Swarm and Evolutionary Computation 17 (2014) 37–59

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

Population statistics for particle swarm optimization: Resampling methods in noisy optimization problems

Juan Rada-Vilela [a,b,*], Mark Johnston [a,c], Mengjie Zhang [a,b]

[a] Evolutionary Computation Research Group, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
[b] School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
[c] School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand

ARTICLE INFO

Article history:
Received 19 September 2013
Received in revised form
9 January 2014
Accepted 21 February 2014
Available online 12 March 2014

Keywords:
Particle swarm optimization
Population statistics
Noisy optimization problems
Resampling methods

ABSTRACT

Particle Swarm Optimization (PSO) is a metaheuristic whose performance deteriorates significantly when utilized on optimization problems subject to noise. On these problems, particles eventually fail to distinguish good from bad solutions because their objective values are corrupted by noise. Specifically, the effect of noise causes particles to suffer from *deception* when they do not select their true neighborhood best solutions, from *blindness* when they ignore better solutions, and from *disorientation* when they prefer worse solutions. Resampling methods reduce the presence of these conditions by re-evaluating the solutions multiple times and better estimating their true objective values with a sample mean over the evaluations. PSO with Equal Resampling (PSO-ER) finds better solutions than the regular PSO thanks mainly to the reduction of deception and blindness, as has been found by utilizing a set of *population statistics* that track the presence of these conditions throughout the search process. However, the solutions of PSO-ER have been reported to be worse than those of state-of-the-art resampling-based PSO algorithms, and the underlying reasons are not known because the population statistics for such algorithms have never been computed. In this article, we study the population statistics for a new extension to PSO-ER that further reduces the presence of blindness, and for state-of-the-art resampling-based PSO algorithms. Experiments on 20 large-scale benchmark functions subject to different levels of noise show that our new algorithm succeeds at reducing blindness and finding better solutions than PSO-ER. However, the population statistics for state-of-the-art resampling-based PSO algorithms show that their particles suffer even less from deception, blindness and disorientation, and therefore find much better solutions.

1. Introduction

Particle Swarm Optimization (PSO) is a metaheuristic designed by Eberhart and Kennedy [1,2] to find good solutions to optimization problems. Inspired by social models and swarming theory, it consists of a swarm of particles that collectively explore the solution search space of an optimization problem and stores the best solutions found. Particles are attracted at each iteration to the (personal) best solutions found by themselves and to the (neighborhood) best solutions found by their neighbors, thereby encouraging the exploration of nearby solutions to potentially find better ones. The simplicity of PSO, the quality of its results and other characteristics have encouraged its application to a number of problems in different fields of research [3]. However, a topic that remains largely unexplored is the deterioration of the quality of its results when problems are subject to noise [4].

In optimization problems subject to noise, particles are often unable to correctly distinguish the quality of their solutions because their objective values are corrupted with sampling noise, hence causing particles to suffer from three conditions known as *deception*, *blindness* and *disorientation* [4]. A particle suffers from deception when it selects the estimated best solution within its neighborhood but it is not the true best solution therein, from blindness when it fails to recognize a better solution and hence misses out on an opportunity to improve its personal best one, and from disorientation when it mistakenly replaces its personal best solution with a worse solution. These conditions are responsible for the deterioration of PSO on optimization problems subject to noise, and therefore need to be addressed via noise mitigation mechanisms in order to prevent, or at least reduce, such a deterioration.

One type of noise mitigation mechanism comprises *resampling methods*, which in PSO serve to better estimate the true objective values of the solutions by taking a sample mean over multiple

* Corresponding author.
E-mail addresses: juan.rada-vilela@ecs.vuw.ac.nz (J. Rada-Vilela), mark.johnston@msor.vuw.ac.nz (M. Johnston), mengjie.zhang@ecs.vuw.ac.nz (M. Zhang).

http://dx.doi.org/10.1016/j.swevo.2014.02.004
2210-6502/

re-evaluations. As such, the more times a solution is re-evaluated, the more accurate its objective value will be. However, since these evaluations are taken from a fixed and limited computational budget of evaluations available to the algorithm, improving the accuracy of the solutions sacrifices the number of iterations to perform. In spite of such a tradeoff, the most basic resampling-based PSO algorithm, namely PSO with Equal Resampling (PSO-ER), finds better solutions than the regular PSO thanks to the better estimated objective values of the current solutions in the swarm [4]. However, the quality of its results is generally worse than state-of-the-art resampling-based PSO algorithms such as PSO with Optimal Computing Budget Allocation (PSO-OCBA) [5–7] and PSO with Equal Resampling and Allocations to the Top-*N* Solutions (PSO-ERN) [8], hence failing to compete against them.

The *population statistics* proposed in [4] showed that PSO-ER finds better solutions than the regular PSO mostly because its particles suffer less often from deception, blindness and disorientation, and yet these conditions are still present in 92.74%, 36.13% and 2.20% of the iterations, respectively. To reduce the extents to which these conditions affect particles in PSO-ER, we propose a new PSO with Extended Equal Resampling (PSO-EER) in which the personal best solutions are also re-evaluated, thereby improving the accuracy of their objective values and hence directly reducing blindness and deception which are the most common causes of deterioration in PSO-ER. In doing so, we expect PSO-EER to find better solutions than PSO-ER, but we are not certain how the quality of its results will compare against state-of-the-art resampling-based PSO algorithms because their population statistics have never been computed.

