Dual Evolutionary Optimization

INSA Technical report No. 01 / 2001

May 2001

Rodolphe Le Riche[*] and Frédéric Guyon[†]

Abstract

The most general strategy for handling constraints in evolutionary optimization is through penalty functions. The choice of the penalty function is critical to both success and efficiency of the optimization. Many strategies have been proposed for formulating penalty functions, most of which rely on arbitrary tuning of parameters. An new insight on function penalization is proposed in this paper that relies on the dual optimization problem. Use of duality is particularly relevant to evolutionary optimization since it applies to any type of variable and function spaces (continuous or discrete, linear or not, with one or many extrema) as evolutionary algorithms do. An evolutionary algorithm for approximately solving dual optimization problems is first presented. It calculates Lagrange multipliers at the optimum, which have great practical importance for cost sensitivity analysis. Next, an exact penalty function without extra parameter to be tuned is proposed. Among a Lagrangian based class of penalty functions, it applies a minimal penalty, which contributes to keeping the evolutionary search efficient. Numerical tests are provided for continuous variables and inequality constraints.

Keywords: Constrained optimization, evolutionary optimization, duality, Lagrange multipliers, penalization.

1 Introduction

Evolutionary optimization ([1]) stands today as one of the primary method for tackling difficult optimization problems. Most practical applications of optimization are constrained problems. The issue of how to handle constraints in evolutionary optimization is therefore central and has received a lot of attention in the last decade. The efficiency of the method and its ability to generate optimal feasible solution is at stake.

[*]R. Le Riche is with the Lab. de Mécanique de Rouen, INSA / CNRS UMR 6138, Rouen, France. E-mail: Rodolphe.Leriche@insa-rouen.fr .

[†]F. Guyon is with the Bio-statistics Bio-mathematics lab., Paris 7 univ., France. E-mail: guyon@urbb.jussieu.fr .

Four types of methods for handling constraints exist: penalization of infeasible solutions([8],[11], [23], [25], [29], [2], [6]), projection of infeasible solutions onto the feasible domain ([20] [27] [28]), co-evolution of populations which together solve the constrained optimization problem ([22]), and constraints representation building in the course of the search ([21],[24]). These approaches are related and have been coupled, like co-evolution and penalty methods ([13] and [30]), or penalty and projection [16]. Reviews on constraints handling in evolutionary optimization can be found in [17] and [18].

Among penalization strategies, one distinguishes static, dynamic and adaptive methods. Static penalties depend neither on the number of points sampled during the search nor on their performance ([8], [13]). Dynamic penalties ([16], [10]) are function of the number of points sampled while adaptive penalties ([2], [6],[23],[11]) vary with points evaluations. Mixed approaches exist, e.g. in [29].

Duality and related concepts such as Lagrange multipliers have yielded some of the most efficient general purpose methods of mathematical programming for continuous, differentiable and locally convex problems ([19],[15],[26]). Of particular practical importance are augmented Lagrangian functions and Lagrange multipliers updating techniques. On particular cases, it has been possible to formulate exact penalty functions ([9], [14]).

Augmented Lagrangian functions and Lagrange multipliers updating have been applied to derive adaptive penalty functions in evolutionary algorithms. Bean and Hadj-Alouane ([2]) have proposed a penalty adaptation scheme which resembles Lagrange multipliers updating strategies. Kim and Myung ([11]) and Tahk and Sun ([30]) have used augmented Lagrangian penalty functions in evolutionary optimization, calculating Lagrange multipliers as a by-product of the search. In [30], a co-evolutionary algorithm simultaneously evolves a population of unknowns variables and a population of Lagrange multipliers.

The current work is also concerned with solving the dual optimization problem as a way to adapt a penalty function. Fundamentally, it differs from previous works in two aspects. First, an evolutionary is devised that explicitly solves the dual optimization problem. Second, the penalty function is not an augmented Lagrangian. Indeed, augmented Lagrangians were originally derived for mathematical programming. They are continuously differentiable functions. They depend on the choice of a penalty parameter (the “augmented” term), which, if taken too small, leaves local optima. Continuous differentiability is not needed in evolutionary optimization. The freedom gained in the formulation of the penalty permits removing the parameter and obtaining global optimality properties.

The text starts with a review of dual optimization principles. Then, linear and evolutionary algorithms are coupled to solve the dual optimization problem. Third, the non-equivalence of primal and dual problems is analyzed. It results in a new discontinuous exact penalty function that satisfies a minimal penalty rule. Finally, numerical tests are carried out where dual and primal problems are successively solved.

2 Dual optimization

Dual optimization principles underlying the rest of the discussion are now reviewed.

2.1 Duality: definitions and fundamental properties

The primal constrained optimization problem (P) is,

$$(P) \qquad \begin{cases} \min_{x \in S} \ f(x), \\ \text{such that } g(x) \ \leq \ 0. \end{cases} \tag{1}$$

where f, the objective function, and g, the constraint, are bounded functions (not necessarily continuous or differentiable). We further assume that there is at least one feasible point in S, i.e., a point x such that $g(x) \ \leq \ 0$. The search is performed in the (primal) space of the design variables $x \in S$, which is a closed and bounded set. For the sake of simplicity, the number of constraints is limited to one. It should be noted that problems having $m \ > \ 1$ constraints can always be set in terms of a single constraint by taking the most critical constraint,

$$\begin{cases} \min_{x \in S} \ f(x), \\ \text{such that } g(x) \ = \ \max_{i=1,m}\{g_i(x)\} \ \leq \ 0. \end{cases} \tag{2}$$

The set of solutions of (P) is denoted X^*. The Lagrangian formulation (P_λ) of the primal problem is,

$$(P_\lambda) \qquad min_{x \in S} L(x, \lambda), \tag{3}$$

where,

$$L(x, \lambda) \ = \ f(x) + \lambda g(x). \tag{4}$$

λ is a Lagrange multiplier. The set of solutions of (P_λ) is X_λ. We further assume that for each $\lambda \ \leq \ 0$, there exists at least a bounded solution $x_\lambda \ \in \ X_\lambda$. This assumption is fulfilled, for example, if f and g are continuous (Weierstrass Theorem, [19]).
The dual function is,

$$\phi(\lambda) \ = \ \min_{x \in S} \ L(x, \lambda), \tag{5}$$

and the dual problem is stated as,

$$(D) \qquad \max_{\lambda \geq 0.} \ \phi(\lambda). \tag{6}$$

The dual search occurs in the Lagrange multipliers space. The solution of (D) are the Lagrange multipliers at the optimum, λ^*, and associated values of x are X_{λ^*}. An example of dual function is given in Fig. 1. When multiple constraints are handled through the maximization scheme of equation (2), λ^* is the optimal Lagrange multiplier of the most critical constraint.

The motivations for solving the dual optimization problem (D) are i) to directly solve the primal problem (P) when $X^* = X_{\lambda^*}$, ii) to calculate λ^*, which permits formulating exact penalty functions (cf. Section 4). At first glance, the dual problem seems much more complex than the primal problem since calculating the dual function involves solving an optimization problem, $\min_{x \in S} L(x, \lambda)$. However, favorable properties of the dual function added to the possibility of approximately solving (D) make duality a powerful approach for rational constraints handling.

