IMA Journal of Numerical Analysis Page 1 of 23
doi:10.1093/imanum/drnxxx

Inexact block coordinate descent methods with application to the nonnegative matrix factorization

SILVIA BONETTINI
silvia.bonettini@unife.it
Dipartimento di Matematica, Università di Ferrara,
Via Saragat 1, Ferrara, 44100, Italy.

[Received on ...]

This work is concerned with the cyclic block coordinate descent method, or nonlinear Gauss-Seidel method, where the solution of an optimization problem is achieved by partitioning the variables in blocks and successively minimizing with respect to each block. The properties of the objective function that guarantee the convergence of such alternating scheme have been widely investigated in the literature and it is well known that, without suitable convexity hypotheses, the method may fail to locate the stationary points when more than two blocks of variables are employed. In this paper the general constrained non convex case is considered and three contributions are given. First, a general method allowing an approximate solution of each block minimization subproblem is devised and the related convergence analysis is developed, showing that the proposed inexact method has the same convergence properties of the standard nonlinear Gauss-Seidel method. Then, a cyclic block gradient projection method is analyzed, proving that it leads to stationary points for every number of blocks. Finally, the cyclic block gradient method is applied to large scale problems arising from the nonnegative matrix factorization approach. The results of a numerical experimentation on image recognition problems are also reported.

Keywords: Constrained optimization, alternating algorithms, gradient projection methods, nonnegative matrix factorization

1. Introduction

This paper deals with the solution of the constrained optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \Omega = \Omega_1 \times \Omega_2 \times ... \times \Omega_m \subseteq \mathbb{R}^n \end{array} \tag{1.1}$$

where, for all $i = 1,...,m$, Ω_i is a convex subset of $\mathbb{R}^{n_i}$ with $n_1 + ... + n_m = n$ and any vector in Ω can be partitioned into vector components as

$$x = (x_1, x_2, ..., x_m) \quad x_i \in \Omega_i.$$

The nonlinear Gauss–Seidel (GS) method, which is also known as nonlinear block coordinate descent or alternating optimization method, consists in solving (1.1) by successively minimizing the function f with respect to each block of variables over the corresponding constraints: given an initial point $x^{(0)} \in \Omega$, for $k = 0, 1, ...$ the iterate $x^{(k+1)} = (x_1^{(k+1)}, ..., x_m^{(k+1)})$ is computed such that for $i = 1, ..., m$ the block of variables $x_i^{(k+1)}$ is a solution of the subproblem

$$\min_{y \in \Omega_i} f(x_1^{(k+1)}, ..., x_{i-1}^{(k+1)}, y, x_{i+1}^{(k)}, ..., x_m^{(k)}). \tag{1.2}$$

The limit points of the sequence $\{x^{(k)}\}$ generated by the GS method are stationary for problem (1.1) when the minimum problem in (1.2) has an unique solution for $i = 1, ..., m$ [4, 6, 7].
On the other hand, it is well known that, when the function f is not strictly convex with respect to each block of variables, the GS method may fail to locate the stationary points. In particular, Powell in [22] produced an example with $m = 3$ where the limit points of the sequence generated by the GS method are not critical.
In [21, 23, 25] the convergence analysis of the GS method is performed devising appropriate convexity assumptions on the objective function while some variants of the GS method, as the proximal point modification [4, 14], have been proposed to handle the nonconvex case.
A different approach is proposed for the unconstrained case in [13], where the authors devise a set of conditions not necessarily related to the convexity of the objective function, which rather involve the property of the sequence $\{x^{(k)}\}$. Furthermore, they define globally convergent line–search based schemes which allow an approximate solution of the minimization with respect to some components.
The inexact solution of (1.2) has been considered for the case $m = 2$ also in [5], where the authors show the local convergence of an alternating algorithm where the exact minimization with respect to one component is replaced by a single Newton's iteration.
Several variants of the gradient descent method have also been proposed in the literature in the alternating optimization framework. In particular, for the constrained case, we mention the gradient projection method with fixed stepsize proposed in [24] and the analysis given in [10] from the point of view of the decomposition techniques, where the convergence results are based on a line–search condition and on the projected gradient properties.
For the general GS method in the nonconvex, constrained case, the more remarkable result can be found in [14], where the authors prove that when $m = 2$ every limit point of $\{x^{(k)}\}$ is a critical point of (1.1).
This convergence result has a special interest in the framework of the nonnegative matrix factorization (NMF) [18], a matrix decomposition technique with a variety of applications ranging from signal and image processing to document classification, to bioinformatics. This approach leads to optimization problems where the variables are naturally partitioned in two blocks and the objective function restricted to each block is not strictly convex.
Many NMF algorithms are based on the GS method where the solution of (1.2) is approximated by applying an iterative optimization method and stopping the procedure when some heuristic criterion is satisfied [3, 16, 19, 29].
In general, the features of many real life problems (e.g., the large number of variables or the nonlinearity of the objective function) could make the exact solution of (1.2) impractical. Moreover, dealing with applications, the hypotheses needed to prove the stronger convergence results may not be verified.
These observations motivated us to focus our analysis on the algorithmic features rather than on the theoretical properties of the objective function that guarantee the convergence of the alternating scheme. Thus, we deal with the general nonconvex case, revisiting the approaches in [13, 14] by means of the projected gradient properties and, in particular, we are interested on the effect of an approximate solution of (1.2).
To this end, we introduce two inexact alternating schemes where subproblem (1.2) is solved approximately.
The first one is a generalization of the GS method defined by means of a set of practical conditions that express the sufficient accuracy level of an approximate solution of (1.2) in terms of decrease of the objective function and improvement of the optimality.
Based on these conditions, we introduce a general inexact Gauss Seidel scheme allowing an inexact solution with respect to all components and we develop its convergence analysis without convexity hy-

potheses. Indeed, we prove that our inexact scheme has the same convergence properties established in [14] for the GS method and, in particular, when $m = 2$ every limit point is stationary for (1.1). The conditions that we propose here have practical interest, since they can provide robust stopping criterion to every two block alternating method, as the NMF algorithms mentioned above, which solves subproblem (1.2) with an iterative optimization procedure.
Then, we propose an inexact alternating algorithm where the approximate solution of (1.2) is computed by a finite number of gradient projection steps based on an Armijo line–search along the feasible direction with variable stepsize. We develop the related convergence analysis and our main result is the proof that the limit points of the generated sequence are critical for problem (1.1), for every $m \geqslant 1$.
The numerical experimentation is performed on the Powell counterexample and on some large scale NMF problems arising from image recognition problems. Furthermore, in order to better evaluate its practical performances, we compare the proposed method also with other recent NMF algorithms.
The paper is organized as follows: in Section 2 we present some preliminary results related to the Armijo condition and to the projected gradient direction, which are the basis of the convergence analysis; in Section 3 we present a general inexact Gauss-Seidel scheme and we discuss its convergence properties; the cyclic block gradient projection method is introduced in Section 4, where the Powell counterexample is also discussed; in Section 5 we present the numerical results of the proposed algorithm applied to the NMF problem, while our conclusions are offered in Section 6.

2. Definitions and preliminary results

Denote by $\nabla_i f(x)$ the partial gradient of f with respect to the i–th block of variables at the point x. Let us define the *projected gradient* of the function f with respect to the i-th block of variables at the point x as

$$\nabla_i^P f(x) = P_{\Omega_i}(x_i - \nabla_i f(x)) - x_i \tag{2.1}$$

where P_{Ω_i} is the projection operator on the convex set Ω_i, that is $P_{\Omega_i}(x) = \min_{y \in \Omega_i} \|x - y\|$ (here and in the following $\|\cdot\|$ indicates the euclidean norm). Then, a point x^* is stationary for problem (1.1) if and only if for all $i = 1, \ldots, m$ we have

$$\nabla_i^P f(x^*) = 0. \tag{2.2}$$

Thus, the quantity $\|\nabla_i^P f(x)\|$ can be considered as a measure of the optimality of the point x.
An alternating scheme that generates a sequence of points $x^{(k)}$ implicitly defines also m other sequences of the *partial updates*, that are

$$\begin{array}{lcl} z(k,0) & = & x^{(k)} \\ z(k,i) & = & (x_1^{(k+1)}, \ldots, x_i^{(k+1)}, x_{i+1}^{(k)}, \ldots, x_m^{(k)}) \;\; i = 1, \ldots, m-1 \\ z(k,m) & = & x^{(k+1)} \end{array}$$

For convenience we set also $z(k, m+1) = z(k+1, 1)$.
Given a sequence $z^{(k)}$ we define the *Armijo line–search procedure* with respect to the i–th component as in the Algorithm LS.

For the Armijo rule we recall the following basic result.

