2009 Ninth International Conference on Intelligent Systems Design and Applications

Handling Box, Linear and Quadratic-Convex Constraints for Boundary Optimization with Differential Evolution Algorithms

Massimo Spadoni
Department of Mathematics for Economic and Social Sciences
University of Bologna, Italy (spadoni@rimini.unibo.it)

Luciano Stefanini
DEMQ, Department of Economics and Quantitative Methods
University of Urbino, Italy (lucste@uniurb.it)

Abstract

We propose and test the performance of an implicit strategy to handle box, linear and quadratic convex constraints, based on changing the search space from points to directions, suitable to be easily implemented in combination with differential evolution (DE) algorithms for the boundary optimization of a generic continuous function. In particular, we see that DE can be efficiently implemented to find solutions on the boundary of box constraints, linear inequality constraints and quadratic convex constraints, for which the feasible set is convex and bounded. The computational results are performed on different classes of minimization problems.

Keywords: *Constrained Global Optimization, Boundary Optimization, Differential Evolution Algorithm.*

1 Introduction

The literature in the recent years (see [4], [9], [8], [15]) has shown an extended interest toward adding more general constraints to the box ones managed in the first implementations of the DE algorithms (Differential Evolution, see [12], [13]). The paper addresses situations where the constraints are box, linear and/or quadratic convex and the solution of the optimization problem is on the boundary of the feasible region. The approach is relevant both for problems where the solution of interest is on the boundary of the feasible region and for problems (such as the minimization of a concave function) having naturally a boundary solution. The proposed technique is based on changing the search space from points to directions, suitable to be easily implemented in combination with DE for boundary optimization of generic continuous functions. In particular, we see that DE methods can be efficiently implemented to find solutions on the boundary of box constraints, linear inequality constraints and quadratic convex constraints, for which the feasible set is convex and bounded. We claim for the uselessness of penalization methods when dealing with constraints of the type specified ([4],[15]).

In section 2 we give the general setting to find the feasible directions that allow optimization on the boundary. Section 3 introduces DE and shows the results of applying our strategy to a set of computational tests taken from the literature.

2 General setting

We consider the problem of finding the global minimum of a continuous function $f(x_1, ..., x_n)$ where the variables are subject to different types of constraints:

- box constraints: $x_j \in [x_j^-, x_j^+]$, $j = 1, 2, ..., n$;

- linear inequality constraints: $p_i^T x \geq q_i$, $i = 1, ..., m_L$;

- convex quadratic inequality constraints: $x^T Q_k x + r_k^T x + s_k \leq 0$, $k = 1, 2, ..., m_Q$;

If some variables x_j are not boxed, we assume $x_j^- = -BIG$ and/or $x_j^+ = BIG$, where BIG is a large positive number.

If we denote by $\mathbb{X}$ the feasible set, it is convenient to write $\mathbb{X}$ as the intersection $\mathbb{X} = \mathbb{X}_B \cap \mathbb{X}_L \cap \mathbb{X}_Q$ where

- $\mathbb{X}_B = \{(x_1, ..., x_n) | x_j \in [x_j^-, x_j^+], \forall j\}$

- $\mathbb{X}_L = \{(x_1, ..., x_n) | p_i^T x \geq q_i, \forall i\}$

- $\mathbb{X}_Q = \{(x_1, ..., x_n) | x^T Q_k x + r_k^T x + s_k \leq 0, \forall k\}$.

We will handle the box, linear and convex quadratic constraints implicitly; other constraints can eventually be handled via penalization.

We will also distinguish the two situations where the solution is to be located on the boundary or possibly inside the (convex) feasible set $\mathbb{X}_{BLQ}$ obtained by the box, linear and convex quadratic constraints, producing the two problems:

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.231

$(P) : \min\{f(x)|x \in \mathbb{X}_{BLQ}\}$ or, denoting $\partial\mathbb{X}$ the boundary of $\mathbb{X}$,

$(\partial P) : \min\{f(x)|x \in \partial\mathbb{X}_{BLQ}\}$.

A case of particular interest is when function f is concave, as it is well known that the optimal solution is "naturally" on the boundary or at a vertex of $\mathbb{X}$ and, except for special cases, the global minimisation is NP-hard.

