Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 167 (2005) 635–649
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Dykstra's algorithm with strategies for projecting onto certain polyhedral cones

Patricia Mariela Morillas

Instituto de Matemática Aplicada San Luis (UNSL-CONICET), Ejército de los Andes 950, C.P. 5700, San Luis, Argentina

Abstract

We consider Dykstra's alternating projection method when it is used to find the projection onto polyhedral cones of the form $\bigcap_{i=1}^{n}\{x \in \mathscr{H} : \langle v_i, x\rangle \leqslant 0\}$ where $\mathscr{H}$ is a real Hilbert space and $\langle v_i, v_j\rangle > 0$, $i,j = 1,\ldots,n$. Based on some properties of the projection, we propose strategies with the aim to reduce the number of cycles and the execution time. These strategies consist in previous discarding and arrangement, and in projecting cyclically onto the intersection of two halfspaces. Encouraging preliminary numerical results with cut semimetrics as vectors v_i are presented.

Keywords: Alternating projection methods; Dykstra's algorithm; Polyhedral cones; Cut cone

1. Introduction

Dykstra's algorithm [1] is based on a clever modification of the classical alternating projection method first proposed by von Neumann [2], and studied later by Cheney and Goldstein [3]. It is a simple iterative technique that

E-mail address: morillas@unsl.edu.ar

0096-3003/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2004.06.136

guarantees convergence to the closest point in the intersection of closed convex sets that are not necessarily closed subspaces. It was applied to a wide range of problems and it is usually easy to program. For a recent application of Dykstra's algorithm to compute the nearest diagonally dominant matrix see [4–6]. For a complete survey on Dykstra's algorithm and applications see [7].

We consider Dykstra's alternating projection method when it is used to project onto polyhedral cones of the form

$$\bigcap_{i=1}^{n}\{x \in \mathscr{H} : \langle v_i, x\rangle \leqslant 0\},$$

where $\mathscr{H}$ is a real Hilbert space and $\langle v_i, v_j\rangle > 0$, $i,j = 1,\ldots,n$. With the aim to reduce the number of cycles and the execution time, we propose strategies that consist in previous discarding and arrangement, and in projecting cyclically onto the intersection of two halfspaces. These strategies are based on some properties of the projection onto polyhedral cones.

If $V \subseteq \{1,\ldots,n\}$ and $0 \leqslant |V| \leqslant [\frac{n}{2}]$, the *cut semimetric* d_V is the vector defined by $d_V(i,j) = 1$ if $|V \cap \{i,j\}| = 1$, and $d_V(i,j) = 0$ otherwise, for $1 \leqslant i < j \leqslant n$. The cone generated by these vectors, denoted by cut_n, is known as *cut cone*. Cuts are very important in discrete mathematics and its applications (see [8] for more details). In particular, cut_n contains the set of ℓ_2-embeddable distances and it is contained in the set of ℓ_2-embeddable squared distances (associated with *euclidean distance matrices*). So the projection onto cut_n is a relaxation to the projections onto these sets which play an important role for determining molecular conformations [9]. In the numerical experiments we use no null cut semimetrics as vectors v_i. So, we compute the projection onto the polar cone of cut_n and then onto cut_n (see Lemma 3.1).

The rest of this paper is divided into sections as follows. In Section 2, we present Dykstra's algorithm and its most relevant characteristics. Section 3 contains some properties of the projection onto polyhedral cones. In Section 4, we present two versions of Dykstra's algorithm. One of them is simply a formulation of this algorithm for projecting onto a polyhedral cone. The other also includes the strategies to reduce the number of cycles and the execution time. In Section 5, we show numerical results to compare both algorithms. Finally, in Section 6 we present some concluding remarks.

2. Dykstra's algorithm

Let $\mathscr{H}$ be a real Hilbert space with scalar product $\langle x, y\rangle$ and norm $\|x\| = \langle x, x\rangle^{1/2}$, $x, y \in \mathscr{H}$. For a given non-empty closed convex subset C of $\mathscr{H}$ and a given $x_0 \in \mathscr{H}$ there exists a unique solution x^* of

$$\min \quad \|x_0 - x\| \tag{1}$$

$$\text{subject to} \quad x \in C, \tag{2}$$

which is characterized by

$$x^* \in C \text{ and } \langle x_0 - x^*, x - x^* \rangle \leqslant 0 \quad \text{for all } x \in C, \tag{3}$$

(see [10], Theorem 1, p. 69). This solution x^* is called the projection of x_0 onto C, and we write $x^* = P(x_0|C)$. We consider the case that C is the intersection of a finite number of closed convex sets $C_i \subset \mathscr{H}$, $i = 1,\ldots,n$, and that each C_i is simple enough to compute $P(y|C_i)$ for any $y \in \mathscr{H}$, $i = 1,\ldots,n$.

Dykstra's algorithm [1,11], solves the problem (1) and (2) by generating two sequences, $\{x_i^k\}$ and $\{I_i^k\}$, with $k \in \mathscr{N}$ and $i = 1,\ldots,n$. These sequences are defined by the following recursive formulas

$$x_0^k = x_n^{k-1}, \quad k \in \mathscr{N},$$

$$x_i^k = P\left(x_{i-1}^k - I_i^{k-1}|C_i\right) \quad \text{and} \quad I_i^k = x_i^k - \left(x_{i-1}^k - I_i^{k-1}\right), \tag{4}$$

$i = 1,\ldots,n$, $k \in \mathscr{N}$, with initial values $x_n^0 = x_0$, $I_i^0 = 0$, $i = 1,\ldots,n$.

The utility of Dykstra's algorithm is based on the following theorem.