Property 1 (Concavity of ϕ) *The dual function $\phi(\lambda)$ is concave in λ.*

Property 2 (sub-gradient) *For all $\lambda \geq 0$, let us denote $X_\lambda = \{x \in S \ / \ L(x,\lambda) = \phi(\lambda)\}$. Then, for all $x_\lambda \in X_\lambda$, $g(x_\lambda)$ is a sub-gradient[1] of ϕ at λ.*

The two above properties, proofs of which can be found in [19], are valid under very general conditions (f and g bounded). They considerably simplify the resolution of (D) since ϕ is a concave function with a known sub-gradient. Lagrangian based penalty functions, such as the ones introduced in [2], [6], and [11], have penalty adaptation schemes where, schematically, the Lagrange multiplier is increased if the current best solution in terms of the penalized objective function is infeasible ($g(x_b) > 0$) and vice versa. Since $g(x_b)$ is an approximation of the sub-gradient of $\phi(\lambda)$, those penalty adaptation schemes are variations of a gradient based dual search. In the current work, an alternative strategy is taken to solve the dual problem (see Section 3 later).

2.2 Approximate dual problem

In terms of λ, (D) is easy to solve because it is a concave problem with a known subgradient. (D) has no local maximum and many algorithms exist to solve it (nondifferentiable optimization or linear programming, such as presented hereafter). The main difficulty remains the resolution of $\min_{x \in S} \{f(x) + \lambda g(x)\}$ at a given λ in the primal space. For this reason, the dual optimization problem is now approximated by restricting S to a discrete set of points T. It yields an approximate dual function,

$$\phi_T(\lambda) = \min_{x \in T} \{f(x) + \lambda g(x)\}, \tag{7}$$

and an approximate dual problem,

$$(AD) \qquad \max_{\lambda \geq 0.} \phi_T(\lambda), \tag{8}$$

[1] A sub-gradient is a generalization of gradient to non-smooth functions ([31]).

where $T \subset S$ is a set of points of the primal space. By construction ϕ_T is concave and piecewise linear. (AD) can be formulated as a linear programming problem:

$$(AD) \qquad \begin{cases} \max_{w,\lambda\geq 0} \; w \;, \\ \text{such that } f(x_i) \;+\; \lambda g(x_i) \;\geq\; w \;, \forall x_i \in T \;, \\ \lambda \;\leq\; \lambda_{max} \;, \end{cases} \tag{9}$$

where λ_{max} is an arbitrarily large upper bound on Lagrange multiplier meant to ensure the existence of a solution to (AD) even when all points in T are infeasible. The linear programming problem (AD) is efficiently solved by a simplex algorithm (cf. Appendix B). Let its solution be λ_T. An illustration of dual and approximate dual functions is given in Fig. 1. The contribution of a vector of primal variables x_i in the dual space is a line $D_i = f(x_i) + \lambda g(x_i)$, of slope $g(x_i)$ and crossing $\lambda \;=\; 0$ at $f(x_i)$. Feasible points are lines of negative slope, infeasible points lines of positive slope. From the definition of the dual function $\phi(\lambda)$ (cf. Equation (5)), all lines associated to x in the primal space are above the dual function $\phi(\lambda)$. The approximate dual function $\phi_T(\lambda)$ is the lower front of all lines associated to x's $\in T$. Tangents to the dual function at λ have a slope $\phi'(\lambda) \;=\; g(x_\lambda)$, $\forall x_\lambda \in X_\lambda$. In Fig. 1, $x_0 \;=\; x_{\lambda_0}$ is an example of such a point. It is the solution of $\min_{x\in S}\{f(x) + \lambda_0 g(x)\}$.

3 A dual evolutionary algorithm

The dual evolutionary optimizer iterates between the primal problem (P_λ) and the approximate dual problem (AD). Based on a particular choice of λ, P_λ resolution by evolutionary optimization produces points to include in T. Based on T, (AD) resolution by simplex algorithm yields a new Lagrange multiplier λ_T. Most dual optimization methods iterate between primal and dual spaces. Our algorithm bears particular resemblance to Dantzig's algorithm ([3], [4]). The difference lies in the evolutionary primal optimization:

- It can visit different basins of attraction of the Lagrangian, $L(x, \lambda)$, during convergence, i.e., it can yield many judicious points to be included in T at each iteration.
- It can handle non-convex, discontinuous functions.

A flow chart of a dual evolutionary optimizer is given in Fig. 2. An evolutionary optimizer (detailed in Appendix A) uses λ_k, that comes from the previous resolution of (AD), to estimate X_{λ_k} by minimizing $L(x, \lambda_k) \;=\; f(x) + \lambda_k g(x)$. During evolutionary convergence, important points are collected in the primal space to be added to T_k: they are the n_f and n_i best feasible and infeasible points and the feasible point with the lowest objective function f. A simplex algorithm is then used to solve (AD) with the updated T_k. It yields a new λ_k and X_k^f, a set of feasible points associated to the active constraints of (AD)

(Equation (9)). Referring to Fig. 1, $X_k^f = \{x_2\}$. It is the set of points associated to the lines of negative slope ($g \leq 0$) going through $\phi_T(\lambda_k)$. A proof that X_k^f is not empty is given in Appendix D. Simplex implementation details are provided in Appendix B. This completes an iteration of the dual evolutionary optimizer. Note that, in order to improve the efficiency of the algorithm, X_{k-1}^k and X_{k-1}^i (when it exists) are included in the initial population of the k-th evolutionary search.

For ϕ twice differentiable and bounded and exact primal searches, one shows that the algorithm converges to λ^* at a rate at least linear (cf. Appendix C). Convergence in more general cases is not proved but observed in paragraph 5. The outputs of the algorithm are $\lambda_{k_{max}}$ and $X_{k_{max}}^f$. $\lambda_{k_{max}}$ is an estimate of λ^*. $X_{k_{max}}^f$ is an estimate of the feasible points in X_{λ^*}.

T_k gathers information from many potentially important points sampled by all evolutionary runs up to iteration k. Values of Lagrange multipliers are inferred from these points through an exact resolution of the approximate dual problem (AD). Such an approach is thought to be more efficient than gradient based dual searches which change λ_k based on a local information, an approximation of $g(x_{\lambda_k})$, $x_{\lambda_k} \in X_{\lambda_k}$.

4 Primal problem resolution

4.1 Problems with a saddle point

The previous paragraph has introduced a coupled evolutionary / simplex algorithm for solving the dual problem (D). But the goal is to tackle the primal problem (P). In fact, problems having a saddle point at the optimum are readily solved. The saddle point property means that,

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*) \quad , \forall x \in S \ , \ \forall \lambda \geq 0 \ . \tag{10}$$

In this case, the dual and the primal problems are equivalent ([19]). One has $X^* = \{x^*\}$, x^* unique, $X_{\lambda^*} = \{x_{\lambda^*}\}$, x_{λ^*} unique, and $x^* = x_{\lambda^*}$. A valid penalty function is simply the Lagrangian $L(x, \lambda^*)$. Once the dual evolutionary algorithm stops, it provides $X_{k_{max}}^f$ (cf. Fig. 2), which contains an estimate of x^* since $x^* = x_{\lambda^*}$ and $X_{k_{max}}^f$ tends to the feasible points in X_{λ^*}. In practice, a lot of problems have a saddle point. For problems without saddle point, solving (D) does not directly provide a solution to (P), but it generates information for properly penalizing the constraints.

