Knowledge-Based Systems 97 (2016) 144–157

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Monkey King Evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization

Zhenyu Meng[a,*], Jeng-Shyang Pan[a,b]

[a] Department of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
[b] College of Information Science and Engineering, Fujian University of Technology, Fuzhou, China

ARTICLE INFO

Article history:
Received 14 July 2015
Revised 7 December 2015
Accepted 8 January 2016
Available online 20 January 2016

Keywords:
Benchmark function
Fuel consumption
Monkey King Evolutionary Algorithm
Number of function evaluation
Particle swarm variants
Vehicle navigation

ABSTRACT

Optimization algorithms are proposed to tackle different complex problems in different areas. In this paper, we firstly put forward a new memetic evolutionary algorithm, named Monkey King Evolutionary (MKE) Algorithm, for global optimization. Then we make a deep analysis of three update schemes for the proposed algorithm. Finally we give an application of this algorithm to solve least gasoline consumption optimization (find the least gasoline consumption path) for vehicle navigation. Although there are many simple and applicable optimization algorithms, such as particle swarm optimization variants (including the canonical PSO, Inertia Weighted PSO, Constriction Coefficients PSO, Fully Informed Particle Swarm, Comprehensive Learning Particle Swarm Optimization, Dynamic Neighborhood Learning Particle Swarm). These algorithms are less powerful than the proposed algorithm in this paper. 28 benchmark functions from BBOB2009 and CEC2013 are used for the validation of robustness and accuracy. Comparison results show that our algorithm outperforms particle swarm optimizer variants not only on robustness and optimization accuracy, but also on convergence speed. Benchmark functions of CEC2008 for large scale optimization are also used to test the large scale optimization characteristic of the proposed algorithm, and it also outperforms others. Finally, we use this algorithm to find the least gasoline consumption path in vehicle navigation, and conducted experiments show that the proposed algorithm outperforms A∗ algorithm and Dijkstra algorithm as well.

1. Introduction

Optimization algorithms in evolutionary computation are equipped with a meta-heuristic or stochastic optimization or memetic optimization character, and they belong to the family of trial and error problem solvers and distinguished by the use of a population of candidate solutions. Particles of these algorithms have two main character components, one is exploitation, and the other exploration. Particle Swarm Optimization (PSO) is a powerful evolutionary computational algorithm introduced by Kennedy and Eberhart in [1]. The canonical PSO does not use cross over and mutation operations, and particles in the population produce the next generation by learning from their history best and global best of the population experience. The moving velocity is used to make a balance between the exploitation and exploration of a particle.

As the PSO algorithm is simple, easy to implement and it also has been empirically performed well on many optimization problems since its inception, many researches have learned about the technique and proposed many variants, or new versions of PSO. [2] proposed a new optimizer using particle swarm theory, and examined how the changes in the paradigm affected the number of iteration required to meet an error criterion. [3] presented a modified particle swarm optimizer with an inertia weight of particle velocity, a 2-dimension 4000-iteration conducted experiment showed that the smaller inertia weight made it converged fast if PSO could find the global optimum. When the inertia weight was small, PSO paid more attention on exploitation and when the inertia weight was larger, PSO paid more on exploration. Moderate value of weight made PSO had the best chance to find the global optimum with a moderate number of iteration. Empirically, inertia weight was set as a decreasing function of iteration instead of a fixed constant. Eberhart and Shi [4] made a comparison between inertia weights and constriction factors in particle swarm optimization, and the experiments showed that constriction coefficient $k = 0.7298$ and the constant $c_1 = c_2 = 2.05$ was a good choice [5]. PSO trajectories and topologies had been deeply analyzed for the importance of the convergence. Kennedy [6,7] claimed that PSO with a small neighborhood might perform better on complex

* Corresponding author. Tel.: +86 13825100875.
E-mail address: mzy1314@gmail.com, mzy_1314@126.com (Z. Meng).

http://dx.doi.org/10.1016/j.knosys.2016.01.009

problems, while PSO with a large neighborhood would perform better on simple problems. Suganthan [8] proposed a neighbor operator particle swarm optimization, and the operator calculated the particles distance and particles learned from the neighborhood. Mendes et al. [9] proposed a fully informed particle swarm, valuable information was gained from the particle neighbor and the convergence speed improved by this variant of particle swarm optimization. Mendes [10] gave a deep analysis on population topologies and their influence in particle swarm performance in his doctoral thesis. Other variants such as dynamic multi-swarm optimizer [11] and optimizer proposed in [12,13] performed better to solve shifted-rotated benchmark functions, multi-optimization problems and multi-modal functions respectively. There are also all kinds of applications using PSO to tackle different tasks in different fields. Ref. [14] is an application of optimization traffic lights program with PSO, [15] shows an application to tackle complex network clustering by multi-objective discrete particle swarm optimization, and [16] proposed a binary particle swarm optimization algorithm for optimizing the echo state network.

There are more and more optimization algorithms proposed to tackle specific tough problems, and PSO is a popular one with developing variants from its inception in 1995. PSO variants proposed in recent years for specific problem optimization are much more complicated with huge time consumption than the canonical one, but the optimization results are not so satisfying either on optimization accuracy or on convergence speed. A property that is appealing than just being able to convergence to the optimum when elapsed time approaches infinity, is to guarantee that a good solution can be found with a low number of function evaluations [10]. Simple optimization algorithms with powerful capacity and robust [17] are much popular both for academic researches and engineers. So in this paper, we proposed the MKE algorithm, which has a better convergence speed and convergence accuracy with the similar time complexity in comparison with variants of PSO.