Theorem 2.1 [11]. *Let $C_1,\ldots,C_n$ be closed convex subsets of a real Hilbert space such that $C = \bigcap_{i=1}^n C_i \neq \emptyset$. For any $i = 1,\ldots,n$ and any $x_0 \in \mathscr{H}$, the sequence $\{x_i^k\}$ generated by (4) converges strongly to $x^* = P(x_0|C)$ (i.e., $\|x_i^k - x_0\| \to 0$ as $k \to \infty$).*

The rate of convergence of Dykstra's algorithm (2.3) in the polyhedral case (i.e., the intersection of a finite number of closed halfspaces) is linear. Starting with any point $x_0 \in \mathscr{H}$, it is possible to show that the sequence of iterates generated by Dykstra's algorithm (4), satisfies an error bound which will depend on the angles between the linear varieties which comprise the boundaries of the halfspaces involved [12]. More precisely, for each $i = 1,\ldots,n$, the sequence $\{x_i^k\}$ satisfies $\|x_i^k - P(x_0|C)\| \leqslant \eta c^k$ for all $k \in \mathscr{N}$, where $C = \bigcap_{i=1}^n C_i$, η is a constant, and $0 \leqslant c < 1$. In the case when C is the intersection of two halfspaces, a stronger result is proved in [12]: the sequence of iterates is either finite or satisfies $\|x_i^k - P(x_0|C) \leqslant c^{k-1}\|x_0 - P(x_0|C)\|$ for all $k \in \mathscr{N}$, where c is the cosine of the angle between the two functionals which define the halfspaces.

3. Some properties of the projection onto polyhedral cones

We begin recalling some basic notions. For an arbitrary subset $S \subseteq \mathscr{H}$ we consider its *orthogonal complement*

$$S^\perp = \{y \in \mathscr{H} : \langle x, y \rangle = 0 \text{ for all } x \in S\} \tag{5}$$

and its *polar cone*

$$S^0 = \{y \in \mathscr{H} : \langle x, y\rangle \leqslant 0 \text{ for all } x \in S\}. \tag{6}$$

The following results appear in [13] for the finite dimensional case, but it is easy to see that they remain true for an arbitrary real Hilbert space. The next lemma is concerning to the projection onto a convex cone (see [13], (2.7.5) Lemma, p. 51).

Lemma 3.1. *Let* $K \subseteq \mathscr{H}$ *be a convex cone and* $a \in \mathscr{H}$. *If* a *admits an orthogonal decomposition* $a = p + p^*$ *with* $p \in K$, $p^* \in K^0$ *and* $\langle p,p^*\rangle = 0$, *then* $p = P(a|K)$ *and* $p^* = P(a|K^0)$. *Conversely, if* $P(a|K)$ *exists, then* $P(a|K^0)$ *exists and both projections constitute an orthogonal decomposition of* a.

A more specific result is the following (see [13], (2.7.7) Theorem, p. 51 and (2.8.4), p. 55).

Theorem 3.2. *Let* $n \geqslant 1$ *and let* $a, v_1, \ldots, v_n \in \mathscr{H}$, *with* $v_i \neq 0$, $i = 1,\ldots,n$. *Then* $P(a|\bigcap_{i=1}^n \{v_i\}^0)$ *exists, and* $p = P(a|\bigcap_{i=1}^n \{v_i\}^0)$ *if and only if* $p \in \bigcap_{i=1}^n \{v_i\}^0$ *and there exist real numbers* $\lambda_i \geqslant 0$, $i = 1,\ldots,n$, *such that* $p = a - \sum_{k=1}^n \lambda_i v_i$ *and* $\sum_{k=1}^n \lambda_i \langle v_i, p\rangle = 0$.

We now present some properties of the projection onto a polyhedral cone that will be useful to state algorithms in Section 4. We begin with the following elemental fact.

Theorem 3.3. *Let* $n \geqslant 1$ *and let* $a, v_1, \ldots, v_n \in \mathscr{H}$, *with* $v_i \neq 0$, $i = 1,\ldots,n$. *Let* $p = P(a|\bigcap_{i=1}^n \{v_i\}^0)$.

1. *If* $\langle v_1,p\rangle < 0$ *then* $p = P(a|\bigcap_{i=2}^n \{v_i\}^0)$.
2. *If* $J = \{i \in \{1,\ldots,n\}: \langle v_i,p\rangle = 0\}$ *then* $p = P(a|\cap_{i \in J}\{v_i\}^\perp)$.

Proof. (1) Set $p' = P(a|\bigcap_{i=2}^n \{v_i\}^0)$. By Lemma 3.1, $\langle a - p', p'\rangle = 0$ and $a - p' \in (\bigcap_{i=2}^n \{v_i\}^0)^0 \subseteq (\bigcap_{i=1}^n \{v_i\}^0)^0$. Suppose now that $\langle v_1,p'\rangle > 0$. Consider for each $t \in [0,1]$ the point $p(t) = (1 - t)p + tp'$. For each $i = 2,\ldots,n$, we have $\langle v_i,p(t)\rangle \leqslant 0$. Consider the continuous function $g : [0,1] \to \mathscr{R}$ given by $g(t) = \langle v_1,p(t)\rangle$. By hypothesis, $g(0) = \langle v_1,p(0)\rangle = \langle v_1,p\rangle < 0$, and by our supposition, $g(1) = \langle v_1,p(1)\rangle = \langle v_1,p'\rangle > 0$. Then, by Bolzano's theorem, there exists $t_0 \in (0,1)$ such that $g(t_0) = 0$. Since $p \in \bigcap_{i=2}^n \{v_i\}^0$, then $\|p' - a\|_2 \leqslant \|p - a\|_2$. So, $\|p(t_0) - a\|_2 \leqslant (1 - t_0)\|p - a\|_2 + t_0\|p' - a\|_2 \leqslant \|p - a\|_2$. But $p(t_0) \in \bigcap_{i=1}^n \{v_i\}^0$ and p is the projection of a onto $\bigcap_{i=1}^n \{v_i\}^0$, consequently $p(t_0) = p$ and then $\langle v_1,p\rangle = \langle v_1,p(t_0)\rangle = 0$. This equality contradicts the hypothesis. Thus,

$\langle v_1,p'\rangle \leqslant 0$. Since we also have $p' \in \bigcap_{i=2}^n\{v_i\}^0$, then $p' \in \bigcap_{i=1}^n\{v_i\}^0$. We have seen that $p' \in \bigcap_{i=1}^n\{v_i\}^0$, $a - p' \in (\bigcap_{i=1}^n\{v_i\}^0)^0$ and $\langle a-p',p'\rangle = 0$. Thus, by Lemma 3.1, $p' = p$.