PROPOSITION 2.1 Let $\{z^{(k)}\}$ be a sequence of points in Ω. Assume that $z^{(k)}$ converges to some $\bar{z}$ and for $i \in \{1, \ldots, m\}$ let $\{d_i^{(k)}\}$ be a sequence of feasible directions such that

Algorithm LS Line–Search Algorithm

Let $\{z^{(k)}\}$ be a sequence of points in Ω and $\{d_i^{(k)}\}$ a sequence of descent directions, for a given $i \in \{1,...,m\}$. Fix $\delta_i, \beta \in (0,1)$ and compute $\lambda_i^{(k)}$ as follows:

1. Set $\lambda_i^{(k)} = 1$;
2. IF
$$f(z_1^{(k)},...,z_i^{(k)} + \lambda_i^{(k)} d_i^{(k)},...,z_m^{(k)}) \leqslant f(z^{(k)}) + \beta \lambda_i^{(k)} \nabla_i f(z^{(k)})^T d_i^{(k)} \tag{2.3}$$
THEN go to step 3.
ELSE set $\lambda_i^{(k)} = \delta_i \lambda_i^{(k)}$ and go to step 2.
3. END

(A1) there exists a number $M > 0$ such that $\|d_i^{(k)}\| \leqslant M$ for all k;

(A2) we have $\nabla_i f(z^{(k)})^T d_i^{(k)} < 0$ for all k;

(A3) we have $\lim_{k\to\infty} f(z^{(k)}) - f(z_1^{(k)},...,z_i^{(k)} + \lambda_i^{(k)} d_i^{(k)},...,z_m^{(k)}) = 0$, where $\lambda_i^{(k)}$ is computed with the algorithm LS.

Then, for each k the LS procedure terminates in a finite number of steps and, furthermore, $\lim_k \nabla_i f(z^{(k)})^T d_i^{(k)} = 0$.

Proposition 2.1 can be derived from known results (see [4, 14]), thus we omit the proof. It is worth stressing that the previous proposition applies to every sequences $\{z^{(k)}\}$ and $\{d_i^{(k)}\}$ satisfying the assumptions (A1)–(A3), not necessarily defined as $z_i^{(k+1)} = z_i^{(k)} + \lambda_i^{(k)} d_i^{(k)}$.
Consider now the projected gradient as feasible direction: then the following proposition holds (the proof can be found for example in [9, Lemmas 2.3,2.4]).

PROPOSITION 2.2 Set $d_i^{(k)} = \nabla_i^P f(z^{(k)})$.We have that

(i) if $\|d_i^{(k)}\| = \varepsilon > 0$, then $\nabla_i f(z^{(k)})^T d_i^{(k)} \leqslant -\varepsilon^2$;

(ii) if $z^{(k)}$ is bounded, $d_i^{(k)}$ is also bounded.

This means that the projected gradient computed in a non stationary point $z^{(k)}$ is a descent direction for the i–th subproblem (1.2).

3. An inexact Gauss-Seidel scheme

The convergence of many optimization algorithms is ensured by a sufficient decrease of the objective function value over two successive iterates, by means of the Armijo condition.
In this section we introduce the concept of sufficient decrease in the GS method: in particular, we assume that each block of variables $x_i^{(k+1)}$ for $i = 1,...,m$ just satisfies an Armijo condition instead of exactly solving problem (1.2). Based on this condition, we prove a first key property of a more general

Gauss-Seidel scheme.
To this end, we consider as descent direction for the Armijo rule the projected gradient at the point $z(k,i-1)=(x_1^{(k+1)},...,x_{i-1}^{(k+1)},x_i^{(k)},x_{i+1}^{(k)},...,x_m^{(k)})$, that is

$$d_i^{(k)}=\nabla_i^P f(z(k,i-1)).$$

More precisely, we consider any sequence $\{x^{(k)}\}$ where, at each block update, the value of f decreases at least as it would decrease by performing an Armijo line–search along the projected gradient direction. Thus, the sufficient decrease can be expressed as

$$f(x_1^{(k+1)},...,x_i^{(k+1)},...,x_m^{(k)})\leqslant f(x_1^{(k+1)},...,x_i^{(k)}+\lambda_i^{(k)}d_i^{(k)},...,x_m^{(k)}) \tag{3.1}$$

where $\lambda_i^{(k)}$ is computed with the algorithm LS such that

$$f(x_1^{(k+1)},...,x_i^{(k)}+\lambda_i^{(k)}d_i^{(k)},...,x_m^{(k)})\leqslant f(z(k,i-1))+\beta\lambda_k\nabla_i f(z(k,i-1))^T d_i^{(k)}$$

(see also [13, Algorithm 1]). A direct consequence of the condition (3.1) is the monotone decrease of the sequence $\{f(z(k,i))\}_{k,i}$; indeed, using the same arguments as in [14, Proposition 2], we can state the following result.

PROPOSITION 3.1 Let $\{x^{(k)}\}$ be any sequence such that (3.1) holds. Suppose that for some $i\in\{0,...,m\}$ the sequence $\{z(k,i)\}$ admits a limit point $\bar{z}$. Then, for every $j\in\{0,...,m\}$ we have

$$\lim_{k\to\infty} f(z(k,j))=f(\bar{z})$$

We point out that condition (3.1) does not necessarily require that $x_i^{(k+1)}$ is defined as $x_i^{(k)}+\lambda_i^{(k)}d_i^{(k)}$, thus any solution of (1.2) also satisfies (3.1).
However, condition (3.1) alone guarantees the stationarity of the limit points of the partial updates sequence $\{z(k,i)\}$ with respect to the block variable x_{i+1} if $i<m$, with respect to x_1 if $i=m$. This result can be proved by means of Propositions 2.1 and 2.2, and by employing similar arguments as in [14, Proposition 3].

PROPOSITION 3.2 Let $\{x^{(k)}\}$ be any sequence satisfying (3.1). Suppose that for some $i\in\{1,...,m\}$ the sequence $\{z(k,i)\}$ admits a limit point $\bar{z}$. Then, $\nabla_{i+1}^P f(\bar{z})=0$ if $i<m$, $\nabla_1^P f(\bar{z})=0$ if $i=m$.

Proof. Suppose first for simplicity that $i<m$ and assume by contradiction that $\|\nabla_{i+1}^P f(\bar{z})\|=2\varepsilon>0$. Indicating by K the set of indices such that $\{z(k,i)\}_{k\in K}$ converges to $\bar{z}$, thanks to the continuity of the projected gradient, for $k\in K$ sufficiently large we have that

$$\|\nabla_{i+1}^P f(z(k,i))\|>\varepsilon.$$

Then, the vector $d_{i+1}^{(k)}=\nabla_{i+1}^P f(z(k,i))$ satisfies

$$\nabla_{i+1} f(z(k,i))^T d_{i+1}^{(k)}\leqslant-\varepsilon^2<0. \tag{3.2}$$

Moreover, since $\{z(k,i)\}_{k\in K}$ is a convergent sequence, it is also bounded, thus the sequence $d_{i+1}^{(k)}$ is bounded and Proposition 2.1 holds. From (3.1) we have

$$\begin{aligned} f(z(k,i+1)) &\leqslant f(x_1^{(k+1)},...,x_i^{(k+1)},x_{i+1}^{(k)}+\lambda_{i+1}^{(k)}d_{i+1}^{(k)},...,x_m^{(k)}) \\ &\leqslant f(z(k,i)). \end{aligned}$$

Therefore, since the sequences $\{f(z(k,i+1))\}$ and $\{f(z(k,i))\}$ have the same limit (see Proposition 3.1), we obtain that

$$\lim_{k\to\infty, k\in K} f(z(k,i)) - f(x_1^{(k+1)},\ldots,x_i^{(k+1)},x_{i+1}^{(k)}+\lambda_{i+1}^{(k)}d_{i+1}^{(k)},\ldots,x_m^{(k+1)}) = 0.$$

Since $x_{i+1}^{(k)}$ converges to $\bar{z}_{i+1}$, and $\{d_{i+1}^{(k)}\}_{k\in K}$ is bounded, Proposition 2.1 implies that

$$\lim_{k\to\infty, k\in K} \nabla_{i+1} f(z(k,i))^T d_{i+1}^{(k)} = 0$$

which contradicts (3.2).
The same arguments can be applied also when $i=m$, since $z(k,m+1) = z(k+1,1)$. □

Remark 1 The conclusions of Proposition 3.2 are unaffected also if instead of the projected gradient (2.1) we choose the search direction as

$$d_i^{(k)} = P_{\Omega_i,(D_i^{(k)})^{-1}}(x_i^{(k)} - \alpha_i^{(k)} D_i^{(k)} \nabla_i f(z(k,i-i))) - x_i^{(k)},$$

where $\alpha_i^{(k)}$ is a positive parameter chosen in a bounded interval $[\alpha_{min},\alpha_{max}]$, with $\alpha_{min}>0$, and $D_i^{(k)}$ is a symmetric positive definite matrix whose eigenvalues are bounded above and below by two constants independent from k. Furthermore, $P_{\Omega_i,(D_i^{(k)})^{-1}}$ is the projection operator in the norm induced by the inverse of the matrix $D_i^{(k)}$, i.e.,

$$P_{\Omega_i,(D_i^{(k)})^{-1}}(z) = \min_{y\in\Omega_i} (z-y)^T (D_i^{(k)})^{-1}(z-y).$$

Indeed, Proposition 2.2 holds also for this scaled version of the projected gradient (see [8, 9]) and the proof of Proposition 3.2 can be obtained using the same arguments as before. More in general, every descent direction $d_{i+1}^{(k)}$ is allowed provided that (3.2) holds when $z(k,i)$ converges to a non stationary point (see also [13, Assumption 1]).