4.2 Properties of problems without saddle point

Properties of optimization problems devoid of saddle points are now described. These properties justify the penalty function soon to be introduced.

Problems without a saddle point necessarily have a non-differentiable dual function ϕ at λ^*. Non-differentiability at λ^* occurs when X_{λ^*} contains at least two elements with different associated constraints $g(x_{\lambda^*})$, $x_{\lambda^*} \in X_{\lambda^*}$, i.e., different slopes in the dual space. An example of such a situation is given in Fig. 3 where X_{λ^*} is composed of two competing minima of $L(x, \lambda^*)$, x^f and x^i. For all optimization problems, X_{λ^*} contains at least one feasible element denoted x^f, $g(x^f) \leq 0$ (see Proposition 4 in Appendix D). Numerically, x^f is estimated by $x^f_{k_{max}}$ at the end of the dual evolutionary algorithm (Fig. 2). When there are infeasible points in X_{λ^*}, x^i denotes any instance of them.

Contrary to problems with saddle point, other optimization problems have $X_{\lambda^*} \neq X^*$. This can also be seen on the example of Figure 3 where $X_{\lambda^*} = \{1.058\ ,\ 4.58\}$, $x^f = 1.058$ and $X^* = \{4.5\}$. Solving the dual (D) yields X_{λ^*}, which is not equivalent to calculating X^*, solution of (P).

4.3 A minimal, exact, penalty function

Let $F_p(x)$ denote any penalized objective function. The choice of the penalty function has a profound effect on the evolutionary optimization efficiency. When too high a penalty is imposed on infeasible points, the population is prematurely pushed into the feasible domain, often far from optima x^*. Subsequent convergence to x^* can be extremely slow. In evolutionary terms, penalization makes $F_p(x)$ deceptive. Reciprocally, if too low a penalty is enforced, the algorithm converges into the infeasible domain. The optimal penalty function is problem dependent. However, several authors have described a reasonable heuristic, the minimal penalty rule, as a remedy against penalization induced deceptiveness (Davis [5], Richardson et al. [25], Smith and Tate [29]). It says: on the average, it is best to apply the smallest amount of penalty such that the algorithm converges to a feasible optimum, x^*. For calculation purpose, a more precise definition of "amount of penalty" is needed.

Definition 1 (Amount of penalty) *For optimization problems without a saddle point and such that there is an infeasible solution to the dual, x^i, the amount of penalty, r, is defined as,*

$$r = F_p(x^i) - f(x^i) \ , \tag{11}$$

where $F_p(x^i)$ is any penalized objective function.

This definition is based on x^i, the best infeasible point with respect to $L(x, \lambda^*)$. Characterizing the amount of penalty at x^i is particularly relevant for Lagrangian based penalty functions.

A Lagrangian based penalty function is now gradually introduced. First, the function,

$$\begin{aligned} &f_0(x; \lambda^*, x^f) = f(x) + H(g(x))[\lambda^*(g(x) - g(x^f))] \ , \\ &\text{where } H(y) = 0 \ \text{ if } y \leq 0 \ , \ H(y) = 1 \ \text{ otherwise,} \end{aligned} \tag{12}$$

is an exact penalty function. Exact penalty functions have $x^* \in X^*$ as minimum.

Proof: For all $x \ / \ g(x) \ \leq \ 0$,

$$f_0(x;\lambda^*,x^f) \ = \ f(x) \ \geq \ f_0(x^*;\lambda^*,x^f) \ = \ f(x^*) \ , \tag{13}$$

by definition of x^*. For all $x \ / \ g(x) \ > \ 0$,

$$\begin{aligned} f_0(x;\lambda^*,x^f) \ &= \ f(x) + \lambda^* g(x) - \lambda^* g(x^f) \\ &\geq f(x^f) + \lambda^* g(x^f) - \lambda^* g(x^f) \ \geq \ f(x^*) \ . \end{aligned} \tag{14}$$

□

It is however possible to reduce the amount of penalty put on infeasible points while keeping the exact penalty property, as shown in the penalty f_1:

$$f_1(x;\lambda^*,x^f) \ = \ f(x) + H(g(x))[\lambda^*(g(x) - g(x^f)) + f(\widehat{x^*}) - f(x^f)] \ , \tag{15}$$

where $\widehat{x^*}$ is the known feasible point with lowest f.
Proof: The penalization is indeed reduced because $f(\widehat{x^*}) - f(x^f) \ \leq \ 0$, by definition of $\widehat{x^*}$. The penalization is exact because for all $x \ / \ g(x) \ > \ 0$,

$$\begin{aligned} f_1(x;\lambda^*,x^f) \ &= \ f(x) + \lambda^* g(x) - \lambda^* g(x^f) + f(\widehat{x^*}) - f(x^f) & (16) \\ &\geq \ f(x^f) + \lambda^* g(x^f) - \lambda^* g(x^f) + f(\widehat{x^*}) - f(x^f) = f(\widehat{x^*}) & (17) \\ &\geq \ f(x^*) \ = \ f_1(x^*;\lambda^*,x^f) \ . & (18) \end{aligned}$$

□

Finally, a class of exact penalty functions is introduced.

Proposition 1 (A class of exact penalty functions) *For f and g bounded, let $f_p(x;\lambda^+,x^f)$ be defined as,*

$$f_p(x;\lambda^+,x^f) \ = \ f(x) + H(g(x))[\lambda^+ g(x) - \lambda^+ g(x^f) - f(x^f) + f(\widehat{x^*}) + \epsilon] \ , \tag{19}$$

where,

$$\begin{aligned} &H(y) \ = \ 0 \ \text{ if } y \ \leq \ 0 \ , \ H(y) \ = \ 1 \ \text{ otherwise}, \\ &\lambda^+ \ \geq \ \lambda^* \ , \ x^f \in X_{\lambda^*} \ / \ g(x^f) \ \leq \ 0 \ , \ \epsilon \ > \ 0 \ , \\ &\widehat{x^*} \text{ is the known feasible point with lowest } f \ . \end{aligned}$$

$f_p(x;\lambda^+,x^f)$ has an absolute minimum at $x^* \in X^*$.

