Dual Evolutionary Optimization

Rodolphe Le Riche[1] and Frédéric Guyon[2]

[1] Lab. de Mécanique de Rouen, UMR 6138, Saint Etienne du R^{ay}, France
Rodolphe.Leriche@insa-rouen.fr
[2] Lab. de Bio-statistiques Bio-mathématiques, Univ. Paris 7, France
guyon@bach.urbb.jussieu.fr

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. An evolutionary algorithm for approximately solving dual optimization problems is first presented. Next, an efficient and exact penalty function without penalization parameter to be tuned is proposed. Numerical tests are provided for continuous variables and inequality constraints.

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.

Four types of methods for handling constraints exist: penalization of infeasible solutions([2], [3], [4], [5], [6], [7]), projection of infeasible solutions onto the feasible domain, co-evolution of populations which together solve the constrained optimization problem, and constraints representation building in the course of the search. These approaches are related and have been coupled, like co-evolution and penalty methods ([8] and [9]), or penalty and projection. A review on constraints handling in evolutionary optimization can be found in [10].

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]). Dynamic penalties are function of the number of points sampled while adaptive penalties ([6], [7],[3],[2]) vary with points evaluations. Mixed approaches exist, e.g. in [5].

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 ([11]). 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 ([12]).

Augmented Lagrangian functions and Lagrange multipliers updating have been applied to derive adaptive penalty functions in evolutionary algorithms. Bean and Hadj-Alouane ([6]) have proposed a penalty adaptation scheme which resembles Lagrange multipliers updating strategies. Kim and Myung ([2]) and Tahk and Sun ([9]) have used augmented Lagrangian penalty functions in evolutionary optimization, calculating Lagrange multipliers as a by-product of the search. In [9], 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 algorithm 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^*, x^* is any element of 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 $\lambda \ \geq \ 0$, 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, [11]).
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 of ϕ at λ.

The two above properties, proofs of which can be found in [11], 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 [6], [7], and [2],

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,

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

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. [13]). Let its solution be λ_T.

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 the 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 ([14]). 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.

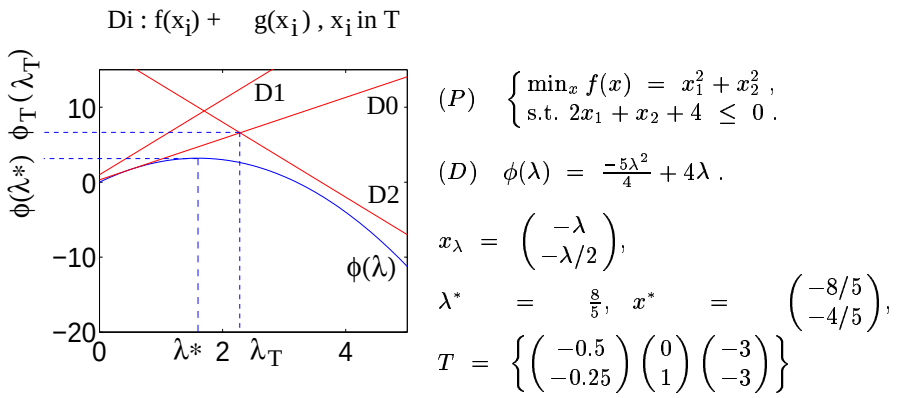


Fig. 1. Example of dual and approximate dual functions, problem with a saddle point.

– It can handle non-convex, discontinuous functions.

A flow chart of a dual evolutionary optimizer is given in Fig. 2. X_k^f and X_k^i are the sets of feasible and infeasible active points of (AD) at iteration k, respectively. The evolutionary algorithm used is a steady-state algorithm with continuous mutation and crossover, and tournament selection ([1]). Evolutionary searches are stopped as soon as an improvement on $L(x, \lambda_k)$ has been observed. This is an important implementation aspect as it saves much computational effort that would otherwise be spent minimizing the Lagrangian with λ_k far from λ^*. As a side effect, this stopping criterion increases the number of resolutions of (AD). The cost of solving (AD) is however negligible because no evaluation of f or g is performed and the simplex algorithm is efficient. In all the tests performed (cf. section 5), the CPU time spent in (AD) is less than a percent of the total CPU time for T_k sets of up to 10000 elements. Further details on the simplex implementation, existence of X_k^f, and convergence rate of the method can be found in [13]. Important outputs of the algorithm are $\lambda_{k_{final}}$ and $X^f_{k_{final}}$. $\lambda_{k_{final}}$ is an estimate of λ^*. $X^f_{k_{final}}$ 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 $g(x_{\lambda_k})$, $x_{\lambda_k} \in X_{\lambda_k}$.

