{2} \begin{pmatrix} \left\|\overline{p}_{j_2}\right\|^2 - p_{j_2 0}^2 - \left\|\overline{p}_{j_1}\right\|^2 + p_{j_1 0}^2 \\ \vdots \\ \left\|\overline{p}_{j_s}\right\|^2 - p_{j_s 0}^2 - \left\|\overline{p}_{j_1}\right\|^2 + p_{j_1 0}^2 \end{pmatrix}.$$

*Otherwise, when $|\mathcal{S}^j| = 1$, $M = [\ ]$, $b = [\ ]$, and $c = [\ ]$.*

*Conditions (7) are equivalent to the following conditions*

$$M^{\mathsf{T}}\overline{x} = b + x_0 c, \tag{10a}$$

$$\left\|\overline{p}_{j_1} - \overline{x}\right\|^2 = (p_{j_1 0} - x_0)^2, \tag{10b}$$

$$x_0 \leq \min_{p_{j_i} \in \mathcal{S}^j} \{p_{j_i 0}\}. \tag{10c}$$

For the general case of $|\mathcal{S}^j| > 1$, let us now define the following matrix and vectors

$$M^+ = (M^{\mathsf{T}}M)^{-1}M^{\mathsf{T}},$$

$$u = (M^{\mathsf{T}}M)^{-1}(b - M^{\mathsf{T}}\overline{p}_{j_1}),$$

$$v = (M^{\mathsf{T}}M)^{-1}c,$$

$$w = -M^+(\overline{p}^* - \overline{p}_{j_1}),$$

$$z = (I - MM^+)(\overline{p}^* - \overline{p}_{j_1}).$$

Matrix $M^+$ is the Moore-Penrose inverse, or pseudo-inverse, of $M$. Since $M$ is full column rank, $M^+$ is a left-inverse ($M^+M = I$). When $|\mathcal{S}^j| = 1$, simply consider

$$u = v = w = [\ ] \quad \text{and} \quad z = \overline{p}^* - \overline{p}_{j_1}$$

**Theorem 13.** *Conditions (7) and (9) define points $(x_0; \overline{x})$ such that*

$$\overline{x} = M(u + x_0 v) + \alpha^* z + \overline{p}_{j_1}, \tag{11}$$

*for $x_0 \leq \min_{p_{j_i} \in \mathcal{S}^j} \{p_{j_i 0}\}$, $\alpha^* \in \mathbb{R}$, and $x_0$ and $\alpha^*$ such that*

$$(\alpha^*)^2 \|z\|^2 + \|M(u + x_0 v)\|^2 - (p_{10} - x_0)^2 = 0. \tag{12}$$

*Proof.* From (9) we know that there exist $\alpha_1, \ldots, \alpha_s, \alpha^*$ such that

$$\overline{x} = \sum_{i=1}^{s} \alpha_i \overline{p}_{j_i} + \alpha^* \overline{p}^* \quad \text{and} \quad \sum_{i=1}^{s} \alpha_i + \alpha^* = 1,$$

that is,

$$\overline{x} = M\alpha_{2:s} + \alpha^*(\overline{p}^* - \overline{p}_{j_1}) + \overline{p}_{j_1} \tag{13}$$

with $\alpha_{2:s}$ the vector with $\alpha_2, \ldots, \alpha_s$. Substituting in (10a), we have

$$M^{\mathsf{T}}M\alpha_{2:s} + \alpha^* M^{\mathsf{T}}(\overline{p}^* - \overline{p}_{j_1}) + M^{\mathsf{T}}\overline{p}_{j_1} = b + x_0 c,$$

which yields

$$\alpha_{2:s} = u + x_0 v + \alpha^* w, \tag{14}$$

with $u$, $v$, and $w$ as defined above. Combining (13) and (14) we obtain (11). Finally, by plugging (11) in (10b) we obtain (12). $\qquad\square$

10

**The curve: the affinely independent case**

When $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely independent then we always have $z \neq 0$, thus the quadratic equation (12) can be solved for $\alpha^*$, the coefficient of the affine combination associated with $p^*$, obtaining

$$\alpha^*(x_0) = \pm \frac{1}{\|z\|}\sqrt{(p_{10} - x_0)^2 - \|M(u + x_0 v)\|^2}. \tag{15}$$

We then get the points that satisfy (7,9) as a function of a single variable $x_0 \in [-\infty, \min_{p_{j_i} \in \mathcal{S}^j}\{p_{j_i 0}\}]$, thus defining a curve $\Gamma$. This is shown by Corollary 14.

**Corollary 14.** *Consider $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ affinely independent. Then, conditions (7,9) define a curve parameterized by*

$$x_0 \in \left[-\infty, \min_{p_{j_i} \in \mathcal{S}^j}\{p_{j_i 0}\}\right],$$

*such that*

$$\overline{x} = \Gamma(x_0) := M(u + x_0 v) \pm \frac{z}{\|z\|}\sqrt{(p_{10} - x_0)^2 - \|M(u + x_0 v)\|^2} + \overline{p}_{j_1}.$$

Geometrically, the curve is either a hyperbola (Figure 5a) or a degenerate hyperbola, see Figure 5b). The curve is also symmetric with respect to a reflection through the hyperplane $\{(x_0; \overline{x}) : \overline{x} \in \text{aff}(\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}), x_0 \in \mathbb{R}\}$ and intersects the boundary of $\text{conv}(\overline{\mathcal{S}}^j \cup \{\overline{p}^*\})$ at two points (one of them in $\text{conv}(\overline{\mathcal{S}}^j)$).



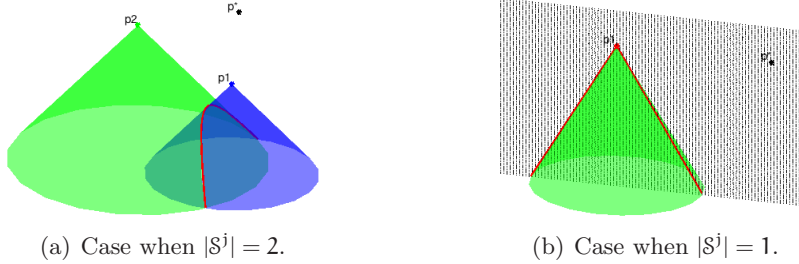(a) Case when $|\mathcal{S}^j| = 2$.  (b) Case when $|\mathcal{S}^j| = 1$.