(2) Applying (1) we obtain $p = P(a|\cap_{i\in J}\{v_i\}^0)$. Set $p' = P(a|\cap_{i\in J}\{v_i\}^\perp)$. By definition of J, $p \in \cap_{i\in J}\{v_i\}^\perp$. By Lemma 3.1, $a-p\in(\cap_{i\in J}\{v_i\}^0)^0 \subseteq (\cap_{i\in J}\{v_i\}^\perp)^0$ and $\langle a-p,p\rangle = 0$. Thus, using again Lemma 3.1, we obtain $p = p'$. □

Remark 3.4

1. From part 1 of Theorem 3.3, to compute $p = P(a|\bigcap_{i=1}^n\{v_i\}^0)$ it is possible discard halfspaces $\{v_i\}^0$ for which $\langle v_i,p\rangle < 0$.
2. Part 2 of Theorem 3.3 states that project onto an intersection of halfspaces is reduced to project onto an intersection of hyperplanes. The inequalities $\langle v_i,x\rangle \leqslant 0$ with $i \in J$ are called *active constraints* for $P(a|\bigcap_{i=1}^n\{v_i\}^0)$, whereas the other inequalities are called *inactive constraints*.

The following theorem is based on part 1 of Theorem 3.3 and provides a criterion for discarding halfspaces when we project onto certain polyhedral cones.

Theorem 3.5. *Let $n \geqslant 1$ and let $a, v_1,\ldots,v_n \in \mathscr{H}$, with $v_i \neq 0$, $i = 1,\ldots,n$ and $p = P(a|\bigcap_{i=1}^n\{v_i\}^0)$. If*

$$\langle v_1,a\rangle = 0 \quad \textit{and} \quad \langle v_1,v_j\rangle > 0,\ j = 2,\ldots,n \tag{7}$$

or if

$$\langle v_1,a\rangle < 0 \quad \textit{and} \quad \langle v_1,v_j\rangle \geqslant 0,\ j = 2,\ldots,n \tag{8}$$

then $p = P(a|\bigcap_{i=2}^n\{v_i\}^0)$.

Proof. If $p = a$ then the conclusion of the theorem is clearly satisfies regardless of the signs of the products $\langle v_i,v_j\rangle$, $i,j = 1,\ldots,n$. Suppose now that $p \neq a$. By part 1 of Theorem 3.3 we only need to see that $\langle v_1,p\rangle < 0$. Since $p \neq a$, by Theorem 3.2, $p = a - \sum_{i=1}^n\lambda_i v_i$ with $\lambda_i \geqslant 0$, $i = 1,\ldots,n$, and at least one of the λ_i is no null. Thus, from (7) or (8), $\langle v_1,p\rangle = \langle v_1,a\rangle - \sum_{i=1}^n\lambda_i\langle v_1,v_j\rangle < 0$. □

Remark 3.6

1. Consider the vectors $a = (-1,-1,-2)$, $v_1 = (1,-1,-1)$, $v_2 = (-1,1,-1)$ and $v_3 = (-1,-1,1)$. We have $\langle v_1,v_2\rangle = \langle v_1,v_3\rangle = \langle v_2,v_3\rangle = -1$, $\langle v_3,a\rangle = 0$, $\langle v_1,a\rangle = \langle v_2,a\rangle = 2$. Since, $a = \frac{3}{2}v_1 + \frac{3}{2}v_2 + v_3$, by Theorem 3.2, then $P(a|\{v_1\}^0 \cap \{v_2\}^0 \cap \{v_3\}^0) = (0,0,0)$. On the other hand, since $\langle v_1, P(a|\{v_2\}^0)\rangle = \langle v_2, P(a|\{v_1\}^0)\rangle = 8/3$, by Theorem 3.7, $P(a|\{v_1\}^0 \cap \{v_2\}^0) = P(a|\{v_1\}^\perp \cap \{v_2\}^\perp) = (-1,-1,0)$. Let now $a = (0,1,1)$, $v_1 = (1,0,0)$,

$v_2 = (0,1,0)$ and $v_3 = (-1,1,1)$. We have $\langle v_1, v_2\rangle = 0$, $\langle v_1, v_3\rangle = -1$, $\langle v_2, v_3\rangle = 1$, $\langle v_1, a\rangle = 0$, $\langle v_2, a\rangle = 1$, $\langle v_3, a\rangle = 2$. Since $a = v_1 + v_3$, by Theorem 3.2, $P(a|\{v_1\}^0 \cap \{v_2\}^0 \cap \{v_3\}^0) = (0,0,0)$. On the other hand, by Theorem 3.7, $P(a|\{v_2\}^0 \cap \{v_3\}^0) = \left(\frac{1}{2}, 0, \frac{1}{2}\right)$. These two examples show that we cannot omit the condition on the vectors v_i in (7).
2. To see that we cannot omit the condition on the vectors v_i in (8), consider the vectors $a = (1,2)$, $v_1 = (0,1)$ and $v_2 = (1,-1)$. Then $\langle v_1, v_2\rangle = -1 < 0$, $\langle v_1, a\rangle = 2$, $\langle v_2, a\rangle = -1$, $P(a|\{v_1\}^0 \cap \{v_2\}^0) = (0,0)$ and $P(a|\{v_1\}^0) = (1,0)$.

The following theorem is about the projection onto the intersection of two halfspaces.

