Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Lecturas en Ciencias de la Computación
ISSN 1316-6239

A generalization of Dykstra's algorithm and a least-squares matrix application

R.Escalante y M. Raydan

RT 2006-02

Centro de Cálculo Científico y Tecnológico de la UCV
CCCT-UCV
Caracas, Febrero, 2006.

A generalization of Dykstra's algorithm and a least-squares matrix application

René Escalante [*] Marcos Raydan[†]

February 12, 2006

Abstract

Dykstra's algorithm is a cyclic scheme for finding the projection of a given point onto the intersection of a finite number of closed convex sets. It has recently been generalized to allow a random ordering of the projections over the convex sets, rather than cyclic, and to allow the projection onto separating half spaces without sacrificing the convergence properties of the algorithm. Using a duality approach, we extend further the algorithm to skip, in a systematic way, some of the involved separating half spaces at each cycle. This generalized version is applied to a constrained least-squares matrix problem obtaining a significant reduction in computational cost when compared to the classical Dykstra's algorithm. The main advantage of the new version is that it avoids the calculation of several eigenvalues and eigenvectors of a square symmetric matrix at every cycle.

[*] Departamento de Cómputo Científico y Estadística, División de Ciencias Físicas y Matemáticas, Universidad Simón Bolívar, Ap. 89000, Caracas, 1080-A, Venezuela (rene@cesma.usb.ve). This author was partially supported by the Centro de Estadística y Software Matemático (CESMa) and by the Decanato de Investigación y Desarrollo (DID) at USB.

[†] Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (mraydan@kuaimare.ciens.ucv.ve). This author was partially supported by the Scientific Computing Center at UCV.

1 Introduction

Dykstra's alternating projection algorithm [7] is a well-known cyclic scheme for finding the projection of a given point onto the intersection of a finite number of closed convex sets. It has been generalized by Hundal and Deutsch [11] to allow a random ordering, instead of cyclic, of the projections onto the closed convex sets. More recently, it has also been generalized by Bregman et al. [4] to avoid the projection onto each one of the convex sets in every cycle. Instead, projections onto separating half spaces are used. In this work, using a duality approach, we extend further the algorithm to skip, in a systematic way, some of the involved separating half spaces at each cycle. In particular we illustrate the generalized version solving the constrained least-squares matrix problem considered by Escalante and Raydan ([8] and [9]). In that case, it is required to project at every cycle onto the ε-positive definite cone of matrices, and that implies the calculation of several eigenvalues and eigenvectors of the projected matrix. These calculations are in general very expensive. By using the generalized version, we obtain a version of the algorithm that only requires the computation of several eigenvalues and corresponding eigenvectors of a matrix at a prescribed subsequence of cycles, and still guarantees convergence.

Dykstra's algorithm belongs to the general family of alternating projection methods. The original alternating projection method dates back to von Neumann [12] who treated the problem of finding the projection of a given point in a Hilbert space onto the intersection of two closed subspaces. Later, Cheney and Goldstein [5] extended the analysis of von Neumann's alternating projection scheme to the case of two closed and convex sets. In particular, they established convergence under mild assumptions. However, the limit point need not be the closest in the intersection. Therefore, the alternating projection method, proposed by von Neumann, is not useful for finding the projection onto convex sets. Fortunately, Dykstra [7] found a clever modification of von Neumann's scheme. It guarantees convergence to the closest point in the intersection of closed and convex sets that are not necessarily closed subspaces. Dykstra's algorithm can be obtained via duality, and more powerful versions can be developed.

This paper is organized as follows. In Section 2 we develop a generalized Dykstra's algorithm via duality, and discuss its theoretical properties. In Section 3, we present the generalized alternating projection algorithm when applied to a constrained least-squares matrix problem. In Section 4 we present preliminary numerical results to illustrate the properties of the proposed algorithm.

