A Hybrid Genetic Pattern Search Augmented Lagrangian Method for Constrained Global Optimization

Lino Costa[a], Isabel A.C.P. Espírito Santo[a], Edite M.G.P. Fernandes[b]

[a] *Department of Production and Systems*
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
[b] *Algoritmi R&D Center*
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

Abstract

Hybridization of genetic algorithms with local search approaches can enhance their performance in global optimization. Genetic algorithms, as most population based algorithms, require a considerable number of function evaluations. This may be an important drawback when the functions involved in the problem are computationally expensive as it occurs in most real world problems. Thus, in order to reduce the total number of function evaluations, local and global techniques may be combined. Moreover, the hybridization may provide a more effective trade-off between exploitation and exploration of the search space. In this study, we propose a new hybrid genetic algorithm based on a local pattern search that relies on an augmented Lagrangian function for constraint-handling. The local search strategy is used to improve the best approximation found by the genetic algorithm. Convergence to an ε-global minimizer is proved. Numerical results and comparisons with other stochastic algorithms using a set of benchmark constrained problems are provided.

Keywords: Global Optimization; Augmented Lagrangian; Genetic Algorithm; Pattern Search

Email addresses: `lac@dps.uminho.pt` (Lino Costa), `iapinho@dps.uminho.pt` (Isabel A.C.P. Espírito Santo), `emgpf@dps.uminho.pt` (Edite M.G.P. Fernandes)

1. Introduction

In this paper, the following problem is under consideration:

$$\begin{array}{lll} \underset{x\in\Omega}{\text{minimize}} & f(x) & \\ \text{subject to} & c_i(x) = 0, & i = 1, \ldots, m \\ & g_j(x) \leq 0, & j = 1, \ldots, p \end{array} \tag{1}$$

where x is an n dimensional vector and $\Omega \subset \mathbb{R}^n$ ($\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$), $f(x)$ is the objective function, $c(x) = 0$ are the equality constraints and $g(x) \leq 0$ are the inequality constraints. We aim at finding a solution with the most extreme objective function value. This type of constrained global minimization problem has many applications in engineering; for example, in Robotics [33], Mechanical Engineering [1] and Materials Science [24]. There is a variety of methods to solve a problem like (1). A common technique in global optimization literature transforms the constrained problem into an unconstrained one, by adding a penalty term to the objective function that aims to penalize constraint violation and depends on a penalty parameter [5, 18]. The implementation of this technique is not easy, since the selection of appropriate penalty parameters for the problem at hand is not a straightforward issue. To overcome this difficulty, some authors proposed the use of variable penalty schemes, e.g., dynamic scheme [17, 22, 25], self-adaptive scheme [10, 19] or superiority of feasible over infeasible solutions based scheme [7]. A comprehensive survey on constraint-handling techniques can be found in [5]. Recently, some other constraint-handling techniques have been implemented with stochastic algorithms. For instance, the filter simulated annealing [15] that uses a filter-set based procedure, the gradient based repair genetic algorithm [4] that derives information from the constraint set for repairing infeasible solutions, the evolutionary algorithm [27] that uses a multiobjective approach coupled with stochastic ranking to deal with constraints, the genetic algorithm with modified genetic operators to preserve the feasibility that includes a stochastic application of a local search procedure and a stopping rule based on asymptotic considerations [31], a hybrid particle swarm optimization with a feasibility-based rule [13], a self-organizing migrating genetic algorithm [9], and a flexible tolerance genetic algorithm [28].

The augmented Lagrangian is an interesting penalty function that avoids the side-effects associated with ill-conditioning of simpler penalty and barrier functions. Lewis and Torczon [21] proposed an augmented Lagrangian technique, where a pattern search algorithm is used to solve the unconstrained

problem, based on the augmented Lagrangian function presented by Conn *et al.* [6]. Although the inequality constraints are not considered in [6], we extended the therein proposed augmented Lagrangian to the inequality constraints, following the ideas presented in [2, 11].

Hybridization of population based algorithms with either deterministic or random local search have appeared in the literature, for instance, the memetic particle swarm optimization algorithm by Petalas *et al.* [25], the particle swarm optimization algorithm proposed by Zahara and Hu [32], the hybrid evolutionary and gradient search by Tahk *et al.* [29], the heuristic pattern search simulated annealing by Hedar and Fukushima [14], the genetic algorithm used by Tsoulos [31], the hybridization of particle swarm and simulated annealing proposed in [13], and the hybridization of the genetic algorithm and the self-organizing migrating algorithm presented in [9]. However, to the best of our knowledge, the hybridization of an augmented Lagrangian strategy, genetic algorithms and local pattern search has not been attempted yet. Thus, we propose a Hybrid Genetic Pattern Search Augmented Lagrangian (HGPSAL) algorithm that hybridizes a genetic algorithm with a derivative-free pattern search method to refine the best solution found by the genetic search. Equality and inequality constraints of the problem are treated by an augmented Lagrangian framework. This is also the first attempt to combine the convergence theory from real analysis and stochastic convergence from the probability theory to analyze some convergence properties of the proposed hybrid algorithm.

This paper is organized as follows. Section 2 describes the HGPSAL algorithm. It introduces the augmented Lagrangian technique for constraint-handling and provides details concerning the genetic algorithm and the pattern search method. The convergence to an ε-global minimizer is proved. In Section 3 we present and analyze the experimental results on 13 constrained test problems. Comparisons with other optimization algorithms are also included in this section. Finally, we conclude the paper with a few remarks and future work in Section 4.

2. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

So far, a paradigm based on a hybridization of augmented Lagrangians with genetic algorithms has not been attempted. Based on the good results

obtained by algorithms that rely on augmented Lagrangian penalties, a very promising framework seems to emerge.

On the other hand, hybridization of global and local optimizers may provide a more effective trade-off between exploitation and exploration of the search space. It is well-known that overall successful and efficient general solvers do not exist. Moreover, stochastic population based algorithms like genetic algorithms [12] are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation). Conversely, local search algorithms like pattern search are good at improving approximations to the minimum. Thus, a promising idea that may provide a reduction on the total number of function evaluations is the combination of local and global optimization techniques.

2.1. Augmented Lagrangian technique

A list of notation used in this section follows.

Notation. $\|.\|$ represents the Euclidean norm, and the component i of vector $[v]_+$ is defined by $\max\{0, v_i\}$ where $v = (v_1, \ldots, v_p)^T$. We use $\eta^j \downarrow 0$ to indicate that the sequence $\{\eta^j\}$ of non-negative decreasing numbers tends to zero.

