Swarm and Evolutionary Computation 18 (2014) 22–37

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

# A hybrid particle swarm with a time-adaptive topology for constrained optimization

Mohammad Reza Bonyadi *, Xiang Li, Zbigniew Michalewicz

School of Computer Science, The University of Adelaide, Australia

ARTICLE INFO

Article history:
Received 25 January 2014
Received in revised form
3 May 2014
Accepted 7 June 2014
Available online 16 June 2014

Keywords:
Particle swarm optimization
Continuous space optimization
Constrained optimization problems
Disjoint feasible regions

ABSTRACT

For constrained optimization problems set in a continuous space, feasible regions might be disjointed and the optimal solution might be in any of these regions. Thus, locating these feasible regions (ideally all of them) as well as identifying the most promising region (in terms of objective value) at the end of the optimization process would be of a great significance. In this paper a time-adaptive topology is proposed that enables a variant of the particle swarm optimization (PSO) to locate many feasible regions at the early stages of the optimization process and to identify the most promising one at the latter stages of the optimization process. This PSO variant is combined with two local searches which improve the ability of the algorithm in both finding feasible regions and higher quality solutions. This method is further hybridized with covariance matrix adaptation evolutionary strategy (CMA-ES) to enhance its ability to improve the solutions at the latter stages of the optimization process. Results generated by this hybrid method are compared with the results of several other state-of-the-art methods in dealing with standard benchmark constraint optimization problems.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

A constrained optimization problem (COP) in a continuous space is formulated as follows:

$$\text{Find } x \in S \subseteq R^D \text{ such that} \begin{cases} \forall y \in \mathcal{F}\ f(x) \le f(y) & (a) \\ g_i(x) \le 0, \quad \text{for } i=1 \text{ to } q & (b) \\ h_i(x) = 0, \quad \text{for } i=1 \text{ to } p & (c) \end{cases} \tag{1}$$

In this formulation, $f$, $g_i$, and $h_i$ are real-valued functions defined on the search space $S$, $q$ is the number of inequalities, and $p$ is the number of equalities. The search space $S$ is defined as a $D$ dimensional rectangle in $R^D$ such that $l(j) \le x(j) \le u(j)$, $j=1,\ldots, D$ ($l(j)$ and $u(j)$ are the lower and upper bounds of the $j$th variable). The set of all feasible points which satisfy constraints (b) and (c) are denoted by $\mathcal{F}$ [1]. Usually in COPs, equalities are replaced by inequalities [2] as follows:

$$|h_i(x)| \le \xi, \quad \text{for } i=1 \text{ to } p \tag{2}$$

where $\xi$ is a small positive value. In all experiments reported in this paper, the value of $\xi$ is equal to $1E-4$, the same as it was adopted in [2,3]. Accordingly, by considering $g_{i+q}(x) = |h_i(x)| - \xi$ for all $1 \le i \le p$, the COP defined in Eq. 1 can be written as

$$\text{Find } x \in S \subseteq R^D \text{ such that} \begin{cases} \forall y \in \mathcal{F}\ f(x) \le f(y) & (a) \\ g_i(x) \le 0, \quad \text{for } i=1 \text{ to } q+p & (b) \end{cases} \tag{3}$$

From now on, the term COP refers to this formulation.

Any method that deals with a COP consists of two parts: an optimization algorithm and a constraint handling technique (CHT). The optimization algorithm can be particle swarm optimization (PSO) [4], genetic algorithm (GA) [5], differential evolution (DE) [6], covariance matrix adaptation evolutionary strategy (CMA-ES) [7], conjugate gradient [8], linear programming [9], etc. However, whatever the optimization algorithm is, evaluation of individuals is one of the challenges in solving COPs [10]. Indeed, unlike unconstrained optimization problems in where evaluation is simply done based on the value of the objective function for each individual, evaluation procedure for COPs includes some complexities because it is necessary to consider both constraints and objective value (see [11] for detailed discussion on this complexity). There are several categories of techniques in handling constraints that can be incorporated into optimization algorithms [12]; these categories include: penalty functions, special operators, repairs, decoders, and hybrid techniques (see also [1] and [10] for details).