Recent literature on global optimization of general (continuous) objective functions has dedicated extended attention to the Differential Evolution (DE) algorithms, introduced in recent years for the box-constrained problem and modified in different ways to handle more general constraints; the inclusion of non-box constraints into the problems solved by DE is still an open problem. We suggest a simple technique to handle linear and convex quadratic constraints, directly by transforming the sampling of x on the search domain into the sampling on a set of directions u,

$$x = a + tu \tag{1}$$

where a is a given point (called the *base point*), u is the sampling direction and t is a parameter to be determined such that x is feasible.

It is well known from convex analysis that, if $\mathbb{X}_{BLQ}$ (in particular $\mathbb{X}_B, \mathbb{X}_L$ and $\mathbb{X}_Q$) are (convex and) bounded, then the set of values t for which $x \in \mathbb{X}_{BLQ}$ is a closed interval $[t^-, t^+]$ (eventually empty if $a + tu$ does not intersect $\mathbb{X}_{BLQ}$).

In our implementation, we distinguish two cases for the *base point* a:

i) $a \in \mathbb{X}_{BLQ}$; in this case every direction produces feasible x's, for given u, belonging to the segment having extrema $x_u^- = a$ and $x_u^+ = a + t^+u$ (i.e. $t^- = 0$ and $t^+ \geq 0$).

ii) $a \notin \mathbb{X}_{BLQ}$; in this case the feasible points x, for a given u, are the points of the segment having extrema $x_u^- = a + t^-u$ and $x_u^+ = a + t^+u$.

In the following, we will see how to determine t^- and t^+ from the three types of constraints. We will essentially concentrate on case i) and eventually we will use case ii) to find a feasible a and then proceed with i). We suggest a way of handling box and/or linear constraints by using linear trasformations (1) of the variables where $u \neq 0$ are unconstrained variables (directions) and t is a positive real number computed to ensure that $x \in \partial\mathbb{P}$ and a is any point in $\mathbb{P}$.

As we have said, the idea of our implementation requires the calculation of the appropriate values t^- and t^+ for a given direction u and for each type of constraint (box, linear or convex quadratic inequalities).

Consider the box $\mathbb{X}_B$ of $\mathbb{R}^n$

$$\mathbb{X}_B = \left\{x|x_j^- \leq x_j \leq x_j^+, j = 1, ..., n\right\}$$

and a feasible point $a \in \mathbb{X}_B$ (e.g. $a_j = \frac{x_j^- + x_j^+}{2}$); let also $u \in \mathbb{R}^n \backslash \{0\}$ be given. It is easy to see that a point $x = a + tu$, $t \geq 0$, belongs to $\mathbb{X}_B$ for all values of $t \in [0, t^+]$ where

$$t^+ = \min\left\{\min\{\frac{x_j^- - a_j}{u_j}\,|u_j < 0\}, \min\{\frac{x_j^+ - a_j}{u_j}\,|u_j > 0\}\right\}$$

and for $t = t^+$ we get the point $a + t^+u$ at the boundary $\partial\mathbb{X}_B$. Note that the assumption of $u \neq 0$ is essential and the components for which $u_j = 0$ are irrelevat for the computation of t^+. If the *base point* is not in the box, we have to determine also t^- and to see if u is a feasible direction. We obtain the following rule:

- if, for some j, $u_j = 0$ and $x_j^- - a_j > 0$ or $x_j^+ - a_j < 0$, then direction u is not acceptable to reach $\mathbb{X}_B$;

- otherwise, define

$$t' = \max\left\{\max\{\tfrac{x_j^+ - a_j}{u_j}\,|u_j < 0\}, \max\{\tfrac{x_j^- - a_j}{u_j}\,|u_j > 0\}\right\}$$

and

$$t'' = \min\left\{\min\{\tfrac{x_j^+ - a_j}{u_j}\,|u_j > 0\}, \min\{\tfrac{x_j^- - a_j}{u_j}\,|u_j < 0\}\right\};$$

if $t' < t''$ then $t^- = t'$ and $t^+ = t''$, otherwise no t exists and direction u is not acceptable.