An augmented Lagrangian technique solves a sequence of very simple subproblems where the objective function penalizes all or some of the constraint violation. This objective function is an augmented Lagrangian that depends on a penalty parameter and the multiplier vectors, and works like penalty functions. Using the ideas in [2, 6, 21], the herein implemented augmented Lagrangian function is

$$\Phi(x; \lambda, \delta, \mu) = f(x) + \lambda^T c(x) + \frac{1}{2\mu}\|c(x)\|^2 + \frac{\mu}{2}\left(\left\|\left[\delta + \frac{g(x)}{\mu}\right]_+\right\|^2 - \|\delta\|^2\right)$$

where μ is a positive penalty parameter, $\lambda = (\lambda_1, \ldots, \lambda_m)^T$, $\delta = (\delta_1, \ldots, \delta_p)^T$ are the Lagrange multiplier vectors associated with the equality and inequality constraints respectively. Function Φ aims to penalize solutions that violate only the equality and inequality constraints. Note that we do not include the simple bounds $l \leq x \leq u$ in the augmented Lagrangian function. The solution method for solving the subproblems will ensure that the bound constraints are always satisfied. Hence, the corresponding subproblem is formulated as:

$$\underset{x \in \Omega}{\text{minimize}} \;\; \Phi(x; \lambda^j, \delta^j, \mu^j) \tag{2}$$

where, for each set of fixed λ^j, δ^j and μ^j, the solution of subproblem (2) provides an approximation to the solution of (1). Denote this approximation by x^j. Here the index j is the iteration counter of the outer iterative process. To simplify the notation, we replace $\Phi(x; \lambda^j, \delta^j, \mu^j)$ by $\Phi^j(x)$ from here on.

An important issue related with the performance of augmented Lagrangian algorithms is the choice of the penalty parameter. If the algorithm is struggling to become feasible, it is convenient to decrease the penalty parameter. However, very small values of μ can cause the algorithm to converge very slowly. It is expected that as $j \to \infty$ and $\mu^j \to 0$, the solutions of the subproblems (2) converge to the solution of (1). We refer to [2, 11] for details. In practice, common safeguarded schemes maintain the sequence of penalty parameters far away from zero so that solving the subproblem (2) is an easy task.

To see when the equality and inequality constraints and the complementarity condition are satisfied, the following error function is used:

$$E(x, \delta) = \max \left\{ \frac{\|c(x)\|_\infty}{1 + \|x\|}, \frac{\|[g(x)]_+\|_\infty}{1 + \|\delta\|}, \frac{\max_i \delta_i |g_i(x)|}{1 + \|\delta\|} \right\}. \tag{3}$$

Since in finite-dimensional space, infinity and Euclidean norms are equivalent, the use of the infinity norm in (3) aims only to simplify the numerical computations. The Lagrange multipliers λ^j and δ^j are estimated in this iterative process using the first-order updating formulae

$$\bar{\lambda}_i^{j+1} = \lambda_i^j + \frac{c_i(x^j)}{\mu^j}, \quad i = 1, \ldots, m \tag{4}$$

and

$$\bar{\delta}_i^{j+1} = \max \left\{ 0, \delta_i^j + \frac{g_i(x^j)}{\mu^j} \right\}, \quad i = 1, \ldots, p \tag{5}$$

where:

Definition 1. *For all $j \in \mathbb{N}$, and for $i = 1, \ldots, m$ and $l = 1, \ldots, p$, λ_i^{j+1} is the projection of $\bar{\lambda}_i^{j+1}$ on the interval $[\lambda_{\min}, \lambda_{\max}]$ and δ_i^{j+1} is the projection of $\bar{\delta}_i^{j+1}$ on the interval $[0, \delta_{\max}]$, where $-\infty < \lambda_{\min} \leq \lambda_{\max} < \infty$ and $0 \leq \delta_{\max} < \infty$.*

After the new approximation x^j has been computed, the Lagrange multiplier vector δ associated with the inequality constraints is updated, in all iterations, since δ^{j+1} is required in the error function (3) to measure constraint

violation and complementarity. We note that the Lagrange multipliers λ_i, $i = 1, \ldots, m$ are updated only when feasibility and complementarity are at a satisfactory level, herein defined by the condition

$$E(x^j, \delta^{j+1}) \leq \eta^j \tag{6}$$

for a positive tolerance η^j. It is required that $\{\eta^j\}$ defines a decreasing sequence of positive values converging to zero, as $j \to \infty$. This is easily achieved by $\eta^{j+1} = \pi\eta^j$ for $0 < \pi < 1$.

We consider that an iteration j failed to provide an approximation x^j with an appropriate level of feasibility and complementarity if condition (6) does not hold. In this case, the penalty parameter is decreased using $\mu^{j+1} = \gamma\mu^j$ where $0 < \gamma < 1$. When condition (6) holds, then the iteration is considered satisfactory. This condition says that the iterate x^j is feasible and the complementarity condition is satisfied within some tolerance η^j, and consequently the algorithm maintains the penalty parameter value. We remark that when (6) fails to hold infinitely many times, the sequence of penalty parameters tends to zero. To be able to define an algorithm where the sequence $\{\mu^j\}$ does not reach zero, the following update is used instead:

$$\mu^{j+1} = \max\{\mu_{\min}, \gamma\mu^j\}, \tag{7}$$

where $\mu_{\min}$ is a sufficiently small positive real value.

The traditional augmented Lagrangian methods are locally convergent if the subproblems (2) are approximately solved, for instance within a predefined tolerance ε^j, for sufficiently small values of the penalty parameter [6, 21]. However, our approach aims at converging to a global solution of problem (1). Thus, global optimization methods ought to be used when solving the subproblems (2). The main differences between augmented Lagrangian algorithms are located on the framework used to find an approximation to a global minimizer of the augmented Lagrangian function subject to the bound constraints. For example, Birgin *et al.* [3] proposed a global optimization method based on the Rockafellar's augmented Lagrangian function, where the αBB method is used to find the approximate global solutions to the subproblems. We remark the following:

Remark 1. *When finding the global minimum of a continuous objective function $f(x)$ over a bounded space $\Omega \subset \mathbb{R}^n$, the point $\bar{x} \in \Omega$ is a solution to the minimization problem if*

$$f(\bar{x}) \leq \min_{y \in \Omega} f(y) + \varepsilon, \tag{8}$$

where ε is the error bound which reflects the accuracy required for the solution.

The herein proposed technique for solving (2) uses a stochastic population based algorithm, known as genetic algorithm, followed by a local search procedure to refine the best solution found thus far. The general form for the bound constrained algorithm to solve subproblems (2) is presented in Algorithm 1.

Algorithm 1 Hybrid Genetic Pattern Search Bound Constrained Algorithm

Require: $x^{j-1} \in \Omega$;

1: Find $y^j \leftarrow GA(x^{j-1})$, using the genetic algorithm presented in Subsection 2.3.1;

2: Find $x^j \leftarrow HJ(y^j)$, using the Hooke and Jeeves version of the pattern search algorithm described in Subsection 2.3.2;

3: **return** x^j.

Since the genetic algorithm is a population based method, y^j is the point with best fitness found by the algorithm. Details concerning each step of the algorithm are presented in Subsection 2.3. Issues related with convergence properties of a genetic algorithm to a global optimum are summarized below.