The decrease of the objective function required by the condition (3.1) implies only $\nabla_{i+1}^P f(\bar{z}) = 0$, when $\bar{z}$ is a limit point of $\{z(k,i)\}$. Stronger convergence results can be obtained, without convexity assumptions, by adding appropriate hypothesis on the sequence generated by the alternating algorithm, as proposed in [13] for the unconstrained case.
In the following section we consider two different strategies. First, we define a general inexact framework by means of a set of conditions related to the optimality of each iterate. Such scheme is a generalization of the standard GS method, which allows an approximate minimization with respect to all components and which is independent from the algorithm employed to compute an approximate solution of (1.2). Indeed, we prove in the inexact case the same properties established in [14] for the GS method.
A different kind of conditions is considered in section 3.2 and, even if they can be stated in a general form, they are more related to the algorithm employed to approximately solve (1.2). The inexact framework associated to these conditions can not be considered a generalization of the GS method, but, on the other hand, it has stronger convergence properties, as we will show in section 3.2.

3.1 *The optimality residual conditions*

The aim of this section is to show that the properties of the GS method can be achieved also when the subproblem (1.2) is inexactly solved for all $i = 1,...,m$. We define a conceptual scheme by means of a set of conditions on the iterates that can be easily implemented and that guarantee the stationarity of the limit points with respect to two consecutive block coordinates.
In particular, besides the sufficient decrease of the objective function expressed by the line–search condition (3.1), we require also an improvement of the optimality at each iterate. As a measure of the optimality of a point with respect to the constrained problem (1.2) we adopt the norm of the projected gradient (2.1). Indeed, through the following conditions we require a sufficient decrease of the projected gradient norm over two successive partial updates:

$$\begin{aligned} \|\nabla_i^P f(z(k,i))\| &\leqslant \eta\|\nabla_i^P f(z(k,i-1))\| \;\; i=1,...,m \\ \|\nabla_i^P f(z(k,i))\| &\leqslant \eta\|\nabla_{i-1}^P f(z(k,i-1))\| \;\; i=2,...,m \\ \|\nabla_1^P f(z(k+1,1))\| &\leqslant \tfrac{1}{\eta^{m-1}}\|\nabla_m^P f(z(k,m))\| \end{aligned} \tag{3.3}$$

where $\eta \in [0,1)$ is a forcing parameter.
We define the IGS method as every method that generates a sequence such that (3.1) and (3.3) hold. It is straightforward to see that the GS method is a special case of the IGS method.
A practical IGS algorithm can be realized by computing the iterates $x^{(k+1)}$ by means of any iterative optimization algorithm to the subproblem (1.2), stopping the inner iterations when the conditions (3.3) and (3.1) are satisfied.
The following theorem states a basic property of the IGS method.

Algorithm IGS Inexact Gauss–Seidel Method

Choose $x^{(0)} \in \Omega$;
FOR $k = 0,1,2,...$

1 Set $z(k,0) = x^{(k)}$;

2 FOR $i = 1,...,m$

2.1 Compute $x_i^{(k+1)}$ such that the conditions (3.1) and (3.3) are satisfied

2.2 Set $z(k,i) = (x_1^{(k+1)},...,x_i^{(k+1)},x_{i+1}^{(k)},...,x_m^{(k)})$

3 Set $x^{(k+1)} = z(k,m)$

THEOREM 3.1 Let $\{x^{(k)}\}$ be any sequence satisfying (3.1) and (3.3) and suppose that for some $i \in \{1,...,m\}$ the sequence $\{z(k,i)\}$ admits a limit point $\bar{z}$. Then, we have

(i) $\nabla_{i+1}^P f(\bar{z}) = 0$ if $i < m$, $\nabla_1^P f(\bar{z}) = 0$ if $i = m$;

(ii) $\nabla_i^P f(\bar{z}) = 0$.

Proof. The part (i) follows directly from Proposition 3.2. Let us prove that $\bar{z}$ is stationary with respect to the i-th block of variables.

For any k, from conditions (3.3) it follows that

$$\begin{aligned}\|\nabla^P_{i+1} f(z(k,i+1))\| &\geqslant \tfrac{1}{\eta}\|\nabla^P_{i+2} f(z(k,i+2))\| \\ &\geqslant \tfrac{1}{\eta^{m-i-1}}\|\nabla^P_m f(z(k,m))\| \\ &\geqslant \eta^i\|\nabla^P_1 f(z(k+1,1))\| \\ &\geqslant \eta\|\nabla^P_i f(z(k+1,i))\|\end{aligned}$$

from which we obtain

$$\|\nabla^P_{i+1} f(z(k,i+1))\| \geqslant \eta\|\nabla^P_i f(z(k+1,i))\| \geqslant \|\nabla^P_{i+1} f(z(k+1,i+1))\|.$$

By repeatedly applying the previous inequality, we get that

$$\|\nabla^P_{i+1} f(z(k,i+1))\| \geqslant \eta\|\nabla^P_i f(z(k+\ell,i))\| \tag{3.4}$$

for any $\ell = 1,2,....$ Let now $\{z(k_j,i)\}_j$ be a subsequence converging to $\bar{z}$ and let ℓ_j be a positive integer such that $k_{j+1} = k_j + \ell_j$. Recalling (3.3) and (3.4) we have that

$$\|\nabla^P_{i+1} f(z(k_j,i))\| \geqslant \frac{1}{\eta}\|\nabla^P_{i+1} f(z(k_j,i+1))\| \geqslant \|\nabla^P_i f(z(k_j+\ell_j,i)\| = \eta\|\nabla^P_i f(z(k_{j+1},i)\|$$

Since from Proposition 2.1 $\|\nabla^P_{i+1} f(z(k_j,i))\|$ goes to zero as $j \to \infty$, then also the sequence $\{\|\nabla^P_i f(z(k_j,i)\|\}$ converges to zero and, by continuity, $\nabla^P_i f(\bar{z}) = 0$. □

Therefore, the limit points of the partial updates generated by the inexact method are critical at least with respect to two consecutive components. Then, the IGS method (3.1)-(3.3) which allows an approximate solution with respect to all the components, has the same property established in [14, Proposition 3] for the exact GS method.
When $m = 2$ the conditions (3.3) can be simplified as follows:

$$\begin{aligned}\|\nabla^P_1 f(x_1^{(k+1)}, x_2^{(k)})\| &\leqslant \tfrac{1}{\eta}\min\left(\|\nabla^P_1 f(x_1^{(k)}, x_2^{(k)})\|, \|\nabla^P_2 f(x_1^{(k)}, x_2^{(k)})\|\right) \\ \|\nabla^P_2 f(x_1^{(k+1)}, x_2^{(k+1)})\| &\leqslant \eta\|\nabla^P_1 f(x_1^{(k+1)}, x_2^{(k)})\|,\end{aligned} \tag{3.5}$$

We call any algorithm that generates a sequence satisfying (3.5) and (3.1) a 2–Blocks Inexact Gauss–Seidel method and the following theorem states our main convergence result, which can be proved employing similar arguments as in the proof of Theorem 3.1.

THEOREM 3.2 Let $m = 2$, and let $\{x^{(k)}\}$ be a sequence such that (3.1) and (3.5) hold. Then, any limit point $\bar{z}$ of $\{x^{(k)}\}$ is a stationary point for problem (1.1).

The previous results are in some sense optimal for the IGS method, since it includes the GS algorithm as special case for $\eta = 0$. Indeed, recalling the Powell counterexample [22], for $m > 2$, we can not guarantee that the limit points of $\{x^{(k)}\}$ are critical for the problem (1.1). The difficulty is that the partial updates $z(k,i)$ may not have the same limit points for all $i = 1,...,m$. In the following section we devise a class of algorithms which overcome this difficulty and which are able to detect critical points of (1.1) for any $m \geqslant 1$.

3.2 *Global convergence conditions*

A sufficient condition such that a $\bar{z} = \lim_{k\to\infty, k\in K} z(k,i)$ is a limit point also for the sequence $\{z(k,i-1)\}$ (and viceversa) is

$$\lim_{k\to\infty, k\in K} \|x_i^{(k+1)} - x_i^{(k)}\| = 0. \tag{3.6}$$

Indeed, by definition of the partial updates, we have

$$\|z(k,i) - z(k,i-1)\| = \|x_i^{(k+1)} - x_i^{(k)}\|.$$

We observe that, in general, if we define $x_i^{(k+1)}$ as a solution of (1.2), property (3.6) is not satisfied. On the other side, (3.6) can be fulfilled by designing inexact algorithms where the new iterate is computed along a suitable direction. The following proposition states the convergence of such a class of algorithms, without restrictions on the number m of blocks.