Behaving this way with an acceptable direction u, the sampling of the objective function is on the border of the constraints; this makes avoidable to sample outside the feasible region and no penalization is needed for the management of a point outside.

Consider now the case of linear constraints and let $\mathbb{X}_L$ be the corresponding polyhedron

$$\mathbb{X}_L = \{x|p_i^T x \geq q_i, i = 1, ..., m_L\}.$$

Also in this case we first assume a feasible $a \in \mathbb{X}_L$. It is easy to see that a point $x = a + tu$, $t \geq 0$ belongs to $\mathbb{X}_L$ for all values of $t \in [0, t^+]$ where

$$t^+ = \min\{\frac{q_i - p_i^T a}{p_i^T u}\,|p_i^T u < 0\} \tag{2}$$

and if the set of indices $\{i\,|p_i^T u < 0\}$is empty, then the polyhedron $\mathbb{X}_L$ is unbounded in the direction u.

Obviously, we get the boundary $\partial\mathbb{X}_L$ for $t = t^+$.

In the general case where $a \in \mathbb{R}^n$ is an arbitrary point (not belonging to $\mathbb{X}_L$) and $u \neq 0$ is a given direction, the points $x = a + tu$ with $t \geq 0$ belong to the polytope $\mathbb{X}_L$ for $t \in [t^-, t^+]$ where t^- and t^+ are calculated as follows (note that $a + tu \in \mathbb{X}_L$ if $tp_i^T u \geq q_i - p_i^T a\ \forall i$).

We have three cases:

1) $p_i^T u = 0$: - if $q_i - p_i^T a > 0$ then no t exists and direction u is not acceptable to reach $\mathbb{X}_L$; - if $q_i - p_i^T a \leq 0$ then any t is valid and the $i - th$ constraint can be ignored.

2) $p_i^T u > 0$: - if $q_i - p_i^T a > 0$ then t must satisfy $t \geq \frac{q_i - p_i^T a}{p_i^T u}$; - if $q_i - p_i^T a \leq 0$ then any t is valid and the $i - th$ constraint can be ignored.

3) $p_i^T u < 0$: - if $q_i - p_i^T a > 0$ then no t exists and direction u is not acceptable; - if $q_i - p_i^T a = 0$ then only $t = 0$ is acceptable for the $i-th$ constraint; - if $q_i - p_i^T a < 0$ then t must satisfy $t \leq \frac{q_i - p_i^T a}{p_i^T u}$.

Consequently, we obtain

$$t^- = \max\{\frac{q_i - p_i^T a}{p_i^T u} \,|p_i^T u > 0, q_i - p_i^T a > 0\}, \quad (3)$$

if the set of indices $\{i\,|p_i^T u > 0, q_i - p_i^T a > 0\}$ is not empty (otherwise $t^- = 0$) and

$$t^+ = \min\{\frac{q_i - p_i^T a}{p_i^T u} \,|p_i^T u < 0, q_i - p_i^T a < 0\} \quad (4)$$

if the set of indices $\{i\,|p_i^T u < 0, q_i - p_i^T a < 0\}$ is not empty (otherwise $t^+ = +\infty$ and $\mathbb{X}_L$ is unbounded).

Finally, in the case of convex quadratic inequalities, let $\mathbb{X}_Q$ be the corresponding convex set

$$\mathbb{X}_Q = \{x | x^T Q_k x + r_k^T x + s_k \leq 0, k = 1, 2, ..., m_Q\}.$$

Also in this case we first assume a feasible $a \in \mathbb{X}_Q$.

After the substitution $x = a + tu$, $t \geq 0$, the constraints can be written as

$$A_k t^2 + B_k t + C_k \leq 0 \quad (5)$$

where

$$\begin{array}{rcl} A_k & = & u^T Q_k u \\ B_k & = & u^T Q_k a + a^T Q_k u + r_k^T u \\ C_k & = & a^T Q_k a + r_k^T a + s_k \end{array}$$

As the constraints are assumed to be convex, we have $u^T Q_k u \geq 0$ i.e. $A_k \geq 0$. If, for some k, the parabolic inequality (5) is never satisfied with respect to $t \geq 0$, then direction u is not acceptable.