Particle Swarm Optimization (PSO) [13] is a population based optimization algorithm of $n > 1$ particles (referred to as *swarm*); each particle is defined by three $D$-dimensional vectors

* Corresponding author.
E-mail addresses: mbonyadi@cs.adelaide.edu.au, vardiar@gmail.com (M. Reza Bonyadi), xiang.li01@cs.adelaide.edu.au (X. Li), zbyszek@cs.adelaide.edu.au (Z. Michalewicz).

http://dx.doi.org/10.1016/j.swevo.2014.06.001
2210-6502/© 2014 Elsevier B.V. All rights reserved.

- *Position* ($\vec{x}^i_t$) – is the position of the $i$th particle in the $t$th iteration. This is used to evaluate the particle's quality.
- *Velocity* ($\vec{v}^i_t$) – is the direction and length of movement of the $i$th particle in the $t$th iteration.
- *Personal best* ($\vec{p}^i_t$) – is the best position[1] that the $i$th particle has visited in its lifetime (up to the $t$th iteration). This vector serves as a memory for keeping knowledge of quality solutions [4].

All of these vectors are updated at every iteration $t$ for each particle ($i$)

$$\vec{v}^i_{t+1} = \mu(\vec{x}^i_t, \vec{v}t^i, N^i_t) \text{ for all } i \tag{4}$$

$$\vec{x}^i_{t+1} = \xi(\vec{x}^i_t, \vec{v}^i_{t+1}) \text{ for all } i \tag{5}$$

In Eq. 4, $N^i_t$ (known as neighbor set of the particle $i$ at iteration $t$) is a subset of personal best positions of some particles which contribute to the velocity updating rule of that particle at iteration $t$, i.e. $N^i_t = \{\vec{p}^k_t | k \in \{T^i_t \subseteq \{1, 2, \ldots, n\}\}\}$ where $T^i_t$ is a set of indices of particles which contribute to the velocity updating for particle $i$ at iteration $t$. Clearly, the strategy of determining $T^i_t$ might be different for various types of PSO algorithms and it is usually referred to as the *topology* of the swarm. Many different topologies have been defined so far [14], e.g., global best topology (gbest), ring topology, non-overlapping, pyramid, and adaptive topology, that are discussed later in this paper. The function $\mu(.)$ calculates the new velocity vector for particle $i$ according to its current position, current velocity $\vec{v}^i_t$, and neighborhood set $N^i_t$. In Eq. 5, $\xi(.)$ is a function which calculates the new position of the particle $i$ according to its previous position and its new velocity. Usually $\xi(\vec{x}^i_t, \vec{v}^i_{t+1}) = \vec{x}^i_t + \vec{v}^i_{t+1}$ is accepted for updating the position of particle $i$. After updating velocity and positions, the personal best vector ($\vec{p}^i_t$) of the particles is also updated.

$$\vec{p}^i_{t+1} = \begin{cases} \vec{p}^i_t & f(\vec{p}^i_t) \le f(\vec{x}^i_{t+1}) \\ \vec{x}^i_{t+1} & \text{otherwise} \end{cases} \tag{6}$$

In Eq. (6), the new personal best position for the $i$th particle is updated according to the objective values of its previous personal best and the current position. In the rest of this paper, these usual forms for the position updating rule (Eq. (5)) and for updating the personal best (Eq. (6)) are assumed. In PSO, updating rules (Eqs. (4) and (5)) are applied to all particles and the personal best for all particles are updated in each iteration until a predefined termination criterion, e.g., maximum number of iterations or deviation from global optimum (if known), is met.