PROPOSITION 3.3 Let $\{x^{(k)}\}$ a sequence and let $\Delta x_i^{(k)}$ the difference between the i–th coordinate block at two successive iterations, so that $x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)}$, $i = 1,...,m$. Assume that

(i) the Armijo condition (3.1) is satisfied for all k and all i;

(ii) if for a given subset K of indices $\lim_{k\to\infty, k\in K} \nabla_i^P f(z(k,i-1)) = 0$, then $\lim_{k\to\infty, k\in K} \Delta x_i^{(k)} = 0$ for all $i = 1,...,m$.

Then, every limit point $\bar{x}$ of $\{x^{(k)}\}$ is a limit point also for the sequences $\{z(k,i)\}$ for all $i = 1,...,m$ and it is stationary for problem (1.1).

Proof. Since (3.1) holds and $\bar{x}$ is a limit point of the sequence $\{z(k,0) = x^{(k)}\}$, Proposition 3.2 ensures that $\nabla_1^P f(\bar{x}) = 0$. Denoting by K the subset of indices such that $\{x^{(k)}\}_{k\in K}$ converges to $\bar{x}$, by hypothesis (ii) and by continuity of the projected gradient we have $\lim_{k\to\infty, k\in K} \Delta x_1^{(k)} = 0$. Then, $\bar{x}$ is a limit point also for the sequence $\{z(k,1)\}$. Invoking again Proposition 3.2, we have $\nabla_2^P f(\bar{x}) = 0$ and, as a consequence, $\lim_{k\to\infty, k\in K} \Delta x_2^{(k)} = 0$. By applying the same arguments, we prove by induction that $\lim_{k\to\infty, k\in K} z(k,i) = \bar{x}$ for all i and $\nabla_i^P f(\bar{x}) = 0$ for all $i = 1,...,m$, i.e. $\bar{x}$ is stationary for problem (1.1). □

The class of algorithms satisfying the hypotheses of Proposition 3.3 can be considered a generalization of the globally convergent line–search–based algorithms presented in [13, Section 7]. In the following section we present a special case of such class where the direction $\Delta x_i^{(k)}$ is the combination of a finite number of gradient projection steps.

4. A cyclic block coordinate gradient projection algorithm

The cyclic block coordinate gradient projection algorithm (CBGP) that is introduced in this section is related to the coordinate descent [21], gradient descent [27] and line–search based methods [13].
The algorithm CBGP is an inexact alternating method where an approximate solution of the i-th subproblem (1.2) is computed by a finite number of gradient projection steps. In the following we prove the convergence result by employing similar arguments as in the proof of Proposition 3.3.

Algorithm CBGP Cyclic Block Coordinate Gradient Projection Algorithm

Given $\delta_i, \beta \in (0,1)$, $L_1, ..., L_m$ finite positive integers. Choose $x^{(0)} \in \Omega$;
FOR $k = 0, 1, 2, ...$

1 Set $z(k,0) = x^{(k)}$;

2 FOR $i = 1, ..., m$

2.1 $x_i^{(k,0)} = x_i^{(k)}$;

2.2 Choose $L_i^{(k)}$ such that $1 \leqslant L_i^{(k)} \leqslant L_1$;

2.3 FOR $\ell = 0, ..., L_i^{(k)}$

2.3.1 Compute $d_i^{(k,\ell)} = \nabla_i^P f(x_1^{(k+1)}, ..., x_i^{(k,\ell)}, ..., x_m^{(k)})$;

2.3.2 Compute $\lambda_i^{(k,\ell)}$ with the algorithm LS

2.3.3 Set $x_i^{(k,\ell+1)} = x_i^{(k,\ell)} + \lambda_i^{(k,\ell)} d_i^{(k,\ell)}$

2.4 Set $x_i^{(k+1)} = x_i^{(k,L_i)}$

2.5 Set $z(k,i) = (x_1^{(k+1)}, ..., x_i^{(k+1)}, x_{i+1}^{(k)}, ..., x_m^{(k)})$

3 Set $x^{(k+1)} = z(k,m)$

THEOREM 4.1 Let $\{x^{(k)}\}$ be the sequence generated by the CBGP algorithm and assume that $\bar{x}$ is a limit point of $\{x^{(k)}\}$. Then, $\bar{x}$ is a limit point also for the sequences $\{z(k,i)\}$ for any $i = 1, ..., m-1$ and it is a stationary point for problem (1.1).

Proof. The proof can be obtained by induction on ℓ and i. Clearly, the sequence generated by the algorithm CBGP satisfies (3.1); then, since $\bar{x}$ is a limit point for $\{x^{(k)} = z(k,0)\}$, from Proposition 3.2 it follows that $\nabla_1^P f(\bar{x}) = 0$. Denoting by K a set of indices such that $\{x^{(k)}\}_{k \in K}$ converges to $\bar{x}$, we have $\lim_{k \to \infty, k \in K} \|d_1^{(k,0)}\| = 0$. By step 2.3.3 of the CBGP algorithm, it follows that $\lim_{k \to \infty, k \in K} \|x^{(k,1)} - x^{(k)}\| = 0$, i.e., $\bar{x_1}$ is a limit point also for the sequence $x_1^{(k,1)}$. From the continuity of the projected gradient we obtain that

$$\lim_{k \to \infty, k \in K} d_1^{(k,1)} = \lim_{k \to \infty, k \in K} P_{\Omega_i}(x_1^{(k,1)} - \nabla_1 f(x_1^{(k,1)}, x_2^{(k)}, ..., x_m^{(k)})) - x_1^{(k,1)} = \nabla_1^P f(\bar{x}) = 0.$$

Using the same arguments, by induction on ℓ we can conclude that, for each $\ell = 1, ..., L_1$, $\lim_{k \to \infty, k \in K} d_1^{(k,\ell)} = 0$, that yields

$$\|\Delta x_1^{(k)}\| = \|x_1^{(k+1)} - x_1^{(k)}\| \leqslant \sum_{\ell=0}^{L_1^{(k)}} \lambda_1^{(k,\ell)} \|d_1^{(k,\ell)}\| \leqslant \sum_{\ell=0}^{L_1} \lambda_1^{(k,\ell)} \|d_1^{(k,\ell)}\| \xrightarrow{k \to \infty, k \in K} 0$$

where the vectors $d_1^{(k,\ell)}$ and the parameters $\lambda_1^{(k,\ell)}$ for $\ell = L_1^{(k)} + 1, ..., L_1$ are defined in the same way as in the step 2.3 (see also [9, Lemma 2.6]). Thus, also the sequence $z(k,1) = (x_1^{(k+1)}, x_2^{(k)}, ..., x_m^{(k)})$ converges

to $\bar{x}$ for $k \in K$ and by Proposition 3.2 it follows $\nabla_2^P f(\bar{x}) = 0$.
Proceeding by induction on i and employing the same arguments used for $i = 1$, we prove that $\bar{x}$ is a limit point of the sequences $\{z(k,i)\}$ for any $i = 1,...,m-1$. As a result of this, invoking again Proposition 3.2, we can conclude that $\bar{x}$ is a stationary point of (1.1). □

Recalling Remark 1, we point out that the previous result still holds if in step 2.3.1 the search direction is a scaled version of the projected gradient

$$d_i^{(k,\ell)} = P_{\Omega_i,(D_i^{(k,\ell)})^{-1}}(x_i^{(k,\ell)} - \alpha_i^{(k,\ell)} D_i^{(k,\ell)} \nabla_i f(x_1^{(k,\ell)},...,x_i^{(k,\ell)},...,x_m^{(k)})) - x_i^{(k,\ell)}, \tag{4.1}$$

provided that the bounds for both the steplength parameter $\alpha_i^{(k,\ell)}$ and the minimum and maximum eigenvalues of the positive definite scaling matrix $D_i^{(k,\ell)}$ are independent from ℓ and k.

Remark 2 In order to show the differences between the Gauss-Seidel method and the inexact IGS and CBGP methods, we consider the following constrained version of the original Powell counterexample (see also [14]):

$$\begin{array}{c} \min \\ -10 \leqslant x_i \leqslant 10 \\ i = 1,2,3 \end{array} \quad f(x) = -x_1x_2 - x_2x_3 - x_1x_3 + \sum_{i=1}^3 \{(x_i-1)_+^2 + (-x_i-1)_+^2\} \tag{4.2}$$

where $(t-c)_+$ is defined as $(t-c)$ if $t \geqslant c$ and 0 otherwise. It is known that, starting from the point $(-1-\varepsilon, 1+\varepsilon/2, -1-\varepsilon/4)$, the GS method generates a sequence whose limit points are not stationary for f.
In Tables 1 and 2 we report the first 10 iterations of the GS, IGS algorithms with different values of the forcing term and of the CBGP method with different numbers of inner iterations, applied to (4.2). The iterates $x_i^{(k+1)}$ of the GS and IGS methods have been computed by applying a bisection method to the equation $\nabla_i^P f(x_1^{(k+1)},...,x_i^{(k+1)},x,x_{i+1}^{(k)},...,x_m^{(k)}) = 0$ with respect to the i–th variable. In the inexact case the bisection procedure has been terminated when conditions (3.1) and (3.3) were satisfied. We observe that both the algorithms IGS with $\eta = 10^{-1}, 10^{-3}$ and CBGP with $L_i^{(k)} = L_i = 1, 5, 5000$ converge to a stationary point of the constrained problem.

