Reduced Gradient Method and its Generalization via Stochastic Perturbation

Abdelkrim El Mouatasim
Department of Mathematics,
Faculty of Sciences, Jazan University,
B.P. 114, Jazan, Saudi Arabia.
E-mail: a.mouatasim@gmail.com

Abstract: In this paper, the global optimization of a nonconvex objective function under linear and nonlinear differentiable constraints is studied, a reduced gradient and GRG descent methods with random perturbation is proposed and it is desired to establish the global convergence of the algorithm. Some numerical examples are also given by the problems of statistical, octagon, mixture, alkylation and pooling.

Keywords: Stochastic perturbation, constraints optimization, reduced gradient method and its generalization, Nanconvex optimization.

Received for JNM on November 19, 2007.

1 Introduction

The general constrained optimization problem is to minimize a nonlinear function subject to linear or nonlinear constraints. Two equivalent formulations of this problem are useful for describing algorithms. They are

\begin{align}
\text{Minimize: } & f(x) \\
\text{subject to: } & g_i(x) = 0, \; i = 1, \ldots, m_l \\
& g_i(x) \leq 0, \; i = m_l + 1, \ldots, m \\
& \ell \leq x \leq u
\end{align}

where \( f \) and each \( g_i : \mathbb{R}^n \rightarrow \mathbb{R} \) is twice continuously differentiable function and the lower- and upper-bound vectors, \( \ell \) and \( u \), may contain some infinite components; and

\begin{align}
\text{Minimize: } & f(x) \\
\text{Subject to: } & h(x) = 0 \\
& x \geq 0
\end{align}
where $h$ maps $\mathbb{R}^m$ to $\mathbb{R}$, the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and the restriction $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \ldots, m$ (the component of $h$) are twice continuously differentiable functions. For the case of multi objective optimization see [10].

The main technique that has been proposed for solving constrained optimization problems in this work is reduced gradient method and its generalization. We are mainly interested in the situation where, on one hand $f$ is not convex and, on the other hand, the constraints are in general not linear; to get equality constraints one might think about introducing slack variables and incorporating them into the functions $g_i$.

The problem (1.2) can be numerically approached by using reduced gradient method or its generalizations, which generates a sequence $\{x^k\}_{k \geq 0}$, where $x^0$ is an initial feasible point and, for each $k > 0$, a new feasible point $x^{k+1}$ is generated from $x^k$ by using an operator $Q_k$ (see section 3). Thus the iterations are given by:

$$k \geq 0 : x^{k+1} = Q_k(x^k).$$

(1.3)

In order to prevent convergence to local minimum, various modifications of these basic methods have been introduced in the literature. For instance, the reduced gradient method has been considered by [1]; the random perturbations projected gradient method in [2]; the sequential convex programming in [24]; the generalized reduced gradient in [5, 7, 25–26, 32, 34]; sequential quadratic programming in [15, 30]; nonlinear programming in [3, 20, 27]; stochastic perturbation of active set [11]. Moreover, stochastic or evolutionary search can be introduced, but these methods are usually considered as expensive and not accurate: on the one hand, the flexibility introduced by randomization implies often a large number of evaluations of the objective function and, on the other hand, the pure random search generates many points which do not improve the value of the objective function. This remark has led some people to introduce a controlled random search (see for instance [2, 4, 6, 9, 23, 28, 33]).

In such a method, the sequence $\{x^k\}_{k \geq 0}$ is replaced by a random vectors sequence $\{X^k\}_{k \geq 0}$ and the iterations are modified as follows:

$$k \geq 0 : X^{k+1} = Q_k(X^k) + P_k$$

(1.4)

where $P_k$ is a suitable random variable, called the stochastic perturbation. The sequence $\{P_k\}_{k \geq 0}$ goes to zero slowly enough in order to prevent convergence to a local minimum (see Section 4). The reduced gradient method and its generalization are recalled in Section 3, while the notations are introduced in Section 2. The results of numerical experiments will be given in Section 5.

## 2 Notations and assumptions

We denote by $\mathbb{R}$ the set of the real numbers $(-\infty, +\infty)$, $\mathbb{R}^+$ the set of positive real numbers $[0, +\infty)$, $E = \mathbb{R}^n$, the $n$-dimensional real Euclidean space. For
\( \mathbf{x} = (x_1, x_2, \ldots, x_n)^t \in E, \) \( \mathbf{x}^t \) denotes the transpose of \( \mathbf{x} \). We denote by \( \| \mathbf{x} \| = \sqrt{\mathbf{x}^t \mathbf{x}} = (x_1^2 + \ldots + x_n^2)^{1/2} \) the Euclidean norm of \( \mathbf{x} \).

The general optimization problem (1.1) is equivalent to global optimization problem (1.2) with equality constrained and nonnegative variables.

**Remark 2.1** The assumed conditions are satisfied by using two phase method, since we can add artificial variables in the constraints.

*Linear case:* \( h(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} \) where \( \mathbf{A} \) is \( m \times n \) matrix with rank \( m \), and \( \mathbf{b} \) is an \( m \)-vector. Any \( m \) columns of \( \mathbf{A} \) are linearly independent, and every extreme point of the feasible region has \( m \) strictly positives variables.

We introduce basic and nonbasic variables according to

\[
\mathbf{A} = [B, N] \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} \quad \mathbf{x}_N \geq 0 \quad \mathbf{x}_B > 0 \quad (2.1)
\]

by nondegeneracy assumption. Furthermore the gradient of \( f \) may be conformally partitioned as

\[
\nabla f(\mathbf{x})^t = [\nabla_B f(\mathbf{x})^t, \nabla_N f(\mathbf{x})^t],
\]

where \( \nabla_B f \) and \( \nabla_N f \) are the basic gradient and nonbasic gradient of \( f \) respectively.

*Nonlinear case:* for simplicity we also assume that the gradients of the constraint functions \( h_j \) are linearly independent for all point \( \mathbf{x} \geq 0 \). This assumption like the assumption of the full rank of the matrix \( \mathbf{A} \) in the linear case ensures that we can choose a basis and express the basic variables as a linear combination of the nonbasic variables and then reduce the problem. Hence a similar generalized reduced gradient procedure applies.

Let a feasible solution \( \mathbf{x}^k \geq 0 \) with \( h_j(\mathbf{x}^k) = 0 \) for all \( j \) be given. By assumption the Jacobian matrix of the constraints \( h(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}))^t \) at each \( \mathbf{x} \geq 0 \) has full rank and, for simplicity at the point \( \mathbf{x}^k \) will be denoted by

\[
\mathbf{A}_k = Jh(\mathbf{x}^k) \quad \text{and} \quad \mathbf{b}_k = \mathbf{A}_k \mathbf{x}^k.
\]

Then a similar construction as in linear case applies.