1. $k = 0$, initialize λ_0, $\phi_0 = DBL\_MAX$.
2. Evolutionary (primal) search minimizing on x $L(x, \lambda_k)$. If $k > 0$, include X^f_{k-1} and X^i_{k-1} in the initial population. Stop when a point x' has been found such that $L(x', \lambda_k) < \phi_k$.
3. Add x' plus other n_f and n_i best feasible and infeasible individuals according to $L(x, \lambda_k)$ to $T_k \longrightarrow T_{temp}$.
4. Simplex exact resolution of dual (AD) with T_{temp} according to formulation (9) $\longrightarrow$ λ_{temp} , X^f_k , X^i_k.
5. $k = k+1$, $T_k = T_{temp}$, $\lambda_k = \lambda_{temp}$, $\phi_k = f(x^f_k) + \lambda_k g(x^f_k)$.
   If cumulated number of analyses > N_{max}, $k_{final} = k$, stop.
   Else go to 2.

Fig. 2. Dual evolutionary optimizer

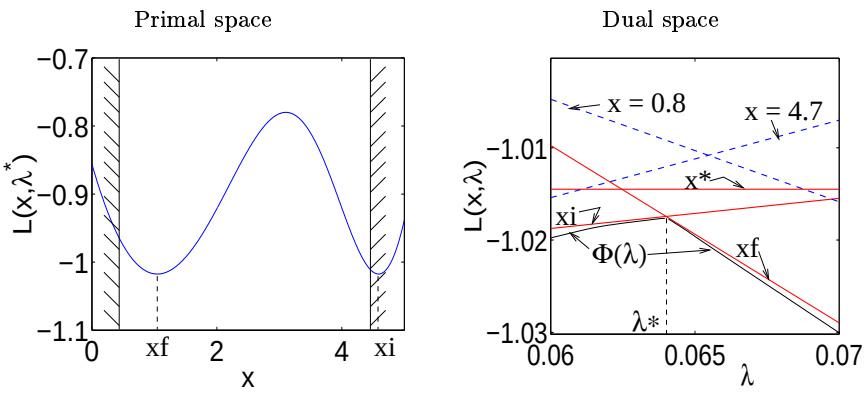
4 A minimal, exact, penalty function

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 because in this case, $X^* = \{x^*\}$, x^* unique, $X_{\lambda^*} = \{x_{\lambda^*}\}$, x_{λ^*} unique, and $x^* = x_{\lambda^*}$ ([11]). In other terms, the dual and the primal problems are equivalent. The dual evolutionary algorithm provides $X^f_{k_{final}}$ (cf. Fig. 2), which includes an estimate of x^*. To sum up, the Lagrangian is a valid penalty function for problems having a saddle-point.

For problems without saddle point, solving (D) does not directly provide a solution to (P), $X_{\lambda^*} \neq X^*$. This can be seen on the example of Fig. 3 where $X_{\lambda^*} = \{1.058\ ,\ 4.58\}, x^f = 1.058$ and $X^* = \{4.5\}$. Problems without a saddle point require using another penalty function. Nevertheless, as will soon be seen, solving (D) still generates information for properly penalizing the constraints : X_{λ^*} contains at least one feasible element denoted x^f, $g(x^f) \leq 0$ (see [13]).

Let F_p 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 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 [15], Richardson et al. [4], Smith and Tate [5]). 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 purposes, 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*



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

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

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

of penalty, r, is defined as,

$$r \;=\; F_p(x^i) - f(x^i)\;, \tag{10}$$

where F_p is any penalized objective function.

A class of Lagrangian based exact penalty function is now introduced.