Since the solutions of the subproblems (2) are obtained by a stochastic method that generates a population of points at each iteration, each point is considered a stochastic vector. The analysis of the convergence properties of our algorithm relies on convergence theory known from elementary real analysis as well as on the stochastic convergence from the probability theory.

The following definition characterizes the stochastic convergence of a genetic algorithm:

Definition 2. *Let $f^k_{best} = \min\{f(x^{(i)})^k : i = 1, \ldots, s\}$ be a sequence of random variables representing the best fitness within a population of size s, at iteration k. A genetic algorithm converges to the global minimum of a continuous function $f(x)$ over a bounded space $\Omega \subset \mathbb{R}^n$ if and only if*

$$\lim_{k\to\infty} \left(Prob[f^k_{best} = f^*]\right) = 1 \tag{9}$$

where $f^* = \min_{y\in\Omega} f(y)$ [26].

The general hybrid genetic pattern search augmented Lagrangian algorithm for solving problem (1) is presented in Algorithm 2.

Algorithm 2 HGPSAL Algorithm

Require: $\lambda_{\min} < \lambda_{\max}$, $\delta_{\max} > 0$, $0 < \gamma < 1$, $\mu_{\min} \ll 1$, $\eta^* \ll 1$, $\varepsilon^* \ll 1$, $\lambda_i^1 \in [\lambda_{\min}, \lambda_{\max}]$, $i = 1, \ldots, m$, $\delta_i^1 \in [0, \delta_{\max}]$, $i = 1, \ldots, p$, $\mu^1 > 0$, $\eta^1 > 0$, $0 < \pi < 1$, $0 < \tau < 1$, $j_{\max} > 0$;

Compute ε^1;
Randomly generate $x^0 \in \Omega$; Set $j = 1$;
while the stopping criterion is not met **do**
  Find an ε^j-global minimizer x^j of subproblem (2) using Algorithm 1 so that

$$\Phi(x^j; \lambda^j, \delta^j, \mu^j) \leq \Phi(x; \lambda^j, \delta^j, \mu^j) + \varepsilon^j \tag{10}$$

  for all $x \in \Omega$;
  Update $\delta_i^{j+1} = \max\left\{0, \min\left\{\delta_i^j + \dfrac{g_i(x^j)}{\mu^j}, \delta_{\max}\right\}\right\}, i = 1, \ldots, p$;
  if $E(x^j, \delta^{j+1}) \leq \eta^j$ **then**
    Update $\lambda_i^{j+1} = \lambda_i^j + \max\left\{\lambda_{\min}, \min\left\{\dfrac{c_i(x^j)}{\mu^j}, \lambda_{\max}\right\}\right\}, i = 1, \ldots, m$;
    Set $\mu^{j+1} = \mu^j$;
  else
    Set $\lambda_i^{j+1} = \lambda_i^j$;
    Update $\mu^{j+1} = \max\{\mu_{\min}, \gamma\mu^j\}$;
  end if
  Update $\eta^{j+1} = \pi\eta^j$; Compute ε^{j+1};
  Set $j = j + 1$;
end while

The stopping criterion is also based on the error function $E(x, \delta)$ and on the tolerance ε. For sufficiently small positive values η^* and ε^*, if the algorithm finds a pair (x^j, δ^{j+1}) for which

$$E(x^j, \delta^{j+1}) \leq \eta^* \text{ and } \varepsilon^{j+1} \leq \varepsilon^* \tag{11}$$

then the algorithm stops; otherwise, the algorithm runs until a maximum of (outer) iterations, $j_{\max}$, is reached. The tolerance ε varies with the Lagrange multipliers and penalty parameter values according to

$$\varepsilon^j = \max\left\{\varepsilon^*, \tau\left(1 + \|\lambda^j\| + \|\delta^j\| + (\mu^j)^{-1}\right)^{-1}\right\}, \ 0 < \tau < 1.$$

Note that a decreasing sequence of μ values will yield a decreasing sequence of ε values forcing more and more accurate solutions to the subproblems (2)

as proposed in [21]. As shown in the HGPSAL algorithm, the Lagrange multiplier estimates are obtained by safeguarded first-order updates aiming to maintain the multiplier vectors bounded throughout the process.

2.2. Convergence to an ε^*-global minimizer

Here we aim at providing a convergence analysis of the Hybrid Genetic Pattern Search Augmented Lagrangian algorithm. At each outer iteration j, an ε^j-global minimizer of the subproblem (2) is obtained, where $\varepsilon^j \to \varepsilon^*$. The convergence analysis of Algorithm 2 has some similarities with that of the global augmented Lagrangian method presented in [3].

We now state the assumptions that are needed to show convergence to an ε^*-global minimum for the HGPSAL algorithm. Let $\{x^j\}$ be the sequence generated by the algorithm.

Assumption A 1. *A global minimizer z of the problem* (1) *exists.*

Assumption A 2. *The sequence $\{x^j\}$ generated by the Algorithm 2 is well defined and there exists a set of indices $\mathcal{N} \subseteq \mathbb{N}$ so that $\lim_{j\in\mathcal{N}} x^j = x^*$.*

Assumption A 3. *The functions $f : \mathbb{R}^n \to \mathbb{R}$, $c : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are continuous in a neighborhood of x^*.*

Since the set Ω is compact and $\Phi(x; \lambda^j, \delta^j, \mu^j)$ is continuous, an ε^j-global minimizer of subproblem (2), x^j, necessarily exists. The above Definition 1 together with updates (4) and (5) define safeguarded first-order Lagrange multiplier estimates. The Lagrange multiplier vectors at x^* are herein represented by λ^* and δ^* and we will assume the following:

Assumption A 4. *For all $i = 1, \ldots, m$ and $l = 1, \ldots, p$, $\lambda_i^* \in [\lambda_{\min}, \lambda_{\max}]$ and $\delta_l^* \in [0, \delta_{\max}]$.*

We now investigate the properties of the limit points of sequences generated by the Algorithm 2.

Theorem 1. *Assume that the Assumptions A 1 through A 4 hold. Then every limit point x^* of the sequence $\{x^j\}$ generated by the Algorithm 2 is feasible.*

Proof. Since $x^j \in \Omega$ and Ω is closed then $x^* \in \Omega$. We now consider two cases: (a) when $\{\mu^j\}$ is bounded; (b) when $\mu^j \to 0$.

In case (a), there exists an index j_f such that $\mu^j = \mu^{j_f} = \bar{\mu}$ for all $j \geq j_f$. This means that for all $j \geq j_f$, (6) holds. The fact that $\eta^j \downarrow 0$ implies that $\|c(x^j)\| \to 0$, as well as $\|[g(x^j)]_+\| \to 0$ and $\delta_i^j |g_i(x^j)| \to 0$, for all $i = 1, \ldots, p$, and we conclude that the limit point is feasible.