Let

\[
C = \{ \mathbf{x} \in E \mid h(\mathbf{x}) = 0, \mathbf{x} \geq 0 \}.
\]

The objective function is \( f : E \longrightarrow \mathbb{R} \), its lower bound on \( C \) is denoted by \( l^* : l^* = \min_C f \). Let us introduce

\[
C_\alpha = S_\alpha \cap C; \quad S_\alpha = \{ \mathbf{x} \in E \mid f(\mathbf{x}) \leq \alpha \}.
\]
We assume that
\[ f \text{ is twice continuously differentiable on } E \]  
\[ \forall \alpha > l^* : \ C_\alpha \text{ is not empty, closed and bounded} \]  
\[ \forall \alpha > l^* : \ \text{meas}(C_\alpha) > 0, \]  
where \( \text{meas}(C_\alpha) \) is the measure of \( C_\alpha \).

Since \( E \) is a finite dimensional space, the assumption (2.3) is verified when \( C \) is bounded or \( f \) is coercive, i.e., \( \lim_{||x|| \rightarrow +\infty} f(x) = +\infty \). Assumption (2.3) is verified when \( C \) contains a sequence of neighborhoods of a point of optimum \( x^* \) having strictly positive measure, i.e., when \( x^* \) can be approximated by a sequence of points of the interior of \( C \). We observe that the assumptions (2.2)-(2.3) yield that
\[ C = \bigcup_{\alpha > l^*} C_\alpha \]  
i.e., \( \forall x \in C : \ \exists \alpha > l^* \text{ such that } x \in C_\alpha. \)

From (2.2) and (2.3) we get
\[ \gamma_1 = \sup \{ ||\nabla f(x)|| : x \in C_\alpha \} < +\infty. \]

Then
\[ \gamma_2 = \sup \{ ||d|| : x \in C_\alpha \} < +\infty, \]
where \( d \) is the direction of reduced gradient method (see section 3). Thus,
\[ \beta(\alpha, \varepsilon) = \sup \{ ||y - (x + \eta d)|| : (x, y) \in C_\alpha \times C_\alpha, 0 \leq \eta \leq \varepsilon \} < +\infty, \quad (2.5) \]
where \( \varepsilon, \eta \) are positive real numbers.

## 3 Reduced Gradient and its Generalization

### 3.1 Reduced Gradient Method

The reduced gradient method is coldly related to the simplex method of linear programming in that the problem variables are partitioned into basic and nonbasic groups. From a theoretical viewpoint, the method can be shown to have very much like the gradient projection method [1]. Some variants of the method applied to convex nonlinear problems with linear constraints are also referred to as convex simplex method.

Again we are dealing with the same basic descent algorithm structure and are interested in still other means for computing descent directions. We use the following form of the linearly constrained with \( f(.) \) continuously differentiable on \( \mathbb{R}^n \):

\[
\begin{aligned}
\text{Minimize :} & \quad f(x) \\
\text{subject to :} & \quad Ax = b \\
& \quad x \geq 0
\end{aligned}
\]  
\[ (3.1) \]
where $A$ is an $m \times n$ matrix with full row rank and $b \in \mathbb{R}^m$.

The reduced gradient method begins with a basis $B$ and a feasible solution $x^k = (x^k_B, x^k_N)$ such that $x^k_B > 0$. The solution $x$ is not necessarily a basic solution, i.e. $x_N$ do not have to be identically zero. Such a solution can be obtained e.g. by the usual first phase procedure of linear optimization. Using the basis $B$ from $Bx_B + N x_N = b$, we have

$$x_B = B^{-1}b - B^{-1}N x_N,$$

hence the basic variables $x_B$ can be eliminated from the problem (3.1)

$$\begin{array}{ll}
\text{Minimize :} & f_N(x_N) \\
\text{subject to :} & B^{-1}b - B^{-1}N x_N > 0 \\
& 0 \leq x_N
\end{array} \quad (3.2)$$

where $f_N(x_N) = f(x) = f(B^{-1}b - B^{-1}N x_N, x_N)$. Using the notation

$$\nabla f(x)^t = [\nabla_B f(x)^t, \nabla_N f(x)^t],$$

the gradient of $f_N$, which is the so-called reduced gradient, can be expressed as

$$\nabla f_N(x)^t = -(\nabla_B f(x)^t B^{-1}N) + (\nabla_N f(x)^t).$$

Now let us assume that the basis is nondegenerate, i.e. only the nonnegativity constraints $x_N \geq 0$ might be active at the current iterate $x^k$. Let the search direction be a vector $d^t = (d^t_B, d^t_N)$ in the null space of the matrix $A$ defined as $d_B = -B^{-1}N d_N$ and $d_N \geq 0$. If we define so, then the feasibility of $x^k + \eta d$ is guaranteed as long as $x^k_B + \eta d_B \geq 0$, i.e. as long as

$$\eta \leq \eta_{max} = \min_{i \in B, d_i < 0} \left\{ \frac{-x^k_i}{d_i} \right\}.$$ 

We still need to define $d_N \geq 0$ such that it is a descent direction of $f_N$ projected to the coordinate hyperplane active at the current point $x^k_N$. So we have

$$d^k_j = \begin{cases} 
0 & \text{if } x^k_j = 0 \text{ and } \frac{\partial f_N(x^k_N)}{\partial x_j} \geq 0, \\
-\frac{\partial f_N(x^k_N)}{\partial x_j} & \text{otherwise}
\end{cases} \quad j \in N.$$

To complete the description of the algorithm we make a line search to obtain the new point.

$$x^{k+1} = \arg \min_{0 \leq \eta \leq \eta_{max}} f(x^k + \eta d^k).$$

If all the coordinates $x^k_B + \eta d^k$ stay strictly positive we keep the basis, else a pivot is made to eliminate the zero variable from the basis and replace it by a positive but currently nonbasic coordinate. For examples of this type see the statistical problem in Section 5.
3.2 Generalized reduced gradient (GRG)

The reduced gradient method can be generalized to nonlinearly constrained optimization problems. Similar to the linearly constrained case we consider the problem with equality constraints and nonnegative variables.

The reduced gradient method and its generalization try to extend the methods of linear optimization to the nonlinear case. These methods are close to, or equivalent to projected gradient methods; just the presentation of methods is frequently quite different.

