Expert Systems with Applications 41 (2014) 7077–7094

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A modified Covariance Matrix Adaptation Evolution Strategy with adaptive penalty function and restart for constrained optimization

Vinícius Veloso de Melo [a,*], Giovanni Iacca [b]

[a] Institute of Science and Technology, Universidade Federal de São Paulo, UNIFESP, São José dos Campos, São Paulo 12231-280, Brazil
[b] INCAS[3], P.O. Box 797, 9400 AT Assen, The Netherlands

ARTICLE INFO

Article history:
Available online 21 June 2014

Keywords:
Constrained optimization
Covariance Matrix Adaptation Evolution Strategy
Adaptive penalty function

ABSTRACT

In the last decades, a number of novel meta-heuristics and hybrid algorithms have been proposed to solve a great variety of optimization problems. Among these, constrained optimization problems are considered of particular interest in applications from many different domains. The presence of multiple constraints can make optimization problems particularly hard to solve, thus imposing the use of specific techniques to handle fitness landscapes which generally show complex properties. In this paper, we introduce a modified Covariance Matrix Adaptation Evolution Strategy (CMA-ES) specifically designed for solving constrained optimization problems. The proposed method makes use of the restart mechanism typical of most modern variants of CMA-ES, and handles constraints by means of an adaptive penalty function. This novel CMA-ES scheme presents competitive results on a broad set of benchmark functions and engineering problems, outperforming most state-of-the-art algorithms as for both efficiency and constraint handling.

1. Introduction

In real-world, several problems are modeled as nonlinear optimization problems, where objective functions have to be optimized under some given constraints. In this kind of problem, one wants to find $\vec{x}$ that optimizes $f(\vec{x})$ which is

$$\text{subject to}: \begin{array}{ll} h_i(\vec{x}) = 0, & i = 1,2,\ldots,m \\ g_i(\vec{x}) \leq 0, & i = 1,2,\ldots,p \end{array} \qquad (1)$$

where $f(\vec{x})$ is the objective (or fitness) function to be optimized, and $\vec{x} \in \mathbb{R}^n$ is an n-dimensional vector of design variables, $\vec{x} = [x_1, x_2, \ldots, x_n]$. This vector can contain mixed variables such as integer, discrete and continuous ones. Each x_k, $k = 1,\ldots,n$ can be bounded by lower and upper limits $L_i \leqslant x_k \leqslant U_k$ (box constraints); $h_i(\vec{x})$ and $g_i(\vec{x})$ are called equality and inequality constraints, being their number m and p, respectively. Both kinds of constraint can be linear or nonlinear.

Over the years, several meta-heuristics have been proposed to solve constrained problems and find global optimum solutions. Among these, Evolutionary Algorithms, such as Genetic Algorithms (GA), see (Coello Coello, 2000; Coello Coello & Mezura-Montes, 2002; Michalewicz, 1996), and Evolution Strategies (ES), see (Mezura-Montes & Coello Coello, 2005), have attracted most of the research interest. Swarm Intelligence has also been investigated as tool for solving constrained optimization problems, see for instance the studies on Particle Swarm Optimization (PSO) (Aguirre, Muñoz Zavala, Villa Diharce, & Botello Rionda, 2007; Cagnina, Esquivel, & Coello, 2008; Coelho, 2010; He & Wang, 2007; Pant, Thangaraj, & Abraha, 2009; Hu, Eberhart, & Shi, 2003), Cuckoo Search (Yang & Deb, 2010), or Artificial Bee Colonies (Brajevic, Tuba, & Subotic, 2011). Another method which is becoming increasingly popular in this field is Differential Evolution (DE): in the last decade, several variants of DE specifically designed for constrained optimization have been proposed, see the papers (Mezura-Montes, Coello Coello, Tun-Morales, Computación, & Tabasco, 2004, ; Mezura-Montes & Palomeque-Ortiz, 2009a, 2009b), or the most recent works (de Melo & Carosio, 2012; Mohamed & Sabry, 2012; Wang & Cai, 2011).