Proof: For all $x \ / \ g(x) \ > \ 0$, one shows that $f_p(x;\lambda^+,x^f) \ > \ f_p(x^*;\lambda^+,x^f) = f(x^*)$. Since $\lambda^+ \ \geq \ \lambda^*$,

$$\begin{aligned} f_p(x;\lambda^+,x^f) \ &= \ f(x) + \lambda^+ g(x) - \lambda^+ g(x^f) - f(x^f) + f(\widehat{x^*}) + \epsilon \\ &\geq f(x) + \lambda^* g(x) - \lambda^* g(x^f) - f(x^f) + f(\widehat{x^*}) + \epsilon \ . \end{aligned} \tag{20}$$

Using $f(x^f) + \lambda^* g(x^f) \ \leq \ f(x) + \lambda^* g(x)$, one obtains,

$$f_p(x; \lambda^+, x^f) \ \geq \ f(\widehat{x^*}) + \epsilon \ > \ f(x^*) \ . \tag{21}$$

□

It is now possible to solve (P) by first applying the dual evolutionary algorithm of Fig. 2, then approximating λ^* and x^f by $\lambda_{k_{max}}$ and $x^f_{k_{max}}$, respectively, and running a final evolutionary search on $f_p(x; \lambda_{k_{max}}, x^f_{k_{max}})$. No parameter is arbitrarily set in f_p. ϵ is an arbitrarily small positive scalar meant to ensure a strict inequality in Equation (21). It prevents x^i from being a minimum of f_p. If ϵ is neglected, $f_p(x; \lambda^*, x^f)$ is an extended Lagrangian according to Rockafellar's definition ([26]). When the problem has a saddle point, $x^f = x^*$, $\lambda^* g(x^*) = 0$, therefore the penalty function $f_p(x; \lambda^+, x^f)$ degenerates to a Lagrangian $L(x, \lambda^+)$ (neglecting ϵ).

Proposition 2 (A minimal penalty function) *Among exact penalty functions, L_p, based on the addition of a step, p, to a Lagrangian,*

$$L_p(x, \lambda^*) \ = \ f(x) + \lambda^* g(x) + p \ , \tag{22}$$

$f_p(x; \lambda^*, x^f)$ uses the smallest amount of penalty.

Proof: The smallest amount of penalty is such that,

$$\begin{aligned} & f(x^*) \ = \ L_p(x^i, \lambda^*) && (23) \\ \Leftrightarrow \ & f(x^*) \ = \ f(x^i) + \lambda^* g(x^i) + p && (24) \\ & (\ f(x^f) + \lambda^* g(x^f) \ = \ f(x^i) + \lambda^* g(x^i)\ ) && (25) \\ \Leftrightarrow \ & p \ = \ f(x^*) - f(x^f) - \lambda^* g(x^f) && (26) \end{aligned}$$

$f(\widehat{x^*})$ is the closest known upper bound on $f(x^*)$, therefore the smallest achievable value of p is $f(\widehat{x^*}) - f(x^f) - \lambda^* g(x^f)$, which is precisely the amount of penalty used in $f_p(x; \lambda^*, x^f)$. □

Proposition 1 explains how a constrained evolutionary optimization using $f_p(x; \lambda^*, x^f)$ as penalty function converges to an optimum. In addition, because it is a minimal penalty strategy (Proposition 2), fast convergence is favored.

4.4 An evolutionary optimizer based on f_p

In the preceding, it has been seen that the extended Lagrangian $f_p(x; \lambda^*, x^f)$ is an exact and minimal penalty function. An implementation of an evolutionary optimizer handling constraints through f_p is outlined in Fig. 4. The process starts with a dual problem resolution using the algorithm of Fig. 2. At a numerical cost of k_{max} evolutionary searches, $\lambda_{k_{max}}$ and $x^f_{k_{max}}$ approximate the parameters of the extended Lagrangian, λ^* and x^f. A final primal evolutionary search on $f_p(x; \lambda_{k_{max}}, x^f_{k_{max}})$ yields an approximate optimum x^*. Note that in our implementation, the best feasible and infeasible solutions of the dual problem, $x^f_{k_{max}}$ and $x^i_{k_{max}}$, are included in the initial population of the final primal search. Such practice speeds up the search because $x^f_{k_{max}}$ and $x^i_{k_{max}}$ typically surround x^*.

5 Numerical tests

Results on 3 test problems are presented. Each of them averages 50 independent runs. Usual evolutionary parameters are given in Appendix A. Other parameters of the algorithm are $\lambda_0 = 20$, $k_{max} = 20$, $n_f = n_i = 20$.

The first test, stated in Fig. 3, has one variable and one constraint. It does not have a saddle point. The feasible solution of the dual problem, $x^f = 1.058$, is far from the optimum, $x^* = 4.5$. Convergence of the algorithm is illustrated in Fig. 5. The left hand side plot shows that λ^* is found in practically all cases after $k = 3$ primal dual iterations, i.e., $3 \times 10000 = 30000$ analyses. The right hand side of the Figure describes convergence in the x space using the smallest distance between x^* and one of the active points, x_k^i or x_k^f. After 30000 analyses, this distance is stabilized around 0.08 because x_k^i converges to x^i and

$$\min_{x \in X_k^i \bigcup X_k^f} ||x^* - x|| \longrightarrow ||x^* - x^i|| = ||4.5 - 4.58|| = 0.08 . \qquad (27)$$

During the last evolutionary search which minimizes the penalized objective function, the best feasible point in terms of f_p tends to x^*, therefore the curve drops to 0.

The second test (from 2) has two variables and two constraints, that are reduced to one constraint through formulation. It is formulated as ([18]),

$$\left\{ \begin{array}{l} \min_{x_1,x_2 \in [0.001,20]} -\frac{\sin((2\pi x_1)^3)\sin(2\pi x_2)}{x_1^3(x_1+x_2)} , \\ \text{such that } g(x_1,x_2) = \max(g_1(x_1,x_2), g_2(x_1,x_2)) \leq 0 , \\ \ g_1(x_1,x_2) = x_1^2 - x_2 + 1 , \\ \ g_2(x_1,x_2) = 1 - x_1 + (x_2-4)^2 . \end{array} \right. \qquad (28)$$

Solutions of the primal and dual problems are :

$$x^* = \begin{pmatrix} 1.228 \\ 4.245 \end{pmatrix} , \lambda^* = 87.348 , x^f = \begin{pmatrix} 1.604 \\ 4.155 \end{pmatrix} , x^i = \begin{pmatrix} 0.001 \\ 0.006 \end{pmatrix} .$$

Test 2 is known as a problem where finding the right penalty is difficult ([18]). This can be understood by looking at a plot of the objective function on Fig. 6. There is a strong infeasible attractor (which, logically, is x^i) with a very low objective function, $f \approx -2000$, near the origin. The optimum is far from that point and has a much higher objective function, $f \approx -0.1$. It can only be seen on the right side zoom of Fig. 6 (where infeasible points have been removed). Feasible local optima are also visible on that plot. Since the infeasible attractor has such a low objective function, a large penalty is required. However, too large a penalty make the search likely to converge to a local feasible optimum. Test 2 has a large "duality gap", $f(x^*) - L(x^f, \lambda^*)$, and a large λ^*. The left side of Fig. 7 shows that the dual evolutionary optimizer robustly locates λ^* within 1% accuracy in about 50000 analyses. The right plot of Fig. 7 illustrates the convergence of the method to x^f during the first primal dual iterations, and the robust determination of x^* as the Lagrangian is replaced by the penalized function f_p.