2 General framework

Let us consider the following optimization problem

$$(P1) \quad \begin{cases} \min f(x) \\ \text{s.t. } x \in \cap_{i=1}^{m} C_i, \end{cases}$$

where $f : V \rightarrow I\!R$, V is a vector space, and $C_i \subseteq V$ is a closed and convex set for every i. Associated to problem $(P1)$ we introduce some required notation. Let $C = \cap_{i=1}^{m} C_i$, and let $V^{\#}$ be the set of affine continuous functionals $\gamma : V \longrightarrow I\!R$. We define the set $V^{\#}(C)$ by

$$V^{\#}(C) \equiv \{\gamma \in V^{\#} : \gamma(x) \leqslant 0, \ \ \forall x \in C\}.$$

Let us now consider an algorithm that produces a sequence $\{x^k\}$, whose k-th iterate is given by

$$x^k = \text{argmin } f(x) + \sum_{i=1}^{m} \gamma_i^k(x),$$

where

$$\gamma_i^k \in V^{\#}(C_i), \ i = 1, \ldots, m.$$

Let $\mathcal{F} : V^{\#} \longrightarrow I\!R$ be defined as $\mathcal{F}(\gamma) = \inf_{x\in V}\ f(x)+\gamma(x)$, and consider the following iterative algorithm.

Generalized Dykstra's algorithm

Step 1. Choose $i(k) \in \{1, \ldots, m\}$ (for example, $i(k) = (k \bmod m) + 1$).

Step 2. Choose $\Gamma_k \subseteq V^{\#}(C_{i(k)})$ such that $\gamma^k_{i(k)} \in \Gamma_k$ and Γ_k is a closed convex cone in $V^{\#}$.

Step 3. Solve the optimization problem (D_k) to obtain g^{k+1}

$$(D_k) \quad \begin{cases} \max \mathcal{F}(\sum_{i\neq i(k)} \gamma^k_i + g) \\ \text{ s.t. } \ g \in \Gamma_k \end{cases}$$

Step 4. Set

$$\begin{cases} \gamma^{k+1}_{i(k)} & = & g^{k+1} \\ \gamma^{k+1}_i & = & \gamma^k_i, \ \ i \neq i(k). \end{cases}$$

Step 5. Finally, set

$$x^{k+1} = \text{argmin } f(x) + \sum_{i=1}^{m} \gamma^{k+1}_i .$$

Now, if we define

$$\gamma^k = \sum_{i=1}^{m} \gamma^k_i ,$$

where $\gamma^k_i \in V^{\#}(C_i)$, then $\gamma^k \in V^{\#}(C)$. Notice that the sequence $\{\gamma^k\}$ satisfy the following ascent properties

$$\begin{cases} \mathcal{F}(\gamma^k) \leqslant f(x), \quad \forall\, x \in C \\ \mathcal{F}(\gamma^1) \leqslant \mathcal{F}(\gamma^2) \leqslant \ldots \end{cases}$$

We now establish a lemma associated with the sequence $\{\gamma^k\}$.

Lemma 2.1 *If $C \neq \emptyset$, then the gradient of γ^k, $\nabla\gamma^k$, is bounded for all k.*

Proof: Let $\hat{x} \in C$. Since γ^k is affine,

$$\gamma^k(x) = \nabla\gamma^k(x - \hat{\mathrm{x}}) + \gamma(\hat{\mathrm{x}}) \leqslant \nabla\gamma^k(x - \hat{\mathrm{x}}).$$

Thus,

$$\mathcal{F}(\gamma^1) \leqslant \mathcal{F}(\gamma^k) = \inf_x f(x) + \gamma^k(x) \leqslant \inf_x f(x) + \nabla\gamma^k(x - \hat{\mathrm{x}})$$

□

We can illustrate the importance of the dual ascent variables with constrained least-squares minimization problems. Consider the following optimization problem

$$(P2) \quad \begin{cases} \min \frac{1}{2}\|x - \overline{x}\|^2 \\ \text{ s.t. } x \in \cap_{i=1}^m C_i, \end{cases}$$

where $\|z\|^2 = \langle z, z\rangle$.

Associated with problem $(P2)$ consider Dykstra's Algorithm [3, 7]:

$I_i^0 = 0$, $i = 1, \ldots, m$, $y^0 = \overline{x}$. For $k = 0, 1, \ldots$

$$\begin{aligned} a^{k+1} &= y^k - I_{i(k)}^k \\ y^{k+1} &= P_{C_{i(k)}}(a^{k+1}) \\ I_{i(k)}^{k+1} &= y^{k+1} - a^{k+1} \\ I_i^{k+1} &= I_i^k, \quad i \neq i(k), \end{aligned}$$

where P_{C_i} represents the projection onto C_i. This process can be written as:

$$(Dykstra) \quad \begin{cases} I_i^0 = 0, \quad i = 1, \ldots, m, \quad x^0 = \overline{x} \\ a^{k+1} = y^k - I_{i(k)}^k \\ y^{k+1} = P_{C_{i(k)}}(a^{k+1}) \\ I_i^{k+1} = \begin{cases} I_i^k, & i \neq i(k) \\ y^{k+1} - a^{k+1}, & i = i(k) \end{cases} \end{cases}$$

Lemma 2.2 $y^k = \overline{x} + \sum_{i=1}^m I_i^k$.