Figure 5: Illustration of the curve in $\mathbb{R}^3$.

When we are interested in the part of the curve that corresponds to a dual feasible solution, we need all coefficients of the affine combination to be non-negative. Therefore the portion of the curve we are interested on corresponds to $\alpha^*(x_0) \geq 0$, that is

$$\overline{x} = \Gamma^+(x_0) := M(u + x_0 v) + \frac{z}{\|z\|}\sqrt{(p_{10} - x_0)^2 - \|M(u + x_0 v)\|^2} + \overline{p}_{j_1}.$$

Corollary 15 is a direct consequence of Theorem 13.

11

**Corollary 15.** *Consider $\overline{S}^j \cup \{\overline{p}^*\}$ affinely independent. The points $x$ that satisfy (7,9) and for which $\alpha^*(x_0) \geq 0$ can be written as an affine combination*

$$\overline{x} = \sum_{i=1}^{s} \alpha_i(x_0)\overline{p}_{j_i} + \alpha^{*+}(x_0)\overline{p}^*, \tag{16}$$

*for $x_0 \in [-\infty, \min_{p_{j_i} \in S^j}\{p_{j_i 0}\}]$, such that*

$$\alpha^{*+}(x_0) = \frac{1}{\|z\|}\sqrt{(p_{10} - x_0)^2 - \|M(u + x_0 v)\|^2}, \tag{17a}$$

$$\alpha_1(x_0) = 1 - 1_{s-1}^{\mathsf{T}}(u + x_0 v) - \alpha^{*+}(x_0)\left(1_{s-1}^{\mathsf{T}}w + 1\right), \tag{17b}$$

$$\alpha_{2:s}(x_0) = u + x_0 v + \alpha^{*+}(x_0)\frac{w}{\|z\|}, \tag{17c}$$

*with $1_{s-1}$ a $(s-1)-$vector with entries all $1$, and $\alpha_{2:s}$ the vector with $\alpha_2, \ldots, \alpha_s$.*

Using the formulas from Corollary 6, it is now possible to write explicitly the dual variables as functions of $x_0$ corresponding to each point $(x_0; \Gamma^+(x_0))$ on the curve.

### Case when $\overline{S}^j \cup \{\overline{p}^*\}$ is affinely dependent

When $\overline{S}^j \cup \{\overline{p}^*\}$ is affinely dependent, the linear system (10a) together with condition (9) define an affine space of dimension $1$, that is, a line, which intersects the manifold defined by (10b-10c), on a single point, $x^j$. That is because $x^j$ satisfies both (7) and $\overline{x}^j \in \text{conv}(\overline{S}^j) = \text{conv}(\overline{S}^j \cup \{\overline{p}^*\})$ if and only if $x^j$ is the solution to $\text{Inf}_Q(S^j)$, which is unique.

**Theorem 16.** *If $\overline{S}^j \cup \{\overline{p}^*\}$ is affinely dependent then conditions (7,9) have as solution the single point $x^j$ that corresponds to a dual feasible solution.*

When $\overline{S}^j \cup \{\overline{p}^*\}$ is affinely dependent, Theorem 13 still holds with $z = 0$. In particular (13) also still holds. But in this case $\alpha_i$, $i = 1, ..., s$, and $\alpha^*$ cannot be written as functions of $x_0$ as in Corollary 15, but they can be written instead in terms of $\alpha^*$. Since, in this case, $x_0(\alpha^*) = x_0^j$ for any value of $\alpha^*$, from (14) we have

$$\alpha_{2:s}(\alpha^*) = u + x_0^j v + \alpha^* w. \tag{18a}$$

$$\alpha_1(\alpha^*) = 1 - \alpha^* - 1^{\mathsf{T}}(u + x_0^j v + \alpha^* w), \tag{18b}$$

## 4.2 The curve search

At the beginning of a curve search step, let $S^j$ be the support set and $x^j$ the current iterate. If we are starting a major iteration (we come from step (b)) then $(S^j, x^j)$ is a dual feasible S-pair, otherwise, if we come from a previous curve search, it is not. Regardless of the case, at the beginning of a curve search we always have the following

- $\overline{S}^j \cup \{\overline{p}^*\}$ is affinely independent;

- $\overline{x} \in \text{conv}(\overline{S}^j \cup \{\overline{p}^*\})$.

This fact will be proved in section 4.4.

The curve search consists on, starting at $x^j$, moving on the curve defined by $\Gamma^+(x_0)$ in the direction of decrease of $x_0$, until either dual feasibility is lost or the violated

constraint becomes feasible. Using the conclusions from the previous section, that corresponds to solving the following problem

$$
\begin{aligned}
\min \quad & x_0 \\
\text{s.t.} \quad & \|\overline{p}^* - \overline{x}\| \geq p_0^* - x_0, \\
& \alpha_i(x_0) \geq 0, \quad i = 1, ..., s, \\
& \alpha^*(x_0) \geq 0, \\
& x_0 \leq \min_{p_{j_i} \in S^j}\{p_{j_i 0}\}.
\end{aligned}
$$

Note that since $\overline{x}^j \in \text{conv}(\overline{S}^j \cup \{\overline{p}^*\})$ we know that $\alpha_i(x_0^j) \geq 0$, for all $i = 1, ..., s$, and $\alpha^*(x_0^j) \geq 0$.

Therefore, the solution to the problem above, is always smaller or equal to $x_0^j$ which always satisfies the last inequality.

Unless stated otherwise, for the remainder of this section, we shall consider $|S^j| > 1$ and $\overline{S}^j \cup \{\overline{p}^*\}$ affinely independent.

