Swarm and Evolutionary Computation 22 (2015) 15–29

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

Population statistics for particle swarm optimization: Hybrid 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 7 September 2014
Received in revised form
23 November 2014
Accepted 22 January 2015
Available online 17 February 2015

Keywords:
Particle swarm optimization
Population statistics
Noisy optimization problems
Hybrid methods
Resampling methods
Single-evaluation methods

ABSTRACT

Particle swarm optimization (PSO) is a metaheuristic designed to find good solutions to optimization problems. However, when optimization problems are subject to noise, the quality of the resulting solutions significantly deteriorates, hence prompting the need to incorporate noise mitigation mechanisms into PSO. Based on the allocation of function evaluations, two opposite approaches are generally taken. On the one hand, *resampling-based* PSO algorithms incorporate resampling methods to better estimate the objective function values of the solutions at the cost of performing fewer iterations. On the other hand, *single-evaluation* PSO algorithms perform more iterations at the cost of dealing with very inaccurately estimated objective function values. In this paper, we propose a new approach in which *hybrid* PSO algorithms incorporate noise mitigation mechanisms from the other two approaches, and the quality of their results is better than that of the state of the art with a few exceptions. The performance of the algorithms is analyzed by means of a set of *population statistics* that measure different characteristics of the swarms throughout the search process. Amongst the hybrid PSO algorithms, we find a promising algorithm whose simplicity, flexibility and quality of results questions the importance of incorporating complex resampling methods into state-of-the-art PSO algorithms.

1. Introduction

Particle swarm optimization (PSO) is a metaheuristic where a swarm of particles explores the search space of an optimization problem to find good solutions. Designed by Eberhart and Kennedy [1,2], it takes inspiration from swarming theory [3] and social models [4] by having particles interact with each other in order to improve the quality of their solutions. Each particle has a *position* that encodes a potential solution to the problem at hand, a *velocity* that will change the position of the particle at the next iteration, and a memory to remember where the particle found the best solution. Particles start at random positions and iteratively adjust their velocities such that they become partially attracted towards the positions of the best solutions found by themselves and their neighbors. At each step, particles evaluate their newly found positions and decide whether to store them in memory replacing previous findings. This is the regular PSO algorithm that has been adapted to address many optimization problems in different fields of research [5–9].

In optimization problems subject to noise, the performance of PSO is an aspect that has not been as thoroughly studied as the performance of other metaheuristics like genetic algorithms [10–13] and evolution strategies [14,15]. In this type of problems, the objective function values that determine the quality of the solutions are corrupted by the effect of sampling noise, hence resulting in differently *estimated* objective function values every time the solutions are evaluated. As a consequence, particles eventually fail to distinguish good from bad solutions, leading to three conditions known as *deception*, *blindness* and *disorientation* [16]. Particles suffer from deception when they fail to select their true neighborhood best solutions, from blindness when they ignore truly better solutions, and from disorientation when they prefer truly worse solutions.

The deterioration of the quality of the results found by PSO on optimization problems subject to noise prompts the need to incorporate noise mitigation mechanisms in order to prevent (or at least reduce) such a deterioration. Based on the use of the computational budget of function evaluations, the literature has distinguished two conceptually different approaches to mitigate the effect of noise on PSO. On the one hand, *resampling-based* PSO

* 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.2015.01.003
2210-6502/

algorithms [17] allocate multiple function evaluations to the solutions in order to better estimate their objective function values by a sample mean over the evaluations [18–21]. On the other hand, *single-evaluation* PSO algorithms [22] do not allocate additional function evaluations to the solutions and focus instead on reducing the effect of having solutions with very inaccurately estimated objective function values [23–27]. Furthermore, since the computational budget is fixed and limited, resampling-based and single-evaluation PSO algorithms present opposite tradeoffs: resampling-based PSO algorithms better estimate the objective function values of the solutions at the cost of performing fewer iterations, whereas single-evaluation PSO algorithms perform more iterations at the cost of dealing with solutions whose objective function values are very inaccurately estimated.

Recently, resampling-based and single-evaluation PSO algorithms have been studied by means of a set of *population statistics* that measure different characteristics of the swarms throughout the search process [16,17,22,28]. The population statistics have revealed that swarms often suffer from deception, blindness and disorientation, for which different algorithms have been designed to reduce the presence of these conditions and hence improve the quality of their resulting solutions. While previous works have focused exclusively on either resampling-based [17] or single-evaluation [22] PSO algorithms, in this article we perform a direct comparison between their population statistics. More importantly, in spite of the opposite tradeoffs of resampling-based and single-evaluation PSO algorithms, we merge their noise mitigation mechanisms into a new group of *hybrid* PSO algorithms. In doing so, we expect that the joint efforts of noise mitigation mechanisms in the new hybrid PSO algorithms will lead to a better quality of results than the purely resampling-based and single-evaluation PSO algorithms, respectively.