Proof. It follows directly from the properties of Dykstra's algorithm. $\square$

Let $\gamma_i^0 \equiv 0$, for $i = 1, \ldots, m$, and $x^0 = \overline{x}$. According to the Generalized Dykstra's Algorithm, if g^{k+1} is the solution of the problem

$$\begin{cases} \max \mathcal{F}(\sum_{i \neq i(k)} \gamma_i^k + g) \\ \text{s.t.} \quad g \in V^{\#}(C_{i(k)}) \end{cases}$$

$$\gamma_i^{k+1} = \quad \begin{cases} \gamma_i^k, & i \neq i(k) \\ g^{k+1}, & i = i(k) \end{cases}$$

then

$$x^{k+1} = \text{argmin} f(x) + \sum_{i=1}^m \gamma_i^{k+1}.$$

Lemma 2.3 $I_i^k = -\nabla \gamma_i^k$, $i = 1, \ldots, m$, $k = 0, 1, \ldots$.

Proof. It follows by induction. It is clear that the thesis is true when $k = 0$. Suppose it is valid for k. Then, since

$$f_k(x) = f(x) + \sum_{i \neq i(k)} \gamma_i^k(x), \text{ where } f(x) = \frac{1}{2}\|x - \overline{x}\|^2$$

$$\begin{aligned} \nabla f_k(x) &= x - \overline{x} + \textstyle\sum_{i \neq i(k)} \nabla \gamma_i^k \\ &\quad \text{(by inductive hypothesis)} \\ &= x - \overline{x} - \textstyle\sum_{i \neq i(k)} I_i^k \\ &= x - \overline{x} - \textstyle\sum_{i=1}^m I_i^k + I_{i(k)}^k \\ &\quad \text{(by Lemma 2.2)} \\ &= x - y^k + I_{i(k)}^k \\ &= x - a^{k+1}, \end{aligned}$$

and therefore,

$$f_k(x) = \frac{1}{2}\|x - a^{k+1}\|^2 + c_k,$$

So, if

$$\mathcal{F}_f(\sum_{i \neq i(k)} \gamma_i^k + \gamma) = \mathcal{F}_{f_k}(\gamma),$$

$$(D_k) \quad \begin{cases} \max \mathcal{F}_{f_k}(\gamma) \\ \text{s.t.} \quad \gamma \in V^{\#}(C_{i(k)}) \end{cases}$$

is equivalent to

$$\begin{cases} \min f_k(x) = \frac{1}{2}\|z - a^{k+1}\|^2 + c_k \\ \text{s.t.} \quad x \in C_{i(k)}, \end{cases}$$

whose solution is given by $P_{C_{i(k)}}(a^{k+1})$.
(We could define

$$\begin{array}{rcl} y^{k+1}(x) & = & \langle x - P_{C_{i(k)}}(a^{k+1}), a^{k+1} - P_{C_{i(k)}}(a^{k+1})\rangle \\ & = & \langle x - y^{k+1}, a^{k+1} - y^{k+1}\rangle). \end{array}$$

Finally,

$$\nabla\gamma_{i(k)}^{k+1} = a^{k+1} - y^{k+1} = -I_{i(k)}^{k+1}, \quad i = i(k).$$

If $i \neq i(k)$ the result follows by the inductive hypothesis. □

We are now ready to establish a global convergence result for the Generalized Dykstra's algorithm when solving $(P2)$.

Theorem 2.1 *If the interior of C is not empty, and there exists a positive integer $p \geq m$ such that for any $i \in \{1, \dots m\}$ there exists at least one $k \in \{1, \dots p\}$ such that $i = i(k)$, and $\Gamma_k = V^{\#}(C_{i(k)})$, then the sequence $\{x^k\}$ generated by the Generalized Dykstra's Algorithm for solving $(P2)$ satisfy*

$$\lim_{k\to\infty} \|x^k - P_C(\overline{x})\| = 0.$$

Proof. From Lemma 2.1 and Lemma 2.3 we obtain Theorem 13 in [4], which yields our convergence result as a particular case of the convergence Theorem 3.1 in [11]. □

Roughly speaking, Theorem 2.1 claims that if all the convex sets are chosen often enough, and we project onto the associated separating affine sets, then the sequence generated by the Generalized Dykstra's Algorithm converges to the closest point in the intersection.