We generate a reduced gradient search direction by virtually keeping the linearized constraints valid. This direction by construction will be in the null space of $A_k$. More specifically for the linearized constraints we have

$$h(x^k) + J_h(x^k)(x - x^k) = 0 + A_k(x - x^k) = 0.$$ 

From this, one has

$$B_k x_{B_k} + N_k x_{N_k} = A_k x^k,$$

and by introducing the notation $b_k = A_k x^k$, we have

$$x_{B_k} = B_k^{-1} b_k - B_k^{-1} N_k x_{N_k}.$$ 

Hence the basic variables $x_{B_k}$ can be eliminated from the linearization of the problem (1.2) to result the problem (3.2).

From this point the generation of the search direction $d$ goes exactly to the same way as in the linearly constrained case. Due to the nonlinearity of the constraints $h(x^{k+1}) = h(x^k + \eta d) = 0$ will not hold. Hence something more has to be done to restore feasibility.

In old versions of the GRG Newton’s method is applied to the nonlinear equality system $h(x) = 0$ from the initial point $x^{k+1}$ to produce gradient direction, which is combined by a direction from the orthogonal subspace (the range space of $A_k^T$) and then a modified (nonlinear, discrete) line search is performed. These schemes are quite complicated and not discussed here in details.

Let us recall briefly the essential points of the reduced gradient method: an initial feasible guess $x^0 \in C$ is given and a sequence $\{x^k\}_{k \geq 0} \subset C$ is generated by using iterations of the general form:

$$\forall k \geq 0 \quad x^{k+1} = Q_k(x^k) = x^k + \eta_k d^k. \quad (3.3)$$

The optimal choice for $\eta_k$ is

$$\eta_k \in \arg \min \{ f(x^k + \eta d^k) : 0 \leq \eta \leq \eta_{max} \}, \quad (3.4)$$

where

$$\eta_{max} = \left\{ \begin{array}{ll} \min_{1 \leq j \leq n} \left\{ \frac{-x_j^k}{d_j^k} : d_j^k < 0 \right\} & \text{if } d^k < 0 \\ \infty & \text{if } d^k \geq 0. \end{array} \right.$$ 

We have $f(x^k + \eta_k d^k) \leq f(x^k)$. 

4 Stochastic perturbation of the reduced gradient

The main difficulty remains the lack of convexity: if $f$ is not convex, the Kuhn-Tucker points may not correspond to global minimum. In the following we shall improve this point by using an appropriate random perturbation.

The sequence of real numbers $\{x^k\}_{k \geq 0}$ is replaced by a sequence of random variables $\{X^k\}_{k \geq 0}$ involving a random perturbation $P_k$ of the deterministic iteration (3.3). Then we have $X^0 = x^0$;

$$\forall k \geq 0 \quad X^{k+1} = Q_k(X^k) + P_k = X^k + \eta_k d^k + P_k, \quad (4.1)$$

where

$$\forall k \geq 1 \quad P_k \quad \text{is independent from} \quad (X^{k-1}, \ldots, X^0), \quad (4.2)$$

$$x \in C \Rightarrow Q_k(x) + P_k \in C. \quad (4.3)$$

Equation (4.1) can be viewed as perturbation of the descent direction $d^k$, which is replaced by a new direction $D_k = d^k + \frac{P_k}{\eta_k}$ and the iterations (4.1) become

$$X^{k+1} = X^k + \eta_k D_k.$$

General properties defining convenient sequences of perturbation $\{P_k\}_{k \geq 0}$ can be found in the literature: usually, a sequence of Gaussian laws may be used in order to satisfy these properties.

We introduce a random vector $Z_k$, we denote by $\Phi_k$ and $\phi_k$ the cumulative distribution function and the probability density of $Z_k$, respectively.

We denote by $F_{k+1}(y|X^k = x)$ the conditional cumulative distribution function

$$F_{k+1}(y|X^k = x) = P(X^{k+1} < y|X^k = x);$$

the conditional probability density of $X^{k+1}$ is denoted by $f_{k+1}$.

Let us introduce a sequence of $n$-dimensional random vectors $\{Z_k\}_{k \geq 0} \in C$. We consider also $\{\xi_k\}_{k \geq 0}$, a suitable decreasing sequence of strictly positive real numbers converging to 0 and such that $\xi_0 \leq 1$. The optimal choice for $\eta_k$ is

$$\eta_k \in \arg \min \{f(X^k + \eta D_k) : 0 \leq \eta \leq \eta_{\max}\}, \quad (4.4)$$

where

$$\eta_{\max} = \begin{cases} \min_{1 \leq j \leq n} \left\{ \frac{-X^k_j}{D^k_j} : D^k_j < 0 \right\} & \text{if } D^k < 0 \\ \infty & \text{if } D^k \geq 0. \end{cases}$$

Then

$$F_{k+1}(y|X^k = x) = P(X^{k+1} < y|X^k = x).$$

$$F_{k+1}(y|X^k = x) = P(Z_k < \frac{y - Q_k(x)}{\xi_k}).$$
and we have
\[ F_{k+1}(y|X^k = x) = \Phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) \]
\[ f_{k+1}(y|X^k = x) = \frac{1}{\xi_k^n} \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) \quad y \in C. \quad (4.5) \]

Note that (2.5) shows that \( \|y - Q_k(x)\| \leq \beta(\alpha, \varepsilon) \) for \( (x, y) \in C_\alpha \times C_\alpha \).

We assume that there exists a decreasing function \( t \mapsto g_k(t) \), \( g_k(t) > 0 \) on \( \mathbb{R}^+ \) such that
\[ y \in C_\alpha \Rightarrow \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) \geq g_k \left( \frac{\beta(\alpha, \varepsilon)}{\xi_k} \right) \quad (4.6) \]

For simplicity, let
\[ Z_k = 1_C(Z_k)Z_k. \quad (4.7) \]

The procedure generates a sequence \( U_k = f(X^k) \). By construction this sequence is decreasing and lower bounded by \( l^* \)
\[ \forall k \geq 0 \ l^* \leq U_{k+1} \leq U_k. \quad (4.8) \]

Thus there exists \( U \geq l^* \) such that \( U_k \to U \) for \( k \to +\infty \).

**Lemma 4.1** Let \( P_k = \xi_kZ_k \) and \( \gamma = f(x^0) \) if \( Z_k \) is given by (4.7). Then there exists \( \nu > 0 \) such that
\[ P(U_{k+1} < \theta|U_k \geq \theta) \geq \frac{\text{meas}(C_\gamma - C_{\theta})}{\xi_k^n} g_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) > 0 \quad \forall \theta \in (l^*, l^* + \nu], \]
where \( n = \dim(E) \).