The overall goal of this paper is to study the population statistics for new hybrid PSO algorithms and compare them against state-of-the-art resampling-based and single-evaluation PSO algorithms on optimization problems whose objective functions are subject to different levels of multiplicative Gaussian noise. Specifically, we will focus on the following objectives:

- Merge noise mitigation mechanisms from single-evaluation and resampling-based PSO algorithms into different hybrid PSO algorithms.
- Study the population statistics for the new hybrid PSO algorithms.
- Contrast the population statistics for the new hybrid PSO algorithms against the population statistics for the respective resampling-based and single-evaluation PSO algorithms.
- Contrast the population statistics of resampling-based and single-evaluation PSO algorithms.

foreach particle i **in** swarm $\mathcal{S}$ **do**
  initialize $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{v}_i)$
foreach iteration t **do**
  foreach particle i **in** swarm $\mathcal{S}$ **do**
    if $f(\mathbf{x}_i^t) < f(\mathbf{y}_i^{t-1})$
    then $\mathbf{y}_i^t \leftarrow \mathbf{x}_i^t$
    else $\mathbf{y}_i^t \leftarrow \mathbf{y}_i^{t-1}$
  foreach particle i **in** swarm $\mathcal{S}$ **do**
    $\hat{\mathbf{y}}_i^t \leftarrow \mathbf{y}_\omega^t \mid \omega = \underset{j \in \mathcal{N}_i}{\arg\min} f(\mathbf{y}_j^t)$
  foreach particle i **in** swarm $\mathcal{S}$ **do**
    foreach dimension j **in** particle i **do**

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

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

Fig. 1. Particle swarm optimization for a minimization problem.

The remainder of this paper 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 the new group of hybrid PSO algorithms. 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) is a metaheuristic designed by Kennedy and Eberhart [1,2] with inspiration from swarming theory [3] and social models [4]. It consists of a population of individuals that collectively explore the search space of an optimization problem to find good solutions. The population is referred to as a *swarm*, the individuals are referred to as *particles*, and the collective behavior results from the interactions between the particles. Specifically, each particle i consists of a position vector $\mathbf{x}_i^t$ at iteration t that encodes a solution to the problem, a velocity vector $\mathbf{v}_i^t$ to change $\mathbf{x}_i^t$ in order to explore new solutions, and a memory vector $\mathbf{y}_i^t$ to store the personal best position found. In addition, particle i belongs to a neighborhood of particles $\mathcal{N}_i$ from which the neighborhood best position $\hat{\mathbf{y}}_i^t$ is selected.

The PSO algorithm is presented in Fig. 1 for a minimization problem, where $f(\mathbf{x})$ is the objective function value of the solution represented by position $\mathbf{x}$. Usually, for each particle i, the position $\mathbf{x}_i$ is initialized with random values sampled from a uniform distribution $U(\mathbf{x}_{\min}, \mathbf{x}_{\max})$, where $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are the boundaries of the optimization problem; the velocity $\mathbf{v}_i$ is initialized to a null vector; and the position $\mathbf{y}_i$ is initialized to an empty vector whose $f(\mathbf{y}_i) = \infty$. Eqs. (1) and (2), v_{ij}^{t+1} and x_{ij}^{t+1}, refer to the values of velocity and position (respectively) of particle i at dimension j for the next iteration, w refers to the inertia coefficient [29], c_1 and c_2 are positive acceleration 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 personal best position found by particle i, and $\hat{y}_{ij}^t$ is the value of dimension j of the neighborhood best position selected by particle i. Hereinafter, we refer to the positions of the particles mostly as *solutions*.

The network topology of the swarm defines the neighborhoods to which particles belong, thereby establishing links between the particles from which they can select their neighborhood best solutions. The most commonly used topologies are the *ring* and the *star* [30], but others have also been proposed in the literature [31]. On the one hand, the ring topology defines each neighborhood $\mathcal{N}_i$ as the set of n particles adjacent to i, usually with $n=2$. On the other hand, the star topology defines each neighborhood $\mathcal{N}_i$ as the entire set of particles in the swarm, for which the star topology is equivalent to the ring topology when $n = |\mathcal{S}|$. Therefore, the network topology influences the quality of the neighborhood best solutions as well as the diversity of the solutions in the swarm. Specifically, the ring topology encourages exploration as more particles are partially attracted towards different neighborhood solutions, whereas the star topology encourages exploitation as more particles are partially attracted towards the same neighborhood solutions [31–33].

2.2. Optimization problems subject to noise

Optimization problems subject to noise are a type of problem in which the objective function values of the solutions are corrupted by the effect of noise. As such, the objective function values of the