Environmental Modelling & Software 104 (2018) 215–235

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Shuffled Complex-Self Adaptive Hybrid EvoLution (SC-SAHEL) optimization framework

Matin Rahnamay Naeini [a], Tiantian Yang [a, b, *], Mojtaba Sadegh [a, c], Amir AghaKouchak [a], Kuo-lin Hsu [a], Soroosh Sorooshian [a], Qingyun Duan [d], Xiaohui Lei [e]

[a] Center for Hydrometeorology and Remote Sensing (CHRS) & Department of Civil and Environmental Engineering, University of California, Irvine, CA, USA
[b] Deltares USA Inc., Silver Spring, MD, USA
[c] Department of Civil Engineering, Boise State University, Boise, ID, USA
[d] Beijing Normal University, Faculty of Geographical Sciences, Beijing, China
[e] China Institute of Water Resources and Hydropower Research, Beijing, China

ARTICLE INFO

Article history:
Received 12 October 2017
Received in revised form
23 March 2018
Accepted 24 March 2018

Keywords:
Shuffled Complex Evolution (SCE)
Hybrid optimization
Evolutionary Algorithm (EA)
Reservoir operation
Hydropower

ABSTRACT

Simplicity and flexibility of meta-heuristic optimization algorithms have attracted lots of attention in the field of optimization. Different optimization methods, however, hold algorithm-specific strengths and limitations, and selecting the best-performing algorithm for a specific problem is a tedious task. We introduce a new hybrid optimization framework, entitled Shuffled Complex-Self Adaptive Hybrid EvoLution (SC-SAHEL), which combines the strengths of different evolutionary algorithms (EAs) in a parallel computing scheme. SC-SAHEL explores performance of different EAs, such as the capability to escape local attractions, speed, convergence, etc., during population evolution as each individual EA suits differently to various response surfaces. The SC-SAHEL algorithm is benchmarked over 29 conceptual test functions, and a real-world hydropower reservoir model case study. Results show that the hybrid SC-SAHEL algorithm is rigorous and effective in finding global optimum for a majority of test cases, and that it is computationally efficient in comparison to algorithms with individual EA.

Software availability

Name of software: SC-SAHEL
Developer: Matin Rahnamay Naeini
Contact address: rahnamam@uci.edu
Program language: MATLAB
Year first available: 2018
Availability: Freely available to public at http://chrs.web.uci.edu/resources.php and MathWorks website
Software requirements: MATLAB 9.0

1. Introduction

Meta-Heuristic optimization algorithms have gained a great deal of attention in science and engineering (Blum and Roli, 2003; Boussaïd et al., 2013; Lee and Geem, 2005; Maier et al., 2014; Nicklow et al., 2010; Reed et al., 2013). Simplicity and flexibility of these algorithms, along with their robustness make them attractive tools for solving optimization problems (Coello et al., 2007; Lee and Geem, 2005). Many of the meta-heuristic algorithms are inspired by a physical phenomenon, such as animals social and foraging behavior and natural selection. For example, Simulated Annealing (Kirkpatrick et al., 1983), Big Bang-Big Crunch (Erol and Eksin, 2006), Gravitational Search Algorithm (Rashedi et al., 2009), Charged System Search (Kaveh and Talatahari, 2010) are inspired by various physical phenomena. Ant Colony Optimization (Dorigo et al., 1996), Particle Swarm Optimization (Kennedy, 2010), Bat-inspired Algorithm (Yang, 2010), Firefly Algorithm (Yang, 2009), Dolphin Echolocation (Kaveh and Farhoudi, 2013), Grey Wolf Optimizer (Mirjalili et al., 2014), Bacterial Foraging (Passino, 2002), Genetic Algorithm (Golberg, 1989; Holland, 1992), and Differential Evolution (Storn and Price, 1997) are examples of algorithms inspired by animal's social and foraging behavior, and the natural selection mechanism of Darwin's evolution theorem. According to the No-Free-Lunch (NFL) (Wolpert and Macready, 1997) theorem,

* Corresponding author. Center for Hydrometeorology and Remote Sensing (CHRS) & Department of Civil and Environmental Engineering, University of California, Irvine, CA, USA.
E-mail address: tiantiay@uci.edu (T. Yang).

https://doi.org/10.1016/j.envsoft.2018.03.019
1364-8152/

none of these algorithms are consistently superior to others over a variety of problems, although some of them may outperform others on a certain type of optimization problem.

The NFL theorem has been a source of motivation for developing optimization algorithms (Mirjalili et al., 2014; Woodruff et al., 2013). It has encouraged scientists and researchers to combine the strengths of different algorithms and devise more robust and efficient optimization algorithms that suit a broad class of problems (Qin and Suganthan, 2005; Vrugt and Robinson, 2007; Vrugt et al., 2009; Hadka and Reed, 2013; Sadegh et al., 2017). These efforts led to emergence of multi-method and self-adaptive optimization algorithms such as Self-adaptive DE algorithm (SaDE) (Qin and Suganthan, 2005), A Multialgorithm Genetically Adaptive Method for Single Objective Optimization (AMALGAM-SO) (Vrugt and Robinson, 2007; Vrugt et al., 2009) and Borg (Hadka and Reed, 2013). They all regularly update the search mechanism during the course of optimization according to the information obtained from the response surface.