**Proof.** Let \( \hat{C}_\theta = \{x \in C|f(x) < \theta\} \), for \( \theta \in (l^*, l^* + \nu] \). Since \( C_\alpha \subset \hat{C}_\theta \), \( l^* < \alpha < \theta \), it follows from (2.4) that \( \hat{C}_\theta \) is non empty and has a strictly positive measure.

If \( \text{meas}(C - \hat{C}_\theta) = 0 \) for any \( \theta \in (l^*, l^* + \nu] \), the result is immediate, since we have \( f(x) = l^* \) on \( C \).

Let us assume that there exists \( \varepsilon > 0 \) such that \( \text{meas}(C - \hat{C}_\varepsilon) > 0 \). For \( \theta \in (l^*, l^* + \varepsilon] \), we have \( \hat{C}_\theta \subset \hat{C}_\varepsilon \) and \( \text{meas}(C - \hat{C}_\theta) > 0 \).

\[ P(X^k \notin \hat{C}_\theta) = P(X^k \in C - \hat{C}_\theta) = \int_{C - \hat{C}_\theta} P(X^k \in dx) > 0 \quad \text{for any } \theta \in (l^*, l^* + \varepsilon], \]

since the sequence \( \{U_i\}_{i \geq 0} \) is decreasing, we have also
\[ \{X^i\}_{i \geq 0} \subset C_\gamma. \quad (4.9) \]

Therefore
\[ P(X^k \notin \hat{C}_\theta) = P(X^k \in C_\gamma - \hat{C}_\theta) = \int_{C_\gamma - \hat{C}_\theta} P(X^k \in dx) > 0 \quad \text{for any } \theta \in (l^*, l^* + \varepsilon]. \]
Let \( \theta \in (l^*, l^* + \varepsilon) \), we have from (4.8)
\[
P(U_{k+1} < \theta | U_k \geq \theta) = P(X^{k+1} \in \hat{C}_\theta | X^i \notin \hat{C}_\theta, \ i = 0, \ldots, k).
\]
But (4.4) and (4.5) yield that
\[
P(X^{k+1} \in \hat{C}_\theta | X^i \notin \hat{C}_\theta, \ i = 0, \ldots, k) = P(X^{k+1} \in \hat{C}_\theta | X^k \notin \hat{C}_\theta).
\]
Thus
\[
P(X^{k+1} \in \hat{C}_\theta | X^k \notin \hat{C}_\theta) = \frac{P(X^{k+1} \in \hat{C}_\theta, X^k \notin \hat{C}_\theta)}{P(X^k \notin \hat{C}_\theta)}.
\]
Moreover
\[
P(X^{k+1} \in \hat{C}_\theta, X^k \notin \hat{C}_\theta) = \int_{C-\hat{C}_\theta} P(X^k \in dx) \int_{\hat{C}_\theta} f_{k+1}(y | X^k = x) dy.
\]
Using (4.9) we get
\[
P(X^{k+1} \in \hat{C}_\theta, X^k \notin \hat{C}_\theta) = \int_{C_\gamma - \hat{C}_\theta} P(X^k \in dx) \int_{\hat{C}_\theta} f_{k+1}(y | X^k = x) dy
\]
and
\[
P(X^{k+1} \in \hat{C}_\theta, X^k \notin \hat{C}_\theta) \geq \inf_{x \in C_\gamma - \hat{C}_\theta} \left\{ \int_{\hat{C}_\theta} f_{k+1}(y | X^k = x) dy \right\} \int_{C_\gamma - \hat{C}_\theta} P(X^k \in dx).
\]
Thus
\[
P(X^{k+1} \in \hat{C}_\theta | X^k \notin \hat{C}_\theta) \geq \inf_{x \in C_\gamma - \hat{C}_\theta} \left\{ \int_{\hat{C}_\theta} f_{k+1}(y | X^k = x) dy \right\}.
\]
Taking (4.5) into account, we have
\[
P(X^{k+1} \in \hat{C}_\theta | X^k \notin \hat{C}_\theta) \geq \frac{1}{\xi_k} \inf_{x \in C_\gamma - \hat{C}_\theta} \left\{ \int_{\hat{C}_\theta} \phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) dy \right\}.
\]
Note that (2.5) shows that
\[
\| y - Q_k(x) \| \leq \beta(\gamma, \varepsilon)
\]
and (4.6) yields that
\[
\phi_k \left( \frac{y - Q_k(x)}{\xi_k} \right) \geq g_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right).
\]
Hence
\[
P(X^{k+1} \in \hat{C}_\theta | X^k \notin \hat{C}_\theta) \geq \frac{1}{\xi_k} \inf_{x \in C_\gamma - \hat{C}_\theta} \int_{\hat{C}_\theta} g_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) dy.
\]
It follows that
\[
P(X^{k+1} \in \hat{C}_\theta | X^k \notin \hat{C}_\theta) \geq \frac{\text{meas}(C_\gamma - C_\theta)}{\xi_k} g_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right).
\]

The global convergence is a consequence of the following result, which yields from the Borel-Catelli’s lemma (see, for instance, [28]).
Lemma 4.2 Let \( \{U_k\}_{k \geq 0} \) be a decreasing sequence, lower bounded by \( l^* \). Then there exists \( U \) such that \( U_k \to U \) for \( k \to +\infty \). Assume that there exists \( \nu > 0 \) such that for any \( \theta \in (l^*, l^* + \nu) \), there is a sequence of strictly positive real numbers \( \{c_k(\theta)\}_{k \geq 0} \) such that

\[
\forall k \geq 0 \quad P(U_{k+1} < \theta | U_k \geq \theta) \geq c_k(\theta) > 0; \quad \sum_{k=0}^{+\infty} c_k(\theta) = +\infty. \tag{4.10}
\]

Then \( U = l^* \) almost surely.

For the proof we refer the reader to [21] or [28].

Theorem 1 Let \( \gamma = f(x^0) \) and \( \eta_k \) satisfy (4.4). Assume that \( x^0 \in C \), the sequence \( \xi_k \) is non increasing and

\[
\sum_{k=0}^{+\infty} g_k \left( \frac{\beta(\gamma, \epsilon)}{\xi_k} \right) = +\infty. \tag{4.11}
\]

Then \( U = l^* \) almost surely.