The third and last test problem (from [7]) has 7 variables, 4 constraints,

$$\begin{cases} \min_{-20 \leq x_i \leq 20} \ (x_1 - 10.)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + \\ \qquad\qquad 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \ , \\ \text{such that,} \\ -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \ \leq \ 0 \ , \\ -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \ \leq \ 0, \\ -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \ \leq \ 0, \\ 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \ \leq \ 0 \ . \end{cases} \tag{29}$$

Two of the constraints (the first and the last) are active at the optimum. $x^* = (2.330499,\ 1.951372,\ -0.4775414,\ 4.365726,\ -0.624487,\ 1.038131,\ 1.594227)^T$,$\lambda^* = 1.493$. Although proof of existence of a saddle point is difficult to establish formally for non-convex problems, Test 3 seems to have a saddle point because the numerically determined x^f and x^i are close to x^*. Constraints are handled by the max scheme of (2). Convergence of the method on Test 3 is depicted in Fig. 8. λ^* is found faster than x^*, which is due to the higher dimensionality of the x space (7 variables). In all cases, the method is robust.

6 Concluding remarks

A general method for handling inequality constraints in evolutionary optimization has been proposed. It is an adaptive penalty strategy based on duality. Beyond the ability, shared by all evolutionary methods, to tackle non-convex optimization problems, this approach has the following advantages: it does not require any penalty parameter to be tuned, it converges to a solution of the problem, the amount of penalty put on infeasible points is minimal, and it yields optimal Lagrange multipliers as a by-product of the search. Lagrange multipliers are important because they describe the sensitivity of the objective function at the solution to a change in constraints.

In the current implementation of dual evolutionary optimization, emphasis has been put on clarity rather than on search speed (the minimal penalty property excepted, but it only applies to the last search on f_p). For this reason, dual and primal problems have been treated successively. As a result, the method is not cost effective. Solving the dual problem typically involves 5 additional evolutionary optimizations (cf. numerical tests). It is not necessary to carry out complete evolutionary search at the beginning of the primal / dual iterations. Futur work should define strategies to stop the first primal searches earlier.

References

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford Univ. Press, New York, 1996.

[2] J. C. Bean and A. B. Hadj-Alouane, *A Dual Genetic Algorithm for Bounded Integer Programs*, Technical Report TR 92-53, Dept. of Industrial and Operations Eng., The Univ. of Michigan, 1992.

[3] G.B. Dantzig, *General Convex Objective Forms*, The Rand Corporation, report P. 1664, 1959.

[4] G.B. Dantzig and P. Wolfe, "The decomposition algorithm for linear programming", *Econometrica*, vol. 29, no. 4, pp. 767–778, 1961.

[5] L. Davis, *Genetic Algorithms and Simulated Annealing*, Pitman, London, 1987.

[6] A. B. Hadj-Alouane and J. C. Bean, *A Genetic Algorithm for the Multiple-Choice Integer Program*, Technical Report TR 92-50, Dept. of Industrial and Operations Eng., The University of Michigan, 1992.

[7] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Springer-Verlag, Lecture Notes in Economics and Mathematical Systems, vol. 187, 1981.

[8] A. Homaifar, S. H.-Y. Lai, and X. Qi, "Constrained optimization via genetic algorithms", *Simulation*, vol. 62, no. 4, pp. 242 – 254, 1994.

[9] S. Howe, "New conditions for exactness of a simple penalty function", *SIAM Journal of Control*, vol. 11, no. 2, pp. 378–381, 1973.

[10] J. Joines and C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained problems with GAs" *Proc. of the First IEEE International Conf. on Evolutionary Computation*, Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, Ed. IEEE Press, 1994, pp. 579–584.

[11] J.-H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization", *IEEE Trans. on Evolutionary Computation*, pp. 129–140, July 1997.

[12] H.P. Kuenzi, H.G. Tzschach, and C.A. Zehnder, *Numerical Methods of Mathematical Optimization*, New York, Academic Press, 1971.

[13] R. Le Riche, C. Knopf-Lenoir, and R.T. Haftka, "A segregated genetic algorithm for constrained structural optimization", *Genetic Algorithms: Proc. of the Sixth International Conference on* (ICGA95), L.Eschelman, Ed. Morgan Kaufman, San Francisco, CA, 1995, pp. 558–565.

[14] H. A. Le Thi and P. Dinh Tao, "Exact penalty in D.C. programming", submitted to *Journal of Convex Analysis*, 1997.

[15] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, USA, 1984.

[16] Z. Michalewicz and N. Attia, "Evolutionary optimization of constrained problems", *Proc. of the 3rd Annual Conf. on Evolutionary Programming*, World Scientific, 1994, pp. 98–108.

[17] Z. Michalewicz, D. Dasgupta, R. Le Riche, and M. Schoenauer, "Evolutionary algorithms for constrained engineering problems", *Computers & Industrial Engineering Journal*, vol. 30, no. 4, pp. 851–869, 1996.

[18] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization", *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1997.

[19] M. Minoux, *Programmation Mathématique, Théorie et Algorithmes*, (in French), vol. 1 and 2, Dunod, Paris, 1983.

[20] D. Orvosh, and L. Davis, "Shall we repair? genetic algorithms, combinatorial optimization, and feasibility constraints", *Proc. of the Fifth International Conference on Genetic Algorithms*, Univ. of Illinois at Urbana-Champaign, Morgan Kaufmann, July 17-21, 1993, pp. 650.

[21] J. Paredis, "Exploiting constraints as background knowledge for genetic algorithms: a case-study for scheduling", *Parallel Problem Solving from Nature*, vol. 2, R. Männer and B. Manderick, Ed. Elsevier Science, 1992, pp. 229–238.

[22] J. Paredis, "Coevolutionary constraint satisfaction", *Proc. of the Third Conference on Parallel Problem Solving from Nature (PPSN 94)*, Lecture Notes in Computer Science, vol. 866, Y. Davidor, H.-P. Schwefel, and R. Manner, Eds. Springer Verlag, 1994, pp. 46–55.

[23] D. Powell and M.M. Skolnick, "Using genetic algorithms in engineering design optimization with non-linear constraints", *Proc. of the Fourth International Conference on Genetic Algorithms*, San Mateo, CA, Morgan Kaufmann, 1991, pp. 424–431.

[24] R. G. Reynolds, Z. Michalewicz, and M. J. Cavaretta, "Using cultural algorithms for constraints handling in GENOCOP", *Proc. of the 4th Annual Conf. on Evolutionary Programming*, J. R. McDownnell, R. G. Reynolds and D. B. Fogel, Eds. MIT Press, 1995, pp. 298–305.

[25] J.T. Richardson, M.R. Palmer, G. Liepins, and M. Hilliard, "Some guidelines for genetic algorithms with penalty functions", *Proc. of the Third International Conference on Genetic Algorithms*, George Mason Univ., Morgan Kaufmann, June 4-7, 1989, pp. 191–197.

[26] R.T. Rockafellar, "Lagrange multipliers in optimization", *Nonlinear Programming, Proc. SIAM-AMS*, vol. 9, R.W. Cottle and C.E. Lemke, Eds New York, 1976, pp. 145–168.