In the excellent comparative study proposed in Mezura-Montes and Lopez-Ramirez (2007), DE outperformed PSO, GA, ES on a broad set of numerical and engineering constrained optimization problems, although it was also observed that PSO tends to reach the feasible region faster, while ES is very effective in improving upon feasible solutions previously found. Motivated by this result, recent studies focused on hybrid techniques, in the attempt of overcoming the specific limitations of each meta-heuristic: for example, Liu, Cai, and Wang (2010) embedded three mutation rules typical of DE (rand/1, current-to-best/1, and rand/2) into a

* Corresponding author. Tel.: +55 12 3309 9576; fax: +55 12 3309 9500.
E-mail addresses: vinicius.melo@unifesp.br (V.V. de Melo), giovanniiacca@incas3.eu (G. Iacca).

http://dx.doi.org/10.1016/j.eswa.2014.06.032
0957-4174/

PSO framework to force it jump out of stagnation; in a similar way, He and Wang (2007) hybridized PSO with Simulated Annealing, which was used as local-search to improve upon the best solution of the swarm. Similar ideas were investigated in Sun and Garibaldi (2010) and Ullah et al. (2007), where different kinds of memetic algorithms were introduced, combining a global search scheme with one or more local-search methods. More recently, Pescador Rojas and Coello Coello (2012) presented instead an agent based memetic algorithm including a co-evolution approach.

One of the most popular—and perhaps intuitive—constraint-handling technique known from the literature is the use of a penalty function which worsens the fitness of unfeasible functions obtained during the search (Michalewicz, 1996). According to this strategy, feasible solutions (i.e., solutions belonging to the feasible set, in the following denoted as F) are simply evaluated based on their function value (i.e. the original objective function), while a penalty function is applied to unfeasible solutions to artificially generate a poor fitness. The above scheme can be formulated as:

$$f(\vec{x}) = \begin{cases} objfun(\vec{x}) & if\, \vec{x} \in F \\ objfun(\vec{x}) + \text{PENALTY}(\vec{x}) & otherwise \end{cases} \tag{2}$$

where $objfun(\vec{x})$ and $penalty(\vec{x})$ obviously indicate the objective and penalty function, respectively. Despite its simplicity, the main drawback of this method lies in the determination of suitable penalty functions, which in turn affects the efficiency of the search. Several specific mechanisms have been designed to solve this issue, see for example (Coelho, 2010; Coello Coello & Mezura-Montes, 2002; He & Wang, 2007; Pant et al., 2009). Other studies focused on self-adapting schemes, see (Coello Coello, 2000; Mezura-Montes, Coello Coello, & Velázquez-Reyes,; Michalewicz, 1996; Noman & Iba, 2008), where the penalty function is adapted during the optimization process. An alternative approach, called filter-based EA (Clevenger, Ferguson, & Hart, 2005), aggregates the constraint violations and tries to solve an equivalent bi-objective problem, minimizing both the original objective function and the aggregated constraint violation function.

It can be easily seen that all these works have in common the underlying idea that population-based meta-heuristics, due to their parallel nature, intrinsically form an efficient tool for solving constrained optimization problems. On the other hand, in order to apply population-based algorithms to this kind of problems, it is imperative to endow them of proper ways to handle constraints, so to avoid unwanted situations such as stagnation and premature convergence, and guide the optimization process towards (and within) the bounds of the feasible region (s) of the search-space.