Theorem 3.7. *Let $a, v_1, v_2 \in \mathscr{H}$, with $v_1 \neq 0$ and $v_2 \neq 0$. Then $P(a|\{v_1\}^0 \cap \{v_2\}^0)$ is characterized as follows:*

1. *If $a \in \{v_1\}^0 \cap \{v_2\}^0$ then $P(a|\{v_1\}^0 \cap \{v_2\}^0) = a$.*
2. *If $\langle v_1, a\rangle > 0$ and $P(a|\{v_1\}^\perp) \in \{v_2\}^0$ then $P(a|\{v_1\}^0 \cap \{v_2\}^0) = P(a|\{v_1\}^\perp)$.*
3. *If $\langle v_2, a\rangle > 0$ and $P(a|\{v_2\}^\perp) \in \{v_1\}^0$ then $P(a|\{v_1\}^0 \cap \{v_2\}^0) = P(a|\{v_2\}^\perp)$.*
4. *If $P(a|\{v_1\}^\perp) \notin \{v_2\}^0$ and $P(a|\{v_2\}^\perp) \notin \{v_1\}^0$ then $P(a|\{v_1\}^0 \cap \{v_2\}^0) = P(a|\{v_1\}^\perp \cap \{v_2\}^\perp)$.*

Proof. Without loose of generality, suppose that $\|v_1\|_2 = \|v_2\|_2 = 1$. Set

$$p = P\left(a|\{v_1\}^0 \cap \{v_2\}^0\right), \tag{9}$$

$$p_1 = P\left(a|\{v_1\}^\perp\right) = a - \langle v_1, a\rangle v_1, \tag{10}$$

$$p_2 = P\left(a|\{v_1\}^\perp\right) = a - \langle v_2, a\rangle v_2. \tag{11}$$

Therefore

$$\langle v_2, p_1\rangle = \langle v_2, a\rangle - \langle v_1, a\rangle\langle v_1, v_2\rangle, \tag{12}$$

$$\langle v_1, p_2\rangle = \langle v_1, a\rangle - \langle v_2, a\rangle\langle v_1, v_2\rangle. \tag{13}$$

If $a \notin \{v_1\}^0 \cap \{v_2\}^0$, then one and only one of the following cases hold:

$$\langle v_1, p\rangle = \langle v_2, p\rangle = 0, \tag{14}$$

$$\langle v_1, p\rangle = 0 \quad \text{and} \quad \langle v_2, p\rangle < 0, \tag{15}$$

$$\langle v_1, p\rangle < 0 \quad \text{and} \quad \langle v_2, p\rangle = 0. \tag{16}$$

We are going to prove first each implication. Statement 1 is trivial.

Suppose now that $\langle v_1, a\rangle > 0$ and $p_1 \in \{v_2\}^0$. We have $p_1 \in \{v_1\}^{\perp} \subseteq \{v_1\}^0$. Thus $p_1 \in \{v_1\}^0 \cap \{v_2\}^0$. We also have $\langle a - p_1, p_1\rangle = 0$ and $\langle a - p_1, x\rangle = \langle v_1, a\rangle \langle v_1, x\rangle \leqslant 0$ for all $x \in \{v_1\}^0 \cap \{v_2\}^0$. Then, by Lemma 3.1, $p_1 = p$. This proves 2 and in a similar manner it can be proved 3.

Suppose now that $p_1 \notin \{v_2\}^0$ and $p_2 \notin \{v_1\}^0$. We have $\langle a - p, p\rangle = 0$ and $\langle a - p, x\rangle \leqslant 0$ for all $x \in \{v_1\}^0 \cap \{v_2\}^0$. Then $\langle a - p, x\rangle \leqslant 0$ for all $x \in \{v_1\}^{\perp} \cap \{v_2\}^{\perp}$. So, by Lemma 3.1, to prove that $p = P(a|\{v_1\}^{\perp} \cap \{v_2\}^{\perp})$ it is sufficient to see that $p \in \{v_1\}^{\perp} \cap \{v_2\}^{\perp}$, i.e., that (14) holds. Suppose that (14) does not hold. For example, suppose that (16) holds. Consider $p(t) = (1 - t)p + t\ p_2$, $t \in [0, 1]$. The function $g(t) = \langle v_1, p(t)\rangle$from $[0, 1]$ to $\mathscr{R}$ is continuous. We also have, $g(0) < 0$ and $g(1) > 0$. Then, there exists $t_0 \in (0, 1)$ such that $g(t_0) = 0$. Therefore $\langle v_1, p(t_0)\rangle = \langle v_2, p(t_0)\rangle = 0$, i.e., $p(t_0) \in \{v_1\}^{\perp} \cap \{v_2\}^{\perp} \subseteq \{v_1\}^0 \cap \{v_2\}^0$. Since $p \in \{v_2\}^{\perp}$, then $\|p(t_0) - a\|_2 \leqslant (1 - t_0)\|p - a\|_2 + t_0\|p_2 - a\|_2 \leqslant \|p - a\|_2$. Since p is the projection onto $\{v_1\}^0 \cap \{v_2\}^0$ and $p(t_0) \in \{v_1\}^0 \cap \{v_2\}^0$, then $p = p(t_0)$. But then $\langle v_1, p\rangle = 0$ which contradicts (16). In a similar manner it can be seen that (15) does not hold. So (14) must holds.

Now we are going to see that at least one of the cases 1, 2, 3 or 4 always holds. If $\langle v_1, a\rangle \leqslant 0$ and $\langle v_2, a\rangle \leqslant 0$, then 1 holds. Suppose now that $\langle v_1, a\rangle > 0$ and $\langle v_2, a\rangle \leqslant 0$ (the case $\langle v_1, a\rangle \leqslant 0$ and $\langle v_2, a\rangle > 0$ is similar). If $\langle v_2, p_1\rangle \leqslant 0$ then 2 holds. If $\langle v_2, p_1\rangle > 0$ and $\langle v_1, p_2\rangle > 0$ then 4 holds. If $\langle v_2, p_1\rangle > 0$ and $\langle v_1, p_2\rangle \leqslant 0$, by (12) and (13)

$$\langle v_2, a\rangle > \langle v_1, a\rangle\langle v_1, v_2\rangle, \tag{17}$$

$$\langle v_1, a\rangle \leqslant \langle v_2, a\rangle\langle v_1, v_2\rangle. \tag{18}$$

By (17), $\langle v_1, v_2\rangle < 0$. Since $\langle v_1, v_2\rangle < 0$, by (18), $\langle v_1, a\rangle/\langle v_1, v_2\rangle \geqslant \langle v_2, a\rangle$. Then, by (17), $\langle v_1, a\rangle/\langle v_1, v_2\rangle > \langle v_1, a\rangle\langle v_1, v_2\rangle$. Since $\langle v_1, a\rangle > 0$, from the last inequality $1 < \langle v_1, v_2\rangle$which is absurd since $\|v_1\|_2 = \|v_2\|_2 = 1$.