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

$$f_p(x;\lambda^+,x^f) \;=\; f(x) + H(g(x))[\lambda^+ g(x) - \lambda^+ g(x^f) - f(x^f) + f(\widehat{x^*}) + \epsilon]\;, \tag{11}$$

where,

$$H(y) \;=\; 0 \;\; \textit{if } y \;\le\; 0\;,\; H(y) \;=\; 1 \;\; \textit{otherwise},$$
$$\lambda^+ \;\ge\; \lambda^*\;,\; x^f \in X_{\lambda^*} \;/\; g(x^f) \;\le\; 0\;,\; \epsilon \;>\; 0\;,$$
$$\widehat{x^*} \;\textit{is the known feasible point with lowest } f\;.$$

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

Proof: The result is obvious if $g(x) \;\le\; 0$. For all $x \;/\; g(x) \;>\; 0$, one shows that $f_p(x;\lambda^+,x^f) \;>\; f_p(x^*;\lambda^+,x^f) = f(x^*)$. Since $\lambda^+ \;\ge\; \lambda^*$,

$$\begin{aligned} f_p(x;\lambda^+,x^f) \;&=\; f(x) + \lambda^+ g(x) - \lambda^+ g(x^f) - f(x^f) + f(\widehat{x^*}) + \epsilon \\ &\ge f(x) + \lambda^* g(x) - \lambda^* g(x^f) - f(x^f) + f(\widehat{x^*}) + \epsilon\;. \end{aligned} \tag{12}$$

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

$$f_p(x; \lambda^+, x^f) \geq f(\widehat{x^*}) + \epsilon > f(x^*) . \tag{13}$$

□

This class of penalty functions contains a minimal penalty function.

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

$$L_p(x, \lambda^*) = f(x) + \lambda^* g(x) + p , \tag{14}$$

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

The proof of Proposition 2 along with a more gentle introduction to f_p can be found in [13].

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), it promotes fast convergence. An evolutionary optimizer for general constrained optimization problems is described in Fig. 4. Finally, we emphasize that no parameter of the penalty function is arbitrarily set since λ^* and x^f have a precise definition in terms of (D). λ_0, N_{max}, n_f and n_i control the rate of convergence in the dual space. These parameters have little influence compared to penalty parameters.

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

Fig. 4. Evolutionary optimization based on f_p.

5 Numerical tests

Results on 4 test problems are presented. Each of them averages 50 independent runs. The mutation operator adds to the variables a Gaussian noise $N(0, \sigma)$, $\sigma^2 = (x_{max} - x_{min})^2/16$, $x_{min} = -20$, $x_{max} = 20$. The probabilities of crossover and mutation are $pc = 0.7$ and $pm = 0.4$, tournaments of size 2 select individuals. Population sizes, *npop*, and search lengths, N_{max}, are the same for the dual and primal evolutionary searches. The pairs $(npop, N_{max})$ are $(200, 10000)$, $(200, 10000)$, $(300, 100000)$ and $(300, 100000)$ for Tests 1 to 4, respectively. Other parameters of the algorithm are $\lambda_0 = 20$, $n_f = n_i = 20$.

5.1 Comparison of approaches

Different penalty approaches are compared to illustrate three claims. Firstly, the minimal penalty function promotes fast and reliable convergence as compared to arbitrarily tuned static penalty functions. Secondly, the dual evolutionary optimizer is a better strategy than an adaptive linear penalty. Thirdly, the dual evolutionary optimizer is not sensitive to its parameters setting. Those claims are checked using the hoop problem, which has an objective function composed of two linear functions and a narrow curved feasible domain :

$$\min_{x_1,x_2} f(x_1,x_2) \quad \text{such that} \quad \begin{cases} R^2-(x_1-A)^2-x_2^2 \;\leq\; 0\;, \\ (x_1-A)^2+x_2^2-(R+E)^2 \;\leq\; 0\;, \\ 0 \;\leq\; x_1 \;\leq\; A \quad , \quad 0 \;\leq\; x_2 \;\leq\; A\;, \end{cases} \tag{15}$$

where,

$$f(x_1,x_2) \;=\; \begin{cases} \alpha(x_1+x_2) & \text{if} \quad x_1+x_2-H \;\leq\; 0\;, \\ -(x_1+x_2)+(\alpha+1)H & \text{otherwise}\;, \end{cases} \tag{16}$$

and $A = 20$, $R = 18$, $E = 0.1$. The solution is $x^* = (1.9\ ,\ 0)^T$. H is a parameter that controls the size of the basin of attraction of a local optimum. If $H = 5$, there is a local optimum at $(20, 18.1)^T$. If $H = 40$, there is no local optimum.