Proof. Let

\[
c_k(\theta) = \frac{\text{meas}(C_\gamma - C_{\theta})}{\xi_k^n} g_k \left( \frac{\beta(\gamma, \epsilon)}{\xi_k} \right) > 0. \tag{4.12}
\]

Since the sequence \( \{ \xi_k \}_{k \geq 0} \) is non increasing,

\[
c_k(\theta) \geq \frac{\text{meas}(C_\gamma - C_{\theta})}{\xi_0^n} g_k \left( \frac{\beta(\gamma, \epsilon)}{\xi_k} \right) > 0.
\]

Thus, equation (4.11) shows that

\[
\sum_{k=0}^{+\infty} c_k(\theta) \geq \frac{\text{meas}(C_\gamma - C_{\theta})}{\xi_0^n} \sum_{k=0}^{+\infty} g_k \left( \frac{\beta(\gamma, \epsilon)}{\xi_k} \right) = +\infty.
\]

Using Lemma 1 and Lemma 2 we have \( U = l^* \) almost surely. \( \blacksquare \)

Theorem 2 Let \( Z_k = 1_{C}(Z)Z \) where \( Z \) is a random variable following \( N(0, \sigma \text{Id}) \), \( \sigma > 0 \) and let

\[
\xi_k = \sqrt{\frac{a}{\log(k+d)}}, \tag{4.13}
\]

where \( a > 0 \), \( d > 0 \) and \( k \) is the iteration number. If \( x^0 \in C \) then, for \( a \) large enough, \( U = l^* \) almost surely.

Proof. We have

\[
\phi_k(z) = \frac{1}{(\sigma \sqrt{2\pi})^n} \exp \left( -\frac{1}{2} \left\| \frac{z}{\sigma} \right\|^2 \right) = g_k \left( \| z \| > 0. \right.
\]
So

\[ g_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) = \frac{1}{(\sigma \sqrt{2\pi})^n} (k + d)^{\beta(\gamma, \varepsilon)^2/(2\sigma^2a)} \]

For \( a \) such that

\[ 0 < \frac{\beta(\gamma, \varepsilon)^2}{2\sigma a} < 1, \]

we have

\[ \sum_{k=0}^{\infty} g_k \left( \frac{\beta(\gamma, \varepsilon)}{\xi_k} \right) = +\infty, \]

and, from the preceding theorem, we have \( U = l^* \) almost surely. ■

5 Numerical results

In order to apply the method presented in Section 4, we start at the initial value \( X^0 = x^0 \in C \). At step \( k \geq 0 \), \( X^k \) is known and \( X^{k+1} \) is determined.

We generate \( k_{sto} \) the number of perturbation, the case \( k_{sto} = 0 \) corresponds to the unperturbed descent method. We add slash variables \( z_1, z_2, \ldots \) to inequality constraints in order to opting equality constraints.

We shall introduce a maximum iteration number \( k_{max} \) such that the iterations are stopped when \( k = k_{max} \) or \( x^k \) is a Kuhn-Tucker point. For each descent direction, the optimal \( \eta_k \) is calculated by unidimensional exhaustive search with a fixed step on the interval defined by (3.4). This procedure could imply a large number of calls of the objective function. The global number of function evaluations is \( n_{eval} \) and includes the evaluations for unidimensional search. We denote by \( k_{end} \) the value of \( k \) when the iterations are stopped (it corresponds to the number of evaluations of the gradient of \( f \)). The optimal value and optimal point are \( f_{opt} \) and \( x_{opt} \) respectively. The perturbation is normally distributed and samples are generated by using the log-trigonometric generator and the standard random number generator of FORTRAN library. We use

\[ \xi_k = \sqrt{\frac{a}{\log(k+2)}}, \text{ where } a > 0. \]

The maximum of iterations has been fixed at \( k_{max} = 100 \). The results for \( k_{sto} = 0, 250, 500, 1000 \) and \( 2000 \) are given in the tables below for each problems. Concern experiments performed on a workstation HP Intel(R) Celeron(R) M processor 1.30GHz, 224 Mo RAM. The row cpu gives the mean CPU time in seconds for one run.

5.1 Statistical problem

The statistical problem of estimating a bounded mean with a minimax procedure is nonconvex and nonlinear; we will restrict ourselves to this problem (see for instance, [16] and [17]). Thus, our problem is reduced to the following: for a suitable \( m_2 > m_1 \simeq 1.27 \) and for each \( m \in (m_1, m_2] \) find the \( a_1, a_2 \). In this case, the problem reduces to a global optimization problem with 5 unknown \((\theta, a_1, a_2, a_3, b)\) and 3 equality constraints. This
problem is equivalent to the maximization of the convex combination of the several \( g_i \) function without the constraint on \( \theta \) (see for instance [14] and [22]). Using the equality \( a_1 + a_2 + a_3 = 1 \), one variable, \( a_1 \) for example, can be eliminated. The problem we eventually studied is the following:

\[
\begin{aligned}
\text{Minimize : } & - \{(1 - a_2 - a_3)g_1 + a_2g_2 + a_3g_3\} \\
\text{subject to: } & \quad a_2 + a_3 + z_1 = 1 \\
& \quad a_i + z_i = 1 \quad i = 2, 3 \\
& \quad b + z_4 = 1 \\
& \quad 0 \leq a_i \quad i = 2, 3 \\
& \quad 0 \leq b, \quad 0 \leq z_i \quad i = 1, ..., 4,
\end{aligned}
\]

where

\[
\begin{aligned}
g_1(a_2, a_3, b) & : \delta(1)^2 = \theta, \\
\delta(1) & = m a_2 b^2 \exp(-m) + a_3 exp(-m) \\
\delta(x) & = m \frac{\frac{a_2 b x \exp(1 - b) m + a_3}{a_2 b^{x - 1} \exp(1 - b) m + a_3}}{1 - a_2 + a_3 + a_2 \exp(-bm) + a_3 \exp(-m)} \\
\end{aligned}
\]

The table below gives the optimal solutions:

<table>
<thead>
<tr>
<th>( k_{sto} )</th>
<th>0</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{opt} )</td>
<td>-0.3401</td>
<td>-0.4022</td>
<td>-0.4030</td>
<td>-0.4044</td>
<td>-0.4044</td>
</tr>
<tr>
<td>( \eta_{eval} )</td>
<td>1</td>
<td>1183</td>
<td>4773</td>
<td>10743</td>
<td>17635</td>
</tr>
<tr>
<td>( k_{end} )</td>
<td>2</td>
<td>7</td>
<td>12</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>cpu</td>
<td>0.00</td>
<td>37.16</td>
<td>134.72</td>
<td>288.91</td>
<td>491.86</td>
</tr>
</tbody>
</table>

Table 1: Results for Statistics

\( a_1^* = 0.0958 \quad a_2^* = 0.9042 \quad a_3^* = 0.00000 \quad b^* = 0.907159 \)
5.2 Octagon problem

Consider polygon in the plane with 5 sides (5-gons for short) and unit diameter. Which one of them has a maximum area? (see for instant [18,31]). Using QP and geometric reasoning, the optimal octagon is determined in (6, 10) to have an area about 2.8% larger than the regular octagon.