Suppose now that $\langle v_1, a\rangle > 0$ and $\langle v_2, a\rangle > 0$. If $\langle v_2, p_1\rangle \leqslant 0$ then 2 holds. If $\langle v_1, p_2\rangle \leqslant 0$ then 3 holds. If $\langle v_1, p_2\rangle > 0$ and $\langle v_2, p_1\rangle > 0$ then 4 holds.

Now we are going to see that 1, 2, 3 and 4 are mutually exclusive or coincide. Clearly, it is sufficient to see that 1 and 4 cannot simultaneously hold, neither 2 and 3; because the other possibilities are trivial.

If 1 and 4 hold, i.e., using (12) and (13)

$$\langle v_1, a\rangle \leqslant 0 \quad \text{and} \quad \langle v_2, a\rangle \leqslant 0, \tag{19}$$

$$\langle v_2, a\rangle > \langle v_1, a\rangle\langle v_1, v_2\rangle, \tag{20}$$

$$\langle v_1, a\rangle > \langle v_2, a\rangle\langle v_1, v_2\rangle. \tag{21}$$

If $\langle v_1, a\rangle = 0$, by (20), $\langle v_2, a\rangle > 0$, which contradicts (19). Analogously it can be seen that $\langle v_2, a\rangle \neq 0$. Therefore we can replace (19) by

$$\langle v_1, a\rangle < 0 \quad \text{and} \quad \langle v_2, a\rangle < 0. \tag{22}$$

By (22) and (20)

$$\langle v_2, a\rangle / \langle v_1, a\rangle < \langle v_1, v_2\rangle \tag{23}$$

and by (22) and (21)

$$\langle v_1, a\rangle / \langle v_2, a\rangle < \langle v_1, v_2\rangle. \tag{24}$$

If $\langle v_2, a\rangle \leqslant \langle v_1, a\rangle$by (23), $1 < \langle v_1, v_2\rangle$. If $\langle v_1, a\rangle \leqslant \langle v_2, a\rangle$, by (24), then $1 < \langle v_1, v_2\rangle$. Thus (20)–(22) cannot hold. This shows that 1 and 4 cannot simultaneously hold.

If $v_1 = v_2$ then $\{v_2\}^0 = \{v_1\}^0$. Thus 2 and 3 coincide. If $v_1 = -v_2$ then $\{v_2\}^\perp = \{v_1\}^\perp$ and $\{v_1\}^0 \cap \{v_2\}^0 = \{v_1\}^\perp$. Thus 2 and 3 are mutually exclusive.

Now suppose that $v_1 \neq \pm v_2$. If 2 and 3 hold then, using (12) and (13)

$$0 < \langle v_2, a\rangle \leqslant \langle v_1, a\rangle\langle v_1, v_2\rangle, \tag{25}$$

$$0 < \langle v_1, a\rangle \leqslant \langle v_2, a\rangle\langle v_1, v_2\rangle. \tag{26}$$

By (25) and (26), $1 \leqslant \langle v_1, v_2\rangle$. Since $\|v_1\|_2 = \|v_2\|_2 = 1$ then $\langle v_1, v_2\rangle = 1$, which is absurd since $v_1 \neq \pm v_2$. Consequently 2 and 3 cannot simultaneously hold. □

4. Algorithms

To find the projection onto $\bigcap_{i=1}^n \{v_i\}^0$, we use the following version of Dykstra's algorithm.

Algorithm 4.1. Given $x_0 \in \mathscr{H}$ set $x_n^0 = x_0, I_i^0 = 0$, $i = 1, \ldots, n$

For $k = 1, 2, \ldots$, until convergence
  Set $x_0^k = x_n^{k-1}$
  For $i = 1, \ldots, n$
    Set $x_i^k = P(x_{i-1}^k - I_i^{k-1} | \{v_i\}^0)$
    Set $I_i^k = x_i^k - (x_{i-1}^k - I_i^{k-1})$
  End For
End For

Here for $a \in \mathscr{H}$, $P(a|\{v_i\}^0)$ is given by

$$P\Big(a|\{v_i\}^0\Big) = \begin{cases} a & \text{if } \langle a, v_i\rangle \leqslant 0, \\ a - \frac{\langle a, v_i\rangle}{\langle v_i, v_i\rangle} v_i & \text{if } \langle a, v_i\rangle > 0. \end{cases}$$

In Algorithm 4.1 we project cyclically onto the n convex sets $\{v_i\}^0$, to find the projection onto $\bigcap_{i=1}^n \{v_i\}^0$. We now describe and motivate strategies with the aim to reduce the number of cycles and the execution time. In the sequel we are going to suppose that $\langle v_i, v_j \rangle > 0$, $i,j = 1,\ldots,n$.

S1. *Previous discarding*: Based on Theorem 3.5, this strategy consists in forming the set $\{i: \langle x_0, v_i \rangle > 0\}$ that contains the indexes of the sets $\{v_i\}^0$ on which projections will be performed.