3 Solving least-squares matrix problems

In this section we illustrate the use of the theory described in the previous section to a constrained least squares matrix problem. In particular we apply the generalized Dykstra's algorithm developed in Section 2 to the least-squares matrix problem considered by Escalante and Raydan ([8] and [9]). For that consider $V = I\!R^{n\times n}$ the subspace of real square matrices, and the following optimization problem

$$\min \|X - A\|_F^2 \ , \tag{1}$$

subject to

$$X^T = X \ ,$$

$$L \leqslant X \leqslant U \ ,$$

$$\lambda_{\min}(X) \geqslant \varepsilon > 0 \ ,$$

$$X \in \mathcal{P} \ ,$$

where A, L and U are given $n \times n$ real matrices, and X is the symmetric $n \times n$ matrix that we wish to find. Likewise, $\lambda_{\min}(X)$ represents the smallest eigenvalue of X, ε is a given positive constant, and $\mathcal{P}$ is the set of $n \times n$ matrices having a particular linear pattern. Throughout this paper, the notation $A \leqslant B$, for any two real $n \times n$ matrices, means that $A_{ij} \leqslant B_{ij}$ for all $1 \leqslant i, j \leqslant n$. Also, $\|A\|_F$ denotes the Frobenius norm of a real matrix A, defined as

$$\|A\|_F^2 = \langle A, A \rangle = \sum_{i,j=1}^{n} (A_{ij})^2 \ ,$$

where the inner product is given by $\langle A, B \rangle = trace(A^T B)$.

Problem (1) arises in statistics and mathematical economics. A classical example in statistics is the problem of finding the nearest symmetric positive definite patterned matrix to a sample covariance matrix. Patterned covariance matrices arise frequently from the models in physical and social sciences, see Hu and Olkin [10]. Another example, discussed by Dantzig [6], is the problem of deriving utility functions for the economy. In this case, the fitting matrix X has to be a symmetric and bounded matrix, whose smallest eigenvalue must be greater than a specified positive parameter ε.

We now define a collection of sets in $I\!R^{n\times n}$ whose intersection is the feasible region of problem (1). These sets are denoted by $\mathcal{B}$ox ($\mathcal{B}$), ε-positive definite (εpd), and $\mathcal{P}$attern ($\mathcal{P}$); and are given by

$$\mathcal{B} = \{X \in I\!R^{n\times n} : L \leqslant X \leqslant U\} \ ,$$

$$\varepsilon pd = \{X \in I\!R^{n\times n} : X^T = X, \ \lambda_{\min}(X) \geqslant \varepsilon > 0\} \ ,$$

and,

$$\mathcal{P} = \{X \in I\!R^{n\times n} : X = \sum_{i=1}^{m} \alpha_i G_i \ \text{ for some } \alpha_i \in I\!R, \ 1 \leqslant i \leqslant m\} \ .$$

In the definition of $\mathcal{P}$, $1 \leqslant m \leqslant n(n+1)/2$ (usually in the applications $m \leqslant n$), and $G_1, ..., G_m$ are given $n \times n$ nonzero symmetric matrices whose entries are either 0 or 1, and have the following property: for each st-entry, $1 \leqslant s, t \leqslant n$, there exists one and only one k ($1 \leqslant k \leqslant m$) such that $(G_k)_{st} = 1$.

Problem (1) can now be stated as follows:

$$\min \left\{ \|X - A\|_F^2 : \; X \in \mathcal{B} \cap \varepsilon pd \cap \mathcal{P} \right\} . \tag{2}$$

We observe that the feasible region of problem (2) is the intersection of closed and convex sets in the inner product space $I\!\!R^{n\times n}$. Furthermore, $\mathcal{P}$ is a closed subspace included in the subspace of symmetric matrices, and $\{G_1, ..., G_m\}$ form a basis of $\mathcal{P}$.