**The partial step**

The partial step consists on finding the point corresponding to the maximum step on the curve that can be made without losing dual feasibility. When $\overline{S}^j \cup \{\overline{p}^*\}$ is affinely independent, Corollary 15 describes the dual variables as we move on the curve. Therefore, the point where dual feasibility is lost corresponds to the value of $x_0$ that solves

$$
\begin{aligned}
\min \quad & x_0 \\
\text{s.t.} \quad & \alpha_i(x_0) \geq 0, \quad i = 1, ..., s, \\
& \alpha^*(x_0) \geq 0, \\
& x_0 \leq x_0^j.
\end{aligned} \tag{19}
$$

Note that, if we consider the definitions of $\alpha_i$, $i = 1, ..., m$ given by (17), the constraint $\alpha^*(x_0) \geq 0$ is already implied. The solution to (19) is then found by simply solving the $s$ equations (17b-17c) subject to the inequality $x_0 \leq x_0^j$ and picking the largest of the solutions. These are radical equations that can be easily solved by isolating the square root term in one side, squaring both sides, solving the squared equation, and finally discarding any extraneous solutions. Note that, for some $i$ the corresponding equation may not have a solution, since the curve may not intersect the supporting hyperplane of the corresponding facet of $\text{conv}(\overline{S}^j \cup \{\overline{p}^*\})$.

The partial step procedure is summarized in Algorithm 1. Note that this procedure will only be done when $|S^j| > 1$, as we shall see ahead.

**The full step**

The full step procedure finds the point corresponding to the minimum step on the curve for which $p^*$ is feasible, that is, the first point on the curve where the primal constraint corresponding to $p^*$ is feasible is active:

$$
\|\overline{p} - \overline{x}\| = p_0^* - x_0. \tag{20}
$$

The system of equations (10a,10b,20) is equivalent to the system of equations (10a,10b,21), with (21) being

$$
(\overline{p}^* - \overline{p}_{j_1})^\mathsf{T}\overline{x} = b^* + x_0 c^*, \tag{21}
$$

13

**Algorithm 1** Partial step procedure

---

**Input:** Matrix $M$, vectors $u, v, w, z$, point $x^j$ and set $\mathcal{S}^j$.
**Output:** $x_0^{partial}$ and index $k$.

1: **for** $i = 1, ..., s$ **do**
2:      Define $\alpha_i(x_0)$ as in (17);
3:      $\hat{\mathcal{A}}_i \leftarrow$ solutions to $\alpha_i(x_0) = 0$;
4:      $\mathcal{A}_i \leftarrow \left\{ \delta \in \hat{\mathcal{A}}_i : \delta \text{ is real} \wedge \delta \leq x_0^j \right\}$;
5: **end for**
6: $x_0^{partial} \leftarrow \max\{\cup_{i=1,...,s} \mathcal{A}_i\}$;
7: $k \leftarrow i$   s.t.   $x_0^{partial} \in \mathcal{A}_i$.

---

such that

$$c^* = p_0^* - p_{j_1 0} \quad \text{and} \quad b^* = \frac{1}{2}\left( \|\overline{p}^*\|^2 - (p_0^*)^2 - \|\overline{p}_{j_1}\|^2 + p_{j_1 0}^2 \right),$$

and for $x_0$ satisfying (10c) and $x_0 \leq p_0^*$. Therefore, the full step consists on solving

$$\begin{aligned} \max \quad & x_0 \\ \text{s.t.} \quad & \overline{x} = \Gamma^+(x_0) \\ & (\overline{p}^* - \overline{p}_{j_1})^T \overline{x} = b^* + x_0 c^* \\ & x_0 \leq p_0^* \\ & x_0 \leq x_0^j. \end{aligned} \tag{22}$$

Note that the solution to (22) to may not exist. It is easy to see that equations (7,9) together with (20) seek the position of the vertex of a second-order cone that has points $\overline{\mathcal{S}}^j \cup \{p^*\}$ on its boundary, which is not guaranteed to exist. Additionally, such point may not be unique.

To solve (22), we could plug $\overline{x} = \Gamma^+(x_0)$ in (20), and solve for $x_0$. However we realized this approach would result in rather intricate calculations involving square roots. So, instead, we go back to the results of Lemma 12 and Theorem 13 and proceed as we explain next. We start by plugging (11) in (21), and solve for $\alpha^*$. We then obtain $\alpha^*$ in terms of $x_0$:

$$\alpha^*(x_0) = \frac{1}{(\overline{p}^* - \overline{p}_{j_1})^T z}(b^* + x_0 c^* - (\overline{p}^* - \overline{p}_{j_1})^T(M(u + x_0 v) + \overline{p}_{j_1})), \tag{23}$$

Now, we could solve (12) for $x_0$. However, the following is easier: we plug (23) back in (11), which now allows $\overline{x}$ to be written solely as a linear function of $x_0$ simply as

$$\overline{x}(x_0) = q + x_0 r + \overline{p}_{j_1},$$

with

$$q = Mu + \frac{b^* - (\overline{p}^* - \overline{p}_{j_1})^T(Mu + \overline{p}_{j_1})}{(\overline{p}^* - \overline{p}_{j_1})^T z} \quad \text{and} \quad r = Mv + \frac{c^* - (\overline{p}^* - \overline{p}_{j_1})^T(Mv)}{(\overline{p}^* - \overline{p}_{j_1})^T z},$$

and we solve (10b). This results in a much simpler quadratic equation on $x_0$:

$$\|q + x_0 r\|^2 = (p_{j_1 0} - x_0)^2. \tag{24}$$

14

Note that, in this process, we never imposed that $\alpha^*(x_0) \geq 0$. If the solution(s) to (24) are real, we only keep the one(s) that satisfy both

$$\alpha^*(x_0) \geq 0 \quad \text{and} \quad x_0 \leq \min\left\{x_0^j, p_0^*\right\},$$

for $\alpha^*(x_0)$ given by (23). If there are more than one such solution, we pick the one with the maximum value of $x_0$ (the one that "occurs first" as we move the curve), which will become $x_0^{\text{full}}$, the solution of (22). If we are left with no solutions or the solutions were not real, then, as explained previously, that means that such point does not exist. In this case, we shall consider $x_0^{\text{full}} = -\infty$. The details of this procedure are summarized in Algorithm 2.