S2. *To project cyclically onto the intersection of two halfspaces*: If n is even then

$$\bigcap_{i=1}^{n} \{v_i\}^0 = \bigcap_{i=1}^{[n/2]} \left(\{v_{2i-1+c(n)}\}^0 \cap \{v_{2i+c(n)}\}^0 \right),$$

and if n is odd then

$$\bigcap_{i=1}^{n} \{v_i\}^0 = \{v_1\}^0 \cap \bigcap_{i=1}^{[n/2]} \left(\{v_{2i-1+c(n)}\}^0 \cap \{v_{2i+c(n)}\}^0 \right),$$

where

$$c(n) = n - 2\left[\frac{n}{2}\right]$$

and $[\frac{n}{2}]$ is the greater integer less or equal to $\frac{n}{2}$. Hence this strategy consists in projecting cyclically onto the sets $\{v_{2i-1+c(n)}\}^0 \cap \{v_{2i+c(n)}\}^0$ for $i = 1,\ldots,[\frac{n}{2}]$ (if n is odd we also project onto $\{v_1\}^0$). To compute $P(x_0|\{v_{2i-1+c(n)}\}^0 \cap \{v_{2i+c(n)}\}^0)$ we use Theorem 3.7. It is easy to see that case 4 of Theorem 3.7 can be presented only if $v_{2i-1+c(n)}$ and $v_{2i+c(n)}$ are linearly independent. In this case, if each $v_i \in R^d$ for some integer $d \geqslant 2$ and R_i is the matrix which rows are $v_{2i-1+c(n)}$ and $v_{2i+c(n)}$, then $P(x_0|\{v_{2i-1+c(n)}\}^\perp \cap \{v_{2i+c(n)}\}^\perp) = x_0 - R_i^t(R_iR_i^t)^{-1}R_ix_0$. Note that $R_iR_i^t$ is a 2×2 symmetric matrix, and then its inverse is very easy to compute.

S3. *Previous arrangement*: When we use the above strategy, we can expect to reduce the execution time if we can reduce the cases that must be considered to project onto $\{v_{2i-1+c(n)}\}^0 \cap \{v_{2i+c(n)}\}^0$ (see Theorem 3.7). If

$$\langle v_{2i+c(n)}, x_0 \rangle \leqslant \left(\langle v_{2i-1+c(n)}, x_0 \rangle / \langle v_{2i-1+c(n)}, v_{2i-1+c(n)} \rangle \right) \langle v_{2i-1+c(n)}, v_{2i+c(n)} \rangle \quad (27)$$

and $\langle v_{2i-1+c(n)}, x_0 \rangle > 0$, by Theorem 3.7, part 2, then

$$P\left(x_0|\{v_{2i-1+c(n)}\}^0 \cap \{v_{2i+c(n)}^0\}\right) = P\left(x_0|\{v_{2i-1+c(n)}\}^\perp\right). \quad (28)$$

In order (27) to be valid and if $\langle v_i, x_0 \rangle > 0$, $i = 1,\ldots,n$, this strategy consists in arranging the indexes of the sets $\{v_i\}^0$ in such a way that

$$\langle v_i, x_0 \rangle \geqslant \langle v_{i+1}, x_0 \rangle \quad \text{for } i = 1,\ldots,n-1. \quad (29)$$

Here we suppose that the successive projections minus the respective increments that generate Dykstra's algorithm, could have a similar behavior than x_0.

S4. *Previous arrangement with intercalation*: If $\langle v_{2i-1+c(n)}, v_{2i+c(n)}\rangle / \langle v_{2i-1+c(n)}, v_{2i-1+c(n)}\rangle \geqslant 1$, then (29) implies (27). If $\langle v_{2i-1+c(n)}, v_{2i+c(n)}\rangle / \langle v_{2i-1+c(n)}, v_{2i-1+c(n)}\rangle < 1$, we need to increment the difference between $\langle v_{2i+c(n)}, x_0\rangle$ and $\langle v_{2i-1+c(n)}, x_0\rangle$ to try to obtain (27). Thus, if $\langle v_i, x_0\rangle > 0$, $i = 1,\ldots,n$, this strategy consists in arranging the indexes of the sets $\{v_i\}^0$ in such a way that (29) is verified and then project cyclically onto $\{v_i\}^0 \cap \{v_{i+[\frac{m}{2}]}\}^0$ for $i = 1 + c(m), \ldots, [\frac{m}{2}] + c(m)$ (if n is odd we also project onto $\{v_1\}^0$).

The previous strategies can be combined in different manners for fitting into distinct algorithms. Here we only present one of these algorithms, which combines strategies S1, S2 and S4, because it produces the most efficient results. The rest of the algorithms and additional numerical experiments that permit us to study the behavior of each strategy individually, can be seen in [14].

Algorithm 4.2. Given $x_0 \in \mathscr{H}$

Step 1: *Build the set* $\{i\colon \langle x_0, v_i\rangle > 0\} = \{i_1,\ldots,i_m\}$.
Step 2: *Build the order set* $(j_1,\ldots,j_m)$ *arranging* $\{i_1,\ldots,i_m\}$ *in such a manner that* (29) *hold.*
Step 3: *Projecting*

Set $x^0_{\left[\frac{m}{2}\right]+c(m)} = x_0$, $I_i^0 = 0$, $i = 1, \ldots, [\frac{m}{2}] + c(m)$

For $k = 1,2,\ldots,$ until convergence

Set $x_0^k = x^{k-1}_{\left[\frac{m}{2}\right]+c(m)}$

If $c(m) = 1$ **then**
  Set $x_1^k = P(x_0^k - I_1^{k-1} | \{v_{j_1}\}^0)$
End If
For $i = 1 + c(m), \ldots, [\frac{m}{2}] + c(m)$
  Set $x_i^k = P(x_{i-1}^k - I_i^{k-1} | \{v_{j_i}\}^0 \cap \{v_{j_{i+[\frac{m}{2}]}}\}^0)$
  Set $I_i^k = x_i^k - (x_{i-1}^k - I_i^{k-1})$
End For
End For

It is important to note that both algorithms converge to the projection onto $\bigcap_{i=1}^n \{v_i\}^0$. When convergence is attained the process is stopped and the solution is given by x^k. In practice, the algorithms are usually stopped whenever the distance between two consecutive projections onto the same convex set, reaches a pre-established tolerance. For example, the process might be stopped when $\|x_0^{k+1} - x_0^k\| \leqslant \text{TOL}$.

5. Numerical results

We compare Algorithm 4.1 with Algorithm 4.2. These algorithms were implemented in MATLAB Version 6.0.0.88 Release 12 and ran in a Pentium 4 processor at 1.5 GHz. When using Algorithm 4.1, we report the CPU time in seconds (CPU), the number of iterations (ITER), and the elapsed time for iteration (CPU/ITER). For Algorithm 4.2 we report the CPU time in seconds (CPU), the number of iterations (ITER), the elapsed time for iteration (CPU/ITER) and the percentage of saved iterations (SAVED ITER %).