In the original version of PSO [13], the function $\mu(\cdot)$ in Eq. (4) was defined as

$$\vec{v}^i_{t+1} = \vec{v}^i_t + \varphi_1 R^i_{1t} \underbrace{(\vec{p}^i_t - \vec{x}^i_t)}_{\substack{\text{Personal} \\ \text{Influence (PI)}}} + \varphi_2 R^i_{2t} \underbrace{(\vec{g}_t - \vec{x}^i_t)}_{\substack{\text{Social} \\ \text{Influence (SI)}}} \tag{7}$$

where $\varphi_1$ and $\varphi_2$ are two real numbers known as acceleration coefficients[2] and $\vec{p}^i_t$ and $\vec{g}_t$ are the personal best (of particle $i$) and the global best vectors, respectively, at iteration $t$. Also, the role of vectors $PI = \vec{p}^i_t - \vec{x}^i_t$ (Personal Influence) and $SI = \vec{g}_t - \vec{x}^i_t$ (Social Influence) is to *attract* the particles to move toward known quality solutions, i.e. personal and global best. Moreover, $R_{1t}$ and $R_{2t}$ are two $d \times d$ diagonal matrices[3] [15,16], where their elements are random numbers distributed uniformly ($\sim U(0,\ 1)$) in $[0, 1]$. Note that matrices $R_{1t}$ and $R_{2t}$ are generated at each iteration for each particle separately.

In 1998, Shi and Eberhart [17] introduced a new coefficient $\omega$ (known as *inertia weight*) to control the influence of the previous velocity value on the updated velocity. Indeed, Eq. 7 was written as

$$\vec{v}^i_{t+1} = \omega \vec{v}^i_t + \varphi_1 R^i_{1t}(\vec{p}^i_t - \vec{x}^i_t) + \varphi_2 R^i_{2t}(\vec{g}_t - \vec{x}^i_t) \tag{8}$$

The coefficient $\omega$ controls the influence of the previous velocity ($\vec{v}^i_t$) on the movement of the particle (this variant is called Standard PSO, SPSO, throughout this paper). One of the issues in SPSO was that, for some values of the coefficients, velocity may grow to infinity. Some studies analyzed the dynamic of the particles to understand why velocity might grow to infinity. It was proven that by setting the coefficients in specific boundaries, velocity shrinks during the time and hence, it does not grow to infinity [18–20]. In SPSO, if the random matrices are replaced by random values then the new variant is known as linear PSO (LPSO).

There are several well-studied issues in the standard PSO such as stagnation [21–24], line search [25,26], and swarm size [21,22]. Apart from these issues in PSO, there have been some attempts to extend the algorithm to work with COPs [3,27–38] and to support niching[4] [39–42]. See Section 2 for a brief review on the issues and extensions of SPSO.

In this paper, different topologies for a PSO variant proposed in our earlier paper [11] are analyzed and their abilities in locating disjoint feasible regions of a COP are tested. Consequently, this variant is extended by a new time-adaptive topology which enables the algorithm to locate feasible regions at the early stages of iterations and to find the region with the highest quality (in terms of the objective function) at the latter stages of the optimization process. Also, this extended method is combined further with two local searches and a covariance matrix adaptation evolutionary strategy (CMA-ES) [43] to improve the quality of the found solutions. The hybrid approach is applied to standard benchmark COPs (usually known as CEC2010 [44]) and its results are compared with three other recently proposed approaches [2,45,46].

The rest of this paper is organized as follows. Section 2 provides an overview of PSO including discussion on some identified issues of this technique, its topology, niching capabilities, and its applicability for COPs. Section 3 discusses two constraint handling methods as well as some relevant optimization methods to deal with COPs. In Section 4 a PSO variant is extended by a new time adaptive topology and the extended method is combined with local searches. Experimental results are reported and analyzed in Section 5 and Section 6 concludes the paper.

## 2. Particle swarm optimization

In this section we provide an overview of PSO, including issues in the algorithm, topology, niching abilities, and its ability to deal with COPs.

[1] In general, personal best can be a *set* of best positions, but all PSO types listed in this paper use single personal best.

[2] These two coefficients control the effect of personal and global best vectors on the movement of particles and they play an important role in the convergence of the algorithm. They are usually determined by a practitioner or by the dynamic of particles' movement.

[3] Alternatively, these two random matrices are often considered as two random vectors. In this case, the multiplication of these random vectors by *PI* and *SI* is element-wise.

[4] Niching is the ability of the algorithm to locate different optima rather than only one of them. The niching concept is used usually in the multi-modal optimization.

Download English Version:

https://daneshyari.com/en/article/494031

Download Persian Version:

https://daneshyari.com/article/494031

Daneshyari.com