The proof in case (b) is by contradiction. We assume that x^* is not feasible and that a global minimizer z exists in Ω (the same for all j) such that $\|c(z)\| = \|[g(z)]_+\| = 0$. Thus

$$\|c(x^*)\|^2 + \|[g(x^*)]_+\|^2 > \|c(z)\|^2 + \|[g(z)]_+\|^2$$

and since c and g are continuous, λ^j and δ^j are bounded, $\mu^j \to 0$, and there exists a constant $K > 0$ and a set of indices $\mathcal{N} \subset \mathbb{N}$ such that $\lim_{j \in \mathcal{N}} x^j = x^*$, then for a large enough $j \in \mathcal{N}$ we have

$$c(x^j)^T c(x^j) + \|[g(x^j)]_+\|^2 - \|\delta^j\|^2 > c(z)^T c(z) + \|[g(z)]_+\|^2 - \|\delta^j\|^2 + K$$

and

$$\begin{aligned} c(x^j)^T \quad & \left(\lambda^j + \frac{c(x^j)}{2\mu^j}\right) + \frac{\mu^j}{2}\left(\left\|\left[\frac{g(x^j)}{\mu^j} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) \\ & > c(z)^T \left(\lambda^j + \frac{c(z)}{2\mu^j}\right) + \frac{\mu^j}{2}\left(\left\|\left[\frac{g(z)}{\mu^j} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) + K. \end{aligned}$$

We also have

$$\begin{aligned} & f(x^j) + c(x^j)^T \left(\lambda^j + \frac{c(x^j)}{2\mu^j}\right) + \frac{\mu^j}{2}\left(\left\|\left[\frac{g(x^j)}{\mu^j} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) \\ & > f(z) + c(z)^T \left(\lambda^j + \frac{c(z)}{2\mu^j}\right) + \frac{\mu^j}{2}\left(\left\|\left[\frac{g(z)}{\mu^j} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) + \varepsilon^j \end{aligned}$$

where for large enough $j \in \mathcal{N}$, $K + f(x^j) - f(z) > \varepsilon^j$, implying that

$$\Phi(x^j; \lambda^j, \delta^j, \mu^j) > \Phi(z; \lambda^j, \delta^j, \mu^j) + \varepsilon^j,$$

which contradicts the definition of x^j in (10). □

We now prove that a sequence of iterates generated by the algorithm converges to an ε^*-global minimizer of the problem (1).

Theorem 2. *Assume that the Assumptions A 1 through A 4 hold. Then every limit point x^* of a sequence $\{x^j\}$ generated by Algorithm 2 is an ε^*-global minimizer of the problem* (1).

Proof. Again, we consider the two cases: (a) when $\{\mu^j\}$ is bounded; (b) when $\mu^j \to 0$. Let $\mathcal{N} \subset \mathbb{N}$ be the set of indices such that $\lim_{j\in\mathcal{N}} x^j = x^*$.

First, we consider case (a). By the definition of x^j in the Algorithm 2, and since $\mu^j = \mu^{j_f} = \bar{\mu}$ for all $j \geq j_f$, we have:

$$\begin{aligned} & f(x^j) + c(x^j)^T \left(\lambda^j + \frac{c(x^j)}{2\bar{\mu}}\right) + \frac{\bar{\mu}}{2}\left(\left\|\left[\frac{g(x^j)}{\bar{\mu}} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) \\ & \leq f(z) + c(z)^T \left(\lambda^j + \frac{c(z)}{2\bar{\mu}}\right) + \frac{\bar{\mu}}{2}\left(\left\|\left[\frac{g(z)}{\bar{\mu}} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) + \varepsilon^j \end{aligned}$$

where $z \in \Omega$ is a global minimizer of problem (1). Since $c(z) = 0$, $g(z) \leq 0$, and $\delta^j, \bar{\mu} > 0$ for all $j \geq j_f$, we get

$$\left\|\left[\frac{g(z)}{\bar{\mu}} + \delta^j\right]_+\right\|^2 \leq \|\delta^j\|^2$$

and then

$$\begin{aligned} & f(x^j) + c(x^j)^T \left(\lambda^j + \frac{c(x^j)}{2\bar{\mu}}\right) + \frac{\bar{\mu}}{2}\left(\left\|\left[\frac{g(x^j)}{\bar{\mu}} + \delta^j\right]_+\right\|^2 - \|\delta^j\|^2\right) \\ & \leq f(z) + \varepsilon^j. \end{aligned}$$

Now, let $\mathcal{N}_1 \subset \mathcal{N}$ be a set of indices such that $\lim_{j\in\mathcal{N}_1} \lambda^j = \lambda^*$ and $\lim_{j\in\mathcal{N}_1} \delta^j = \delta^*$. Taking limits for $j \in \mathcal{N}_1$ and also using $\lim_{j\in\mathcal{N}_1} \varepsilon^j = \varepsilon^*$, we obtain:

$$f(x^*) + \frac{\bar{\mu}}{2}\left(\left\|\left[\frac{g(x^*)}{\bar{\mu}} + \delta^*\right]_+\right\|^2 - \|\delta^*\|^2\right) \leq f(z) + \varepsilon^*.$$

Since x^* is feasible, either $g_i(x^*) = 0$ and $\delta_i^* > 0$, or $g_i(x^*) < 0$ and $\delta_i^* = 0$, for each i, and therefore

$$f(x^*) \leq f(z) + \varepsilon^*$$

which proves the claim that x^* is an ε^*-global minimizer, since z is a global minimizer.

For case (b), we have

$$
\begin{aligned}
& f(x^j) + c(x^j)^T \left(\lambda^j + \frac{c(x^j)}{2\mu^j} \right) + \frac{\mu^j}{2} \left(\left\| \left[\frac{g(x^j)}{\mu^j} + \delta^j \right]_+ \right\|^2 - \|\delta^j\|^2 \right) \\
& \leq f(z) + c(z)^T \left(\lambda^j + \frac{c(z)}{2\mu^j} \right) + \frac{\mu^j}{2} \left(\left\| \left[\frac{g(z)}{\mu^j} + \delta^j \right]_+ \right\|^2 - \|\delta^j\|^2 \right) + \varepsilon^j
\end{aligned}
$$

for all $j \in \mathbb{N}$. Since z is feasible, and using an argument similar to that used in case (a), we get:

$$
\begin{aligned}
& f(x^j) + c(x^j)^T \left(\lambda^j + \frac{c(x^j)}{2\mu^j} \right) + \frac{\mu^j}{2} \left(\left\| \left[\frac{g(x^j)}{\mu^j} + \delta^j \right]_+ \right\|^2 - \|\delta^j\|^2 \right) \\
& \leq f(z) + \varepsilon^j.
\end{aligned}
$$

Now, taking limits for $j \in \mathcal{N}$, and using the convergence of x^j, $c(x^j) \to 0$, and the fact that $\lim \mu^j = 0$, we obtain the desired result

$$
f(x^*) \leq f(z) + \varepsilon^*.
$$

□

2.3. Global optimization of the subproblems

An ε^j-global minimum of subproblem (2) can be obtained by applying the Hooke and Jeeves pattern search method to enhance a genetic algorithm with elitism. We remark that the implementation of elitism in a genetic algorithm, maintaining the best fitness over the entire iterative process, is a crucial issue related with the stochastic convergence properties of the algorithm. The goal is to guarantee that the probability of converging to a globally optimal point approaches one as $k \to \infty$, so that condition (9) is fulfilled [26]. As an immediate consequence, an approximation to the global minimum of the augmented Lagrangian may be obtained with a pre-defined accuracy. The solution may be further improved applying a local pattern search, namely the Hooke and Jeeves algorithm, to that solution.