[27] M. Schoenauer and Z. Michalewicz, "Evolutionary computation at the edge of feasibility", *Proc. of the 4th Conference on Parallel Problems Solving from Nature*, W. Ebeling and H.-M. Voigt, Eds. Springer Verlag, 1996.

[28] M. Schoenauer and Z. Michalewicz, "Boundary operators for constrained parameter optimization problems", *Proc. of the 7th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, July 1997, pp. 322–329.

[29] A.E. Smith and D.M. Tate, "Genetic optimization using a penalty function", *Proc. of the Fourth International Conference on Genetic Algorithms*, San Mateo, CA, Morgan Kaufmann, 1991, pp. 499–505.

[30] M.-J. Tahk and B.-C. Sun, "Co-evolutionary augmented Lagrangian methods for constrained optimization", submitted for publication in *IEEE Trans. on Evolutionary Computation*, Feb. 1999.

[31] J. Zowe, "Nondifferentiable Optimization", *Computational Mathematical Programming*, K. Schittkowski, Ed. Springer, Berlin, Germany, pp. 323–359, 1985.

A Evolutionary Algorithm

The evolutionary algorithm used is a steady state algorithm with continuous mutation and crossover ([1]). The flow chart of the method is:

> *Create at random a set of npop points, evaluate them,* $n = npop$.
> *While* $n < l$ *do,*
>
> - *select 2 points from the current set of points by tournaments of size 2,*
> - *cross them with probability pc* $\rightarrow$ *a new point,*
> - *mutate the new point with probability pm per variable,*
> - *evaluate the new point,* $n = n + 1$,
> - *replace the worst point of the set (in the sense of its penalized objective function) by the new point,*
>
> *end do.*

l is the maximal number of point evaluations. The continuous crossover is defined as,

$$x_{new} = \alpha x_1 + (1 - \alpha) x_2 \ , \tag{30}$$

where $\alpha \in [0, 1]$. The continuous mutation is implemented as,

$$x_{new} = x + \delta x , \tag{31}$$

where δx follows a Gaussian distribution $N(0, \sigma)$ and $\sigma^2 = (x_{max} - x_{min})^2/16$. Parameters used in the numerical experiments are $pc = 0.7$, $pm = 0.4$, $x_{min} = -20$, $x_{max} = 20$. Population sizes of $npop = 50$ and $npop = 200$ are taken for the dual and the last (primal) evolutionary searches, respectively. Similarly, the search length, l, is increased from 10000 to 20000 analyses in the last evolutionary search.

B Approximate Dual Problem Optimization using the Simplex Method

The approximate dual optimization problem (AD) can be formulated as the linear programming problem of Equation (9). The numerical implementation, based on the simplex algorithm proposed in [12], requires changing the formulation (9) in order to obtain a restricted normal form of the linear programming problem.

In restricted normal form, all variables and right-hand sides of constraints must be positive. Among the variables, the Lagrange multipliers λ are, by definition, positive. The variable w are made positive by the change of variable $w \longrightarrow w' - K$, where K is a positive constant. By construction of the approximate dual problem (cf. Fig. 1), it is seen that the value

$$K = - \min_{x_i \in T}(0, f(x_i)) \tag{32}$$

is sufficient to enforce $w' > 0$ when solving (AD). The restricted normal form of the approximate dual problem is,

$$\begin{cases} \max_{w' \geq 0, \lambda \geq 0} \ w' \ , \\ \text{such that } z_i = \lambda_{i,max} - \lambda_i - y_i \ , \ i = 1, sizeof(g) \ , \\ z_i = f(x_j) + K - w' + \lambda g(x_j) - y_i \ , \forall x_j \in T \mid f(x_j) + K \geq 0 \ , \\ z_i = -f(x_j) - K + w' - \lambda g(x_j) + y_i \ , \forall x_j \in T \mid f(x_j) + K < 0 \ , \end{cases} \tag{33}$$

where $y_i, i = 1, sizeof(T)+1$ are positive slack variables and $z_i, i = 1, sizeof(T)+ sizeof(g)$ are artificial variables. Equation (33) is given on input to the simplex algorithm under the form of a tableau. On output, all z_i's are equal to 0 if the constraints are consistent, and slack variables y_i equal to 0 characterize active constraints. Information on active constraints is passed to the calling program so that points x_i of T involved in active constraints are not removed from T.

C Convergence of the dual evolutionary optimizer

We assume in this section that the primal optimization problem (P) is exactly solved (until a $x_\lambda \in X_\lambda$ is found). In this case, each primal search provides a tangent to the dual function at λ. The difference between exact primal resolution and approximate primal resolution was illustrated in Fig. 1 where x_0 was the only point corresponding to an exact primal problem resolution.

Proposition 3 (Rate of Convergence of the Dual Stochastic Algorithm) *If ϕ'' exists and is bounded, the algorithm converges to λ^*, solution of the dual problem, at a rate at least linear.*

Proof: We accept without proof that $\lambda_k \le \lambda_{k+2} \le \lambda_{k+1}$ and $\phi'(\lambda_k) \le 0$ and $\phi'(\lambda_{k+1}) \ge 0$. This can be seen by construction of the λ_k's as intersections of tangents to a concave function. Existence of ϕ' stems from the existence of ϕ''. By definition of the dual part of the algorithm, λ_{k+2} is at the intersection of the two tangents to the dual function ϕ at λ_k and λ_{k+1},

$$\phi(\lambda_k) + \phi'(\lambda_k)(\lambda_{k+2} - \lambda_k) = \phi(\lambda_{k+1}) + \phi'(\lambda_{k+1})(\lambda_{k+2} - \lambda_{k+1}) \ . \tag{34}$$

There exists $c_0 \in \,]\lambda_k, \lambda_{k+2}[$ and $c_1 \in \,]\lambda_{k+2}, \lambda_{k+1}[$ such that

$$\phi(\lambda_{k+2}) = \phi(\lambda_k) + \phi'(\lambda_k)(\lambda_{k+2} - \lambda_k) + \frac{1}{2}\phi''(c_0)(\lambda_{k+2} - \lambda_k)^2 \ , \tag{35}$$

$$\phi(\lambda_{k+2}) = \phi(\lambda_{k+1}) + \phi'(\lambda_{k+1})(\lambda_{k+2} - \lambda_{k+1}) + \frac{1}{2}\phi''(c_1)(\lambda_{k+2} - \lambda_{k+1})^2 \tag{36}$$

Substituting (35) and (36) in (34), one obtains,

$$\frac{1}{2}\phi''(c_0)(\lambda_{k+2} - \lambda_k)^2 = \frac{1}{2}\phi''(c_1)(\lambda_{k+2} - \lambda_{k+1})^2 \ , \tag{37}$$

or,

$$\lambda_{k+1} - \lambda_{k+2} = \frac{1}{1 + \sqrt{\frac{\phi''(c_1)}{\phi''(c_0)}}}(\lambda_{k+1} - \lambda_k) \ . \tag{38}$$