5. Application to the nonnegative matrix factorization

In order to better evaluate the effects of inexact solution of subproblem (1.2), we consider large scale problems arising from the nonnegative matrix factorization technique [18], which leads to problems of the following form:

$$\min_{W,H \geqslant 0} f(W,H) \equiv \frac{1}{2}\|WH - V\|_F^2 \tag{5.1}$$

where $V \in \mathbb{R}^{n \times p}$ with $V \geqslant 0$, $W \in \mathbb{R}^{n \times r}$, $H \in \mathbb{R}^{r \times p}$ and $\|\cdot\|_F$ denotes the Frobenius norm. The NMF determines a lower rank approximation of the matrix V and provides an approximate representation of the columns of V as combination of non negative basis vectors (the columns of W) with non negative coefficients (the rows of H). This factorization is a useful tool for several applications in data analysis.
There exists several variants of the NMF approach; for example, a different metric evaluating the distance between V and WH can be adopted and regularization terms or sparsity constraints can be included

Table 1. Behaviour of the GS and IGS algorithms on the constrained Powell counterexample (4.2).

	GS			IGS					
				$\eta = 10^{-1}$			$\eta = 10^{-3}$		
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	-2	1.5	-1.25	-2	1.5	-1.25	-2	1.5	-1.25
1	1.13	-1.06	1.03	1.09	-1.08	1.01	1.13	-1.06	1.03
2	-1.02	1.01	-1	-1.04	-1.01	-2.02	-1.02	1.01	-1
3	1	-1	1	-2.52	-3.27	-3.9	1	-1	1
4	-1	1	-1	-4.58	-5.24	-5.91	1	2	2.5
5	1	-1	1	-6.58	-7.24	-7.91	3.25	3.88	4.56
6	-1	1	-1	-8.58	-9.24	-9.91	5.22	5.89	6.56
7	1	-1	1	-10	-10	-10	7.22	7.89	8.56
8	-1	1	-1	-10	-10	-10	9.22	9.89	10
9	1	-1	1	-10	-10	-10	10	10	10
10	-1	1	-1	-10	-10	-10	10	10	10

[15, 17]. We will focus our numerical experience on the NMF problem written as in (5.1), but in principle also the variants mentioned above can be handled by the methods introduced in this paper.
Problem (5.1) is nonlinear and it has a natural structure of a two block problem. Many algorithms that have been recently proposed for solving the NMF problem consist in alternating methods which sequentially solve the two subproblems

$$W^{(k+1)} \leftarrow \min_{W \geqslant 0} f(W, H^{(k)}) \tag{5.2}$$

$$H^{(k+1)} \leftarrow \min_{H \geqslant 0} f(W^{(k+1)}, H) \tag{5.3}$$

Problems (5.2)–(5.3) consist in nonnegativity constrained least squares problems with multiple right hand side and are convex but, in general, not strictly convex, thus their solution may be not unique. The partial gradients of (5.1) are

$$\nabla_W f(W,H) = W(HH^T) - VH^T \tag{5.4}$$

$$\nabla_H f(W,H) = (W^T W)H - W^T V. \tag{5.5}$$

5.1 *Description of the algorithm*

For the solution of the NMF problem we propose the algorithm CBGP, where the search direction in step 2.3.1 is the projected gradient with a variable stepsize parameter along the negative gradient direction (see formula (4.1) with $D_i = I$). More precisely, for problem (5.2), the projected gradient direction at the inner iteration ℓ is computed as

$$d_W^{(k,\ell)} = P(W^{(k,\ell)} - \alpha_\ell \nabla_W f(W^{(k,\ell)}, H^{(k)})) - W^{(k,\ell)}$$

where P indicates the projection on the non negative orthant.
Furthermore, the steplength parameter α_ℓ is computed by an adaptive alternation of the Barzilai–Borwein

Table 2. Behaviour of the CBGP algorithm on the constrained Powell counterexample (4.2).

	CBGP								
	$L_i = 1$			$L_i = 5$			$L_i = 5000$		
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	-2	1.5	-1.25	-2	1.5	-1.25	-2	1.5	-1.25
1	0.25	-0.5	-1.15	1.1	-0.25	1.44	1.13	-1.06	1.03
2	-1.4	-3.05	-2.81	1.59	2.52	3.06	-1.02	1.01	-1
3	-3.42	-3.9	-4.29	3.79	4.42	5.1	1	-1	1
4	-4.76	-5.2	-5.65	5.76	6.43	7.1	-1	0.221	-1.39
5	-6.09	-6.54	-6.98	7.76	8.43	9.1	-1.58	-2.49	-3.04
6	-7.42	-7.87	-10	9.77	10	10	-3.76	-4.4	-5.08
7	-10	-10	-10	10	10	10	-5.74	-6.41	-7.07
8	-10	-10	-10	10	10	10	-7.74	-8.41	-9.07
9	-10	-10	-10	10	10	10	-9.74	-10	-10
10	-10	-10	-10	10	10	10	-10	-10	-10

(BB) rules [1], whose explicit expressions are

$$\alpha_\ell^{(1)} = \frac{{s^{(\ell-1)}}^T s^{(\ell-1)}}{{s^{(\ell-1)}}^T y^{(\ell-1)}}, \quad \alpha_\ell^{(2)} = \frac{{s^{(\ell-1)}}^T y^{(\ell-1)}}{{y^{(\ell-1)}}^T y^{(\ell-1)}},$$

where

$$s^{(\ell-1)} = \mathrm{vec}(W^{(k,\ell)} - W^{(k,\ell-1)}),$$
$$y^{(\ell-1)} = \mathrm{vec}(\nabla_W f(W^{(k,\ell)}, H^{(k)}) - \nabla_W f(W^{(k,\ell-1)}, H^{(k)}))$$

and $\mathrm{vec}(\cdot)$ denotes the vectorization operation. The previous choice of the steplength is motivated by the quasi-Newton approach where the inverse of the hessian is replaced by a multiple of the identity matrix $B(\alpha) = \alpha I$. Then, omitting the iteration number, the two BB formulas are given by $\alpha^{(1)} = \arg\min \|B(\alpha)s - y\|$ and $\alpha^{(2)} = \arg\min \|s - B(\alpha)^{-1}y\|$.
The recent literature on the steplength selection in gradient methods suggests to design steplength updating strategies by alternating the two BB rules. Here we will use the adaptive alternation strategy proposed in [9, 12], that has given remarkable convergence rate improvements in many different applications. Given an initial value α_0, the steplengths α_ℓ are defined by the following criterion:

IF $\alpha_\ell^{(2)}/\alpha_\ell^{(1)} \leqslant \tau_\ell$ THEN
  $\alpha_\ell = \min\left\{\alpha_j^{(2)},\ j = \max\{1, \ell - M_\alpha\}, \ldots, \ell\right\}$; $\quad \tau_{\ell+1} = \tau_\ell * 0.9$;
ELSE
  $\alpha_\ell = \alpha_\ell^{(1)}$; $\quad \tau_{\ell+1} = \tau_\ell * 1.1$;
ENDIF

where M_α is a prefixed non-negative integer and $\tau_1 \in (0,1)$. The same techniques are employed for the computation of $H^{(k+1)}$ starting from $(W^{(k+1)}, H^{(k)})$.
From the computational point of view, the main task required by the algorithm is the matrix-vector product, needed for updating the gradient and the objective function. Following the suggestion in [19], when $r \ll p$ (and $r \ll n$), it is convenient to save the $r \times r$ matrix W^TW (and HH^T respectively) at the

Table 3. Test problems

Problem	n	p	r	Problem	n	p	r
`cbcl` [1]	361	2429	49	`essex` [5]	9000	150	40
`orl` [2]	10304	400	25	`yale` [6]	77760	75	15
`nat` [3]	288	10000	72	`georgia` [7]	17316	750	50
`umist` [4]	10304	300	20				

[1] http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html;

[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html;

[3] http://www.cs.helsinki.fi/u/phoyer/software.html;

[4] http://www.shef.ac.uk/eee/research/vie/research/face.html;

[5] http://cswww.essex.ac.uk/mv/allfaces/index.html;

[6] http://cvc.yale.edu/projects/yalefaces/yalefaces.html;

[7] http://www.anefian.com/research/face_reco.htm.

