Swarm and Evolutionary Computation 26 (2016) 157–174

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

Unknown environment exploration of multi-robot system with the FORDPSO

Dongshu Wang [a,*], Haitao Wang [b], Lei Liu [c]

[a] School of Electrical Engineering, Zhengzhou University, 450001 Zhengzhou, China
[b] The 716 Institute, China Shipbuilding Industry Corporation, JiangSu Automation Institute, 222006 Lianyungang, Jiangsu, China
[c] Department of Research, The People's Bank of China, Zhengzhou Central Sub-Branch, 450040 Zhengzhou, China

ARTICLE INFO

Article history:
Received 11 December 2014
Received in revised form
6 September 2015
Accepted 21 September 2015
Available online 1 October 2015

Keywords:
Multi-robot system
Unknown environment exploration
Particle swarm optimization
Darwinian particle swarm optimization
Fractional order calculus
Fuzzy inferring system

ABSTRACT

Effective environment exploration in unknown environment is precondition of constructing the environment map and carrying out other tasks for multi-robot system. Due to its excellent performance, particle swarm optimization (PSO) has been widely used in multi-robot exploration field. To deal with its drawback – easily trapped in local optima, Darwinian PSO (DPSO) optimization is proposed by Tillett et al. [1] with the natural selection function and first used in real world robot exploration by Couceiro et al. [2], forming the robotic DPSO (RDPSO). To increase the algorithm performance and control its convergence rate, fractional calculus is used to replace inertia component in RDPSO for its “memory” ability and forming the fractional order RDPSO (FORDPSO). This paper presents a formal analysis of RDPSO and studies the influence of the coefficients on FORDPSO algorithm. To satisfy the requirement of dynamically changing robots' behaviors during the exploration, fuzzy inferring system is designed to achieve better control coefficients. Experiment results obtained in two complex simulated environments illustrate that biological and sociological inspiration is effective to meet the challenges of multi-robot system application in unknown environment exploration, and the exploration effect of the fuzzy adaptive FORDPSO is better than that of the fixed coefficient FORDPSO. Furthermore, the performance of FORDPSO with different neighborhood topologies are studied and compared with other six PSO variations. All the results demonstrate the effect of the FORDPSO on the multi-robot environment exploration.

1. Introduction

Unknown environment exploration is one of the most important problems in mobile robot research field, and it is also a research hotspot in recent years. In many practical cases which are too dangerous or difficult for human to reach (e. g., military assignments, interstellar exploration, searching for victims after earthquake, and nuclear/biological accidents), whether the robot can complete the tasks successfully depends on whether the environment exploration can be carried out smoothly. Comparing with the single robot system, multi-robot system has been widely used in many complex and abominable scenarios, due to its strong adaptation, excellent flexibility, and high reliability. Multi-robot system is getting more and more emphasis and studies [3–5]. Unknown environment exploration of the multi-robot system is also achieving more and more researches [6–8].

Traditional exploration strategies of multi-robot system adopt concentrating or semi-distributing control modes which use a few leading nodes to undertake swarm decision task, and these strategies suit the multi-robot system with a small swarm population. When the swarm population grows, the calculation of the leading nodes and communication burden will increase exponentially which make it unsuitable in practical application. Furthermore, in many conventional multi-robot exploration methods, due to the node's heterogeneity, one node fault can result in complicated task re-allocation and dynamic equilibrium. Moreover, traditional multi-robot exploration methods also bring some new problems, such as data fusion. Recently, swarm robot systems based on swarm intelligence become a new research direction for its robustness, excellent expandability and good performance on communication data control [9].

Swarm intelligence is a kind of collective behavior of distributed, self-organized system. It is a representative method of behaviorism artificial intelligence. The intuitive notion of “Swarm Intelligence” is that of a “swarm” of agents (biological or artificial) which, without central control, collectively (and only collectively) carry out (unknowingly, and in a somewhat-random way) tasks normally

* Corresponding author.
E-mail addresses: wangdongshu@126.com (D. Wang), waahait@163.com (H. Wang), luckyliulei@126.com (L. Liu).

http://dx.doi.org/10.1016/j.swevo.2015.09.004
2210-6502/

requiring some form of "intelligence" [10]. It originates the collective behavior research of social insects (e. g., ants, bees and fish school). Swarm intelligent behavior is reflected by simple rules and collective cooperation among a swarm of simple low intelligence individuals. Swarm intelligence has the following characteristics: (1) each individual communicates with others among the swarm or their surrounding environment in local scope; (2) each individual is very simple but isomorphic; (3) through individual's simple local feedback, the swarm can complete complex tasks.

Swarm intelligence has produced two famous algorithms: ant colony optimization [11,12] and particle swarm optimization (PSO) [13,14]. The former simulates the ant colony searching the food and has been applied in many discrete optimization problems [15–17]. PSO is inspired by birds flocking or fish school in search of food. Now the PSO algorithm has become an excellent optimization tool. These two optimization algorithms have been widely used in multi-robot cooperation [18–20].