2.3.1. Genetic Algorithm

A Genetic Algorithm (GA) is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover [12]. Thus, unlike conventional algorithms, GAs start

from a population of points P of size s. In spite of the traditional binary representation used by GAs, our implementation uses a real representation since we are dealing with continuous problems. Therefore, each point of the population $z^{(l)}$, for $l = 1, \ldots s$, is an n dimensional vector.

A fitness function is defined as evaluation function in order to compare the points of the population and to apply a stochastic selection that guarantees that better points are more likely to be selected. The fitness function, $\Phi^j(x)$, corresponds to the objective function of the subproblem (2). A tournament selection was considered, i.e., tournaments are played between two points and the best point (with the lowest fitness value) is chosen for the pool.

New points in the search space are generated by the application of genetic operators (crossover and mutation) to the selected points from population. Elitism was implemented by maintaining, during the search, a given number, e, of the best points in the population.

Crossover combines two points in order to generate new ones. A Simulated Binary Crossover (SBX) [8] that simulates the working principle of single-point crossover operator for binary strings was implemented. Two points, $z^{(1)}$ and $z^{(2)}$, are randomly selected from the pool and, with probability p_c, two new points, $w^{(1)}$ and $w^{(2)}$ are generated according to

$$\begin{aligned} w_i^{(1)} &= 0.5\left((1+\beta_i)z_i^{(1)} + (1-\beta_i)z_i^{(2)}\right) \\ w_i^{(2)} &= 0.5\left((1-\beta_i)z_i^{(1)} + (1+\beta_i)z_i^{(2)}\right) \end{aligned}$$

for $i = 1, \ldots, n$. The values of β_i are obtained from the following distribution:

$$\beta_i = \begin{cases} (2r_i)^{\frac{1}{\eta_c+1}} & \text{if} \quad r_i \leq 0.5 \\ \left(\frac{1}{2(1-r_i)}\right)^{\frac{1}{\eta_c+1}} & \text{if} \quad r_i > 0.5 \end{cases}$$

where $r_i \sim U(0,1)$ and $\eta_c > 0$ is an external parameter of the distribution. This procedure is repeated until the number of generated points equals the number of points in the pool.

A Polynomial Mutation is applied, with a probability p_m, to the points produced by the crossover operator. Mutation introduces diversity in the population since crossover, exclusively, could not assure the exploration of new regions of the search space. This operator guarantees that the probability of creating a new point $t^{(l)}$ closer to the previous one $w^{(l)}$ $(l = 1, \ldots, s)$ is larger than the probability of creating one away from it. It can be expressed

by:

$$t_i^{(l)} = w_i^{(l)} + (u_i - l_i)\iota_i$$

for $i = 1, \ldots, n$. The values of ι_i are given by:

$$\iota_i = \begin{cases} (2r_i)^{\frac{1}{\eta_m+1}} - 1 & \text{if} \quad r_i < 0.5 \\ 1 - (2(1-r_i))^{\frac{1}{\eta_m+1}} & \text{if} \quad r_i \geq 0.5 \end{cases}$$

where $r_i \sim U(0,1)$ and $\eta_m > 0$ is an external parameter of the distribution. The GA proceeds as the following algorithm.

Algorithm 3 Genetic Algorithm

Require: x^{j-1} (current approximation), e (number of best points in the population), s (population size), p_c (crossover probability), p_m (mutation probability), $k_{\max} > 0$;

Set $z^{(1)} = x^{j-1}$ and randomly generate $z^{(l)} \in \Omega$, for $l = 2, \ldots, s$ *(Initialization of P)*;
Set $k = 0$;
while the stopping criterion is not met **do**
  Compute $\Phi^j(z^{(l)})$, for $l = 1, \ldots, s$ *(Fitness Evaluation)*;
  Select by tournaments $s - e$ points from P *(Selection)*;
  Apply SBX crossover with probability p_c *(Crossover)*;
  Apply polynomial mutation with probability p_m *(Mutation)*;
  Replace the worst $s - e$ points of P *(Elitism)*;
  Set $y^j = z_{best}^k$;
  Set $k = k + 1$;
end while
return y^j.

To guarantee that the Algorithm 3 converges to an ε^j-global minimizer of the augmented Lagrangian, the stopping criterion relies on the condition:

$$\Phi^j(z_{best}^k) - \Phi(\bar{z}) \leq \varepsilon^j \tag{12}$$

where $\Phi^j(z_{best}^k)$ is the fitness value of the best point in population (z_{best}^k), at iteration k, and in a practical context $\Phi(\bar{z})$ is the smallest functional value considering all the algorithms that found a feasible point. However, if it happens that the previous criterion is not satisfied in $k_{\max}$ iterations, the algorithm is terminated and the best point in the population is returned.

2.3.2. The Hooke and Jeeves Pattern Search

A pattern search method is a derivative-free method that performs, at each iteration k, a series of exploratory moves around a current approximation, z^k, in order to find a new approximation $z^{k+1} = z^k + \Delta^k s^k$, with a lower fitness value. We use k for the iteration counter of this inner iterative process. For $k = 0$, the initial approximation to begin the search is $z^0 = y^j$ (see Algorithm 1). The scalar Δ^k represents the step length and the vector s^k determines the direction of the step. The exploratory moves to produce $\Delta^k s^k$ and the updating of Δ^k and s^k define a particular pattern search method and their choices are crucial to the success of the algorithm. When $\Phi^j(z^{k+1}) < \Phi^j(z^k)$ then the iteration is considered successful; otherwise it is unsuccessful. When an iteration is successful, the step length is not modified, while in an unsuccessful iteration Δ^k is reduced. See for example [20, 30].