---

**Algorithm 2** Full step procedure

---

**Input:** Matrix $M$, vectors $u, v, w, z$, points $x^j, p^*$, and set $\mathcal{S}^j$
**Output:** $x_0^{\text{full}}$.

1: Define $c^*, b^*, q$ and $r$;
2: Get $\widehat{\mathcal{A}}^*$ the set of solutions to $\|q + x_0 r\|^2 = (p_{j_1 0} - x_0)^2$.
3: $\mathcal{A}^* \leftarrow \{\delta \in \widehat{\mathcal{A}}^* : \alpha^*(\delta) \geq 0 \wedge \delta \leq p_0^* \wedge \delta \leq x_0\}$, for $\alpha^*$ as in (23).
4: **if** $\mathcal{A}^* = \emptyset$ **then**
5: $\quad x_0^{\text{full}} \leftarrow -\infty$
6: **else**
7: $\quad x_0^{\text{full}} \leftarrow \max\{\mathcal{A}^*\}$
8: **end if**

---

**Taking a step**

If $x_0^{\text{partial}} < x_0^{\text{full}}$, then the $p^*$ becomes feasible while all dual variables associated with $\mathcal{S}^j$ are feasible too. That implies that $x^{\text{full}}$ is the solution to $\text{Inf}_{\mathcal{Q}}(\mathcal{S}^j \cup \{p^*\})$. The algorithm now starts a new major iteration with a dual feasible S-pair $(\mathcal{S}^j \cup \{p^*\}, x^{\text{full}})$.

On the other hand, if $x_0^{\text{partial}} \leq x_0^{\text{full}}$, then a dual variable became $0$ before $p^*$ was primal feasible. Geometrically, that means the curve hits a facet of $\text{conv}(\overline{\mathcal{S}}^j \cup \{\overline{p}^*\})$ before hitting the translated cone $\overline{p}^* - \mathcal{Q}$. The point $p_k$ that results from applying (26) is an opposite point to that hit facet. When this happens, $p_k$ is dropped from $\mathcal{S}^j$: $\mathcal{S}^j = \mathcal{S}^j \setminus \{p_k\}$, and the algorithm then performs a new curve search with the new $\mathcal{S}^j$ starting at $x^{\text{partial}}$. It may happen that there may be more than one dual variable that became $0$ simultaneously, that is, at line 6 of Algorithm 1, $x_0^{\text{partial}}$ belongs to more than one $\mathcal{A}_i$. When that happens, after dropping one of such points at the first curve search, the next curve search will consist on a partial step with zero length, that is, the objective function value will be maintained, and a point corresponding to one of those dual variables that are $0$ will be picked to be removed from the support set. This is done as many times as necessary.

### 4.2.1 $\quad \overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely dependent

When $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely dependent, we have seen that conditions (7,9) do not define a curve. That means, a movement in the primal space is not possible. However, it is possible to move in the dual space to different dual solutions corresponding to the

same $x^j$. Once one the dual variables becomes $0$, we drop that point from $S^j$, fixing the issue of the affine dependence. In order to do that, we will solve the following problem

$$
\begin{aligned}
\max \quad & \alpha^* \\
\text{s.t.} \quad & \alpha_i(\alpha^*) \geq 0, \ i = 1, ..., s \\
& \alpha^* \geq 0
\end{aligned}
\tag{25}
$$

for $\alpha_i(\alpha^*)$ defined as in (18). This problem finds an affine combination of $x^j$ where $\alpha^* > 0$ and $\alpha_i = 0$ for some $i = 1, ..., s$. Problem (25) can easily be solved using a minimum ratio rule, see Algorithm 3.

---

**Algorithm 3** Aff Dep Case procedure

---

**Input:** Vectors $u, v, w$, point $x^j$.
**Output:** Index $k$.

1: Define vectors $\rho$ and $\sigma$:

$$
\rho = \begin{pmatrix} 1 - 1^T(u + x_0^j v) \\ u + x_0^j v \end{pmatrix} \quad \text{and} \quad \sigma = \begin{pmatrix} -1 - 1^T w \\ w \end{pmatrix}.
$$

2: Get $k$ such that

$$
-\frac{\rho_k}{\sigma_k} := \min_{j=1,...,s} \left\{ -\frac{\rho_j}{\sigma_j} : \sigma_j < 0 \right\}.
\tag{26}
$$

---

Let $p_k \in S^j$ be the point resulting from (26). In section 4.4 it will be proved that $\overline{S}^j \setminus \{\overline{p}_k\} \cup \{\overline{p}^*\}$ is affinely independent, and so, a curve search can then be performed.

### 4.2.2 The special case when $p^*$ is the solution and the relation to the $|S^j| = 1$ case

Note that the curve search always assumes that the solution to $\text{Inf}_Q(S^j \cup \{p^*\})$ will satisfy (7) for a subset of $S^j$. But that is not the case when $p^*$ happens to be the solution. In fact, as we show next, this case fails to be identified by the curve search procedure.

Consider that $p^*$ is the solution to $\text{Inf}_Q(S^j \cup \{p^*\})$. After a series of partial steps where a point from $S^j$ is dropped each time, the algorithm finally performs a curve search with $|S^j| = 1$, in which, the full step essentially consists on solving the following system of equations

$$
\|\overline{p}_{j_1} - \overline{x}\| = p_{j_1 0} - x_0
\tag{27a}
$$

$$
\|\overline{p}^* - \overline{x}\| = p_0^* - x_0
\tag{27b}
$$

$$
\overline{x} \in \text{aff}(\{\overline{p}_{j_1}, \overline{p}^*\}).
\tag{27c}
$$

When $p^*$ is the solution to $\text{Inf}_Q(\{p_{j_1}\} \cup \{p^*\})$, the solution to equations (27) does not exist in general (unless it happens that $\|\overline{p}_1 - \overline{p}^*\| = p_{10} - p_0^*$), and so from the full step we would have $x_0^{\text{full}} = -\infty$. On the other hand, the partial step consists of

16

solving $\alpha_1(x_0) = 0$ given simply by

$$\alpha_1(x_0) = 1 - \frac{p_{j_1 0} - x_0}{\left\| \overline{p}^* - \overline{p}_{j_1} \right\|} = 0,$$

(note that $M = [\,]$) for $x_0 \leq p_{j_1 0}$. This yields

$$x_0 = p_{10} - \left\| \overline{p}^* - \overline{p}_{j_1} \right\| \quad \text{and} \quad \overline{x} = \overline{p}^*. \tag{28}$$

The partial step in this case does not get the correct value of $x_0$, which would be $p_0^*$. That happens because, in the curve search we made the assumption that $\|\overline{p}_1 - \overline{x}\| = p_{10} - x_0$. Note that the result of the partial step is independent of whether $p^*$ is the solution or not.