beginning and use it for all the subsequent updates.
To assess the performance of the algorithm, we consider the image databases described in Table 3. The columns of V are formed by stacking the images columnwise and then normalizing the result. This is the same technique employed in [15], and the Matlab source codes for the generation of the matrix V related to the `cbcl`, `orl` and `nat` databases are included in the `nmfpack` package downloadable from *http://www.cs.helsinki.fi/u/phoyer/software.html*.
The accuracy of the results is evaluated on both the objective function value and the norm of the projected gradient

$$f(W^{(k)},H^{(k)}), \qquad \|\nabla^P f(W^{(k)},H^{(k)})\|.$$

As we proved in Section 4, the convergence of the CBGP method is ensured for every number of inner iterations, whose maximum value, L_W and L_H, is *a priori* defined. However, we include also an adaptive stopping condition based on the decrease of the projected gradient norm, with the aim of avoiding unnecessary computations.
We experienced several stopping criteria, but the more effective rule is similar to the one suggested in [19] and consists, for the problem (5.2) in

$$\|\nabla^P_W f(W^{(k,\ell)},H^{(k)})\| \leqslant \eta_W^{(k)}$$

where the adaptive tolerance $\tau_W^{(k)}$ is initialized as $\eta_W^{(0)} = 10^{-3}\|\nabla^P f(W^{(0)},H^{(0)})\|$ and

$$\eta_W^{(k)} = \begin{cases} 0.1 \cdot \eta_W^{(k-1)} & \text{if } \eta_W^{(k-1)} \geqslant \min(\|\nabla^P f(W^{(k)},H^{(k)})\|, \|\nabla^P_W f(W^{(k)},H^{(k)})\|) \\ \eta_W^{(k-1)} & \text{otherwise} \end{cases} \tag{5.6}$$

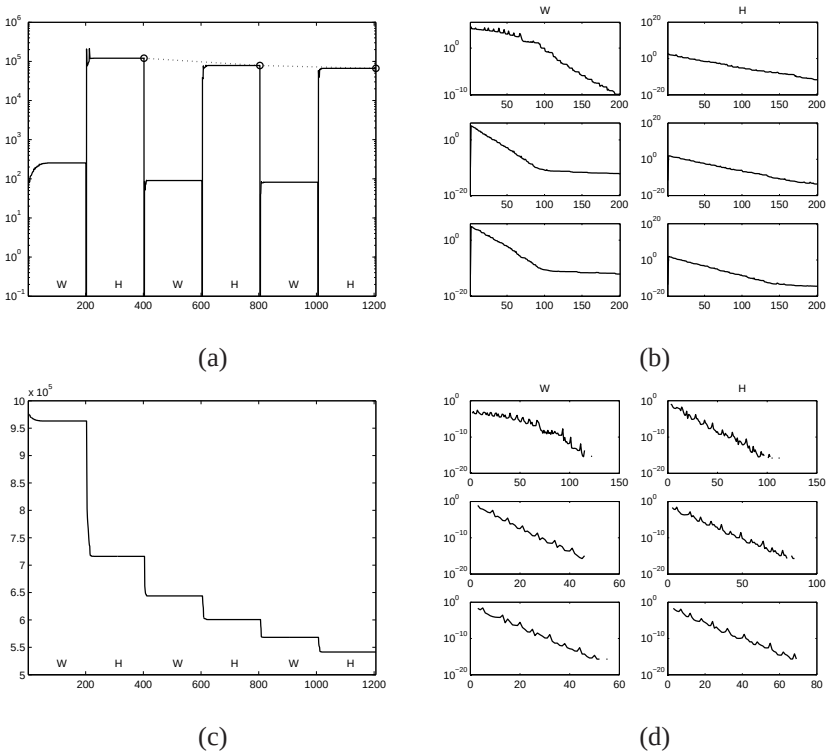
The same condition is introduced also for problem (5.3).

5.2 *Preliminary experiments*

In order to show the behaviour of the alternating algorithm, we report the result of an experiment where we monitor the optimality of each inner iteration not only with respect to the related subproblem, but

also with respect to the leading problem (5.1) by explicitly computing the norm of the projected gradient for each W and H step.
The result in Figure 1 is related to the first 3 iterations of CBGP applied to the problem `nat` with $L_W^{(k)} = L_H^{(k)} = 200$ gradient projection iterations. It can be observed that, as one should expect, the norm of the partial projected gradient improves over the inner iterations (top right panel); on the other hand, the optimality with respect to the leading problem (5.1) is not improved after the first few tens of iterations (top left). A similar behaviour is observed on the objective function value.
This suggests that exactly solving subproblems (5.2)–(5.3), which is not needed from a theoretical point of view, is not convenient in a practical framework either.

FIG. 1. Projected gradient and objective function value over the inner iterations. (a) Plot of the projected gradient norm $\|\nabla^P f(W,H)\|$ over the inner W and H iterations; the circles indicates the values at the first three outer iterations. (b) Plot of the norm of the partial projected gradients $\nabla^P_W f(W,H)$ and $\nabla^P_W f(W,H)$ over the inner iterations (columns W and H respectively). (c) Objective function value over the inner iterations. (d) Relative difference of the objective function values over two successive inner iterations.



5.3 *Benchmark tests*

We compare the performances of the CBGP algorithm also with other algorithms recently proposed for the NMF, which are based on the alternating solution of subproblems (5.2) and (5.3). The first algorithm [19], denoted by 'Lin' in the following, consists in an iterative solution of the W and H

subproblems by means of a gradient projection method; such method differs from the one presented in the previous section since it computes the new point along the projection arc instead of the feasible direction (4.1) (see also [4]). The steplenght is adaptively chosen as the largest one satisfying the Armijo condition. The inner iterations are stopped when the partial projected gradient is reduced by some factor with respect to the initial point $(W^{(0)},H^{(0)})$. We adopt the Matlab implementation given by the author of the Lin gradient projection algorithm, named `nlssubprob`, which is downloadable from `http://www.csie.ntu.edu.tw/~cjlin/nmf/nmf.m`.
The other algorithm ('KP') is based on the exact solution of the W and H subproblems by means of an active set method using a combinatorial approach (see [17] and references therein). Also in this case, we adopt the implementation provided by the authors available on their web page.
We consider also the 'GPSR-BB' method proposed in [29, 11], which is a gradient projection method based on the first BB formula. In particular, we adopt the implementation given in [29, Algorithm 4], named `nmf_gpsr_bb`, to find a solution of both problems (5.2)–(5.3), stopping the gradient iterations with the same tolerance used for the CBGP method (see also [11, Algorithm 5.3]).
All methods showed to be very efficient on different kinds of test sets and here we evaluate their performances on the test problems listed in Table 3.
We initialize the algorithms with the same starting point obtained by generating two nonnegative random matrices $\bar{W}$ and $\bar{H}$ (we take the absolute value of normally distributed random matrices, as in the `nmfpack` codes); then, we perform one step of the multiplicative algorithm proposed in [18], i.e.,

$$W^{(0)} = \bar{W}.* \frac{V\bar{H}^T}{W(HH^T)}\ ,\ H^{(0)} = \bar{H}.* \frac{{W^{(0)}}^T V}{{W^{(0)}}^T W^{(0)}\bar{H}},$$

where the product $.*$ and the quotient are componentwise. The aim of this choice is to produce a more accurate initial guess: in fact, the projected gradient norm at $(\bar{W},\bar{H})$ can be several order of magnitude larger than the same quantity computed in the point $(W^{(0)},H^{(0)})$ defined as above.
As stopping condition for the alternating algorithm we consider the following one:

$$\|\nabla^P f(W^{(k)},H^{(k)})\| \leqslant \varepsilon \|\nabla^P f(W^{(0)},H^{(0)})\| \tag{5.7}$$

but we include also as safeguard values a maximum number of iterations N_{max} and a limit of the computational time T_{max}. All the experiments have been carried out on an Intel Pentium 4 Dual Core, 3GHz, with 1.5Gb RAM running Linux operating system equipped with Matlab 7.0.1(R14).
In Figure 2 we report the norm of the projected gradient over the computational time. The algorithms have been stopped when condition (5.7) was satisfied with $\varepsilon = 10^{-4}$, or when the maximum number of iterations $N_{max} = 1000$ or the time limit of $T_{max} = 14400$ seconds was reached. In the KP algorithm the time of the computation of the projected gradient was excluded since such operation is not actually needed by the algorithm but only performed for benchmark reasons.

From Figure 2 we can observe the oscillating decrease of the projected gradient norm over the iterations of all the considered algorithms. For this reason it could be difficult to compare the results, but for a small value of the tolerance ε in (10^{-3}–10^{-4}) the accuracy of the approximate solution in terms of the objective function value is comparable. To give a more complete insight on the behaviour of the algorithms, in Tables 4–5 we report the number of iterations needed to each algorithm to satisfy the stopping conditions with different values of ε in (5.7); we indicate also the corresponding objective function value and computational time. Furthermore, for the CBGP, Lin and GPSR-BB algorithms the total number of inner iterations needed for the solution of the W and H subproblems ('itW' and 'itH')

FIG. 2. Projected gradient norm over the computational time in seconds.