ϕ'' is bounded and negative (from concavity of ϕ),

$$A \le \phi''(\lambda) \le a < 0, \ \forall \lambda \ge 0 \ , \tag{39}$$

which yields a bound on $\lambda_{k+1} - \lambda_{k+2}$,

$$|\lambda_{k+2} - \lambda_{k+1}| \le \frac{1}{1 + \sqrt{\frac{a}{A}}}|\lambda_{k+1} - \lambda_k| \tag{40}$$

Since $\frac{1}{1+\sqrt{\frac{a}{A}}} \le \alpha < 1$, $|\lambda_{k+1} - \lambda_k|$ tends to 0, at least linearly. Moreover, it is known that $\lambda_k \le \lambda^* \le \lambda_{k+1}$ because $\phi'(\lambda)$ is decreasing and $\phi'(\lambda_k) \le 0$, $\phi'(\lambda_{k+1}) \ge 0$, and $\phi'(\lambda^*) = 0$. Therefore, λ_k tends to λ^*. □

D Existence of feasible primal solutions to (D) and (AD)

Existence of feasible solutions of the dual and approximate dual problems, X^f and $X_k^f \neq \emptyset$, is proved here. It justifies using $x^f \in X^f$ in the penalty function (19).

D.1 Dual problem

Proposition 4 (Existence of feasible solutions to (D)**)** *For f and g bounded with at least one feasible point in S, there is at least one point, $x^f \in X_{\lambda^*}$ such that $g(x^f) \ \leq \ 0$.*

Proof: Non-smooth functions have 0 in the sub-differential at a maximum ([19], Theorem 8, p. 17). More precisely, for the dual function ϕ, one writes: the dual function ϕ being concave and $\{\lambda \geq 0 \mid \phi(\lambda) > -\infty\} \ \neq \ \emptyset$, λ^* maximizes ϕ if and only if $\ 0 \ \in \ \partial\phi(\lambda^*)$, where $\partial\phi(\lambda^*)$ is the sub-differential of ϕ at λ^*. Another fundamental result is that the directional derivative of a non-differentiable function exist, and is equal to an extremum of the sub-gradients γ (in 1 dimension) ([19], Theorem 9, p.18). Applied to ϕ, it means, $\forall \lambda^* \geq 0 \mid \phi(\lambda^*) > -\infty$,

$$\lim_{h \to 0^+} \frac{\phi(\lambda^* + h) - \phi(\lambda^*)}{h} \ = \ \min_{\gamma \in \partial\phi(\lambda^*)} \gamma \ . \tag{41}$$

Because $\phi(\lambda)$ is concave and $g(x_\lambda)$, $x_\lambda \in X_\lambda$, is a sub-gradient of ϕ,

$$\lim_{h \to 0^+} g(x_{\lambda^*+h}) \ \leq \ lim_{h \to 0^+} \frac{\phi(\lambda^* + h) - \phi(\lambda^*)}{h} \ . \tag{42}$$

(41) and (42) induce,

$$\lim_{h \to 0^+} g(x_{\lambda^*+h}) \ \leq \ min_{\gamma \in \partial\phi(\lambda^*)} \gamma \ , \tag{43}$$

or,

$$g(x_{\lambda^*}) \ \leq \ min_{\gamma \in \partial\phi(\lambda^*)} \gamma \ , \tag{44}$$

For the $x_{\lambda^*} \in X_{\lambda^*}$ associated to the left hand side limit. From Property 2, $g(x_{\lambda^*})$ is one of the γ's. Therefore the lower bound of $\partial\phi(\lambda^*)$ is hit by at least one of the points of X_{λ^*}, called x^f. Since $\ 0 \ \in \ \partial\phi(\lambda^*)$,

$$g(x^f) \ \leq \ 0 \ , \tag{45}$$

which proves the Proposition. □

D.2 Approximate dual problem

Proposition 5 *There is at least a feasible point associated to the active constraints of the approximate dual problem (9).*

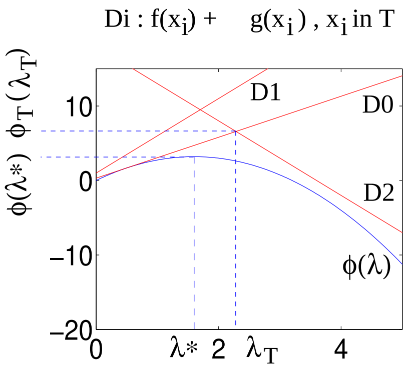
Proof: The approximate dual problem is convex when written in the form,

$$(AD) \qquad \begin{cases} \min_{w,\lambda \geq 0} \ -w \ , \\ \text{such that } w - f(x_i) \ - \ \lambda g(x_i) \ \leq \ 0 \ , \forall x_i \in T \ , \\ \lambda - \lambda_{max} \ < \ 0. \end{cases} \tag{46}$$

The Kuhn and Tucker optimality conditions are necessary and sufficient. Let μ_i denote the positive Lagrange multipliers associated to the constraints of (46). The Kuhn and Tucker condition associated to the λ derivative is,

$$\sum_{i=1}^{card(T)} \mu_i g(x_i) \; = \; 0. \tag{47}$$

Let I denote the set of indices of the active constraints, $\mu_i \; > \; 0 \; , \; \forall i \in I$. Equation (47) stands either if $g(x_i) \; = \; 0 \; , \; \forall i \in I$, or if there is at least one $g(x_i) \; < \; 0$. $\square$



$$(P) \quad \left\{ \begin{array}{rcl} \min_x f(x) & = & x_1^2 + x_2^2 \ , \\ \text{s.t. } 2x_1 + x_2 + 4 & \leq & 0 \ . \end{array} \right.$$

$$(D) \quad \phi(\lambda) \;=\; \tfrac{-5\lambda^2}{4} + 4\lambda \ .$$

$$x_\lambda \;=\; \begin{pmatrix} -\lambda \\ -\lambda/2 \end{pmatrix},$$

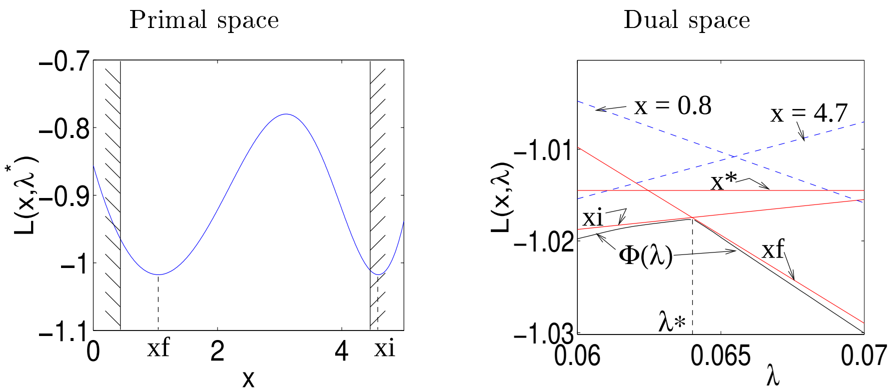
$$\lambda^* \;=\; \tfrac{8}{5}, \quad x^* \;=\; \begin{pmatrix} -8/5 \\ -4/5 \end{pmatrix},$$

$$T \;=\; \left\{ \begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} -3 \\ -3 \end{pmatrix} \right\}$$

Figure 1: Example of dual and approximate dual functions, problem with a saddle point.