So, when $p^*$ is the solution to $\mathrm{Inf}_\Omega(\{p_{j_1}\} \cup \{p^*\})$, following the partial and full step procedures as explained before, we would have $x_0^{partial} < x_0^{full}$. As a consequence, a partial step would be taken, point $p_{j_1 0}$ would correctly be dropped from $\mathcal{S}^j$, but the next iterate would have the incorrect value of $x_0^j$!

This issue conducted us to add an extra step after the Optimality Check where we check whether $p^*$ is the solution, that is,

$$\left\| \overline{p}_{i_j} - \overline{p}^* \right\| \leq p_{j_i 0} - p_0^* \quad \forall p_{i_j} \in \mathcal{S}^j.$$

If $p^*$ is not the solution, then a curve search is performed. This not only avoids issues with the partial step when $\mathcal{S}^j$ has a single point, but it also avoids potentially $n$ curve search procedures where a point would be removed from the support set in each one, only then to find out that $p^*$ is the solution.

This extra step does not fix the fact that, when $\mathcal{S}^j$ has a single point, the partial step will not work correctly. However that will never be an issue once it is known that $p^*$ is not the solution, because when that is the case, at the full step we will obtain

$$x_0^{full} = \frac{1}{2} \left( p_{j_1 0} + p_0^* - \left\| \overline{p}_{j_1} - \overline{p}^* \right\| \right)$$

as per Theorem 7. The value of $x_0^{partial}$ will be as in (28), and so $x_0^{full} > x_0^{partial}$ as a consequence of the fact that

$$\left\| \overline{p}_1 - \overline{p}^* \right\| > p_{10} - p_0^*.$$

Alternatively, and for the sake of simplicity, whenever $|\mathcal{S}^j| = 1$ one can simply use the formulas given by Theorem 7 instead of doing a curve search.

## 4.3   Pseudo-code

We now aggregate the results/discussion of the previous sections on Algorithm 4.

**Algorithm 4** Dual algorithm for the infimum of $\mathcal{P}$ with respect to $\mathcal{Q}$

**Input:** $\mathcal{P}$, dual feasible S-pair $(\mathcal{S}^0, x^0)$.
**Output:** $x$, $\mathcal{S}$, the optimal solution and an support set, respectively.

1: **for** $j = 0, 1, ....$ **do**

    *Optimality check:*

2:    **if** $x^j \preceq_{\mathcal{Q}} p_i$ for all $p_i \in \mathcal{P}$ **then**

3:       $x^j$ is the optimal solution and $\mathcal{S}^j$ an optimal support set. **Stop.**

4:    **else**

5:       Get $p^* \in \mathcal{P}$ s.t. $x^j \succ_{\mathcal{Q}} p^*$.

6:    **end if**

    *Check if solution of* $\mathrm{Inf}_{\mathcal{Q}}(\mathcal{S}^j \cup \{p^*\})$ *is* $p^*$:

7:    **if** $p^* \preceq_{\mathcal{Q}} p_{i_j}$ for all $p_{i_j} \in \mathcal{S}^j$ **then**

8:       $x^{j+1} \leftarrow p^*$; $\mathcal{S}^{j+1} \leftarrow \{p^*\}$; Go to *Optimality check.*

9:    **end if**

    *Special case* $|\mathcal{S}^j| = 1$ *(optional):*

10:   **if** $|\mathcal{S}^j| = 1$ **then**

11:      Set $x^{j+1}$ as in Theorem 7; $\mathcal{S}^{j+1} \leftarrow \mathcal{S}^j \cup \{p^*\}$; Go to *Optimality check.*

12:   **end if**

    $\overrightarrow{\mathcal{S}^j} \cup \{\overline{p}^*\}$ *affinely dependent:*

13:   **if** $\|z\| = 0$ **then**

14:      Get $k$ from Algorithm 3; $\mathcal{S}^j \leftarrow \mathcal{S}^j \setminus \{p_k\}$.

15:   **end if**

    *Curve search:*

16:   Define matrix $M$ and vectors $b, c, u, v, w$ and $z$.

17:   Get $x_0^{\mathrm{partial}}$ from Algorithm 1; Get $x_0^{\mathrm{full}}$ from Algorithm 2.

18:   **if** $x_0^{\mathrm{partial}} \geq x_0^{\mathrm{full}}$ **then**

19:      $x_0^j \leftarrow x_0^{\mathrm{partial}}$; $\overline{x}^j \leftarrow \Gamma^+(x_0^j)$; $\mathcal{S}^j \leftarrow \mathcal{S}^j \setminus \{p_k\}$; Go to *Curve search.*

20:   **else**

21:      $x_0^{j+1} \leftarrow x_0^{\mathrm{full}}$; $\overline{x}^{j+1} \leftarrow \Gamma^+(x_0^{j+1})$; $\mathcal{S}^{j+1} \leftarrow \mathcal{S}^j \cup \{p^*\}$; Go to *Optimality check.*

22:   **end if**

23: **end for**

## 4.4 Finiteness and correctness of the algorithm

For the proof of the correctness of the algorithm presented in Algorithm 4, we will follow a similar approach as in [10]. We first introduce the following definition:

**Definition 17.** The triple $(x, \mathcal{S}, p)$ is said to be a *(violated) V-triple* if the following four conditions hold:

(a) $\overline{\mathcal{S}} \cup \{\overline{p}\}$ is affinely independent,

(b) $\|\overline{p} - \overline{x}\| \geq (p_0 - x_0)$,

(c) $\|\overline{p}_i - \overline{x}\| = (p_{i0} - x_0)$, for $p_i \in \mathcal{S}$,

(d) $\overline{x} \in \text{conv}(\overline{\mathcal{S}} \cup \{\overline{p}\})$.

We now prove a series of theorems that will culminate in the correctness of the algorithm.