We use no null *cut semimetrics* as vectors v_i. Since the number of cut semimetrics is 2^{n-1} and increases considerably with n, we only consider values of n such that $3 \leqslant n \leqslant 12$. The test vectors were obtained from cut_n and from vectorized symmetric matrices with zero diagonal. In this last case, we distinguish between the dense and the sparse cases, and between the non-negative entries and the distinct sign entries cases. Note that when the entries of the test vectors are non-negative no halfspace is discarded.

For each n we generate a set of representative vectors. The tabulated results, for a given n, is the average of the obtained results for each of these vectors. The algorithms used the distance between the two last projection as a stopping criterion when it reaches a pre-established tolerance. For each algorithm the tolerance was 10^{-7}.

Tables 1–5 show the results of the first five experiments, and show that Algorithm 4.2 outperforms Algorithm 4.1 in number of cycles and CPU time. We note that in these experiments, the greater percentage of saved iterations is observed for vectors in cut_n. We also observe that the percentage of saved iterations is greater for the dense case than for the sparse case. The percentage of saved iterations for the non-negative and distinct sign cases is similar.

Table 1
Results for experiment 1: vectors in cut_n

n	Algorithm 4.1			Algorithm 4.2			
	CPU	ITER	CPU/ITER	CPU	ITER	CPU/ITER	SAVED ITER (%)
3	0	20	0.00E+00	0.01	19	5.26E−04	5.00
4	0.02	73	2.74E−04	0.02	57	3.51E−04	21.92
5	0.13	209	6.22E−04	0.06	70	8.57E−04	66.51
6	0.35	286	1.22E−03	0.18	106	1.70E−03	62.94
7	1.43	420	3.40E−03	0.56	114	4.91E−03	72.86
8	4.64	494	9.39E−03	2.03	180	1.13E−02	63.56
9	14.53	583	2.49E−02	3.2	118	2.71E−02	79.76
10	34.79	663	5.25E−02	12.39	257	4.82E−02	61.24
11	80.99	770	1.05E−01	20.81	188	1.11E−01	75.58
12	213.44	891	2.40E−01	79.88	360	2.22E−01	59.60

Table 2
Results for experiment 2: vectorized symmetric dense matrices with zero diagonal and non-negative entries

n	Algorithm 4.1			Algorithm 4.2			
	CPU	ITER	CPU/ITER	CPU	ITER	CPU/ITER	SAVED ITER (%)
3	0	16	0.00E+00	0	16	0.00E+00	0.00
4	0.01	36	2.78E−04	0.02	33	6.06E−04	8.33
5	0.05	77	6.49E−04	0.04	65	6.15E−04	15.58
6	0.15	139	1.08E−03	0.14	118	1.19E−03	15.11
7	0.81	193	4.20E−03	0.66	157	4.20E−03	18.65
8	2.57	297	8.65E−03	1.88	230	8.17E−03	22.56
9	9.59	417	2.30E−02	7.43	387	1.92E−02	7.19
10	27.83	565	4.93E−02	20.44	504	4.06E−02	10.80
11	68.67	723	9.50E−02	54.41	621	8.76E−02	14.11
12	189.83	930	2.04E−01	152.82	752	2.03E−01	19.14

Table 3
Results for experiment 3: vectorized symmetric dense matrices with zero diagonal and distinct sign entries

n	Algorithm 4.1			Algorithm 4.2			
	CPU	ITER	CPU/ITER	CPU	ITER	CPU/ITER	SAVED ITER (%)
3	0	5	0.00E+00	0	3	0.00E+00	40.00
4	0	9	0.00E+00	0	8	0.00E+00	11.11
5	0.01	19	5.26E−04	0.01	16	6.25E−04	15.79
6	0.04	35	1.14E−03	0.03	32	9.38E−04	8.57
7	0.1	46	2.17E−03	0.07	40	1.75E−03	13.04
8	0.3	51	5.88E−03	0.24	48	5.00E−03	5.88
9	1.91	85	2.25E−02	1.36	78	1.74E−02	8.24
10	4.17	87	4.79E−02	2.64	73	3.62E−02	16.09
11	13	132	9.85E−02	9.84	112	8.79E−02	15.15
12	41.65	200	2.08E−01	29.11	156	1.87E−01	22.00

It is worth noticing that Dykstra's algorithm converge faster when the number of active constrains is smaller. Therefore, for both algorithm, the number of cycles required for the sparse case is less than the number of cycles required for the dense case. We also have that the previous discarding does not reduce the number of cycles. It can only reduce the number of projections per cycle and then the CPU time.

Table 6 shows the results of an experiment in which Algorithm 4.1 outperforms Algorithm 4.2. In this experiment, Algorithm 4.1 requires a number of cycles that is more or less the half of the number of cycles that requires Algorithm 4.2. It is important to note that in this case, Algorithm 4.1 in each cycle project first onto the halfspaces corresponding to the active constrains, whereas Algorithm 4.2 project onto the halfspaces corresponding to the active

Table 4
Results for experiment 4: vectorized symmetric sparse matrices with zero diagonal and non–negative entries

n	Algorithm 4.1			Algorithm 4.2			
	CPU	ITER	CPU/ITER	CPU	ITER	CPU/ITER	SAVED ITER (%)
3	0	14	0.00E+00	0	9	0.00E+00	35.71
4	0	20	0.00E+00	0	19	0.00E+00	5.00
5	0.02	32	6.25E−04	0.02	29	6.90E−04	9.38
6	0.05	48	1.04E−03	0.04	44	9.09E−04	8.33
7	0.14	66	2.12E−03	0.12	60	2.00E−03	9.09
8	0.53	90	5.89E−03	0.39	76	5.13E−03	15.56
9	2.57	112	2.29E−02	1.92	102	1.88E−02	8.93
10	7.1	142	5.00E−02	5.64	135	4.18E−02	4.93
11	18.87	184	1.03E−01	15.59	161	9.68E−02	12.50
12	43.3	204	2.12E−01	39.71	196	2.03E−01	3.92