The overall goal of this article is to study the population statistics for the newly proposed PSO-EER and for state-of-the-art resampling-based PSO algorithms on optimization problems subject to different levels of multiplicative Gaussian noise. Specifically, we will focus on the population statistics for PSO-EER, PSO-ER, PSO-OCBA, and PSO-ERN to find and compare the underlying characteristics of the quality of their results.

The remainder of this article is structured as follows. Section 2 provides some background on PSO, optimization problems subject to noise, population statistics for PSO, and related work. Section 3 presents PSO with resampling methods, the state of the art in resampling-based PSO algorithms, and the new PSO-EER algorithm that we propose. Section 4 describes the design of experiments. Section 5 presents the results and discussions. Finally, Section 6 presents the conclusions and suggestions for future work.

2. Background

2.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) [1,2] is an iterative algorithm in which a swarm of particles explores the solution search space of an n-dimensional optimization problem to find good solutions. Each particle i at iteration t is made up of three n-dimensional vectors that represent its position $\mathbf{x}_i^t$, its velocity $\mathbf{v}_i^t$, and its memory $\mathbf{y}_i^t$. The position encodes a potential solution to the problem at hand, the velocity changes the position to explore the solution search space, and the memory stores the (personal) best position found. Every time a particle finds a position that is better than its personal best one, the memory is replaced with such a position. Afterwards, each particle selects the best solution within its neighborhood and refers to it as $\hat{\mathbf{y}}_i^t$. The personal and neighborhood best positions are utilized to update the velocity of each particle and hence change its position at the next iteration according to Eqs. (1) and (2), respectively,

```
foreach iteration t do
    foreach particle i in swarm do
        if  f̃(x_i^t) < f̃(y_i^{t-1})
        then  y_i^t ← x_i^t else y_i^t ← y_i^{t-1}
    foreach particle i in swarm do
        ŷ_i^t ← arg min_{y^t ∈ N_i} f̃(y^t)
    foreach particle i in swarm do
        update v_i^{t+1} and x_i^{t+1} using (1) and (2)
```

Fig. 1. Particle swarm optimization.

$$v_{ij}^{t+1} = wv_{ij}^t + c_1 r_{1j}^t (y_{ij}^t - x_{ij}^t) + c_2 r_{2j}^t (\hat{y}_{ij}^t - x_{ij}^t) \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (2)$$

where w is the inertia weight of the particle [9], c_1 and c_2 are positive coefficients that determine the influence of the personal and neighborhood best positions, r_{1j}^t and r_{2j}^t are random values sampled from a uniform distribution $U(0,1)$, y_{ij}^t is the value of dimension j of the best position found by particle i, and $\hat{y}_{ij}^t$ is the value of dimension j of the best position found by any particle in i's neighborhood $\mathcal{N}_i$. These equations are utilized within the PSO algorithm for a minimization problem as described in Fig. 1, where $\tilde{f}(\mathbf{x})$ is the objective value of solution $\mathbf{x}$.

The neighborhoods in the swarm are determined by the network topology that connects the particles, thereby defining the extent to which particles communicate their personal best positions. The most commonly used topologies are the *star* and the *ring* [10], in which the former defines a single neighborhood to which all particles belong, and the latter defines multiple overlapping neighborhoods that connect each particle to two adjacent neighbors. As such, the star topology favors exploitation because the swarm is partially attracted to a single neighborhood position, whereas the ring topology favors exploration because neighborhoods of particles are partially attracted towards different positions [11,12].

2.2. Optimization problems subject to noise

Noise in optimization problems is common in the real world when variables are affected by imprecise measurements, modeled by probability distributions, or just corrupted by external factors such as communication errors. In this type of problem, noise corrupts the objective values of the solutions each time these are evaluated, hence resulting in different values every time. In controlled experiments, this type of uncertainty can be modeled as sampling noise from a Gaussian distribution [13], thereby corrupting the objective values with an additive (3) or multiplicative (4) effect as follows:

$$\hat{f}_+(\mathbf{x}) = f(\mathbf{x}) + N(0, \sigma^2) \quad (3)$$

$$\hat{f}_\times(\mathbf{x}) = f(\mathbf{x}) \times N(1, \sigma^2) \quad (4)$$

where $N(\mu, \sigma^2)$ is a random value sampled from a Gaussian distribution with mean μ and standard deviation σ. Hereinafter, the true objective value of solution $\mathbf{x}$ is represented as $f(\mathbf{x})$, a single noisy evaluation of solution $\mathbf{x}$ is represented as $\hat{f}(\mathbf{x})$, and the estimated objective value of solution $\mathbf{x}$ is represented as $\tilde{f}(\mathbf{x})$. Thus, in the absence of noise, $f(\mathbf{x}) = \hat{f}(\mathbf{x}) = \tilde{f}(\mathbf{x})$.

The severity of noise is determined by the standard deviation σ and by the effect of noise on the objective values. Optimization under multiplicative noise is very challenging because, unlike additive noise, its effect on the objective values is larger and