In this work we use an inexpensive procedure to project directly on $\mathcal{B} \cap \mathcal{P}$, which is based on a characterization that was obtained in [8]. Similarly, in [9] the authors characterize the projection onto the set εpd. The associated implementation requires the computation of several eigenvalues and eigenvectors of a symmetric matrix X^k at every cycle k, and is given by

$$P_{\varepsilon pd}(X^k) = X^k + \sum_{i=1}^{p} (\varepsilon - \lambda_i{}^k) Z_i{}^k Z_i{}^{kT} \; , \tag{3}$$

where X^k is the matrix obtained at the k-th cycle by projecting onto the set $\mathcal{B} \cap \mathcal{P}$, which in turn will be projected onto the set εpd. Likewise, $\lambda_1, \ldots, \lambda_p$ are the eigenvalues of X^k that are less than ε, and $Z_1{}^k, \ldots, Z_p{}^k$ are the corresponding eigenvectors.

For a connection with problem $(P1)$ in Section 2, set $f(X) = \frac{1}{2}\|X - A\|_F^2$, $C_1 = \mathcal{B} \cap \mathcal{P}$, and $C_2 = \varepsilon pd$. We present a suitable version of the Generalized Dykstra's Algorithm that can be written as follows.

Set $\gamma_1^0 \equiv 0$, $\gamma_2^0 \equiv 0$, $X^0 = A$. For $k = 0, 1, \ldots$

$$X^k = \text{argmin}\; \frac{1}{2}\|X - A\|^2 + \gamma_1^k(X) + \gamma_2^k(X).$$

a) If $k+1$ is odd,
set $i(k) = 1$,
set $\Gamma_k = V^{\#}(C_1)$, and solve

$$(D_k) \quad \begin{cases} \max \mathcal{F}(\gamma_2^k + \gamma) \\ \text{s.t. } \gamma \in V^{\#}(C_1) \end{cases} \leftarrow \begin{cases} \min \frac{1}{2}\|X - (X^k + \nabla \gamma_1^k)\|^2 \\ \text{s.t. } X \in C_1 \end{cases}$$

b) If $k+1$ is even,
set $i(k)=2$.

b1) If $\frac{k+1}{2}$ is odd,
set $\Gamma_k = V^{\#}(C_2)$, and solve

$$(D_k) \quad \left\{ \begin{array}{l} \max \mathcal{F}(\gamma_1^k + \gamma) \\ \text{ s.t. } \gamma \in V^{\#}(C_2) \end{array} \right.$$

$$X^{k+1} = P_{C_2}(X^k + \nabla\gamma_2^k).$$

$$\begin{aligned} \gamma_2^{k+1} &= \langle X - X^{k+1}, (X^k + \nabla\gamma_2^k) - X^{k+1} \rangle \\ &= -\sum (\varepsilon - \lambda_i)_+ v_i v_i^t \\ &= \sum -(\varepsilon - \lambda_i)_+ (v_i^t (X - X^{k+1}) v_i). \end{aligned}$$

Let $I = \{i : \lambda_i < \varepsilon\}$. For $i \in I$, $v_i^t X^{k+1} v_i = \varepsilon$.

b2) If $\frac{k+1}{2}$ is even,

$$(\text{here} \quad \Gamma_k = \{X \to -\sum_{i\in I} t_i(v_i^t X v_i - \varepsilon) : t_i \geqslant 0\})$$

$$X \to -\sum_{i \in I} t_i (v_i^t (X - X^{k+1}) v_i)$$

$\gamma_2^k \in \Gamma_k$

$$\left\{ \begin{array}{l} \max \mathcal{F}(\gamma_1^k + \gamma) \\ \text{ s.t. } \gamma \in \Gamma_k \end{array} \right. \leftarrow \left\{ \begin{array}{l} \min \frac{1}{2}\|X - (X^k + \nabla\gamma_2^k)\|^2 \\ \text{ s.t. } v_i^t X v_i \geqslant \varepsilon, \ \ i \in I. \end{array} \right.$$

4 Numerical experiments

In this section we compare the performance of the new generalized alternating projection algorithm (GA for brevity), obtained in Section 3 for solving problem (2), with the improved version (IA) of Escalante and Raydan [8], using the modified projection (3), that can be written as follows.