However, PSO and other optimization algorithms have a general drawback – they may be trapped in a local optimal solution, which may work well in some problems but fails in others [21]. In order to overcome these drawbacks, Darwinian PSO (DPSO) algorithm was proposed by Tillett et al. [1] who introduced the natural selection in PSO. In this algorithm, multiple swarms of test solutions, each of them performing just like an ordinary PSO, may exist at any time with some rules governing the collection of swarms which are designed to simulate natural selection [22]. Couceiro et al. [2,22,23] firstly applied the DPSO into multi-robot exploration field and proposed the Robotic DPSO (RDPSO), taking into accounting obstacle avoidance. It benefits from the dynamical partitioning of the whole population of robots, thus decreasing the amount of required information exchange among the robots. Further, in order to control the convergence rate of RDPSO, they introduced the fractional order RDPSO (FORDPSO) [24]. It uses the fractional calculus to control the convergence rate of robots toward the optimal solutions.

Performance of FORDPSO greatly depends on its coefficients. In conventional methods, these coefficients are set randomly and subjectively [24], so the algorithm performance cannot be guaranteed well. In order to control the swarm susceptibility to the main mission, obstacle avoidance, etc., FORDPSO should be improved to systematically adjust its parameters. Based on the above analysis, this paper adopts fuzzy inferring system to adjust the coefficients in FORDPSO and compares the algorithm performance of two kinds of FORDPSO: adaptive parameter adjustment with fuzzy inferring and fixed parameter, through two typical environment exploration experiments with the multi-robot system. Simulation results show that performance of FORDPSO with adaptive parameter adjustment is better than that of the algorithm with fixed parameters.

The remainder of the paper is organized as follows. Section 2 introduces the theory for PSO and DPSO. Section 3 describes the theory of RDPSO. Section 4 represents the fractional order RDPSO and discusses the coefficient ranges. Section 5 analyzes the coefficients influence of FORDPSO algorithm and designs the fuzzy inferring system to dynamically adjust the coefficients. In Section 6, we design the simulation procedure, compare and analyze the simulation results of two typical scenarios with fuzzy parameter adjustment FORDPSO and fixed parameter FORDPSO, while the conclusions and future study are given in the last section.

2. Related work

2.1. Particle swarm optimization

The PSO proposed by James Kennedy and R.C. Eberhart is a population-based stochastic optimization method based on the social behavior of groups of organisms like bird flocks or fish schools [25]. The method was inspired by the movement of flocking birds and their interactions with their neighbors in the group. Each individual in the group is regarded as a particle that flies in a given virtual search space, and each of them represents a potential solution. Movement of a particle is determined by not only its own search but also by the search of its neighbors. That is to say, each particle's change of position and velocity is affected by the information of its neighbors.

Assuming a given swarm has n particles, and each particle is characterized by its position vector $x_n(t)$ and velocity vector $v_n(t)$ at each discrete time t. In other words, these two characteristics can depict the particle's pose. If we extend the search space to D dimensions, the position and velocity of the ith particle can be denoted as $x_i=[x_{i1},x_{i2},\ldots,x_{iD}]^T$, $v_i=[v_{i1},v_{i2},\ldots,v_{iD}]^T$. Performance of each particle in each position needs an objective function to evaluate. For a given position, put the x_i into the objective function to get the current objective function value and compare it with the optimal value achieved, we can gain the optimal objective function value (individual historical optimal solution, $P_i=[P_{i1},P_{i2},\ldots,P_{in}]^T$) so far. For this given position, all particles' optimal solutions are compared to obtain the swarm's optimal solution, $P_g=[P_{g1},P_{g2},\ldots,P_{gn}]^T$.

Assuming at current time t, the ith particle's current position and velocity, as well as the individual historical optima and swarm historical optima are known, then at the next time $t+1$, the ith particle's expected position and velocity can be depicted as follows:

$$v_i^{t+1}=wv_i^t+c_1r_1(P_i-x_i^t)+c_2r_2(P_g-x_i^t)+c_3r_3(P_N-x_i^t) \tag{1}$$

$$x_i^{t+1}=x_i^t+v_i^{t+1} \tag{2}$$

In formula (1), the first term at right side denotes inertia effect produced by the current velocity. The second term represents the best search position of particle i in its vicinity, known as "cognitive" component. The third one depicts the global optimal position detected by all particles, known as "social" component. The fourth one denotes the optimal solution in its vicinity, known as "neighborhood social" component. r_1, r_2 and r_3 are positive random numbers between 0 and 1, c_1, c_2 and c_3 are the weights of the cognitive, social and neighborhood social components, respectively.

2.2. Darwinian particle swarm optimization

In order to deal with the drawback of PSO, many variants have been presented [26–28]. Among these variants, DPSO is one of the most representative algorithms. The DPSO was proposed by Tillett et al. [1] as a solution to deal with the problems mentioned above by imposing a "natural selection" mechanism.

Main idea of the DPSO is to run many simultaneous parallel PSO algorithms, each one representing a different swarm, on the same test problem, and a simple selection mechanism is applied. When a search tends to a local optima, the search in that area is discarded and another area is searched instead. To analyze the general state of each swarm, the fitness of all particles is evaluated, and the neighborhood and individual best positions of each particle are updated. For a given swarm, if it finds a better swarm optimal solution, then it iterates and updates, and these behaviors indicate that this swarm has made a progress and should be rewarded by spawning a new particle or extending particle's survival life, if its population does not exceed the maximum number of particles per swarm. Moreover, the longer a swarm lives, the more chance it has to produce offspring, though this chance is very small. On the contrary, if the swarm cannot find a better swarm optimal solution and stagnates, it means this swarm has not updated and should be punished by deleting his worst performing particle, if then the swarm is left with a number of particles below