Table 5
Results for experiment 5: vectorized symmetric sparse matrices with zero diagonal and distinct sign entries

n	Algorithm 4.1			Algorithm 4.2			
	CPU	ITER	CPU/ITER	CPU	ITER	CPU/ITER	SAVE ITER (%)
3	0	3	0.00E+00	0	1	0.00E+00	66.67
4	0	10	0.00E+00	0	8	0.00E+00	20.00
5	0.01	14	7.14E−04	0	12	0.00E+00	14.29
6	0.03	27	1.11E−03	0.02	24	8.33E−04	11.11
7	0.06	30	2.00E−03	0.04	27	1.48E−03	10.00
8	0.17	41	4.15E−03	0.11	38	2.89E−03	7.32
9	0.83	50	1.66E−02	0.51	47	1.09E−02	6.00
10	2.46	51	4.82E−02	1.49	47	3.17E−02	7.84
11	7.26	72	1.01E−01	4.56	66	6.91E−02	8.33
12	19.38	86	2.25E−01	11.07	78	1.42E−01	9.30

Table 6
Results for experiment 6: $x_0 = \alpha d_{\{n\}} + \beta e$ with $\alpha + (n-1)\beta \geqslant 0$ and $2\alpha + \beta \leqslant 0$

n	Algorithm 4.1			Algorithm 4.2		
	CPU	ITER	CPU/ITER	CPU	ITER	CPU/ITER
3	0.01	12	8.33E−04	0	6	0.00E+00
4	0.01	15	6.67E−04	0.01	17	5.88E−04
5	0.01	22	4.55E−04	0.02	30	6.67E−04
6	0.03	28	1.07E−03	0.06	42	1.43E−03
7	0.1	40	2.50E−03	0.16	73	2.19E−03
8	0.26	59	4.41E−03	0.47	107	4.39E−03
9	1.06	90	1.18E−02	2.1	210	1.00E−02
10	3.27	133	2.46E−02	6.05	284	2.13E−02
11	19.05	191	9.97E−02	36.37	390	9.33E−02
12	55.96	265	2.11E−01	101.27	504	2.01E−01

constrains at the end of each cycle (see [14] for more details). This fact probably explains the behavior of the algorithms.

Note that in all experiments the required CPU time per cycle is more or less the same for each algorithm when n is fixed. Indeed, except for experiment 1, it is slightly smaller for Algorithm 4.2.

In spite of the results of experiment 6 and taking into in account the results of the other experiments we can conclude that in general, Algorithm 4.2 is better than Algorithm 4.1.

6. Concluding remarks

We have characterized in a simple manner the projection onto the intersection of two halfspaces. We have also proved that under certain conditions, it is possible to discard halfspaces when we project onto certain polyhedral cones. We have used these two results to state strategies for Dykstra's algorithm with the aim to reduce the number of cycles and the execution time. These strategies consist in previous discarding and arrangement, and in projecting cyclically onto the intersection of two halfspaces. Our preliminary numerical experiments indicate that Dykstra's algorithm with these strategies is better.

In the numerical experiments we have projected onto the polar cone of cut_n, and then onto cut_n. This projection is a relaxation of the projections onto the sets of distances and squared distances that are ℓ_2-embeddable. Both sets play an important role for determining molecular conformations.

Acknowledgments

I am grateful with Prof. Juan Cesco at Universidad Nacional de San Luis, San Luis, Argentina; Prof. Marcos Raydan at Universidad Central de Venezuela, Caracas, Venezuela; and Prof. Pablo Tarazaga at Texas A&M University, Texas, USA, for some helpful discussions.

References

[1] R.L. Dykstra, An algorithm for restricted least-squares regression, J. Amer. Stat. Assoc. 78 (1983) 837–842.
[2] J. von Newmann, The geometry of orthogonal spaces, in: Functional operators, vol. II, Annals of Math. Studies, No. 22, Princeton University Press, 1950. This is a reprint of mimeographed lecture notes first distributed in 1933.
[3] W. Cheney, A. Goldstein, Proximity maps for convex sets, Proc. Amer. Math. Soc. 10 (1959) 448–450.

[4] M. Mendoza, M. Raydan, P. Tarazaga, Computing the nearest diagonally dominant matrix, Numer. Linear Algebra Appl. 5 (1998) 461–474.
[5] M. Raydan, P. Tarazaga, Primal and polar approach for computing the symmetric diagonally projection, Numer. Linear Algebra Appl. 9 (2002) 333–345.
[6] M. Monsalve, J. Moreno, R. Escalante, M. Raydan, Selective alternating projections to find the nearest SDD^+ matrix, Appl. Math. Comput. 145 (2003) 205–220.
[7] F. Deutsch, The method of alternating orthogonal projections, in: S.P. Singth (Ed.), Approximation Theory, Spline Functions and Applications, Kluwer Academic Publishers, Netherlands, 1992, pp. 105–121.
[8] M. Deza, M. Laurent, Geometric of Cuts and Metrics, Springer, 1997.
[9] W. Glunt, T.L. Hayden, M. Raydan, Molecular conformations from distance matrices, J. Comp. Chem. 14 (1993) 114–120.
[10] D.G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, 1969.
[11] J.P. Boyle, R.L. Dykstra, A method for finding projections onto the intersections of convex sets in Hilbert spaces, Lect. Notes Statist. 37 (1986) 28–47.
[12] F. Deutsch, H. Hundal, The rate of convergence of Dykstra's cyclic projections algorithm: the polyhedral case, Numer. Funct. Anal. Optim. 15 (1994) 537–565.
[13] J. Stoer, C. Witzgall, Convexity and Optimization in Finite Dimensions I, Springer-Verlag, Berlin, 1970.
[14] P.M. Morillas, Proyección sobre la relajación CUT para el cono de las matrices de distancia, Tesis de Maestría, Departamento de Matemática, Facultad de Ciencias Físico, Matemáticas y Naturales, Universidad Nacional de San Luis, San Luis, Argentina, 2003.