In this paper, we investigate a well-known meta-heuristic which has proven extremely successful on unconstrained problems, being yet barely studied in constrained optimization, namely the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen, Müller, & Koumoutsakos, 2003). The choice of looking at CMA-ES as tool for constrained optimization is mostly due to the fact that this algorithm represents the current state-of-the-art in optimization, although very few studies focus on its application to constrained problems. Therefore we believe that it is of great interest trying to exploit the powerful features of robustness of CMA-ES also in constrained contexts. To overcome the limitations of CMA-ES on constrained problems, (1) we modify the basic CMA-ES framework by applying some simple changes in its stop criteria and sampling mechanism and (2) we introduce an adaptive penalty function within the algorithm, which efficiently handles constraints and dynamically adjusts the penalty during the optimization. In our view, this adaptive penalty scheme represents the main contribution of this paper. Interestingly, this penalty scheme could be applied also to meta-heuristics other than CMA-ES. However, since our purpose is to extend the range of applications of CMA-ES to constrained optimization, here we prove the applicability of this penalty function only on CMA-ES, and we assess the advantages due to its use. Extensive numerical experiments and comparisons with state-of-the-art methods complete this paper, where we evaluate the performance of the proposed modified CMA-ES with adaptive penalty function on a broad set of benchmark problems and engineering applications take from specialized literature.

The paper is organized as follows: in Section 2, we briefly introduce the basic CMA-ES algorithm. Section 3 describes the proposed algorithm based on modified CMA-ES, with particular focus on the adaptive penalty function. Section 4 includes the description of the test problems, the details of the experiments, the results obtained, and the related discussion. Finally, in Section 5 concludes this work.

2. Covariance Matrix Adaptation Evolution Strategy

The Covariance Matrix Adaptation Evolution Strategy, introduced in Hansen et al. (2003), is a variant of classic Evolution Strategies (ES) (Rechenberg, 1971; Schwefel, 1965) which makes use of a distribution model of the population in order to learn the variable linkages and speed up the evolutionary process. CMA-ES consists of the following. At the beginning of the optimization, a mean vector $\vec{m} \in \mathbb{R}^n$ is randomly initialized inside the problem bounds $L_j \leqslant m_j \leqslant U_j,\ for\ j = 1, \ldots, n$, where n is the number of variables of the problem; additionally, a covariance matrix $cov = \sigma^2 C$ is defined, where $C \in \mathbb{R}^{n \times n}$ is initially set to I, the identity matrix, and σ is the initial step size, a parameter of the algorithm. After the initialization, each step of the algorithm first samples λ new solutions from the multivariate normal distribution $\mathcal{N}(\vec{m}, cov)$; then, $\vec{m}, \sigma$ and C are adaptively updated from a weighted sum of the best μ solutions in the population[1]. As a broadly accepted rule of thumb (see Hansen & Ostermeier (2001)) λ and μ are set, respectively, to $4 + \lfloor 3 \ln(n) \rfloor$ and $\lfloor \lambda/2 \rfloor$, while the initial value of σ is set, in general, to 0.5. Once the distribution is updated, the loop is repeated until a stop condition is met. In the original implementation (Hansen et al., 2003), the optimization is stopped when one of these situations occurs:

- a maximum iteration is reached;
- a stop fitness value is reached;
- the covariance matrix C is numerically not positive definite;
- all the standard deviations are smaller than a given tolerance (1E-12).

For the sake of completeness, we report a simplified pseudo-code illustrating the main steps of CMA-ES, see Algorithm 1. A detailed description of the distribution model update mechanism, which is out of the scope of this paper, can be found in Hansen et al. (2003) and Hansen and Ostermeier (1996, 2001).

Algorithm 1. Basic pseudo-code of CMA-ES

```
initialize covariance matrix C = I, and step-size σ
initialize the mean vector m⃗ to a random solution within the
   problem bounds
while CMA-ES stop criteria are not met do
   sample λ new individuals from distribution N(m⃗, σ²C)
   evaluate individuals and sort them based on their fitness
   update m⃗ based on a weighted sum of the best μ individuals
   update covariance matrix C and step-size σ
end while
return best individual x⃗_best
```

[1] It is interesting to note that this update mechanism, named $(\mu/\mu_w, \lambda)$ (being μ_w the weights on the best μ individuals used to update the distribution model), differs from the classic (μ, λ) or $(\mu + \lambda)$ survivor selection schemes used in ES.