Graham’s conjecture states that the optimal octagon can be illustrated as in Figure 1, in which a solid line between two vertices indicates that the distance between these points is one.

In 1996 Hansen formulates this question in the quadratically constrained quadratic optimization problem defining this configuration, which appears below (see for instance [8]). By symmetry, and without loss of generality, the constraint $x_2 \geq x_3$ is added to reduce the size of the feasible region.

\[ \begin{align*}
A_0 &= (0, 0) \\
A_1 &= (x_1 - x_2, y_1 - y_2) \\
A_2 &= (x_3 - x_1 - x_5, y_1 - y_3 + y_5) \\
A_3 &= (-x_1, y_1) \\
A_4 &= (0, 1) \\
A_5 &= (x_1, y_1) \\
A_6 &= (x_1 - x_2 + x_4, y_1 - y_2 + y_4) \\
A_7 &= (x_3 - x_1, y_1 - y_4)
\end{align*} \]

Figure 1: Definition of variables for the configuration conjectured by Graham to have maximum area
There are 33 variables and 62 constraints.

We use \( a = 5 \), the Fortran code furnishes the following optimal solutions, see also Table 2:

\[
x_{opt} = (0.25682, 0.67357, 0.67060, 0.91187, 0.91342),
\]

\[
y_{opt} = (0.96619, 0.73978, 0.74245, 0.41172, 0.40820)
\]

Table 2: Results for Octagon

<table>
<thead>
<tr>
<th></th>
<th>( k_{sto} = 0 )</th>
<th>( k_{sto} = 250 )</th>
<th>( k_{sto} = 500 )</th>
<th>( k_{sto} = 1000 )</th>
<th>( k_{sto} = 2000 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{opt} )</td>
<td>-0.726637</td>
<td>-0.72712</td>
<td>-0.727377</td>
<td>-0.72738</td>
<td>-0.72739</td>
</tr>
<tr>
<td>( \eta_{eval} )</td>
<td>2</td>
<td>526</td>
<td>1654</td>
<td>2634</td>
<td>4585</td>
</tr>
<tr>
<td>( k_{end} )</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>( cpu )</td>
<td>0.00</td>
<td>1.48</td>
<td>6.70</td>
<td>10.67</td>
<td>12.53</td>
</tr>
</tbody>
</table>
5.3 Mixture problem

In this example of petrochemical mixture we have four reservoirs: $R_1$, $R_2$, $R_3$ and $R_4$. The first two receive three distinct source products. Then their content is combined in the other two reservoirs to create the wanted miscellaneies. The question is to determine the quantity of every product to buy in order to maximize the profits.

The $R_1$ reservoir receives two products of quality (e.g., content in sulfur) 3 and 1 in quantity $q_0$ and $q_1$ (see for instance [12, 19]). The quality of the mixture done contain in $R_1$ is $s = \frac{3q_0 + q_1}{q_0 + q_1} \in (1, 3)$. The $R_2$ reservoir contains a product of the third source, of quality 2 in quantity $q_2$. One wants to get in the $R_3$ reservoirs and $R_4$ of capacity 10 and 20, of the quality products of to the more 2.5 and 1.5. Figure 2, where the variable $x_1$ to $x_2$ represents some quantities, illustrates this situation. The unit prices of the bought products are respectively 60, 160 and 100; those of the finished products are 90 and 150. Is the difference between the costs of purchase and sale therefore $60q_0 + 160q_1 + 100q_2 - 90(x_1 + x_3) - 150(x_2 - x_4)$? The qualities of the final miscellaneies are $\frac{sx_1 + 2x_3}{x_1 + x_3}$ and $\frac{sx_2 + 2x_4}{x_2 + x_4}$. The addition of the constraints of volume conservation $q_0 + q_1 = x_1 + x_2$ and $q_2 = x_3 + x_4$ permits the elimination of the variables $q_0, q_1$ and $q_2$:

$$q_0 = \frac{(s - 1)(x_1 + x_2)}{2}, \quad q_1 = \frac{(3 - s)(x_1 + x_2)}{2}, \quad q_2 = x_3 + x_4.$$  

![Figure 2: A Problem of mixture](image-url)
The mathematical model transformed (the variable $s$ is noted $x_5$) is as follows:

\[
\begin{align*}
\text{Minimize:} & \quad 120x_1 + 60x_2 + 10x_3 - 50x_4 - 50x_1x_5 - 50x_2x_5 \\
\text{Subject to:} & \quad 2.5x_1 + 0.5x_3 - x_1x_5 - z_1 = 0 \\
& \quad 1.5x_2 - 0.5x_4 - x_2x_5 - z_2 = 0 \\
& \quad x_1 + x_3 + z_3 = 10 \\
& \quad x_2 + x_4 + z_4 = 20, \\
& \quad x_5 + z_5 = 3, \\
& \quad x_5 - z_6 = 1, \\
& \quad 0 \leq x_i \quad i = 1, \ldots, 5; \quad 0 \leq z_i \quad i = 1, \ldots, 6.
\end{align*}
\]

There are 11 variables and 17 constraints. We use $a = 0.05$, the Fortran code furnish the following optimal solutions, see also Table 3. $x_{\text{opt}} = (0, 10, 0, 10, 1)$

<table>
<thead>
<tr>
<th>$k_{sto}$</th>
<th>0</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{opt}$</td>
<td>-306</td>
<td>-353</td>
<td>-367</td>
<td>-374</td>
<td>-400</td>
</tr>
<tr>
<td>$\eta_{eval}$</td>
<td>2</td>
<td>28</td>
<td>51</td>
<td>105</td>
<td>28667</td>
</tr>
<tr>
<td>$k_{end}$</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>101</td>
</tr>
<tr>
<td>$cpu$</td>
<td>0.00</td>
<td>0.02</td>
<td>0.36</td>
<td>1.27</td>
<td>38.53</td>
</tr>
</tbody>
</table>

We have the optimum value of objective function ($f_{opt} = -400$) for Mixture problem in $k_{sto} = 2000$.

