Applied Soft Computing 49 (2016) 313–334

Contents lists available at ScienceDirect

Applied Soft Computing

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

Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems

Andrea Serani [a,b], Cecilia Leotardi [a], Umberto Iemma [b], Emilio F. Campana [a], Giovanni Fasano [c], Matteo Diez [a,*]

[a] CNR-INSEAN, National Research Council—Marine Technology Research Institute, Rome, Italy
[b] Department of Mechanical and Industrial Engineering, Roma Tre University, Rome, Italy
[c] Department of Management, Ca' Foscari University of Venice, Venice, Italy

ARTICLE INFO

Article history:
Received 13 November 2015
Received in revised form 15 July 2016
Accepted 12 August 2016
Available online 20 August 2016

Keywords:
Simulation-based design
Derivative-free optimization
Global optimization
Particle swarm optimization
Ship hydrodynamics optimization

ABSTRACT

Deterministic optimization algorithms are very attractive when the objective function is computationally expensive and therefore the statistical analysis of the optimization outcomes becomes too expensive. Among deterministic methods, deterministic particle swarm optimization (DPSO) has several attractive characteristics such as the simplicity of the heuristics, the ease of implementation, and its often fairly remarkable effectiveness. The performances of DPSO depend on four main setting parameters: the number of swarm particles, their initialization, the set of coefficients defining the swarm behavior, and (for box-constrained optimization) the method to handle the box constraints. Here, a parametric study of DPSO is presented, with application to simulation-based design in ship hydrodynamics. The objective is the identification of the most promising setup for both synchronous and asynchronous implementations of DPSO. The analysis is performed under the assumption of limited computational resources and large computational burden of the objective function evaluation. The analysis is conducted using 100 analytical test functions (with dimensionality from two to fifty) and three performance criteria, varying the swarm size, initialization, coefficients, and the method for the box constraints, resulting in more than 40,000 optimizations. The most promising setup is applied to the hull-form optimization of a high speed catamaran, for resistance reduction in calm water and at fixed speed, using a potential-flow solver.

1. Introduction

Particle swarm optimization (PSO) was originally introduced in [1], based on the social-behaviour metaphor of a flock of birds or a swarm of bees searching for food. PSO belongs to the class of heuristic algorithms for single-objective evolutionary derivative-free global optimization. Derivative-free global optimization approaches are often preferred to local approaches when objectives are nonconvex and/or noisy, and when multiple local optima cannot be excluded, as often encountered in simulation-based design (SBD) optimization. The computational burden of global optimization techniques is usually much larger compared to local methods, so that the accuracy of the solution sought often depends on the available computational resources.

Zhang et al. [2] presents a comprehensive survey on the PSO variants and their application in several engineering fields, such as mechanical or chemical. Recent applications of PSO to ship SBD include medium- to high-fidelity hull-form and waterjet design optimization of fast catamarans, by morphing techniques [3,4] and geometry modifications based on Karhunen–Loève expansion (KLE) [5–7], and low- to medium-fidelity optimization of unconventional multi-hull configurations [8]. When global techniques are used in design optimization, with CPU-time expensive solvers, the optimization process is computationally expensive and its effectiveness and efficiency remain an algorithmic and technological challenge. Although complex SBD applications are often solved by metamodels [9,10], their development and assessment require benchmark solutions, with simulations directly connected to the optimization algorithm. These solutions are achieved only if affordable and effective optimization procedures are available.

The original PSO makes use of random coefficients, aiming at sustaining the variety of the swarm dynamics. This property implies that statistically significant results can be obtained only through extensive numerical campaigns. Such an approach can be too expensive in SBD optimization for industrial applications, when CPU-time expensive computer simulations are used directly as

* Corresponding author.
E-mail address: matteo.diez@cnr.it (M. Diez).

http://dx.doi.org/10.1016/j.asoc.2016.08.028
1568-4946/

analysis tools. Furthermore, if the design problem in hand is scheduled within an accurate project planning, time resources might be a tight requirement for the optimization process. For these reasons efficient deterministic approaches (such as deterministic PSO, DPSO) have been developed, and their effectiveness and efficiency in industrial applications in ship hydrodynamics problems have been shown, including comparisons with local methods [11] and random PSO [5]. Moreover, the availability of parallel architectures and high performance computing (HPC) systems has offered the opportunity to extend the original *synchronous* implementation of PSO (SPSO) to CPU-time efficient *asynchronous* methods (APSO), assessed on test functions in [12], and applied in several engineering problems such as multidisciplinary optimization of commercial aircraft [13], biomechanics [14], and swarm robotics [15]. Using distributed computing, synchronous implementations of PSO imply that at iteration $k+1$ the position and velocity of any particle is updated after evaluating the function at *all* the particles positions at iteration k. In an asynchronous implementation of PSO the position and velocity of a particle is possibly based on the fitness value at *a subset of all* particles positions.