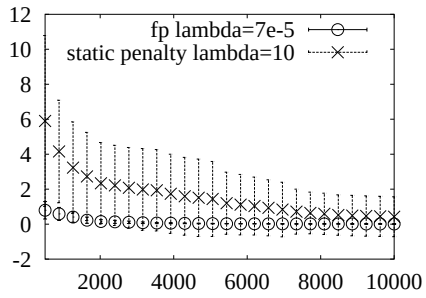


Fig. 5. $||\widehat{x^*} - x^*||$ vs. nb. of analyses. Comparison of two amounts of static penalties, hoop problem.

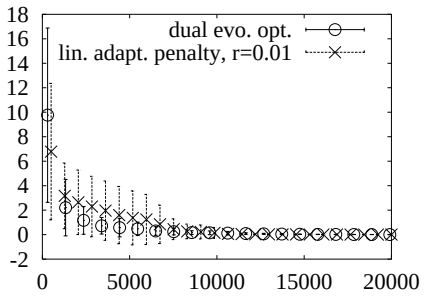


Fig. 6. $||\widehat{x^*} - x^*||$ vs. nb. of analyses. Comparison of two adaptive approaches for estimating the penalty, hoop problem.

Fig. 5 illustrates the effect of the (static) amount of penalty on the convergence to x^*. The minimal penalty function $f_p(x;\lambda^*,x^f)$ is compared to a linear (static, λ is fixed) penalty function,

$$F_p(x;\lambda) \;=\; f(x) + \lambda \max(0, g(x))\;, \tag{17}$$

where λ is arbitrarily set to 10. Note that no dual search is performed on this plot. It is seen that the static penalty is much slower than the minimal penalty

function. Another series of tests has been performed with $H = 5$. The evolutionary algorithm using a static penalty function, $\lambda = 10$, converges to the local (false) optimum 38 times out of 50, against 26 times when using the minimal penalty.

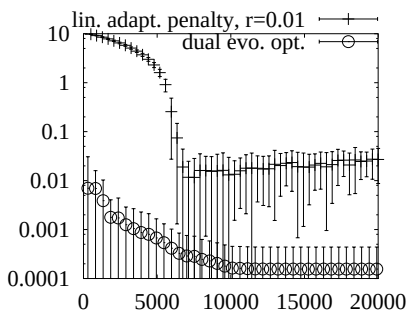


Fig. 7. $||\widehat{\lambda^*} - \lambda^*||$ vs. nb. of analyses. Comparison of two adaptive approaches for estimating the penalty, hoop problem.

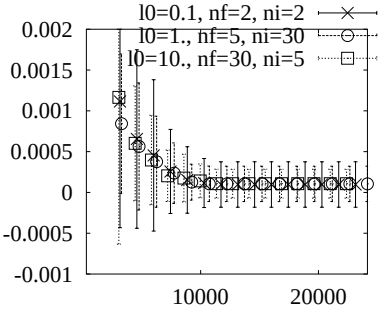


Fig. 8. $||\widehat{\lambda^*} - \lambda^*||$ vs. nb. of analyses. Convergence of the dual evolutionary optimizer for different settings of λ_0, n_f and n_i, hoop problem.

In Fig. 6 and 7, the dual evolutionary optimizer of Fig. 4 is compared to an adaptive linear penalty algorithm. This last algorithm minimizes, at each primal iteration, the linear penalty function $F_p(x;\lambda_k)$ of Equation (17) using an evolutionary optimizer. Let use temporarily denote by $\widehat{x}(\lambda_k)$ the solution estimate. Lagrange multipliers are then updated according to,

$$\lambda_{k+1} = \max(0, (\lambda_k + rg(\widehat{x}(\lambda_k)))) \; . \tag{18}$$

One sees in Fig. 6 that convergence to x^* is faster, between 1000 and 7000 analyses, with the dual evolutionary optimizer than with the linear adaptive penalty. In the space of Lagrange multipliers, convergence to λ^* is faster, more accurate and more stable with the dual evolutionary optimizer than with the linear adaptive penalty (cf. Fig. 7).