Define also the following quantities

$$D_k = B_k^2 - 4 A_k C_k.$$

Note that if a is feasible for the quadratic constraints, then $C_k \leq 0$ and $t = 0$ is always feasible; if only $t = 0$ satisfies the constraints, then direction u must be discarded. Also note that in this case we always have $D_k \geq 0$.

Going into the details, we have three cases.

1. $A_k > 0, D_k > 0$: the constraint is satisfied for all $t \in [0, \frac{-B_k + \sqrt{D_k}}{2A_k}]$; set $n_k = 0$ if $\frac{-B_k + \sqrt{D_k}}{2A_k} = 0$, otherwise $n_k = 1$ and $t'_k = 0$, $t''_k = \frac{-B_k + \sqrt{D_k}}{2A_k}$.

2. $A_k = 0, B_k \neq 0$: in this case feasible t satisfy $t \leq \frac{-C_k}{B_k}$; set $n_k = 0$ if $\frac{-C_k}{B_k} \leq 0$, otherwise $n_k = 1$ and $t'_k = 0$, $t''_k = \frac{-C_k}{B_k}$.

3. $A_k = 0, B_k = 0$: here, as $C_k \leq 0$ then any t is valid (set $n_k = 2$).

The feasible interval $[0, t^+]$ for the values of t is then obtained (if $n_k > 0$ for all k, otherwise u is discarded) by

$$t^+ = \min\{t''_k | n_k = 1\}.$$

The case where a is not feasible requires to determine the interval $[t^-, t^+]$ for the valid values of t and it is possible that no such t exist.

Note that if a is not feasible, then $C_k > 0$ and D_k can be negative. The details are as follows:

1. $A_k > 0, D_k < 0$: no feasible t exists and direction u is discarded; set $n_k = 0$.

2. $A_k > 0, D_k \geq 0$: in this case we need $\frac{-B_k - \sqrt{D_k}}{2A_k} \geq 0$; set $n_k = 0$ if $\frac{-B_k - \sqrt{D_k}}{2A_k} < 0$, otherwise $n_k = 1$ and $t'_k = \frac{-B_k - \sqrt{D_k}}{2A_k}$, $t''_k = \frac{-B_k + \sqrt{D_k}}{2A_k}$.

3. $A_k = 0, B_k > 0$: in this case no feasible t exist; set $n_k = 0$.

4. $A_k = 0, B_k < 0$: in this case feasible t satisfy $t > \frac{-C_k}{B_k}$; set $n_k = 1$ and $t'_k = \frac{-C_k}{B_k}$, $t''_k = +\infty$.

5. $A_k = 0, B_k = 0$: here, as $C_k > 0$ then no t is valid (set $n_k = 0$).

The feasible interval $[t^-, t^+]$ for the values of t is then obtained (if $n_k > 0$ for all k, otherwise u is discarded) by

$$t^- = \max\{t'_k | n_k = 1\} \text{ and } t^+ = \min\{t''_k | n_k = 1\}.$$

3 Implementation and computational tests

As we have seen in the previous section, in cases of box, linear and convex quadratic constraints, the computation of t_u such that $x = a + t_u u \in \partial\mathbb{X}_{BLQ}$ or of $[t_u^-, t_u^+]$ such that $x = a + tu \in \mathbb{X}_{BLQ}$ $\forall t \in [t_u^-, t_u^+]$ is easy; starting with a feasible point $a \in \mathbb{X}_{BLQ}$ the transformation (1) can be directly substituted into the objective function obtaining the equivalent problem

$$\min F(u) = f(a + t_u u), u \in [-1, 1]^n$$

where t_u is the unique values of $t > 0$ such that $a + t_u u \in \partial\mathbb{X}_{BLQ}$.

Our idea of the algorithm implemented for DE is to find a solution of problem (∂P) by determining a valid direction u, computing the current value of t_u and the corresponding $x_u = a + t_u u$ and evaluating the objective function there.