With the development of industry technology, there are more and more cars driving on the road. Traffic navigation becomes a hot topic for city governors and researchers. Different approaches have been advanced to tackle congestion and traffic emergency, which aim for better performance of the traffic networks. The desires of different roles in the traffic networks are different. City governors often emphasize on the output of the total networks, while some of the single drivers pay more attention to least travel time or travel distance and most of them pay attention to least fuel expense. As we know that large fuel consumption occurs when cars are in traffic jams, and how to make a navigation while avoiding congestion in the traffic networks not only achieve good output of the total traffic networks, but also save the drivers' money, and this is an optimization problem. Ref. [18] shows some evolutionary thoughts to tackle vehicle routing problem using a different model of a static network. In this paper we advance models and fitness functions of the traveling fuel consumption for a path in vehicle navigation. The conducted experiments show that our method outperforms A∗ [19] and Dijkstra [20] in finding the least fuel consumption path of the real-time navigation. The main contributions of the paper include:

1. A new memetic evolutionary algorithm is advanced for global optimization and it outperforms state-of-the-art PSO variants not only on the robustness and accuracy but also on the convergence speed (Test on BBoB2009 [21] and CEC2013 [22] benchmark functions on real parameter optimization).
2. The proposed algorithm has a large scale optimization property that can be well used to tackle large scale optimization problems and it can be easily paralleled on distributed computing systems to boost the calculation speed (Test on CEC2008 [23] benchmark functions on large scale optimization).
3. A traffic networks model based on wireless sensor network environment is proposed with gasoline consumption function (of navigation paths) proposed to be optimized for individual navigation with regarding to least congestion in restricted traffic networks.
4. The navigation result of the proposed algorithm outperforms A∗ and Dijkstra algorithm on gasoline consumption for real-time navigation.

The rest of the paper is organized as follows, Section 2 presents the related works. Section 3 presents the detailed algorithm of Monkey King Evolution. Section 4 presents the navigation model and fuel consumption fitness function. Section 5 gives a comparative view and analysis and Section 6 shows the final conclusion.

2. Related works

Considerable developments have occurred since the inception of canonical PSO [1]. The canonical PSO is based on swarm intelligence and was inspired by the seeking food behavior of a flock of birds. Individual bird is only influenced by its historical best and the global best of the population, and the evolution equation is shown in Eq. (1). The canonical PSO is simple and easy to implement, but the convergence is not good enough or even rather bad to complicated problems.

$$\begin{cases} V_i^{t+1} \leftarrow V_i^t + c_1 * (X_{fb}^t - X_i^t) + c_2 * (X_{gb}^t - X_i^t); \\ X_i^{t+1} \leftarrow X_i^t + V_i^{t+1}; \end{cases} \tag{1}$$

In order to accelerate the convergence speed of the canonical PSO, an inertial weighted PSO [3] was proposed with the evolution equation shown in Eq. (2). Almost all the PSO variants like constraint coefficient PSO (Eq. (3)), FIPS (Eq. (4)), Comprehensive Learning PSO [12] (CLPSO, learning form personal best and others' best), Cooperative PSO [24] (CPSO, decomposing dimension vector as multiple swarm), Dynamic Neighborhood Learning PSO [13] (dynamic neighborhood topology enabled exploration) use particle topology/relationship for evolution, we can also get the topology perception from Eqs. (1)–(4).

$$\begin{cases} v_i^{t+1} \leftarrow \omega * v_i^t + c_1 * (X_{fb}^t - X_i^t) + c_2 * (X_{gb}^t - X_i^t); \\ X_i^{t+1} \leftarrow X_i^t + v_i^{t+1}; \end{cases} \tag{2}$$

$$\begin{cases} v_i^{t+1} \leftarrow \chi * (v_i^t + \varphi_1 * (X_{fb} - X_i^t) + \varphi_2 * (X_{gb} - X_i^t)); \\ X_i^{t+1} \leftarrow X_i^t + v_i^{t+1}; \end{cases} \tag{3}$$

$$\begin{cases} v(t+1) = \chi * (v(t) + \varphi * (p_m - x(t))) \\ x(t+1) = x(t) + v(t+1) \\ \varphi_k = U[0, \dfrac{\varphi_{max}}{|N|}], \forall k \in N \\ p_m = \dfrac{\sum_{k \in N} \omega(k)\varphi_k \otimes p_m}{\sum_{k \in N} \omega(k)\varphi_k} \end{cases} \tag{4}$$

Topology/relationship plays a very important role in the performance of a PSO variant, and proposed topologies up to date still do not make the full exploration of the search region. Some of the algorithms (CPSO, SLPSO) mentioned above need extra computation expense. For example, the computation time complexity of CPSO is about D (D is the dimension number) times larger than PSO, IW-PSO, CCPSO, FIPS, CLPSO, and DNLPSO, and the performance of it does not improved significantly. Moreover, the PSO variants also have a fatal weakness that the performance does not improve with the increase of population size, so is the weakness of these algorithms for parallel computing.

For traffic navigation, Wireless Sensor Networks (WSNs) consisting a number of sensor nodes are used to monitoring a local area and getting the traffic information with little infrastructure [25]. There are often two different types of the structure, one