In our algorithm, $\Delta^k s^k$ is computed by the Hooke and Jeeves (HJ) search method [16]. This algorithm differs from the traditional coordinate search since it performs two types of moves: the exploratory move and the pattern move. An exploratory move is a coordinate search - a search along the coordinate axes - around a selected approximation, using a step length Δ^k. A pattern move is a promising direction that is defined by $z^k - z^{k-1}$ when the previous iteration was successful and z^k was accepted as the new approximation. A new trial approximation is then defined as $z^k + (z^k - z^{k-1})$ and an exploratory move is then carried out around this trial point. If this search is successful, the new approximation is accepted as z^{k+1}. We refer to [16, 20] for details. This HJ iterative procedure terminates, providing a new approximation x^j to the problem (1), $x^j \leftarrow z^{k+1}$, when the following stopping condition is satisfied, $\Delta^k \leq \varepsilon^j$. However, if this condition can not be satisfied in $k_{\max}$ iterations, then the procedure is stopped with the last available approximation.

2.3.3. Forcing feasibility

The inner iterative process must return an approximate solution that satisfies the bound constraints (recall (2)). While using the genetic algorithm and the Hooke and Jeeves version of the pattern search, any computed solution x that does not satisfy the bounds is projected onto the set Ω component by component (for all $i, \ldots, n$) as follows:

$$x_i = \begin{cases} l_i & \text{if} \quad x_i < l_i \\ x_i & \text{if} \quad l_i \leq x_i \leq u_i \\ u_i & \text{if} \quad x_i > u_i \end{cases} .$$

Table 1: Test problems.

Prob.		Type of $f(x)$	n	p	m	n_{act}	f_{global}
g01	min	quadratic	13	9	0	6	-15.00000
g02	max	nonlinear	20	2	0	1	0.803619
g03	max	polynomial	10	0	1	1	1.000000
g04	min	quadratic	5	6	0	2	-30665.54
g05	min	cubic	4	2	3	3	5126.498
g06	min	cubic	2	2	0	2	-6961.814
g07	min	quadratic	10	8	0	6	24.30621
g08	max	nonlinear	2	2	0	0	0.095825
g09	min	polynomial	7	4	0	2	680.6301
g10	min	linear	8	6	0	6	7049.248
g11	min	quadratic	2	0	1	1	0.750000
g12	max	quadratic	3	1	0	0	1.000000
g13	min	nonlinear	5	0	3	3	0.053945

3. Numerical results

The HGPSAL algorithm is coded in the MatLab programming language so that the problems also coded in MatLab could be easily read and solved. The numerical results were obtained with a PC Intel Core i3 CPU M350 @ 2.27GHz with 4GB of memory.

3.1. Test Problems

In this study, to evaluate the performance of the HGPSAL algorithm, 13 benchmark problems were considered. This set contains difficult constrained optimization problems with very distinct properties [18, 23]. The characteristics of the problems are summarized in Table 1 that lists the type of objective function, the number of decision variables (n), the number of inequality constraints (p), the number of equality constraints (m), the number of active constraints at the optimum (n_{act}) and the known optimal value (f_{global}).

3.2. Algorithm parameters

All parameters of the HGPSAL algorithm were kept constant for all problems. No effort was made in finding the best parameter setting for each problem. Table 2 shows the parameters of the augmented Lagrangian used in all experiments. In Table 3 the genetic algorithm parameters are listed. The maximum number of iterations $k_{\max}$ for Hooke and Jeeves pattern search was set on 200 iterations.

Table 2: Augmented Lagrangian parameters.

$\lambda_{\min}$	$\lambda_{\max}$	$\delta_{\max}$	μ^1	$\mu_{\min}$	γ	ε^*	η^*	$\lambda_i^1, \delta_i^1, \forall i$	η^1	$j_{\max}$	π	τ
-10^{12}	10^{12}	10^{12}	1	10^{-12}	0.5	10^{-12}	10^{-6}	0	1	300	0.5	0.5

Table 3: Genetic Algorithm parameters.

$k_{\max}$	s	e	p_c	η_c	p_m	η_m
200	20	2	0.9	20	$1/n$	20

3.3. Discussion

Due to the stochastic nature of the Algorithm 2 each problem was run 30 times. We then report the best of the 30 best solutions, 'Best', the worst of the 30 best solutions, 'Worst', the average of the 30 best solutions, 'Average', the standard deviation of the function values, 'St.Dev.' and the average number of function evaluations after the 30 runs, 'Avg.Eval.'.

The most unbiased measure to assess and compare the performance of stochastic algorithms is the average value of the obtained results over all the runs, since it provides the central tendency of the results. A number of runs of the order herein used ensures statistically significance since it is considered sufficiently large and the statistical distribution of the average value asymptotically converges to a normal distribution.

Table 4 shows the results of the test problems in terms of the best, the worst, the average, the standard deviation of the objective value of the solutions and the average number of function evaluations. We report the solutions obtained by HGPSAL, as well as those obtained by the filter simulated annealing (FSA) [15], the gradient based repair genetic algorithm (GRGA) [4], the evolutionary algorithm (EA) [27], the genetic algorithm (GAT) [31], the hybrid particle swarm optimization algorithm (HPSO) [13] and the self-organizing migrating genetic algorithm (C-SOMGA) [9]. These stochastic algorithms use different approaches to handle constraints: FSA uses a filter-set based procedure and simulated annealing; GRGA derives information from the constraint set for repairing infeasible solutions and searches using a genetic algorithm; EA uses a multiobjective approach to deal with constraints and an evolutionary algorithm based on evolution strategies; GAT incorporates modified genetic operators to preserve feasibility; HPSO uses a feasibility-based rule and C-SOMGA uses a penalty free constraint handling

selection. It should be noted that HGPSAL is being compared with other algorithms which use different constraint-handling techniques. Solutions to problems g12 and g13 by GRGA are not given in [4] and therefore they were not included in the table. The solutions found by GAT for the set of problems in [31] are only for the problems g01, g02, g03, g08 and g11, and the worst objective function values are not available. The results presented in [13] of HPSO are only for problems g04, g08 and g12. As to C-SOMGA [9], the available solutions are for problems g01, g02, g08, g11 and g12. We remark that the listed values in [9] are only the average value and the standard deviation of the objective function.

Table 4: Comparison results of test problems using HGPSAL, FSA, GRGA, EA, HPSO and C-SOMGA.