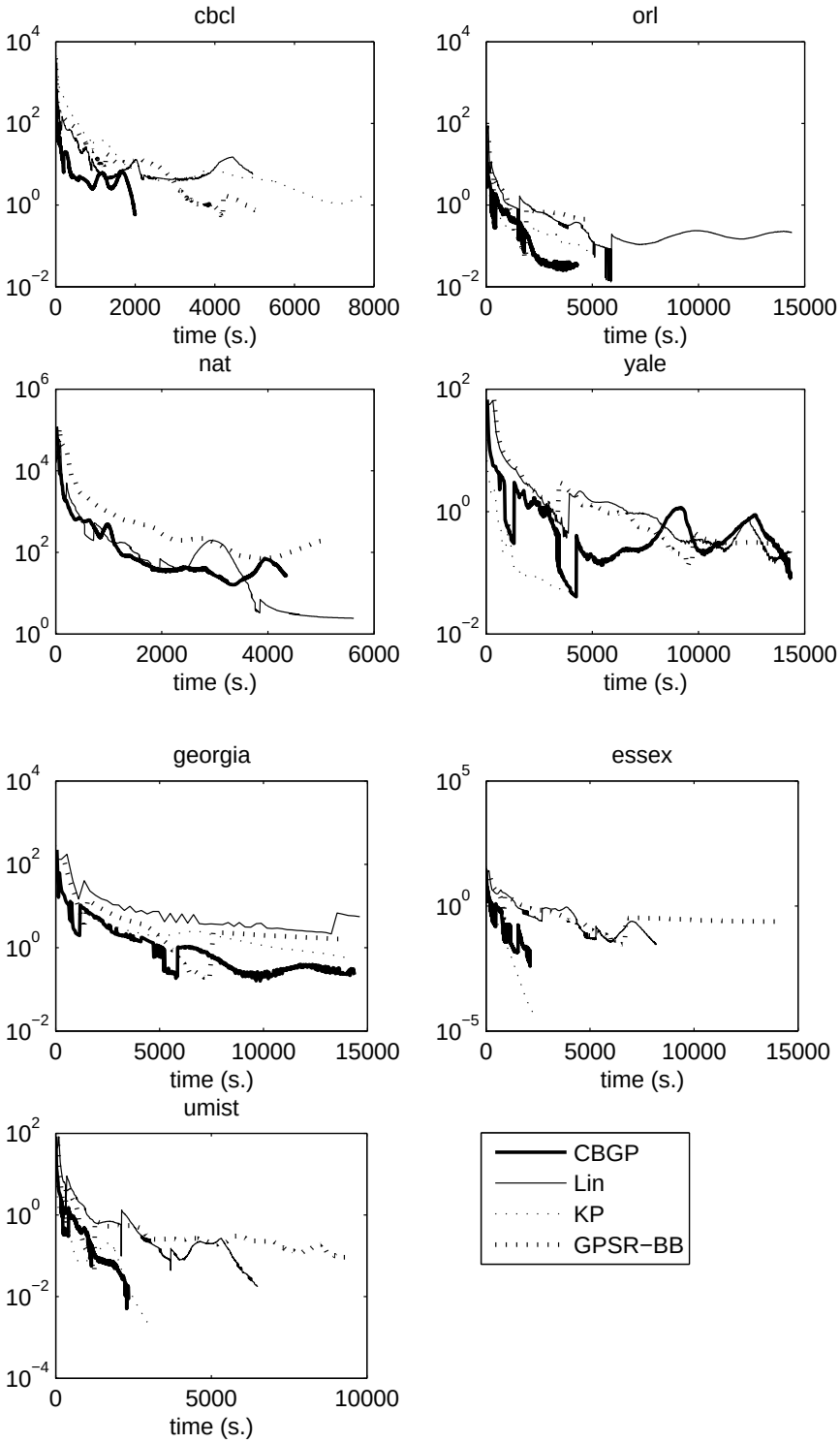


Table 4. Benchmark test on the NMF problem: algorithms CBGP and Lin

		CBGP					Lin				
	ε	it	itW	itH	$f(W,H)$	time (s)	it	itW	itH	$f(W,H)$	time (s)
cbcl	10^{-1}	8	362	70	1003.93	6.8	10	789	239	1005.37	61.8
	10^{-2}	87	2512	795	811.70	75.0	97	4781	2358	812.97	590.1
	10^{-3}	528	8916	4351	795.43	749.7	*	*	*	*	*
orl	10^{-1}	4	160	38	15.30	42.0	5	538	64	15.63	184.0
	10^{-2}	28	757	258	13.81	258.2	66	3515	433	13.67	1459.2
	10^{-3}	347	4887	3172	13.37	1851.2	420	12305	3878	13.37	5420.9
nat	10^{-1}	11	405	61	3.641e5	70.6	12	905	84	3.611e5	118.9
	10^{-2}	53	1623	216	3.443e5	266.9	58	2682	281	3.446e5	398.1
	10^{-3}	249	5276	1014	3.428e5	1220.3	258	7437	1110	3.432e5	1519.8
	10^{-4}	*	*	*	*	*	627	9322	3088	3.431e5	3907.4
yale	10^{-1}	6	246	149	20.57	273.8	7	670	731	20.61	958.6
	10^{-2}	24	716	321	18.23	919.7	46	2088	1864	17.90	3386.8
	10^{-3}	228	2858	3307	16.62	3779.9	*	*	*	*	*
georgia	10^{-1}	2	121	15	43.07	64.7	5	1000	145	35.21	1100.2
	10^{-2}	25	851	182	30.46	955.8	51	9736	666	29.43	13258.5
	10^{-3}	209	4739	1729	28.29	5772.1	*	*	*	*	*
essex	10^{-1}	4	191	32	4.23	44.1	17	3005	342	3.53	1051.4
	10^{-2}	49	1028	416	3.25	298.6	88	7032	1013	3.26	2524.4
	10^{-3}	552	3947	4494	3.09	1212.8	*	*	*	*	*
	10^{-4}	*	*	*	*	*	*	*	*	*	*
umist	10^{-1}	8	268	105	10.18	62.7	6	873	49	11.15	222.4
	10^{-2}	38	862	307	9.43	224.1	51	3788	430	9.49	1106.9
	10^{-3}	333	4069	2488	9.18	1144.6	370	11713	2894	9.21	3611.5
	10^{-4}	934	7645	7760	9.17	2279.0	*	*	*	*	*

Table 5. Benchmark test on the NMF problem: algorithms KP and GPSR-BB

		KP			GPSR-BB				
	ε	it	$f(W,H)$	time (s)	it	itW	itH	$f(W,H)$	time (s)
cbcl	10^{-1}	25	879.34	273.3	8	840	669	1023.44	52.8
	10^{-2}	150	798.86	1321.6	73	3490	6236	812.73	773.7
	10^{-3}	731	796.26	5777.9	349	11314	20389	795.71	3016.0
orl	10^{-1}	6	14.46	17.2	6	662	829	15.86	233.5
	10^{-2}	110	13.45	523.2	46	2680	7209	13.75	1095.7
	10^{-3}	906	13.36	4694.0	102	4153	9089	13.67	1713.1
nat	10^{-1}	f	f	f	14	579	245	3.541e5	279.2
	10^{-2}	f	f	f	49	1201	738	3.441e5	862.1
	10^{-3}	f	f	f	193	3348	2721	3.431e5	3260.2
	10^{-4}	f	f	f	*	*	*	*	*
yale	10^{-1}	9	19.58	52.3	7	761	345	20.60	1081.4
	10^{-2}	119	16.59	590.3	163	4299	6758	16.59	6984.5
	10^{-3}	675	16.49	2863.1	*	*	*	*	*
georgia	10^{-1}	9	31.10	344.4	5	879	671	41.25	892.0
	10^{-2}	75	28.64	4074.1	39	3468	6563	29.80	4855.1
	10^{-3}	*	*	*	88	4911	8226	29.46	6993.0
essex	10^{-1}	13	3.46	53.3	9	1277	1396	3.42	642.4
	10^{-2}	180	3.09	538.7	164	7099	28502	2.15	3932.5
	10^{-3}	360	3.08	964.0	*	*	*	*	*
	10^{-4}	559	3.08	1433.8	*	*	*	*	*
umist	10^{-1}	10	9.78	24.9	6	596	874	11.54	154.7
	10^{-2}	91	9.23	262.5	16	1014	1054	10.99	294.3
	10^{-3}	226	9.18	670.3	105	3558	9140	9.45	1105.0
	10^{-4}	848	9.17	2526.3	*	*	*	*	*

are reported. The asterisk indicates that the corresponding tolerance was not reached within the prefixed iterations and time limits, while the symbol 'f' indicates a failure of the algorithm KP on the problem `nat` (the code exited with an error message).
The best results in terms of computational time are obtained by the CBGP and KP algorithms. In particular, the CBGP algorithm generally requires a smaller number of inner iterations than the Lin and GPSR-BB algorithms.
Indeed, CBGP is based on a two parameters gradient projection method, where $\lambda_i^{(k,\ell)}$ guarantees the sufficient decrease and $\alpha_i^{(k,\ell)}$ can be any bounded positive parameter. The choice of the steplength parameter is crucial for the effectiveness of a gradient method, and the alternation of the two BB formulas can significantly improve the convergence speed.
On the other hand, the Lin algorithm is based on the direction obtained along the projection arc: it involves only one steplength parameter, which is chosen as the largest value guaranteeing the sufficient decrease of the objective function [20]. Moreover, in the GPSR-BB method, the benefits of a suitable alternation between the two BB rules [12, 9] are not exploited, since only the first formula is employed.
To better evaluate the effectiveness of the iterative inner solvers, we compare their performances when applied to the same nonnegative least square problem.
To this end, we save the intermediate results $(W^{(k)},H^{(k)})$ and $(W^{(k+1)},H^{(k+1)})$ obtained with $k = 0,100,200$ iterations of the Lin's NMF algorithm. Then, we solve the W and H problems

$$\min_{W\geqslant 0} f(W,H^{(k)}) \quad \text{and} \quad \min_{H\geqslant 0} f(W^{(k+1)},H)$$

with the three gradient projection methods underlying CBGP, Lin and GPSR-BB algorithms:

- the gradient projection method (`GP`) described in the step 2.3 of Algorithm CBGP and in section 5.1;
- the `nlssubprob` routine given in [19];
- the `nmf_gpsr_bb` function in [29].