Clearly, for a given valid direction u, also all directions αu with positive α and $\alpha u \in [-1, 1]^n$ produce the same x_u at the boundary of the feasible region. On the other hand, DE produces new search points by "sampling" the hypercube, so the possibility of equivalent directions (with the same objective value) exists, unless the search itself is controlled and constrained e.g. by generating only normalized directions ($\|u\| = 1$).

In order to avoid this effect, we insert a simple modification of the objective function so that directions near to normalized are preferred; this is done by using a two-objective strategy that prefers normalized directions as a second objective of the search (see [3] and the references therein).

The idea of DE to find *Min* is to start with an initial "population" $x^{(1)} = (x_1, ..., x_n)^{(1)}, ..., x^{(p)} = (x_1, ..., x_n)^{(p)} \in \mathbb{X}$ of p feasible points for each generation (i.e. for each iteration) to obtain a new set of points by recombining randomly the individuals of the current population and by selecting the best generated elements to continue in the next generation. The initial population is chosen randomly to "cover" uniformly the entire parameter space. Denote by $x^{(k,g)}$ the $k-$th vector of the population at iteration (generation) g and by $x_j^{(k,g)}$ its $j-$th component.

At each iteration, the method generates a set of candidate points $y^{(k,g)}$ to substitute the elements $x^{(k,g)}$ of the current population, if $y^{(k,g)}$ is better. To generate $y^{(k,g)}$ two operations are applied: recombination and crossover.

A typical recombination operates on a single component $j \in \{1, ..., n\}$ by generating a new perturbed vector of the form $v_j^{(k,g)} = x_j^{(r,g)} + \gamma[x_j^{(s,g)} - x_j^{(t,g)}]$, where $r, s, t \in \{1, 2, ..., p\}$ are chosen randomly and $\gamma \in]0, 2]$ is a constant (eventually chosen randomly for the current iteration) that controls the amplification of the variation.

The potential diversity of the population is controlled by a crossover operator, that construct the candidate $y^{(k,g)}$ by crossing randomly the components of the perturbed vector $v_j^{(k,g)}$ and the old vector $x_j^{(k,g)}$ ($j_1, j_2, ..., j_h$ are random),

$$y_j^{(k,g)} = \begin{cases} v_j^{(k,g)} & \text{if } j \in \{j_1, j_2, ..., j_h\} \\ x_j^{(k,g)} & \text{if } j \notin \{j_1, j_2, ..., j_h\} \end{cases}$$

and the components of each individual of the current population are modified to $y_j^{(k,g)}$ by a given probability q.

Typical values are $\gamma \in [0.2, 0.95]$, $q \in [0.7, 1.0]$ and $p \geq 5n$ (the higher p, the lower γ).

The candidate $y^{(k,g)}$ is then compared to the existing $x^{(k,g)}$ by evaluating the objective function at $y^{(k,g)}$: if better then substitution occurs in the new generation $g + 1$, otherwise $x^{(k,g)}$ is retained.

Many variants of the recombination schemes have been proposed and some seem to be more effective than others (see e.g. [12]).

3.1 Computational tests

We have implemented the DE procedure using MATLAB and we have run a series of examples with problems taken from the literature and problems with randomly generated constraints. The problems are classified in three classes of examples: a) Problems with only box constraints and solutions on the boundary; b) Problems with linear and convex quadratic constraints and solutions on the boundary; c) Problems with all types of constraints and internal solutions.

As the basic routines for DE, we have used the one described in [13]. The strategies for DE are numbered from 1 to 6 as 1=DE/rand/1, 2=DE/local-to-best/1, 3=DE/best/1 with jitter, 4=DE/rand/1 with per-vector-dither, 5=DE/rand/1 with per-generation-dither, 6=DE/rand/1 either-or-algorithm (see routine DEOPT in [13]).

The results are illustrated in the tables, where FE$_{best}$ and FE$_{worst}$ specify the function evaluation counts for the given best S$_{best}$ and worst S$_{worst}$ strategy, respectively; γ is the value of the DE parameter with best performance and f^* is the optimum value found.

The first example is:

$\min_x \; 1.45x_1^2 + 0.75x_2^2 + 7x_3^2 - 2x_1x_3 - 2x_2x_3 + 2\sqrt{x_1 + x_2 + x_3}$

s.t.