### 5.4 Alkylation problem

The alkylation process is common in the petroleum industry, it is an important unit that is used in refineries to upgrade light olefins and isobutane into much more highly valued gasoline component. The alkylation process is usually mixed at the gasoline in order to improve his performance; see the problem 5.3. A simplified process flow diagram of an alkylation process is shown in Figure 3 below. The process model seeks to determine the optimum set of operating conditions for the process, based on a mathematical model, which allows the maximization of profit. The problem is formulated as a direct nonlinear programming model with mixed nonlinear inequality and equality constraints and a nonlinear profit function to be maximized (see for instance [3]). A typical reaction [29] is

\[
(CH_3)_2C = CH_2 + (CH_3)_2CH \quad \xrightarrow{\text{Fresh acid}} \quad (CH_3)_2CHCH_2C(CH_3)_3 \quad \xrightarrow{\text{Alkylate product}} (CH_3)_2CHCH_2C(CH_3)_3
\]

As shown in Figure 3, an olefin feed (100% butane), a pure isobutane recycle and a 100% isobutane make up stream are introduced in a reactor together with an acid
catalyst. The reactor product stream is then passed through a fractionator where the isobutane and the alkylate product are separated. The spent acid is also removed from the reactor.

![Simplified Alkylation Process Flow sheet](image)

The notation used is shown in Table 4 along with the upper and lower bounds on each variable. The bounds represent economic, physical and performance constraints.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Variable</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>Olefin feed (barrels/day)</td>
<td>0</td>
<td>2000</td>
</tr>
<tr>
<td>$x_2$</td>
<td>Isobutane recycle (barrels/day)</td>
<td>0</td>
<td>16000</td>
</tr>
<tr>
<td>$x_3$</td>
<td>Acid Addition Rate (X1000 pounds/day)</td>
<td>0</td>
<td>120</td>
</tr>
<tr>
<td>$x_4$</td>
<td>Alkylate yield (barrels/day)</td>
<td>0</td>
<td>5000</td>
</tr>
<tr>
<td>$x_5$</td>
<td>Isobutane make up (barrels/day)</td>
<td>0</td>
<td>2000</td>
</tr>
<tr>
<td>$x_6$</td>
<td>Acid strength (wt.%)</td>
<td>85</td>
<td>93</td>
</tr>
<tr>
<td>$x_7$</td>
<td>Motor octane no.</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>$x_8$</td>
<td>External Isobutane-olefin Ratio</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>$x_9$</td>
<td>Acid dilution factor</td>
<td>1.2</td>
<td>4</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>F-4 performance no.</td>
<td>145</td>
<td>162</td>
</tr>
</tbody>
</table>

The external isobutane- to- olefin ratio, $x_8$, was equal to the sum of the isobutane recycle, $x_2$, and the isobutane make up, $x_5$, divided by the olefin feed, $x_1$.

$$x_8 = \frac{x_2 + x_5}{x_1}.$$
The motor octane number, $x_7$, was a function of the external isobutane-to-olefin ratio, $x_8$, and the acid strength by weight percent, $x_6$, (for the same reactor temperatures and acid strengths as for the alkylate yield $x_4$)

$$x_1 \leq 86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_6 - 89).$$

The acid strength by weight percent, $x_6$, could be derived from an equation that expressed the acid addition rate, $x_3$, as a function of the alkylate yield, $x_4$, and the acid dilution factor, $x_9$, and the acid strength by weight percent, $x_6$ (the addition acid was assumed to have a strength of 98%)

$$1000x_3 = \frac{x_4x_6x_9}{98 - x_6}.$$

This equation repeats itself with the help of quadratic constraints implying an artificial variable $x_{11}$

$$x_3 = x_6x_{11} \quad \text{and} \quad 98000x_{11} - 1000x_3 = x_4x_9.$$

The alkylate yield, $x_4$, was a function of the olefin feed, $x_1$, and the external isobutane-to-olefin ratio, $x_8$. The relationship determined by holding the reactor temperature between 80 F and 90 F and the reactor acid strength by weight percent at 85 to 93 was:

$$x_4 \leq x_1(1.12 + 0.13167x_8 - 0.00667x_8^2).$$

While adding an artificial variable $x_{12}$, one gets the equivalent quadratic constraints:

$$x_4 = x_1x_{12} \quad \text{and} \quad x_{12} \leq 1.12 + 0.13167x_8 - 0.00667x_8^2.$$

The objective function is linear in the cost of the products sources $x_1, x_3$ and $x_5$, and in the cost of the isobutane recycled $x_2$, and depends linearly on the product of the quantity of alkane by the rate of octane $x_4x_7$.

The other constraints of the physical boundary are on the variables and the linear relations. Continuation to the elimination of variables, a stake to the scale and a
renumbering of the indications, one gets the following optimization problem:

Minimize: 
\[ 614.88x_1 - 171.5x_2 - 6.3x_1x_4 + 4.27x_1x_5 - 3.5x_2x_5 + 10x_3x_6 \]
Subject to:
\[ -0.325x_3 + x_4 - 1.098x_5 + 0.038x_5^2 + z_1 = 57.425 \]
\[ -0.13167x_5 + x_7 + 0.00667x_5^2 + z_2 = 1.12 \]
\[ 12.2x_1 - 10x_2 + z_3 = 200, \quad 12.2x_1 - 10x_2 - z_4 = 0.1 \]
\[ -10x_2 + 12.2x_1x_5 - 10x_2x_5 + z_5 = 1600, \quad -10x_2 + 12.2x_1x_5 - 10x_2x_5 - z_6 = 0.1 \]
\[ 65.346x_1 - 980x_6 - 0.666x_1x_4 + 10x_2x_5 = 0 \]
\[ x_1 - 1.22x_1x_7 + x_2x_7 = 0 \]
\[ x_3x_6 + z_7 = 120 \]
\[ x_1 + z_8 = 32.786885, \quad x_1 - z_9 = 0.01 \]
\[ x_5 + z_{10} = 12, \quad x_5 - z_{11} = 3 \]
\[ x_2 + z_{12} = 20, \quad x_6 + z_{13} = 1.411765 \]
\[ x_3 + z_{14} = 95, \quad x_3 - z_{15} = 85 \]
\[ x_4 + z_{16} = 95, \quad x_4 - z_{17} = 92.66666 \]
\[ 0.8196722 + z_{18} = x_7, \]
\[ x_i \geq 0, \quad i = 1, \ldots, 7; \quad z_j, \quad j = 1, \ldots, 18. \]

There are 25 variables and 38 restrictions and we use \( a = 10 \) and \( k_{\text{max}} = 100 \).
The SPRGM algorithm furnishes the following optimal solutions, see also Table 5:

\[ x_{\text{opt}} = (30.4713, 20.0000, 90.9999, 94.1900, 10.4100, 1.0697, 1.7743) \]

<table>
<thead>
<tr>
<th>( n_{\text{sto}} )</th>
<th>0</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{opt}} )</td>
<td>-1161.626</td>
<td>-1176.192</td>
<td>-1176.191</td>
<td>-1176.186</td>
<td>-1176.193</td>
</tr>
<tr>
<td>( n_{\text{eval}} )</td>
<td>2</td>
<td>274</td>
<td>531</td>
<td>1019</td>
<td>2063</td>
</tr>
<tr>
<td>( k_{\text{end}} )</td>
<td>3</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>( \text{cpu} )</td>
<td>0.00</td>
<td>0.97</td>
<td>1.95</td>
<td>3.86</td>
<td>7.86</td>
</tr>
</tbody>
</table>

5.5 Pooling problem

We present in Figure 4 (and Table 6) an example in which the intermediate pools are allowed to be intermediate, see for instance [12]. The pooling is thus extended to the case where exiting blends of some intermediate pools are feeds of others.