**Theorem 18.** *Given a V-triple* $(x^j, \mathcal{S}^j, p^*)$, *if the solution given by the full step,* $x^{\text{full}}$, *exists and is dual feasible, then it is the optimal solution to* $\text{Inf}_Q(\mathcal{S}^j \cup \{p^*\})$. *Moreover,* $x_0^{\text{full}} \leq x_0^j$, *and, if* $x_0^{\text{full}} > x_0^{\text{partial}}$ *then* $(\mathcal{S}^j \cup \{p^*\}, x^{\text{full}})$ *is a dual feasible S-pair.*

*Proof.* The statement is a direct consequence of the fact that the full step finds a point $x^{\text{full}}$ such that

$$\left\|\overline{p}^* - \overline{x}^{\text{full}}\right\| = p_0^* - x_0^{\text{full}} \tag{29}$$

$$\left\|\overline{p}_i - \overline{x}^{\text{full}}\right\| = p_{i0} - x_0^{\text{full}} \quad \forall p_i \in \mathcal{S}, \tag{30}$$

$$\overline{x}^{\text{full}} \in \text{aff}(\overline{\mathcal{S}} \cup \{\overline{p}^*\}). \tag{31}$$

whenever the above conditions are feasible. If, additionally, $\overline{x}^{\text{full}} \in \text{conv}(\overline{\mathcal{S}}^j \cup \{\overline{p}^*\})$ then $x^{\text{full}}$ solves $\text{Inf}_Q(\mathcal{S}^j \cup \{p^*\})$ as per Corollary 4. The fact that $x_0^{\text{full}} \leq x_0^j$ is a consequence of the full step definition.

Now assume $x_0^{\text{full}} > x_0^{\text{partial}}$. It is easy to see that $(\mathcal{S}^j \cup \{p^*\}, x^{\text{full}})$ is a dual feasible S-pair. Clearly $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely independent. Additionally, $\overline{x}^{\text{full}} \in \text{ri conv}(\overline{\mathcal{S}}^j)$, because otherwise there would have been a coefficient of the convex combination that would be 0, implying that $x_0^{\text{full}} = x_0^{\text{partial}}$, which contradicts the assumption. $\square$

From Theorem 18 we conclude that, before the "Optimality Check" procedure, we always have a dual feasible S-pair $(\mathcal{S}^j, x^j)$. After picking a $p^*$ corresponding to an infeasible constraint, if $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely independent, then we have a V-triple. Otherwise, Theorem 20, proves that Algorithm 3 returns a V-triple. Either way, the V-triple at this stage is always such that $\|\overline{p}^* - \overline{x}^j\| > (p_0^* - x_0^j)$. Combined with Theorem 19, we conclude that before a curve search we always have a V-triple.

**Theorem 19.** *Suppose* $x_0^{\text{partial}} \geq x_0^{\text{full}}$. *Given a V-triple* $(x^j, \mathcal{S}^j, p^*)$, *the partial step returns another V-triple* $(x^{\text{partial}}, \mathcal{S}^j \setminus \{p_k\}, p^*)$ *such that* $x_0^{\text{partial}} \leq x_0^j$.

*Proof.* We need to prove properties (a-d) from Definition 17 for $(x^{\text{partial}}, \mathcal{S}^j \setminus \{p_k\}, p)$. Property (a) follows from the fact that $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely independent. Properties (c) and (d) and the fact that $x_0^{\text{partial}} \leq x_0^j$ are a direct consequence of the partial step definition. Property (b) follows from a continuity argument since $\|\overline{p}^* - \overline{x}^j\| \geq (p_0^* - x_0^j)$ and the fact that $x_0^{\text{partial}} \geq x_0^{\text{full}}$, implying that $\|\overline{p}^* - \overline{x}^{\text{partial}}\| \geq (p_0^* - x_0^{\text{partial}})$. $\square$

**Theorem 20.** *When $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely dependent, $(x^j, \mathcal{S}^j \setminus \{p_k\}, p^*)$ for $p_k$ satisfying (26) is a V-triple.*

*Proof.* In order to prove that $(x^j, \mathcal{S}^j \setminus \{p_k\}, p^*)$ we need to prove the four properties from Definition 17. Properties (b) and (c) are trivial since $(\overline{\mathcal{S}}^j, x^j)$ is an S-pair and $p^*$ corresponds to a primal infeasible constraint at $x^j$. Now, in order to prove (a), suppose, by contradiction, that $\overline{\mathcal{S}}^j \setminus \{\overline{p}_k\} \cup \{\overline{p}^*\}$ is affinely dependent. Then, since $\overline{\mathcal{S}}^j \setminus \{\overline{p}_k\}$ is affinely independent, there exists $\beta_i$, $i = 1, ..., s$ and $i \neq k$ s.t.

$$\overline{p}^* = \sum_{i=1, i\neq k}^{s} \beta_i \overline{p}_i \quad \text{with} \quad \sum_{i=1, i\neq k}^{s} \beta_i = 1.$$

With loss of generality consider $k \neq 1$. Since $z = Mw + (\overline{p} - \overline{p}_1) = 0$ (a consequence of the affine dependence of $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$), we have

$$\overline{p}^* = \sum_{i=2}^{s} w_i(\overline{p}_i - \overline{p}_1) + \overline{p}_1 = \sum_{i=1}^{s} \delta_i \overline{p}_i$$

for $\delta_1 = 1 - \sum_i w_i$ and $\delta_i = w_i$ for $i = 2, ..., s$. As a consequence

$$\overline{p}_k = \sum_{i=1, i\neq k}^{s} \frac{\beta_i - \delta_i}{\delta_k} \overline{p}_i.$$

We have that $\sum \frac{\beta_i - \delta_i}{\delta_k} = 1$, and note that $\delta_k \neq 0$. That is, $\overline{p}_k$ is an affine combination of $\overline{\mathcal{S}} \setminus \{\overline{p}_k\}$, which contradicts the assumption.

Finally, consider $\beta = \rho - \frac{\rho_k}{\sigma_k}\sigma$ and $\beta^* = \frac{\rho_k}{\sigma_k}$. It is easy to see that $\beta \geq 0$ and, in particular, $\beta_k = 0$. We now prove that $\overline{x}^j = M\beta_{2:s} + \beta^*(\overline{p} - \overline{p}_1) + \overline{p}_1$, that is property (d), by observing that

$$M\beta_{2:s} + \beta^*(\overline{p} - \overline{p}_1) + \overline{p}_1 = M\left(u + x_0 v - \frac{\rho_k}{\sigma_k}w\right) - \frac{\rho_k}{\sigma_k}(\overline{p} - \overline{p}_1) + \overline{p}_1 \tag{32}$$

$$= M(u + x_0 v) + \overline{p}_1 - \frac{\rho_k}{\sigma_k}\left(Mw + (\overline{p} - \overline{p}_1)\right) \tag{33}$$