Here, we propose a new self-adaptive hybrid optimization framework, entitled Shuffled Complex-Self Adaptive Hybrid EvoLution (SC-SAHEL). The SC-SAHEL framework employs multiple Evolutionary Algorithms (EAs) as search cores, and enables competition among different algorithms as optimization run progresses. The proposed framework differs from other multi-method algorithms as it grants independent evolution of population by each EA. In this framework, population is partitioned into equally sized groups, so-called complexes; each assigned to different EAs. Number of complexes assigned to each EA is regularly updated according to their performance. In general, the newly developed framework has two main characteristics. First, all the EAs evolve population in a parallel structure. Second, each participating EA works independent of other EAs. The architecture of SC-SAHEL is inspired by the concept of the Shuffled Complex Evolution algorithm - University of Arizona (SCE-UA) (Duan et al., 1992). The SCE-UA algorithm is a population-evolution based algorithm (Madsen, 2003), which evolves individuals by partitioning population into different complexes. The complexes are evolved for a specific number of iterations independent of other complexes, and then are forced to shuffle.

The SCE-UA framework employs Nelder-Mead simplex (Nelder and Mead, 1965) technique along with the concept of controlled random search (Price, 1987), clustering (Kan and Timmer, 1987), competitive evolution (Holland, 1975) and complex shuffling (Duan et al., 1993) to offer a global optimization strategy. By employing these techniques, the SCE-UA algorithm provides a robust optimization framework and has shown numerically to be competitive and efficient comparing to other algorithms, such as GA, for calibrating rainfall-runoff models (Beven, 2011; Gan and Biftu, 1996; Wagener et al., 2004; Wang et al., 2010). The SCE-UA algorithm has been widely used in water resources management (Barati et al., 2014; Eckhardt and Arnold, 2001, K. Ajami et al., 2004; Lin et al., 2006; Liong and Atiquzzaman, 2004; Madsen, 2000; Sorooshian et al., 1993; Toth et al., 2000; Yang et al., 2015; Yapo et al., 1996), as well as other fields of study, such as pyrolysis modeling (Ding et al., 2016; Hasalová et al., 2016) and Artificial Intelligence (Yang et al., 2017).

Application of the SCE-UA is not limited to solving single objective optimization problems. The Multi-Objective Complex evolution, University of Arizona (MOCOM-UA), is an extension of the SCE-UA for solving multi-objective problems (Boyle et al., 2000; Yapo et al., 1998). Besides, the SCE-UA architecture has been used to develop Markov Chain Monte Carlo (MCMC) sampling, named Shuffled Complex Evolution Metropolis algorithm (SCEM-UA) and the Multi-Objective Shuffled Complex Evolution Metropolis (MOSCEM) to infer posterior parameter distributions of hydrologic models (Vrugt et al. 2003a, 2003b). The Metropolis scheme is used as the search kernel in the SCEM-UA and MOSCEM-UA (Chu et al., 2010; Vrugt et al. 2003a, 2003b). There is also an enhanced version of SCE-UA, which is developed by Chu et al. (2011) entitled the Shuffled Complex strategy with Principle Component Analysis, developed at the University of California, Irvine (SP-UCI). Chu et al. (2011) found that the SCE-UA algorithm may not converge to the best solution on high-dimensional problems due to “population degeneration” phenomenon. The “population degeneration” refers to the situation when the search particles span a lower dimension space than the original search space (Chu et al., 2010), which causes the search algorithm to fail in finding the global optimum. To address this issue, the SP-UCI algorithm employs Principle Component Analysis (PCA) in order to find and restore the missing dimensions during the course of search (Chu et al., 2011).

Both SCE-UA and SP-UCI start the evolution process by generating a population within the feasible parameters space. Then, population is partitioned into different complexes, and each complex is evolved independently. Each member of the complex has the potential to contribute to offspring in the evolution process. In each evolution step, more than two parents may contribute to generating offspring. To make the evolution process competitive, a triangular probability function is used to select parents. As a result, the fittest individuals will have a higher chance of being selected. Each complex is evolved for a specific number of iterations, and then complexes are shuffled to globally share the information attained by individuals during the search.

The Competitive Complex Evolution (CCE) and Modified Competitive Complex Evolution (MCCE) are the search cores of the SCE-UA and SP-UCI algorithm, respectively. The CCE and MCCE evolutionary processes are developed based on Nelder-Mead (Nelder and Mead, 1965) method with some modification. The evolution process in the SCE-UA is not limited to these algorithms. In fact, several studies have incorporated different EAs into the structure of the SCE-UA algorithm. For example, the Frog Leaping (FL) is developed by adapting Particle Swarm Optimization (PSO) algorithm to the SCE-UA structure for solving discrete problems (Eusuff et al., 2006; Eusuff and Lansey, 2003). Mariani et al. (2011) proposed an SCE-UA algorithm which employs DE for evolving the complexes. These studies revealed the flexibility of the SCE-UA in combination with other types of EAs; however, the potential of combining different algorithms into a hybrid shuffled complex scheme has not been investigated.

The unique structure of the SCE-UA algorithm along with the flexibility of the algorithm for using different EAs, motivated us to use the SCE-UA as the cornerstone of the SC-SAHEL framework. The SC-SAHEL algorithm employs multiple EAs for evolving the population in a similar structure as that of the SCE-UA, with the goal of selecting the most suitable search algorithm at each optimization step. On the one hand, some EAs are more capable of visiting the new regions of the search space and exploring the problem space, and hence are particularly suitable at the beginning of the optimization (Olorunda and Engelbrecht, 2008). On the other hand, some EAs are more capable of searching within the visited regions of the search space, and hence boosting the convergence process after finding the region of interest (Mirjalili and Hashim, 2010). Balancing between these two steps, which are referred to as exploration and exploitation (Moeini and Afshar, 2009), is a challenging task in stochastic optimization methods (Črepinšek et al., 2013). The SC-SAHEL algorithm maintains a balance between exploration and exploitation phases by evaluating the performance of participating EAs at each optimization step. EAs contribute to the population evolution according to their performance in previous steps. The algorithms' performance is evaluated by comparing the evolved complexes before and after evolution. In this process, the