$2x_1 - x_2 - x_3 \geq -1$

$-3x_2 - x_3 \geq -9$

$-3x_1 - 1.5x_2 - x_3 \geq -12$

$-3x_1 + 3x_2 - x_3 \geq -3$

$-x_1 + x_2 - 2x_3 \geq -3.5$

$2x_1 + 3x_2 - x_3 \geq 7$

$x_i \geq 0$

The results are

γ	S$_{best}$	FE$_{best}$	S$_{worst}$	FE$_{worst}$	f^*
0.5	3	930	5	3930	6.5086

For the box constrained problems with solutions at the boundary, we consider the following n-dimensional quartic function with a random noise variable defined by

(P1): $\min \; \sum_{i=1}^{n} [2.2(x_i + e_i)^2 - (x_i + e_i)^4]$

s.t. $\quad x_i \in [-2, 2], i = 1, 2, ..., n$

where e_i is uniformly distributed on $[0.2, 0.4]$. The global minimization (P1) is NP-hard and the solution is at a vertex of the box. We have generated different problems with n from 5 and 10.

The second problem concerns the packing of $p > 1$ equal circles in a square; it is a minimization of a concave function over a box (with p circles we have a problem in $n = 2p$ variables representing the centers):

(P2): $\min_{(x_i, x_{p+i}) \in U} \left(-\max_{i<j} \left[(x_i - x_j)^2 + (x_{p+i} - x_{p+j})^2 \right] \right)$

s.t.

$0 \leq x_i, x_{p+i} \leq 1$ for $i = 1, 2, ..., p$.

where $U = [0, 1]^2$

The third and fourth problems consider the minimization of concave functions over a box $x \in [0, 1]^n$ (from [7]):

(P3): $\min -3\sum_{j=1}^{n} x_j^2 + 2\sum_{j=1}^{n-1} x_j x_{j+1}$

(P4): $\min -\left(\sum_{j=1}^{n} \frac{1}{j}x_j\right) \ln\left(1+\sum_{j=1}^{n}\frac{1}{j}x_j\right)$

The results for DE are illustrated in the following table (FE_{best} and FE_{worst} specify the number of function evaluations for the given best S_{best} and worst S_{worst} strategies, respectively, f^* is the optimum value and f_{worst} is the valued reached by the worst strategy, that may differ from f^* when the number of function evaluations has reached the upper limit of 249600 (corresponding to 5000 generations).

Problem	γ	S_{best}	FE_{best}	f^*	S_{worst}	FE_{worst}	f_{worst}
P1, n=5	0.6	3	5050	-87.3898	4	33550	-87.3898
P1, n=10	0.6	2	44600	-169.6234	1	(249600)	-169.6183
P2, p=5	0.6	3	9100	-0.7071	5	82300	-0.7071
P2, p=9	0.6	3	42400	-0.5	5	>249600	-0.5
P3, n=5	0.5	3	3050	-5.1962	4	20550	-5.1962
P3, n=10	0.5	3	7100	-6.7082	5	(249600)	-6.7081
P4, n=5	0.5	3	2300	-2.7146	5	13300	-2.7146
P4, n=10	0.5	2	9600	-4.0079	4	203100	-4.0079

The results reported in [7] for problems P3 and P4 are not comparable with the ones in the previous table because the authors use other parameters (computational time, number of iterations) and do not show the number of function evaluations; meanwhile, the times are out of our experience obtained with a code running in MATLAB. Notice that the sizes of the problems in the mentioned paper are limited to $n = 5$ or 6, so the case with $n = 10$ is not solved in [7].

The tests for fully constrained problems with boundary solutions are taken from [5], [7], [10], [14], and [15].