Given $A \in I\!R^{n\times n}$, set $A_0 = A$, and $I^0_{\varepsilon pd} = I^0_{\mathcal{B}\cap\mathcal{P}} = 0$

For $i = 0, 1, 2, ...$

$$\begin{aligned} A_i &= P_{\mathcal{B}\cap\mathcal{P}}(A_i) - I^i_{\varepsilon pd} \\ I^{i+1}_{\varepsilon pd} &= P_{\varepsilon pd}(A_i) - A_i \\ A_{i+1} &= P_{\varepsilon pd}(A_i) - I^i_{\mathcal{B}\cap\mathcal{P}} \\ I^{i+1}_{\mathcal{B}\cap\mathcal{P}} &= P_{\mathcal{B}\cap\mathcal{P}}(A_{i+1}) - A_{i+1}\ . \end{aligned} \tag{4}$$

Here $I^i_{\varepsilon pd}$ and $I^i_{\mathcal{B}\cap\mathcal{P}}$ play the role of the increments introduced by Dykstra [7], and $P_{\mathcal{B}\cap\mathcal{P}}(A)$, is the unique solution to the problem

$$\min_{x\in\mathcal{B}\cap\mathcal{P}} \|X - A\|_F\ . \tag{5}$$

The key fact here is that we do not need to compute eigenvalues and eigenvectors at each iteration of the process, instead we compute them, based on the Generalized Dykstra's algorithm of Section 3, in a simple way and without deteriorating the quality of the approximate solutions.

All experiments in this section were run on a HP APOLLO workstation in double precision FORTRAN. The eigenvalues and eigenvectors, required in the projection onto the set εpd, were computed by the subroutine DSYEVX from the LAPACK library [1]. We set $\varepsilon = 0.1$, for all experiments. The iterations in both algorithm were stopped when

$$\|I^{i-1}_{\varepsilon pd} - I^i_{\varepsilon pd}\|^2_F + \|I^{i-1}_{\mathcal{B}\cap\mathcal{P}} - I^i_{\mathcal{B}\cap\mathcal{P}}\|^2_F \leqslant TOL\ ,$$

for different values of TOL. This stopping criterion is robust (see Birgin and Raydan [2]).

In [9] we use the subroutine DSYEVX from the LAPACK library [1] to compute the eigenvalues and eigenvectors. The subroutine DSYEVX provide flexibility in these estimates, allowing the compute of a subset of eigenvalues and eigenvectors of a given matrix A.

Our purpose is to try with matrices that have a different number of eigenvalues smaller than ε (i.e., matrices with a number of eigenvalues smaller than ε, between 1 and $n/2$ approximately). In [9] the authors observed that if the number of eigenvalues is "large", then computing all eigenvalues at every cycle by means of subroutine DSYEV from LAPACK library is a better option.

The tables below show the dimension of the matrix problem (n), the CPU time in seconds (TIME), the number of iterations (IT) required by both implementations for different tolerances, and the number (constant) of iterations in which we do not compute neither the eigenvalues nor the eigenvectors before calling the LAPACK library again to compute the eigenvalues and eigenvectors (YES/NO) (e.g., p/q means that for p iterations we compute the eigenvalues and eigenvectors, and then for the next q iterations we do not compute them).

In the case of the GA algorithm, it is important to point out that IT$/(YES+NO)$ is the approximate number of times that we call the LAPACK library per numerical test. Far from it, in the case of the IA algorithm, we call in each iteration the subroutine DSYEVX from the LAPACK library. For this reason we expect the GA algorithm to be faster than the IA version.

In each of our experiments we define the matrix L as the null matrix and U as the matrix whose ij-entry is given by $i+j$. Lastly, we set the pattern matrix to be a symmetric Toeplitz matrix (see [8, Exp. 2]). In this case, problem (2) is feasible.

Experiment 1

In our first experiment we carried out many trials, and in all of them we use as initial matrix a random matrix A, which we add at the entries of its diagonal the quantities 0, 0.2, 0.3, 0.4, 0.5 and 0.6 ($d(A)$).

Table 1 shows clearly that GA is faster than IA. Moreover, if n, $d(A)$ and NO (in YES/NO) increase, the CPU time for GA is less than the CPU time

			IA		**GA**		
$d(A)$	n	$e < \varepsilon$	IT	TIME	IT	TIME	YES/NO
0	50	25	294	171	415	158	1/1
					571	179	1/2
					733	205	1/3
0.2	50	11	290	120	379	95.6	1/1
					445	88.6	1/2
					481	83	1/3
					521	81	1/4
0.3	50	7	167	54.87	229	46.13	1/1
					301	47.6	1/2
					375	52.3	1/3
0.4	50	6	111	29.6	169	27	1/1
					235	28	1/2
0.5	100	13	84	138	113	119	1/1
					148	127	1/2
0.6	100	2	179	184	205	121	1/1
					235	106	1/2
					269	102	1/3