$$= M(u + x_0 v) + \overline{p}_1 \tag{34}$$

$$= \overline{x}^j \tag{35}$$

since $z = 0$. $\qquad\square$

With the previous theorems, we conclude that, starting from a V-triple $(x^j, \mathcal{S}^j, p^*)$ for which $\|\overline{p}^* - \overline{x}^j\| > (p_0^* - x_0^j)$, one can obtain a dual feasible S-pair $(\hat{\mathcal{S}} \cup \{p\}, \hat{x})$ such that $\hat{\mathcal{S}} \subseteq \mathcal{S}^j$ in at most $|\mathcal{S}^j| - |\hat{\mathcal{S}}| \leq n$ partial steps and a full step. Moreover, $\hat{x}_0 < x_0^j$, since, even though when taking a partial step or a full step the value of $x_0$ may be maintained, we know that the value of $x_0$ must decrease either in one of the partial steps taken or in the full step, because otherwise we would have that $\hat{x} = x^j$, that is, $\text{Inf}_Q(\mathcal{S}^j) = \text{Inf}_Q(\hat{\mathcal{S}} \cup \{p^*\})$ contradicting the fact that $p^*$ is infeasible to $\text{Inf}_Q(\mathcal{S}^j)$. Therefore, since the value of $x_0$ strictly decreases at each major iteration the same S-pair can never reoccur. Since the number of possible S-pairs is finite we conclude:

**Theorem 21.** *The proposed dual algorithm solves problem* $\mathrm{Inf}_Q(\mathcal{P})$ *in a finite number of iterations.*

**An observation on degeneracy**

It is easy to see that in a primal algorithm *cycling* would be a possibility. The current primal feasible solution may correspond to different support sets, that is, different dual solutions. Thus, it could happen that, after a sequence of adding/dropping points from the support set, the algorithm did not move in the primal space and ended up in a support set visited previously. That is a consequence of the lack of freedom of which point to enter the support set in the primal setting, it has to be one corresponding to a primal constraint that is active. In the dual algorithm, the equivalent situation, of when a movement is not possible because a dual variable is zero at the current iterate, is easily dealt by the algorithm by removing the point corresponding to that dual variable from $\mathcal{S}^j$ (either at a partial step or when $\overline{\mathcal{S}}^j \cup \{\overline{p}^*\}$ is affinely dependent). This is done as many times as the number of dual variables that are zero, after which a movement will then be possible in the next iteration. Therefore, as far as degeneracy is concerned no special procedure is required in our algorithm to prevent *cycling*.

# 5 Implementation details

The main computational work that is required in the algorithm happens before each curve search, when three linear systems need solved in order to get vectors $u, v$ and $w$. These linear systems all have the same matrix $M^\mathsf{T}M$. Our implementation is based on the QR factorization of matrix $M$ of size $n \times (s-1)$ such that $s-1 \leq n$, where $s = |\mathcal{S}^j|$. We have that $M = QR$, with $Q$ a $n \times n$ orthogonal matrix and $R$ a $n \times (s-1)$ upper triangular matrix. Let

$$Q = [Q_M \mid Q_\perp] \text{ and } R = \left[\frac{R_\triangle}{0}\right],$$

with $Q_M$ with size $n \times (s-1)$ and $Q_\perp$ with size $n \times (n-s+1)$, and let, and $R_\triangle$ a $(s-1) \times (s-1)$ matrix. As a consequence, $M^\mathsf{T}M = R^\mathsf{T}R$.

Using the QR factorization of $M$ allows the following linear systems

$$(M^\mathsf{T}M)u = b - M^\mathsf{T}\overline{p}_{j_1} \text{ and } (M^\mathsf{T}M)v = c,$$

to be reduced to two triangular linear systems, which can efficiently be computed using *Back Substitution*. Moreover, $M^+ = R_\triangle^{-1}Q_M^\mathsf{T}$, and so

$$w = -M^+(\overline{p}_1^* - \overline{p}_{j_1})$$

can be obtained by solving the triangular linear system

$$R_\triangle w = -Q_M^\mathsf{T}(\overline{p}_1^* - \overline{p}_{j_1}).$$

Matrix $M$ is given by the points of the support set $\mathcal{S}^j$ which is updated throughout the algorithm either by adding a point or removing a point. Next we explain how to update $M$ in those circumstances.

**Adding a point to $\mathcal{S}^j$.** Whenever a point, $p_{i_k}$ is added to $\mathcal{S}^j$, we append column $\overline{p}_{i_k} - \overline{p}_{i_1}$ to matrix $M$.

**Removing a point from $\mathcal{S}^j$.** Whenever a point, $p_{i_k}$ is removed from $\mathcal{S}^j$, there are two cases. If $i_k \geq 1$ then we need to remove the $(k-1)$-th column from $M$. When $i_k = 1$, we need to remove the first column from $M$, obtaining $\hat{M}$, and get the new $M$ by adding the rank-1 matrix $(p_{i_1} - p_{i_k})^{\mathsf{T}} 1_n$ to $\hat{M}$.

Given the above, every time $M$ needs to be updated, we can use the QR factorization of the old matrix $M$ to calculate efficiently the QR factorization of the new $M$ [11, §12.5]. This is accomplished by using *Givens rotations*, which, in the case when $M$ is $n \times n$, has $\mathcal{O}(n^2)$ computational complexity.

## 6    Computational Results

We have implemented our algorithm in MATLAB as explained in Section 5 choosing the most infeasible point at each iteration. We also compared it with solving both the primal (P) and dual (D) problems with the interior point method solver of Gurobi [13] (version 8.0.0) using its MATLAB interface. All Gurobi parameters were kept at their default values. Our experiments were conducted using MATLAB R2016a (version 9.0) on a Mac with an Intel Core i5 1.6 GHz processor, with 8GB RAM, running Mac OS X version 10.11.6. The results are shown in Table 1.