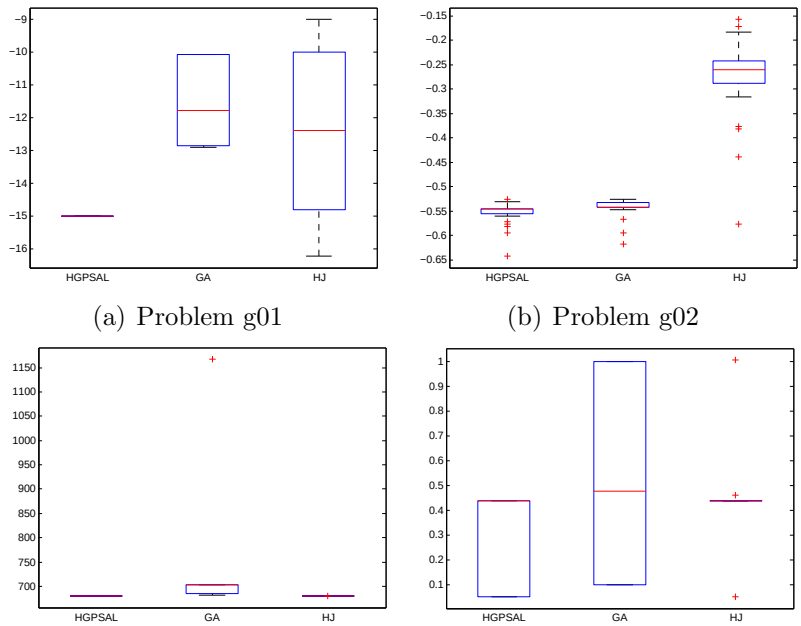
Prob	Optimal	Algorithm	Best	Worst	Average	St.Dev.	Avg.Eval.
g01	-15.00000	HGPSAL	-15.00000	-14.99993	-14.99998	0.00003	87927
		FSA	-14.99911	-14.97998	-14.99331	0.00481	205748
		GRGA	-15.00000	-15.00000	-15.00000	0.00000	95512
		EA	-15.00000	-15.00000	-15.00000	0.00000	122000
		GAT	-15.0000		-14.9999	0.00008	21833
		C-SOMGA			-14.99225	0.00304	150000
g02	0.803619	HGPSAL	0.61133	0.52666	0.556323	0.02501	227247
		FSA	0.754912	0.271311	0.371708	0.09802	227832
		GRGA	0.801119	0.745329	0.785746	0.01370	331972
		EA	0.803619	0.609330	0.753209	0.03700	349600
		GAT	0.79466		0.7555	0.02710	483870
		C-SOMGA			0.77542	0.02739	150000
g03	1.000000	HGPSAL	1.000000	1.000000	1.000000	0.00000	113890
		FSA	1.000000	0.991519	0.999187	0.00165	314938
		GRGA	0.999980	0.999790	0.999920	0.00006	399804
		EA	1.000000	1.000000	1.000000	0.00000	339600
		GAT	1.0125		1.0124	0.00003	15240
g04	-30665.54	HGPSAL	-30665.54	-30665.54	-30665.54	0.00000	106602
		FSA	-30665.54	-30664.54	-30665.47	0.17322	86154
		GRGA	-30665.54	-30665.54	-30665.54	0.00000	26981
		EA	-30665.54	-30665.54	-30665.54	0.00000	66400
		HPSO	-30665.539	-30665.539	-30665.539	0.00000	81000
g05	5126.498	HGPSAL	5126.498	5126.498	5126.498	0.00000	199439
		FSA	5126.498	5126.498	5126.498	0.00000	47661
		GRGA	5126.498	5126.498	5126.498	0.00000	39459
		EA	5126.498	5126.498	5126.498	0.00000	62000
g06	-6961.814	HGPSAL	-6961.814	-6961.809	-6961.814	0.00127	77547
		FSA	-6961.814	-6961.814	-6961.814	0.00000	44538
		GRGA	-6961.814	-6961.814	-6961.814	0.00000	13577
		EA	-6961.814	-6961.814	-6961.814	0.00000	56000
g07	24.30621	HGPSAL	24.30621	24.30621	24.30621	0.00000	81060
		FSA	24.31057	24.64440	24.37953	0.07164	404501
		GRGA	24.32940	24.83520	24.47190	0.12900	428314
		EA	24.30621	24.63500	24.33700	0.04100	350000
g08	0.095825	HGPSAL	0.095825	0.095825	0.095825	0.00000	39381
		FSA	0.095825	0.095825	0.095825	0.00000	56476
		GRGA	0.095825	0.095825	0.095825	0.00000	6217

Table 4: (*continued*)

Prob	Optimal	Algorithm	Best	Worst	Average	St.Dev.	Avg.Eval.
		EA	0.095825	0.095825	0.095825	0.00000	49600
		GAT	0.09582		0.09582	0.00000	3147
		HPSO	0.095825	0.095825	0.095825	0.00000	81000
		C-SOMGA			0.08816	0.01657	150000
g09	680.6301	HGPSAL	680.6301	680.6301	680.6301	0.00000	56564
		FSA	680.6301	680.6983	680.6364	0.01452	324569
		GRGA	680.6303	680.6538	680.6381	0.00661	388453
		EA	680.6301	680.6301	680.6301	0.00000	310000
g10	7049.248	HGPSAL	7049.247	7049.248	7049.248	0.00050	150676
		FSA	7059.864	9398.649	7059.321	542.342	243520
		GRGA	7049.261	7051.686	7049.566	0.57000	572629
		EA	7049.404	7258.540	7082.227	42.0000	344000
g11	0.750000	HGPSAL	0.750000	0.750000	0.750000	0.00000	17948
		FSA	0.749900	0.749900	0.749900	0.00000	23722
		GRGA	0.750000	0.750000	0.750000	0.00000	7215
		EA	0.750000	0.750000	0.750000	0.00000	46400
		GAT	0.7500		0.75003	0.00001	4651
		C-SOMGA			0.82519	0.09761	150000
g12	1.000000	HGPSAL	1.000000	1.000000	1.000000	0.00000	61344
		FSA	1.000000	1.000000	1.000000	0.00000	59355
		EA	1.000000	1.000000	1.000000	0.00000	20400
		HPSO	1.000000	1.000000	1.000000	0.00000	81000
		C-SOMGA			0.88794	0.21215	150000
g13	0.053945	HGPSAL	0.053950	0.438851	0.349041	0.16558	31269
		FSA	0.053950	0.438851	0.297720	0.18865	120268
		EA	0.053942	0.438804	0.111671	0.14000	109200

After analyzing the results, it can be observed that HGPSAL consistently found the global optimal solutions in all problems. For problem g02, HGPSAL did not find any good approximation to the optimum. We remark that this is a very difficult nonconvex problem. For problem g13, HGPSAL obtained good approximations to the optimum in seven of the 30 runs. The small values of standard deviations obtained by HGPSAL in the majority of the problems highlight the consistency of the algorithm. In terms of computational effort (average number of function evaluations), the conclusions are the following. In ten of the 13 problems, HGPSAL requires the smallest, or one of the smallest, effort between all methods in comparison, while in the remaining three problems, HGPSAL requires the biggest effort.

In Figure 1 we can analyze the exploration-exploitation trade-off when using the proposed hybridization scheme. The HGPSAL is compared with both the GA and the HJ methods alone. We note that HJ is a deterministic method that can be trapped in a local optimum. In general, the final solution is heavily dependent on the provided initial approximation. Four



(a) Problem g01 (b) Problem g02

(c) Problem g09 (d) Problem g13

Figure 1: Boxplots of best function values to compare HGPSAL, GA and HJ

example problems were selected from the previous set (g01, g02, g09, g13). Based on the boxplots of the best function values, over the 30 runs, we can observe that the GA gives the least accurate solutions. GA is good in finding the promising area of the search (exploration), but is not effective in fine-tuning the approximation to the minimum (exploitation). We note that the HJ method may also give a set of different solutions due to the randomly generated initial approximations of the algorithm.