(P5): $\min_x \; 5\sum_{j=1}^{4} x_j - 5\sum_{j=1}^{4} x_j^2 - \sum_{j=5}^{13} x_j$

s.t.

$$\begin{aligned}
&-2x_1 - 2x_2 - x_{10} - x_{11} \geq -10\\
&-2x_1 - 2x_3 - x_{10} - x_{12} \geq -10\\
&-2x_2 - 2x_3 - x_{11} - x_{12} \geq -10\\
&2x_4 + x_5 - x_{10} \geq 0\\
&2x_6 + x_7 - x_{11} \geq 0\\
&2x_8 + x_9 - x_{12} \geq 0\\
&8x_1 - x_{10} \geq 0\\
&8x_2 - x_{11} \geq 0\\
&8x_3 - x_{12} \geq 0\\
&0 \leq x_i \leq 1, (i = 1, ..., 9)\\
&0 \leq x_i \leq 100, (i = 10, 11, 12)\\
&0 \leq x_{13} \leq 1
\end{aligned}$$

The global minimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ and $f^* = -15$.

(P6) $\min x^T Q x$

s.t.

$$\sum_{j=1}^{5} x_j \geq 1$$

$$0 \leq x_j \leq 1, j = 1, ..., 5$$

where

$$Q = \begin{bmatrix} 14 & 15 & 16 & 0 & 0 \\ 15 & 14 & 12.5 & 22.5 & 15 \\ 16 & 12.5 & 10 & 26.5 & 16 \\ 0 & 22.5 & 26.5 & 0 & 0 \\ 0 & 15 & 16 & 0 & 14 \end{bmatrix}$$

(P7) Min $x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 + (x_3-3)^2 + 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 + (x_{10}-7)^2 + 45$

s.t.

$$\begin{aligned}
&-105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0\\
&10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0\\
&-12 - 8x_1 + 2x_2 + 5x_9 - 2x_{10} \leq 0\\
&3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0\\
&5x_1^2 + 8x_2 + (x_3-6)^2 - 2x_4 - 40 \leq 0\\
&x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0\\
&0.5(x_1-8)^2 + 2(x_2-2)^2 + 3x_5^2 - 6x_6 - 30 \leq 0\\
&-3x_1 + 6x_2 + 12(x_9-8)^2 - 7x_{10} \leq 0\\
&-10 \leq x_j \leq 10
\end{aligned}$$

Solution is $x^* = (2.172, 2.364, 8.774, 5.096, 0.991, 1.431, 1.322, 9.829, 8.280, 8.376)$ with $f^* = 24.306)$

The results for DE are illustrated in the following table; the structure of the table is mimicked from the previous one.

Problem	γ	S_{best}	FE_{best}	f^*	S_{worst}	FE_{worst}	f_{worst}
P5, n=13	0.4	1	79430	-15.0	6	281520	-15.0
P6, n=5	0.5	3	2300	0.0	4	9050	0.0
P7, n=10	0.6	3	17100	24.3062	5	239600	24.3062

Comparing with the perfomances of DE published in [15], our implementation compares favourably. For example, the number of function evaluations for functions g01 (our P5) and g07 (our P7) are respectively 111034 (our 3100) and 83476 (our 17100).

The test problems for concave minimization with linear constraints are P8 to P13, of the form

$$\min\{f(x)|p_i^T x \geq q_i, \forall i, x_j \in [0, 100], \forall j\},$$

where $f(x)$ is one of the following functions:

P8: $f_1(x) = -\left(x_1 + \sum_{j=2}^{n} \frac{j-1}{j} x_j\right)^{3/2}$

P9: $f_2(x) = -\left(\sum_{j=1}^{n} \frac{1}{j}x_j\right) \ln\left(1 + \left(\sum_{j=1}^{n} \frac{1}{j}x_j\right)\right)$

P10: $f_3(x) = -3\sum_{j=1}^{n} x_j^2 + 2\sum_{j=1}^{n-1} x_j x_{j+1}$

P11: $f_6(x) = -\sum_{j=1}^{n} [\ln(1+x_j) - \exp(-x_j/n)]$

P12: $f_8(x) = -\sqrt{\sum_{j=1}^{n} x_j^2 - \left(\sum_{j=1}^{n} \sqrt{x_j}\right)^{3/2}}$

P13: $f_{10}(x) = -\prod_{j=1}^{n} x_j^{\alpha_j}$, where the $\alpha_j \geq 0$ are randomly generated with $\sum_{j=1}^{n} \alpha_j = k \geq 1$ (it follows that f_{10} is concave) We use $k = 2$.