1. $k = 0$, initialize λ_0.
2. Evolutionary (primal) search minimizing on x $L(x, \lambda_k)$. If $k > 0$, include X^k_{k-1} and X^i_{k-1} in the initial population.
3. Add the n_f and n_i best feasible and infeasible individuals according to $L(x, \lambda_k)$, plus the feasible individual with the lowest f to $T_k \longrightarrow T_{temp}$.
4. Solve exactly dual (AD) with T_{temp} according to formulation (9) $\longrightarrow \lambda_{temp}$, X^f_k , X^i_k.
5. $T_k = T_{temp}$, $\lambda_k = \lambda_{temp}$, $k = k + 1$. If $k \geq k_{max}$, stop, else go to 2.

Figure 2: Dual evolutionary optimizer



$$(P) \quad \begin{cases} \min_{-20 \le x \le 20} f(x) & = & -\exp(-0.1x^2) - \exp(-0.5(x-5)^2) \ , \\ \text{s.t. } (x-0.5)(x-4.5) & \le & 0 \ . \end{cases}$$

$$X^* = \{4.5\} \ , \ \lambda^* = 0.064 \ , \ X_{\lambda^*} = \{1.058 \ , \ 4.58\}$$

Figure 3: Example of dual and approximate dual functions, problem without a saddle point.

1. `Run the dual evolutionary algorithm of Fig. 2`
   $\longrightarrow \quad \lambda_{k_{max}} \ , \ x^f_{k_{max}} \ , \ x^i_{k_{max}}$.
2. `Final evolutionary search minimizing on` x $f_p(x; \lambda_{k_{max}}, x^f_{k_{max}})$.
   $x^f_{k_{max}}$ `and` $x^i_{k_{max}}$ `are included in the initial population.`

Figure 4: Evolutionary optimization based on f_p.

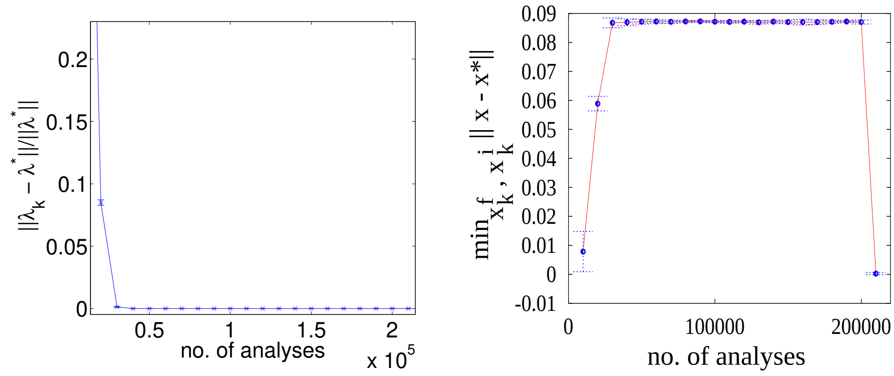


Figure 5: Average convergence ± standard deviation, 50 runs, Test 1.

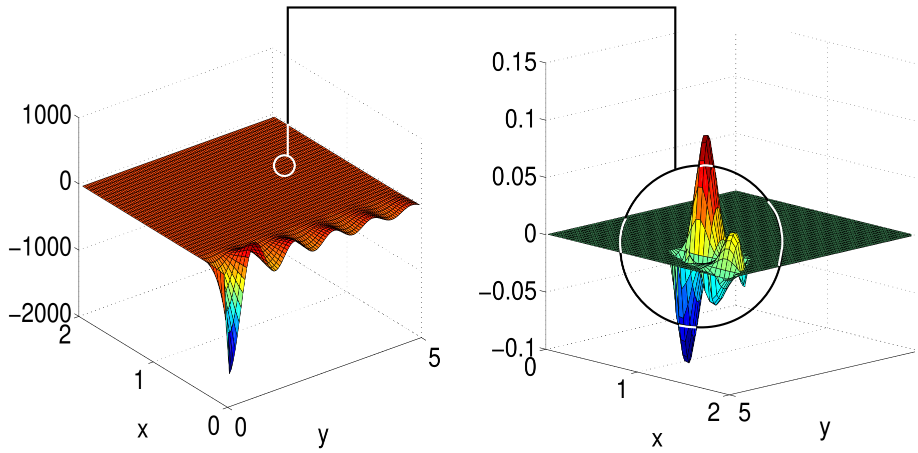


Figure 6: Plots of the objective function of Test 2. On the right, only feasible points are drawn and the view point is changed so that feasible local minima are visible.

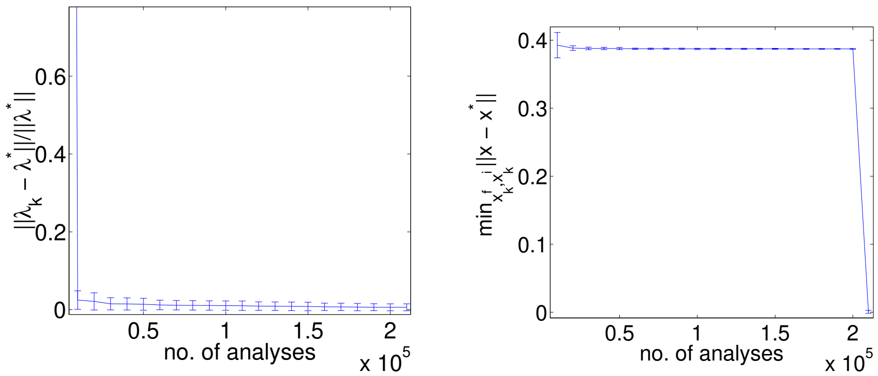


Figure 7: Average convergence ± standard deviation, 50 runs, Test 2.

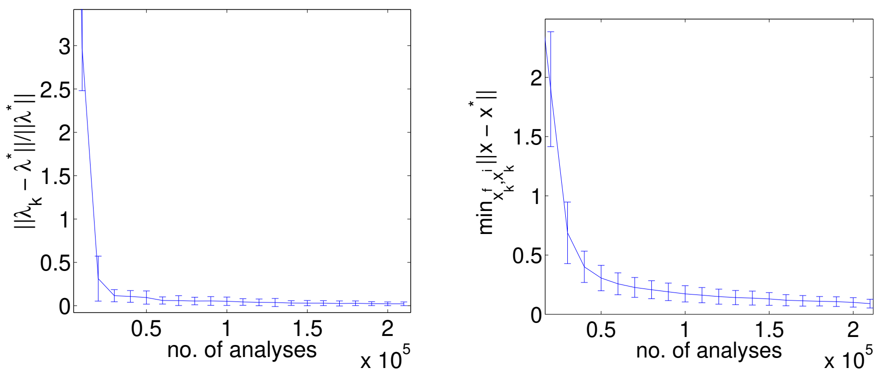


Figure 8: Average convergence ± standard deviation, 50 runs, Test 3.