Fig. 8 shows how the dual evolutionary optimizer converges to λ^* when its parameters setting (λ_0, n_f and n_i) changes. The method does not appear to be sensitive to parameters changes. The only visible feature is a slightly higher variance when $n_f + n_i$ decreases.

5.2 Convergence on various test functions

The second test is the two humps function stated in Fig. 3. It 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, $x^* = 4.5$. The algorithm rapidly converges to λ^*, on the average after 5000 analyses (cf. Table 1). During

the dual iterations, the evolutionary optimizer converges either to x^i or to x^f (similarly for Tests 3 and 4 later). The final primal search using f_p robustly locates x^*. After 10000 analyses, the best search point according to f_p, x_b, is such that $||x_b - x^*|| \;=\; 4.6e-4 \;\pm\; 6.5e-4$.

The third test has two variables and two constraints. The constraints are reduced to one constraint by considering only the most critical. It is formulated as ([10]),

$$\left\{\begin{array}{l} \min_{x_1,x_2\in[0.001,20]} -\frac{\sin((2\pi x_1)^3)\sin(2\pi x_2)}{x_1^3(x_1+x_2)} \;, \\ \text{such that } g(x_1,x_2) \;=\; \max(g_1(x_1,x_2),g_2(x_1,x_2)) \;\leq\; 0 \;, \\ \quad g_1(x_1,x_2) \;=\; x_1^2 - x_2 + 1 \;, \\ \quad g_2(x_1,x_2) \;=\; 1 - x_1 + (x_2-4)^2 \;. \end{array}\right. \tag{19}$$

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 3 is known as a problem where finding the right penalty is difficult ([10]). This can be understood by looking at a plot of the objective function on Fig. 9. 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. 9 (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 3 has a large "duality gap", $f(x^*) - L(x^f,\lambda^*)$, and a large λ^*. Numerical experiments show that the dual evolutionary optimizer locates λ^* within 20% accuracy in 50000 analyses and 15% accuracy in 100000 analyses (cf. Table 1). Those results, which are substantially worse than those obtained on the other tests, illustrate how penalizing Test 3 is difficult. Nevertheless 15% accuracy in λ^* is sufficient for f_p to robustly guide the search around x^* in the final primal search, $||x_b - x^*|| \;=\; 2.e-4 \;\pm\; 8.e-4$.

The fourth and last test problem (from [10]) has 7 variables, 4 constraints,

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

Two of the constraints (the first and the last) are active at the optimum. $x^* \;=\; (2.330499,\;\; 1.951372,\;\; -0.4775414,\;\; 4.365726,\;\; -0.624487,\;\; 1.038131,$

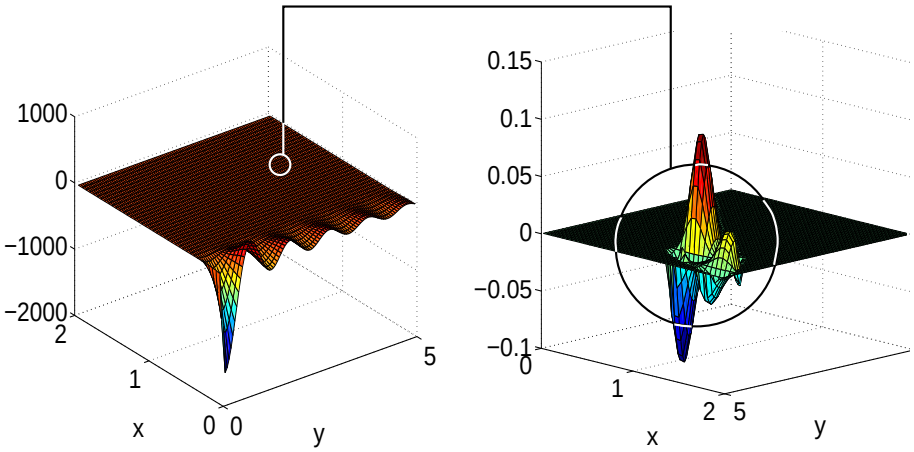


Fig. 9. Plots of the objective function of Test 3. On the right, only feasible points are drawn and the view point is changed so that feasible local minima are visible.