For the test problems `cbcl`, `orl` and `nat`, the iterations number and CPU time (in seconds) needed to reach a fixed tolerance $\eta = 10^{-4}$ on the partial projected gradient norm are listed in table 6 (the symbol '*' indicates that the prefixed tolerance was not satisfied within 10000 iterations). Figure 3 shows the decrease of the projected gradient norm over the iterations number for the problem `cbcl` with $k = 100$.
We can observe that the best results are obtained by the `GP` method, which requires a significantly smaller number of iterations and computational time with respect to the other solvers (see also [2] for a recent comparison of gradient projection methods for nonnegative least squares problems).
Thus, the good performances of CBGP on the NMF problem with respect to Lin and GPSR-BB can be motivated by the effectiveness of the inner gradient projection method. Furthermore, comparing the results of CBGP and KP in terms of both number of iterations and computational time, it turns out that the exact solution of the two subproblems not always leads to a faster convergence. In particular, an efficient iterative solver is convenient when the problem size is large.

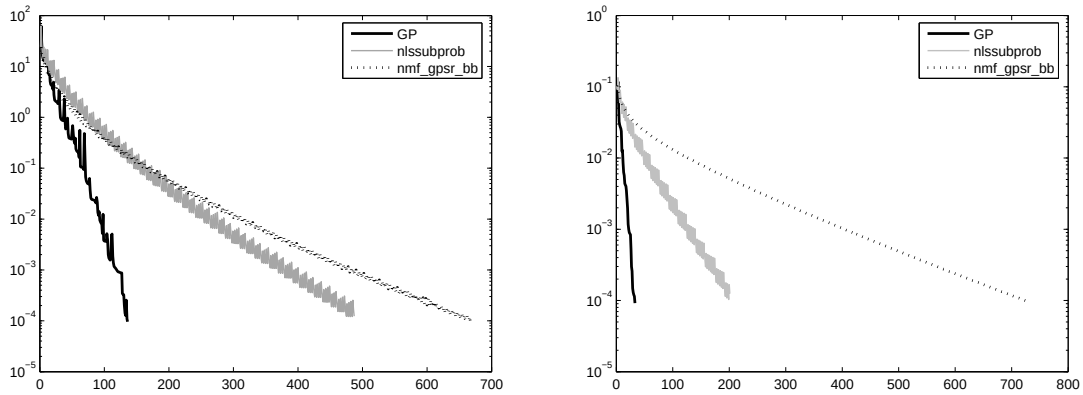
6. Conclusions and perspectives

In this paper we proposed an inexact nonlinear Gauss-Seidel method and we developed the related convergence analysis, proving that the inexact scheme has the same properties of the standard Gauss-Seidel

Table 6. Inner solvers comparison.

		$k=0$				$k=100$				$k=200$			
		W		H		W		H		W		H	
		it.	time	it	time	it.	time	it	time	it	time	it	time
cbcl	GP	113	1.8	330	51.1	136	2.3	33	4.9	138	2.3	29	4.4
	nlssubprob	388	8.1	1288	230.5	488	9.8	202	37.4	492	9.8	170	32.6
	nmf_gpsr_bb	827	13.3	*	*	668	10.4	724	104.5	729	11.3	595	81.3
orl	GP	61	17.7	166	1.4	53	16.9	27	0.2	49	15.6	26	0.2
	nlssubprob	319	118.9	306	3.1	193	72.9	51	0.5	172	68.1	36	0.4
	nmf_gpsr_bb	403	148.4	*	*	196	72.5	1836	14.9	176	66.7	1269	10.5
nat	GP	133	2.8	96	89.3	80	1.7	12	10.6	69	1.5	13	11.3
	nlssubprob	1018	26.5	205	232.0	210	5.0	42	42.8	169	4.1	34	34.0
	nmf_gpsr_bb	*	*	728	759.0	60	1.2	42	44.6	62	1.3	35	37.1

FIG. 3. Inner solver comparison: projected gradient norm over the iteration number.



method.

Furthermore, we proved a stronger property of the cyclic block gradient projection method, that leads to stationary points even with more than two blocks of variables. Finally we analyzed the alternating approach applied to large scale problems arising in the NMF framework, showing that the performances of the CBGP method are comparable to the ones of other recently proposed methods.

The future research will deal with both theoretical and practical issues. In particular, it would be interesting to extend the theoretical results obtained in the present work to non cyclic choices of the optimization variables of each subproblem. Furthermore, the encouraging results obtained on the NMF problems suggest to consider other formulations of such problem, that can also be handled by the CBGP method.

References

[1] J. Barzilai and J.M. Borwein. Two point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[2] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, 26(025004), 2010.

[3] M. Berry, M. Browne, A. Langville, P. Pauca, and R. J. Plemmons. Algorithms and applications for approximation nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173, 2007.

[4] D. P. Bertsekas. *Nonlinear Progrmming*. Athena Scientific, 2nd edition, 1999.

[5] J. C. Bezdek and R. J. Hathaway. Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate. *Journal of Optimization Theory and Applications*, 71:503–516, 1991.

[6] J. C. Bezdek and R. J. Hathaway. Some notes on alternating optimization. In *Advances in Soft Computing AFSS02*, volume 2275 of *Lecture Notes in Computer Science. Proceedings of the 2002 AFSS International Conference on Fuzzy Systems*, pages 288–300, 2002.

[7] J. C. Bezdek and R. J. Hathaway. Convergence of alternating optimization. *Neural, Parallel and Scientific Computing*, 11(4):351–368, 2003.

[8] E. G. Birgin, J. M. Martinez, and M. Raydan. Inexact spectral projected gradient methods on convex sets. *IMA J. Numer. Anal.*, 23:539–559, 2003.

[9] S. Bonettini, R. Zanella, and L. Zanni. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, 25(1):015002 (23pp), 2009.

[10] C.-C. Chang, C.-W. Hsu, and C.-J. Lin. The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 11:1003–1008, 2000.

[11] A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multy-way Data Analysis and Blind Source Separation*. Wiley and Sons, 2009.

[12] G. Frassoldati, G. Zanghirati, and L. Zanni. New adaptive stepsize selections in gradient methods. *J. Industrial and Management Optim.*, 4(2), 2008.

[13] L. Grippo and M. Sciandrone. Globally convergent block–coordinate techniques for unconstrained optimization. *Optimization Methods and Software*, 10:587–637, 1999.

[14] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations Research Letters*, 26:127–136, 2000.

[15] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[16] D. Kim, S. Sra, and I. S. Dhillon. Fast Newton-type methods for the least squares nonnegative matrix approximation problem. In *in Data Mining, Proceedings of SIAM Conference on*, pages 343–354, 2007.

[17] H. Kim and H. Park. Non–negative matrix factorization based on alternating non–negativity constrained least squares and active set method. *SIAM J. Matrix Anal. Appl.*, 30(2):713–730, 2008.

[18] D. D. Lee and H. S. Seung. Learning the part of objects from non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[19] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19:2756–2779, 2007.

[20] C.-J. Lin and J. J. Moré. Newton's method for large scale bound constrained problems. *SIAM J. on Optimization*, 9:1100–1127, 1999.

[21] Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

[22] M. J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4:193–201, 1973.

[23] P. Tseng. Decomposition algorithm for convex differentiable minimization. *Journal of Optimization Theory and Applicazions*, 70:109–135, 1991.

[24] P. Tseng. On the rate of convervence of partially asyncronous gradient projection algorithm. *SIAM J. Optimization*, 1:603–619, 1991.

[25] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applicazions*, 109:475–494, 2001.

[26] P. Tseng and S. Yun. Block coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of Optimization Theory and Applicazions*, 140:513–535, 2009.

[27] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program., Ser. B*, 117:387–423, 2009.

[28] S. Yun and K.-C. Toh. A coordinate gradient descent method for ℓ_1 regularized convex minimization. *Comput. Optim. Appl.*, (to appear), 2009.

[29] R. Zdunek and A. Cichocki. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large–scale problems. *Computational Intelligence and Neuroscience*, 2008:939567(13 pp), 2008.