On the other hand, we can see in problem g02 (a non-convex problem) that the Hooke & Jeeves method totally misses the global maximum, and in problem g13 it misses the global minimum in almost all runs. This highlights the advantage of our hybridization scheme over using GA or HJ alone.

4. Conclusions and Future Work

In this paper, we developed and proposed a hybrid algorithm for constrained global optimization that combines the augmented Lagrangian technique for handling the equality and inequality constraints with a genetic algorithm as global optimizer and a pattern search as local optimizer. A

convergence analysis of the algorithm is also provided. The sequence of iterates generated by the algorithm converges to an ε-global minimizer.

Numerical results for a set of test problems seem to show that hybridization provides a more effective trade-off between exploitation and exploration of the search space. Moreover, the numerical results validate and stress the advantages of using the augmented Lagrangian technique for handling constraints.

In general, HGPSAL exhibited a good performance in terms of accuracy and precision of the obtained optimal approximations. The algorithm has shown to be competitive and efficient when compared with the others.

As future work, we intend to perform comparisons with other stochastic approaches and solve other benchmark problems, to improve the integration of GA and HJ (e.g., using hill climbing strategies) and to tune the parameters of the algorithm.

Acknowledgments

The authors thank an anonymous referee whose comments helped to improve the paper. Financial support from FEDER COMPETE (Programa Operacional Fatores de Competitividade / Operational Programme Thematic Factors of Competitiveness) and FCT (Fundação para a Ciência e a Tecnologia / Portuguese Foundation for Science and Technology) under Project FCOMP-01-0124-FEDER-022674 is gratefully acknowledged.

References

[1] H.-G. Beyer, B. Sendhoff. Robust optimization - a comprehensive survey. Computer Methods in Applied Mechanics and Engineering, 196 (33-34), (2007) 3190–3218.

[2] D.P. Bertsekas, Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999).

[3] E.G. Birgin, C.A. Floudas and J.M. Martinez, Global minimization using an Augmented Lagrangian method with variable lower-level constraints, Mathematical Programming, Ser. A, 125(1), (2010) 139–162.

[4] P. Chootinan, A. Chen, Constraint handling in genetic algorithms using a gradient-based repair method, Computers and Operations Research, 33(8) (2006) 2263–2281.

[5] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering, 191(11) (2002) 1245–87.

[6] A.R. Conn, N.I.M. Gould, P.L. Toint, A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds, Journal on Numerical Analysis, 28 (1991) 545–572.

[7] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering, 186(2-4) (2000) 311–338.

[8] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems, 9(2) (1995) 115–149.

[9] K. Deep and Dipti, A self-organizing migrating genetic algorithm for constrained optimization, Applied Mathematics and Computation, 198 (2008) 237–250.

[10] R. Farmani, J.A. Wright, Self-adaptive fitness formulation for constrained optimization, IEEE Transactions on Evolutionary Computation, 7(5) (2003) 445–455.

[11] A.V. Fiacco, G.P McCormick. Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley (1968).

[12] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley (1989).

[13] Q. He and L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Applied Mathematics and Computation, 186 (2007) 1407–1422.

[14] A.-R. Hedar, M. Fukushima, Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization, Optimization Methods and Software, 19 (2004) 291–308.

[15] A.-R. Hedar, M. Fukushima, Derivative-free filter simulated annealing method for constrained continuous global optimization, Journal of Global Optimization, 35(4) (2006) 521–549.

[16] R. Hooke, T.A. Jeeves, Direct search solution of numerical and statistical problems, Journal on Associated Computation, 8 (1961) 212–229.

[17] J. Joines, C. Houck, On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs, Proceedings of the first IEEE conference on evolutionary computation (1994) 579–84.

[18] S. Koziel, Z. Michalewicz, Evolutionary Algorithms, homomorphous mappings, and constrained parameter optimization, Evol. Comput., 7(1) (1999) 19–14.

[19] A.C.C. Lemonge, H.J.C. Barbosa, An Adaptive Penalty Scheme for Genetic Algorithms in Structural Optimization, International Journal for Numerical Methods in Engineering, 59(5) (2004) 703–736.

[20] R.M. Lewis, V. Torczon, Pattern search algorithms for bound constrained minimization, SIAM Journal on Optimization, 9 (1999) 1082–1099.

[21] R.M. Lewis, V.Torczon, A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, SIAM Journal on Optimization, 12 (2002) 1075–1089.

[22] J.-L. Liu, J.-H. Lin, Evolutionary computation of unconstrained and constrained problems using a novel momentum-type particle swarm optimization, Engineering Optimization, 39(3) (2007) 287–305.

[23] Z. Michalewicz, Genetic Algorithms, numerical optimization and constrains, in Proc. 6th Int. Conf. Genetic Algorithms (1995) 151–158.

[24] W. Paszkowicz, Genetic Algorithms, A Nature-Inspired Tool:Survey of Application in Materials Science and Related Fields, Materials and Manufacturing Processes, 24(2) (2009) 174–197.

[25] Y.G. Petalas, K.E. Parsopoulos, M.N. Vrahatis, Memetic particle swarm optimization, Annals of Operations Research, 156 (2007) 99–127.

[26] G. Rudolph, Convergence Analysis of Canonical Genetic Algorithms, IEEE Transactions on Neural Networks, 5 (1994) 96–101.

[27] T.P. Runarsson, X. Yao, Search biases in constrained evolutionary optimization, IEEE Transactions on Systems, Man and Cybernetics, 35 (2) (2005) 233–243.

[28] W. Shang, S. Zhao, Y. Shen, A flexible tolerance genetic algorithm for optimal problems with nonlinear equality constraints, Advanced Engineering Informatics, 23 (2009) 253–264.

[29] M.-J. Tahk, H.-W. Woo, M.-S. Park, A hybrid optimization method of evolutionary and gradient search, Engineering Optimization, 39 (2007) 87–104.

[30] V. Torczon, On the convergence of pattern search algorithms, SIAM Journal on Optimization, 7 (1997) 1–25.

[31] I. G. Tsoulos, Solving constrained optimization problems using a novel genetic algorithm, Applied Mathematics and Computation, 208 (2009) 273–283.

[32] E. Zahara, C.-H. Hu, Solving constrained optimization problems with hybrid particle swarm optimization, Engineering Optimization, 40 (2008) 1031–1049.

[33] Y. Wan, G. Wang, S. Ji, J. Li, A Survey on the Parallel Robot Optimization, Proceeding on Second International Symposium on Intelligent Information Technology Application, 2 (2008) 655–659.