$1.594227)^T$,$\lambda^* = 1.493$. Although proof of existence of a saddle point is difficult to establish formally for non-convex problems, Test 4 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). Numerical experiments show that λ^* is found after 50000 analyses (cf. Table 1). At the end of the last primal search, $||x_b - x^*|| = 2.7e-1 \pm 1.5e-1$.

These numerical experiments lead us to consider that solving the dual before the primal problem about doubles the price of the search. Convergence to λ^* is achieved in 5000 analyses when $N_{max} = 10000$ analyses in Test 2, and in 50000 analyses when $N_{max} = 100000$ analyses in Test 4, but all $N_{max} = 100000$ analyses are necessary in Test 3. N_{max} is chosen as a typical search length with a static penalty function. Therefore, in this work, it is the length of the last primal search which minimizes f_p.

Table 1. Convergence to λ^* : average $|\lambda_k - \lambda^*|/\lambda^* \pm$ std. deviation, 50 runs.

No. analyses	500	5000	10000	50000	100000
Test 2	0.02 ± 0.02	2.e-5 ± 4.e-5	1.e-5 ± 2.e-5	—	—
Test 3	0.71 ± 0.15	0.41 ± 0.15	0.32 ± 0.15	0.20 ± 0.11	0.15 ± 0.07
Test 4	8.e3 ± 6.e3	122 ± 368	1.49 ± 1.63	0.28 ± 0.20	0.27 ± 0.12

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, 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, solving for the Lagrange multipliers doubles the computational cost of a search using a traditional static penalty function.

References

1. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford Univ. Press, New York (1996)
2. Kim, J.-H., Myung, H.: Evolutionary Programming Techniques for Constrained Optimization. IEEE Trans. on Evolutionary Computation. July (1997) 129–140
3. Powell, D., Skolnick, M.M.: Using Genetic Algorithms in Engineering Design Optimization with Non-linear Constraints. In: Proc. of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo CA (1991) 424–431
4. Richardson, J.T., Palmer, M.R., Liepins, G. Hilliard, M.: Some Guidelines for Genetic Algorithms with Penalty Functions. In: Proc. of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo CA George Mason Univ., June 4-7 (1989) 191–197
5. Smith, A.E., Tate, D.M.: Genetic Optimization using a Penalty Function. In: Proc. of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo CA (1991) 499–505
6. Bean, J.C., Hadj-Alouane, A.B.: A Dual Genetic Algorithm for Bounded Integer Programs. Technical Report TR 92-53. Dept. of Industrial and Operations Eng., The Univ. of Michigan (1992)
7. Hadj-Alouane, A.B., Bean, J.C.: 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)
8. Le Riche, R., Knopf-Lenoir, C., Haftka, R.T.: A Segregated Genetic Algorithm for Constrained Structural Optimization. In: Eschelman, L. (ed.): Proc. of the Sixth International Conference on Genetic Algorithms (ICGA95). Morgan Kaufman, San Francisco CA (1995) 558–565
9. Tahk, M.-J., Sun, B.-C.: Co-evolutionary Augmented Lagrangian Methods for Constrained Optimization. Submitted for publication in: IEEE Trans. on Evolutionary Computation. February (1999)
10. Michalewicz, Z., Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization. Evolutionary Computation. Vol. 4 **1** (1997) 1–32
11. Minoux, M.: Programmation Mathématique, Théorie et Algorithmes. Vol. 1 and 2. Dunod, Paris (1983).

12. Howe, S.: New Conditions for Exactness of a Simple Penalty Function. SIAM Journal of Control. Vol. 11 **2** (1973) 378–381
13. Le Riche, R. Guyon, F.: Dual Evolutionary Optimization. Technical Report no. 01/2001. LMR, INSA de Rouen, France available at `http://meca.insa-rouen.fr/~rleriche` (2001)
14. Dantzig, G.B., Wolfe, P.: The Decomposition Algorithm for Linear Programming. Econometrica. Vol. 29 **4** (1961) 767–778
15. Davis, L.: Genetic Algorithms and Simulated Annealing. Pitman, London (1987)