| Problems | | Our dual algorithm | | | Gurobi (dual) | | Gurobi (primal) | |
|---|---|---|---|---|---|---|---|---|
| | | | S-pair | Time | | Time | | Time |
| $n$ | $m$ | Iters | updates | (secs) | Iters | (secs) | Iters | (secs) |
| 10 | $10^2$ | 6.44 | 6.68 | 0.008 | 9.72 | 0.008 | 8.56 | 0.228 |
| 10 | $10^3$ | 8.12 | 8.76 | 0.010 | 11.56 | 0.094 | 9.16 | 1.627 |
| 10 | $10^4$ | 8.12 | 9.24 | 0.027 | 12.44 | 1.073 | out of memory | |
| $10^2$ | $10^2$ | 15.56 | 15.56 | 0.031 | 9.56 | 0.127 | 8.88 | 7.455 |
| $10^2$ | $10^3$ | 20.96 | 21.16 | 0.145 | could not solve[†] | | 9.36 | 87.601 |
| $10^2$ | $10^4$ | 25.28 | 25.72 | 0.375 | could not solve[†] | | out of memory | |
| $10^2$ | $10^5$ | 29.24 | 29.88 | 3.341 | could not solve[†] | | out of memory | |
| $10^3$ | $10^4$ | 77.32 | 77.44 | 8.641 | could not solve[†] | | out of memory | |
| $10^3$ | $10^5$ | 87.92 | 88.16 | 154.613 | could not solve[†] | | out of memory | |

Table 1: Computational results with the averages corresponding to 25 datasets with points randomly sampled from a standard normal distribution. [†]Matlab stopped responding.

One reason for the good performance of the dual algorithm is that it does not add many points to the support set that are not in the final support set. This can be seen by the fact that the number of iterations (number of points added to the support set) is very close to the number of dual S-pair updates (number of curve searches performed).

# References

[1] F. ALIZADEH AND D. GOLDFARB, *Second-order cone programming*, Mathematical Programming, 95 (2003), pp. 3–51.

[2] M. BÂDOIU AND K. L. CLARKSON, *Smaller core-sets for balls*, in Proc. 14th Annual ACM-SIAM Symp. on Discrete Algorithms, SODA '03, Philadelphia, PA, USA, 2003, SIAM, pp. 801–802.

[3] M. BÂDOIU, S. HAR-PELED, AND P. INDYK, *Approximate clustering via core-sets*, in Proc. 34th Annual ACM Symp. on Theory of Computing, STOC '02, New York, NY, USA, 2002, ACM, pp. 250–257.

[4] M. CAVALEIRO AND F. ALIZADEH, *A faster dual algorithm for the euclidean minimum covering ball problem*, Ann. Oper. Res., to appear.

[5] P. DEARING AND C. R. ZECK, *A dual algorithm for the minimum covering ball problem in $\mathbb{R}^n$*, Oper. Res. Lett., 37 (2009), pp. 171–175.

[6] M. DYER, *A class of convex programs with applications to computational geometry*, in Proc. of the Eighth Annual Symp. on Computational Geometry, SCG '92, New York, NY, USA, 1992, ACM, pp. 9–15.

[7] M. DYER, N. MEGIDDO, AND E. WELZL, *Linear programming*, Chapman and Hall/CRC, Boca Raton, FL, 2nd ed., 2004, ch. 45.

[8] K. FISCHER AND B. GÄRTNER, *The smallest enclosing ball of balls: Combinatorial structure and algorithms*, Internat. J. Comput. Geom. Appl., 14 (2004), pp. 341–387.

[9] B. GÄRTNER AND S. SCHÖNHERR, *An efficient, exact, and generic quadratic programming solver for geometric optimization*, in Proc. 16th annual ACM Symp. on Computational Geometry (SCG), 2000, pp. 110–118.

[10] D. GOLDFARB AND A. IDNANI, *A numerically stable dual method for solving strictly convex quadratic programs*, Math. Program., 27 (1983), pp. 1–33.

[11] G. H. GOLUB AND C. VAN LOAN, *Matrix Computations (3rd Ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[12] 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, Gaithersburg, MD, USA, (1996).

[13] G. O. INC., *Gurobi optimizer reference manual*, 2016.

[14] P. KUMAR, J. S. B. MITCHELL, AND E. A. YILDIRIM, *Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions*, in Proc. 5th Workshop on Algorithm Engineering and Experiments, ALENEX03, Springer Berlin Heidelberg, 2003, pp. 44–55.

[15] T. LARSSON AND L. KÄLLBERG, *Fast and robust approximation of smallest enclosing balls in arbitrary dimensions*, in Proc. 11th Eurographics/ACMSIGGRAPH Symp. on Geometry Processing, SGP '13, Aire-la-Ville, Switzerland, Switzerland, 2013, Eurographics Association, pp. 93–101.

[16] J. MATOUŠEK, M. SHARIR, AND E. WELZL, *A subexponential bound for linear programming*, Algorithmica, 16 (1996), pp. 498–516.

[17] N. MEGIDDO, *On the ball spanned by balls*, Discrete Comput. Geom., 4 (1989), pp. 605–610.

[18] B. MORDUKHOVICH, N. M. NAM, AND C. VILLALOBOS, *The smallest enclosing ball problem and the smallest intersecting ball problem: existence and uniqueness of solutions*, Optim. Lett., 7 (2013), pp. 839–853.

[19] N. M. NAM, T. A. NGUYEN, AND J. SALINAS, *Applications of convex analysis to the smallest intersecting ball problem*, J. Convex Anal., 19 (201), pp. 497–518.

[20] Y. NESTEROV AND M. TODD, *Primal-dual interior-point methods for self-scaled cones*, SIAM Journal on Optimization, 8 (1998), pp. 324–364.

[21] Y. E. NESTEROV AND M. J. TODD, *Self-scaled barriers and interior-point methods for convex programming*, Mathematics of Operations Research, 22 (1997), pp. 1–42.

[22] F. NIELSEN AND R. NOCK, *Approximating smallest enclosing balls with applications to machine learning*, Internat. J. Computat. Geom. Appl., 19 (2009), pp. 389–414.

[23] E. WELZL, *Smallest enclosing disks (balls and ellipsoids)*, in New Results and New Trends in Computer Science Proc., H. Maurer, ed., Berlin, Heidelberg, 1991, Springer Berlin Heidelberg, pp. 359–370.

[24] E. A. YILDIRIM, *Two algorithms for the minimum enclosing ball problem*, SIAM J. Optim., 19 (2008), pp. 1368–1391.

[25] G. ZHOU, K.-C. TOHEMAIL, AND J. SUN, *Efficient algorithms for the smallest enclosing ball problem*, Comput. Optim. Appl., 30 (2005), pp. 147–160.