Table 1: Results for Experiment 2 ($A = RANDOM + d(A).I$ and $TOL = .5 \times 10^{-3}$)

	IA			**GA**			
TOL	IT	TIME	ERROR	IT	TIME	ERROR	YES/NO
10^{-2}	7	4.0×10^{-2}	3.11×10^{-3}	7	2.9×10^{-2}	3.21×10^{-3}	1/2
10^{-7}	24	0.13	3.98×10^{-8}	47	0.1	2.5×10^{-8}	1/5
10^{-7}	24	0.13	3.98×10^{-8}	42	8.0×10^{-2}	4.6×10^{-8}	1/10

Table 2: Results for Experiment 2 ($n = 10$)

for IA to reach the same accuracy. We report also the number of eigenvalues less than ε during each process ($e < \varepsilon$).

Experiment 2

In the second experiment we considered the Experiment 1 which appears in [9]. In this case the matrix A has $n/2$ eigenvalues smaller than ε (first cycle), and few (one or two) during the iterations that follow.

Tables 2 and 3 show the numerical results obtained. They show also the distance (ERROR) in the Frobenius norm between the output matrices and the exact solution X.

In Table 3 we consider another tolerance parameter ($TOLL$). We will use LAPACK library in all the cycles to compute the eigenvalues and eigenvectors (applying the modified projection (3)) until we reach the new tolerance $TOLL$, then we continue with the application of the approach dual. Here we report also the number of eigenvalues less than ε during the whole process.

References

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, and D. SORENSEN. *LAPACK User's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

IA				GA			
IT	TIME	ERROR	TOLL	IT	TIME	ERROR	$e < \varepsilon$
24	0.13	3.98×10^{-8}	10^{-2}	28	0.11	2.42×10^{-8}	10
24	0.13	3.98×10^{-8}	0.5	35	0.1	2.51×10^{-8}	7
24	0.13	3.98×10^{-8}	1.0	44	9.0×10^{-2}	2.68×10^{-8}	5

Table 3: Results for Experiment 2 ($n = 10$, $TOL = 10^{-7}$ and $YES/NO = 1/10$)

[2] E. G. BIRGIN and M. RAYDAN. Robust stopping criteria for Dykstra's algorithm. *SIAM Journal on Scientific Computing*, 26:1405–1414, 2005.

[3] J.P. BOYLE and R.L. DYKSTRA. A method for finding projections onto the intersections of convex sets in hilbert spaces. *Lecture Notes in Statistics*, 37:28–47, 1986.

[4] L.M. BREGMAN, Y. CENSOR, S. REICH, and Y. ZEPKOWITZ-MALACHI. Finding the projection of a point onto the intersection of convex sets via projections onto halfspaces. *Journal of Approximation Theory*, 124:194–218, 2003.

[5] W. CHENEY and A. GOLDSTEIN. Proximity maps for convex sets. *Proc. Amer. Math. Soc.*, 10:448–450, 1959.

[6] G.B. DANTZIG. Deriving an utility function for the economy. Technical Report SOL 85-6R, Department of Operations Research, Stanford University, Stanford, CA, 1985.

[7] R.L. DYKSTRA. An algorithm for restricted least-squares regression. *J. Amer. Stat. Assoc.*, 78:837–842, 1983.

[8] R. ESCALANTE and M. RAYDAN. Dykstra's algorithm for a constrained least-squares matrix problem. *Num. Linear Algebra Appl.*, 3(6):459–471, 1996.

[9] R. ESCALANTE and M. RAYDAN. Dykstra's algorithm for constrained least-squares rectangular matrix problems. *Computers Math. Applic.*, 35(6):73–79, 1998.

[10] H. HU and I. OLKIN. A numerical procedure for finding the positive definite matrix closest to a patterned matrix. *Statistics and Probability letters*, 12:511–515, 1991.

[11] H. HUNDAL and F. DEUTSCH. Two generalizations of Dykstra's cyclic projections algorithm. *Math. Program.*, 77:335–355, 1997.

[12] J. VON NEUMANN. Functional operators vol. II. The geometry of orthogonal spaces. *Annals of Math. Studies*, 22, 1950. Princeton University Press. This is a reprint of mineographed lecture notes first distributed in 1933.