The linear constraints for f_1 to f_{10} are of the form $p_i^T x \geq q_i, i = 1, ..., m_L$ and are generated randomly according to the procedure suggested by [10]. The point $a = (1, ..., 1)$ is always feasible.

The runs for DE are documented in the following table (with the same structure as before); they have the number of linear constraints set to $m_L = 25$

Problem	γ	S_{best}	FE_{best}	f^*	S_{worst}	FE_{worst}	f_{worst}
P8, n=10	0.7	3	32100	-41.7017	1	255100	-41.7017
P9, n=10	0.7	3	26100	-9.7248	1	188100	-9.7248
P10, n=10	0.8	3	58100	-107.2079	2	85100	-107.2079
P11, n=10	0.7	3	15100	0.3255	1	90100	0.3255
P12, n=10	0.7	3	19100	-41.1397	4	124100	-41.1397
P13, n=10	0.7	3	14100	-2.1334	4	112100	-2.1334

Problem P10 comes from [7] where it is solved for instances of n between 5 and 9 and with m_L between 8 and 16 linear constraints (generated in a similar way as us); the authors espose the number of interations and computational times (up to 9000 seconds on a workstation), not the count of function evaluations.

References

[1] M.M. Ali, L.P. Fatti, "A differential free point generation scheme in the differential evolution algorithm", Journal of Global Optimization, 35 (2006) 551-572.

[2] J. Brest, M. Sepesy Maucec, "Population size reduction for the differential evolution algorithm", Appl. Intell. DOI 10.1007/s10489-007-0091-x, Springer, LLC 2007.

[3] Z. Cai, Y. Wang, " A multiobjective optimization-based evolutionary algorithm for constrained optimization", IEEE Transactions on Evolutionary Computation, 10 (2006) 658-675.

[4] C.A.C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state if the art", Comput. Methods Appl. Mech. Eng. 191 (2002) 1245-1287.

[5] C.A. Floudas, P.M. Pardalos et al., Handbook of test problems in local and global optimization, Kluwer Academic Publ., 1999.

[6] G. Leguizamon, C.A. Coello Coello, "Boundary search for constrained numerical optimization problems in ACO algorithms", in M. Dorigo et al. (Eds.), Proceedings of ANTS 2006, Lecture Notes in Computer Science n. 4150, Springer, 2006, 108-119.

[7] M. Locatelli, N.V. Thoai, "Finite exact branch and bound algorithms for concave minimization over polytopes", Journal of Global Optimization, 18 (2000) 107-128.

[8] E. Mezura-Montes, C.A. Coello Coello, E.I. Tun-Morales, "Simple feasibility rules and differential evolution for constrained optimization", Proceeding of IMICAI 2004, Lecture Notes in Artificial Intelligence, 2972, Springer, 2004, 707-716.

[9] E. Mezura-Montes, J. Velazquez-Reyes, C.A. Coello Coello, "Modified differential elovution for constrained optimization", Proceedings of Congress on Evolutionary Computation (CEC), 2006, 25-32.

[10] K. Moshirvaziri, "Construction of test problems for concave minimization under linear and nonlinear constraints", J.O.T.A. 98 (1998) 83-108.

[11] K. Price, "An introduction to differential evolution", in D. Corne, M. Dorigo, F. Glover (Ed.), New Ideas in Optimization, McGraw Hill, 1999, 79-108.

[12] R. Storn, K. Price, "Differential Evolution: a simple and efficient heuristic for global optimization over continuous spaces", ICSI technical report TR-95-012, Berkeley University, 1995. Also, Journal of Global Optimization, 11 (1997) 341-359.

[13] R. Storn, Differential Evolution for MATLAB, International Computer Science Institute (ICSI), Berkeley, CA 94704, http://http.icsi.berkeley.edu/~storn, 1997.

[14] X.L. Sun, J.L. Li, "A branch-and-bound based method for solving monotone optimization problems", Journal of Global Optimization, 35 (2006) 367-385.

[15] M. Zhang, W. Luo, X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization", Information Sciences, 178 (2008) 3043-3074.