The effectiveness and efficiency of PSO for box constrained optimization are significantly influenced by four main setting parameters: (a) the number of swarm particles interacting during the optimization, (b) the initialization of the particles in terms of initial location and velocity, (c) the set of coefficients defining the personal or social behaviour of the swarm dynamics, and (d) the method to handle the box constraints. These parameters and their effects on PSO have been studied by a number of authors [16–18]. More recently the effects of the particle initialization have been studied in [19,20], the effects of the coefficients have been shown in [21], whereas bound handling techniques have been presented in [22]. A comprehensive study on the PSO parameter selection has been presented in [23] and a preliminary assessment of the performances of DPSO, varying (a), (b) and (c), is presented in [24]. A survey of approaches for general constrained optimization problems in industrial design and multidisciplinary design optimization may be found in [25], including also general nonlinear constraints. However, the discussion on the application of DPSO in SBD problems is still limited, lacking a systematic and comparative analysis.

The objective of the present work is the identification of the most effective and efficient parameters for both synchronous and asynchronous deterministic particle swarm optimization (SDPSO and ADPSO), for their use in SBD procedures. The focus is on industrial problems, directly using CPU-time expensive analyses. These make the statistical analysis of the results too expensive and therefore demand for deterministic algorithms. Due to the attractive features of DPSO (such as the simplicity of the heuristics, the ease of implementation, and its often fairly remarkable effectiveness in industrial problems), the current study is limited to DPSO and its implementations. A systematic comparison of DPSO with other deterministic and stochastic methods is beyond the scopes of the present work.

The approach includes a preliminary parametric analysis on 100 analytical test functions [26–29] characterized by different degrees of non-linearities and number of local minima, with full-factorial combination of: (a) number of particles (using a power of two times the number of design variables); (b) initialization of the particle position and velocity (using Hammersley sequence sampling (HSS) [30]); (c) set of coefficients, chosen from literature [12,16,17,31,32]; (d) inelastic and semi-elastic wall-type approach for box constraints [22]. In order to handle the box constraints, wall-type approaches are preferred instead of penalty approaches or Lagrangian functions, which might introduce additional bias in the analysis. The number of design variables ranges from two to fifty and the simulation budget (maximum number of objective function evaluations) is up to 4096 times the number of design variables. The preliminary parametric analysis is conducted on two subsets of problems, respectively with less and more than ten design variables. A Intel Xeon E5-1620 v2 3.70GHz is used for the preliminary tests. Three absolute metrics are defined and applied for the evaluation of the algorithm performances, based on the distance between PSO-found solutions and analytical optima. According with the numerical tests, the most effective parameter choice among (a), (b), (c), and (d) is identified, based on the associated relative variability of the results. Then, the most promising setups for SDPSO and ADPSO are determined and applied to an industrial problem, namely a fast catamaran hull-form optimization, for calm water and fixed speed. The objective function is the ratio R_T/W between the total resistance (R_T) and the ship weight (W). The hull-form modification is performed using a KLE-based morphing approach [5–7], using respectively four- and six-dimensional design spaces. Computer simulations are performed using the potential flow (PF) solver CNR-INSEAN WARP [33], on a cluster of Intel Xeon E5462 2.80GHz. Each function evaluation takes about 10 min per node. Additionally, the optimization results are compared with those obtained in earlier research, based on a high-fidelity URANS solver [5].

2. PSO formulations

Consider the following *objective function*:

$$f(\mathbf{x}) : \mathbb{R}^{N_{dv}} \longrightarrow \mathbb{R} \tag{1}$$

and the global optimization problem

$$\min_{\mathbf{x}\in\mathcal{L}} f(\mathbf{x}), \quad \mathcal{L}\subset\mathbb{R}^{N_{dv}} \tag{2}$$

where $\mathcal{L}$ is a closed and bounded subset of $\mathbb{R}^{N_{dv}}$ and N_{dv} is the number of design variables. The global minimization of the objective function $f(\mathbf{x})$ requires to find a vector $\mathbf{a}\in\mathcal{L}$ such that:

$$\forall \mathbf{b}\in\mathcal{L} : f(\mathbf{a}) \le f(\mathbf{b}) \tag{3}$$

Then, $\mathbf{a}$ is a global minimum for the function $f(\mathbf{x})$ over $\mathcal{L}$. Since the solution of Eq. (2) is in general an NP-hard problem, the exact identification of a global minimum might be very difficult. Therefore, solutions with sufficient good fitness, provided by heuristic procedures, are often considered acceptable for several practical purposes. Among different heuristic procedures, PSO is often the method of choice for its capability to outreach a suitable approximate solution within a few iterations. In PSO, the positions of the particles represent the candidate solutions and will be denoted by $\mathbf{x}\in\mathcal{L}$, with associated fitness $f(\mathbf{x})$. Moreover, in this paper the compact set $\mathcal{L}$ represents the box constraints.

2.1. Original formulation

The original formulation of the PSO algorithm, as presented in [16], reads

$$\begin{cases} \mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1 r_{1,i}^k(\mathbf{x}_{i,pb} - \mathbf{x}_i^k) + c_2 r_{2,i}^k(\mathbf{x}_{gb} - \mathbf{x}_i^k) \\ \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{cases} \tag{4}$$

The above equations update velocity and position of the ith particle at the kth iteration, where w is the *inertia weight*; c_1 and c_2 are respectively the social and cognitive learning rate; $r_{1,i}^k$ and $r_{2,i}^k$ are uniformly distributed random numbers in $[0, 1]$; $\mathbf{x}_{i,pb}$ is the *personal best* position ever found by the ith particle in the previous iterations and $\mathbf{x}_{gb}$ is the *global best* position ever found in the previous iterations, considering all particles.