The proportion model of the generalized pooling problem is not a bilinear program, since the variables are not partitioned into two sets. Therefore, this formulation belongs to the class of quadratically constrained quadratic programs; cf. for instance [19].
Figure 4: A Generalized Pooling Problem

Table 6: Characteristics of GP

<table>
<thead>
<tr>
<th>Feed</th>
<th>Price</th>
<th>Attribute</th>
<th>Max Supply</th>
<th>Pool Max capacity</th>
<th>Blend</th>
<th>Price</th>
<th>Max Demand</th>
<th>Max Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>$/\text{bbl}$</td>
<td>Quality</td>
<td></td>
<td></td>
<td></td>
<td>$$/\text{bbl}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F_1$</td>
<td>6</td>
<td>3</td>
<td>18</td>
<td>$P_1$</td>
<td>20</td>
<td>$B_1$</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>$F_2$</td>
<td>16</td>
<td>1</td>
<td>18</td>
<td>$P_2$</td>
<td>20</td>
<td>$B_2$</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>$F_3$</td>
<td>10</td>
<td>2</td>
<td>18</td>
<td>$B_3$</td>
<td>14</td>
<td>20</td>
<td>1.5</td>
<td></td>
</tr>
</tbody>
</table>

A hybrid model for GP is given below where $v_{12}$ denotes the flow from $P_1$ to $P_2$ (all other variables are defined as before).

\[
\begin{align*}
\max_{q,t,v,w,x,y} & \quad -6(x_{11} + y_{12} + v_{12} - q_{21}(y_{12} + v_{12})) - 16q_{12}(y_{12} + v_{12}) \\
& \quad -10(x_{32} + y_{21} + y_{23} - v_{12}) + 9(x_{11} + y_{21}) + 13(y_{12} + x_{32}) + 14y_{23}
\end{align*}
\]

Subject to:

supply:

\[
\begin{align*}
& x_{11} + y_{12} + v_{12} - q_{21}(y_{12} + v_{12}) + z_1 = 18 \\
& q_{21}(y_{12} + v_{12}) + z_2 = 18 \\
& x_{32} + y_{21} + y_{23} - v_{12} + z_3 = 18 \\
& x_{11} + y_{21} + z_4 = 10
\end{align*}
\]

demand:

\[
\begin{align*}
& y_{12} + x_{32} + z_5 = 15 \\
& y_{23} + z_6 = 20
\end{align*}
\]

capacity:

\[
\begin{align*}
& y_{12} + v_{12} + z_7 = 20 \\
& y_{21} + y_{23} + z_8 = 20 \\
& (3 - 2q_{21})v_{12} + 2(y_{21} + y_{23} - v_{12}) - t_2^1(y_{21} + y_{23}) = 0 \\
& 3x_{11} + t_2^1y_{21} - 2.5(x_{11} + y_{21}) + z_9 = 0 \\
& 2x_{32} + (3 - 2q_{21})y_{12} - 1.75(x_{32} + y_{12}) + z_{10} = 0 \\
& t_2^1 + z_{11} = 1.5
\end{align*}
\]

attribute:

\[
\begin{align*}
& y_{21} + y_{23} - v_{12} - z_{12} = 0 \\
& y_{21} + y_{23} - z_{13} = 1
\end{align*}
\]

pos. flow:

\[
\begin{align*}
& q_{21} + z_{13} = 1 \\
& q \geq 0, t \geq 0, v \geq 0, w \geq 0, x \geq 0, y \geq 0, \quad z_i \geq 0, i = 1, \ldots, 13.
\end{align*}
\]
There are 21 variables and 35 restrictions, we use $a = 50$.

The SPRGM algorithm furnish the following optimal solutions, see also Table 7:

$x_{opt} = (x_{11}, x_{32}) = (0.1506, 2.6564); \quad y_{opt} = (y_{12}, y_{21}, y_{23}) = (0.7425, 0.0005, 19.9988)$

$v_{opt} = v_{12} = 10.0013; \quad q_{opt} = q_{21} = 0.9949; \quad t_{opt} = t_{12} = 1.5000.$

<table>
<thead>
<tr>
<th>$n_{sto}$</th>
<th>0</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{end}$</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$\eta_{eval}$</td>
<td>2</td>
<td>181</td>
<td>441</td>
<td>813</td>
<td>1114</td>
</tr>
<tr>
<td>$cpu$</td>
<td>0.00</td>
<td>0.03</td>
<td>0.03</td>
<td>0.11</td>
<td>0.17</td>
</tr>
</tbody>
</table>

6 Concluding remarks

We have presented a stochastic modification of the reduced gradient method for linear and nonlinear constraints, involving the adjunction of a stochastic perturbation. This approach leads to a stochastic descent method where the deterministic sequence generated by the reduced gradient is replaced by a sequence of random variables. We have established a result of stochastic descent methods.

The numerical experiments show that our method is effective to be calculated for nonconvex optimization problems verifying a nondegeneracy assumption. The use of stochastic perturbations generally improves the results furnished by the reduced gradient.

Here yet, we observe that the adjunction of the stochastic perturbation improves the result, with a larger number of evaluations of the objective function. The final points $X_{k_{end}}$ are very close or practically identical for $k_{sto} \geq 250$.

The main difficulty in the practical use of the stochastic perturbation is connected to the tuning of the parameters. Even if the combination with the deterministic reduced gradient increases the robustness, the values of $a$ and $k_{sto}$ have an important influence on the quality of the result. The random number generator may have also influence aspects such as the number of evaluations. All the sequences $\{\xi_k\}_{k \geq 0}$ tested have furnished analogous results, but for different sets of parameters.

References


