Fitness Proportionate Selection Based Binary Particle Swarm Optimization

Xing Liu  
State Key Laboratory of Novel Software Technology  
Department of Computer Science and Technology  
Nanjing University, Nanjing 210046, China  
lui.xing@outlook.com

Lin Shang  
State Key Laboratory of Novel Software Technology  
Department of Computer Science and Technology  
Nanjing University, Nanjing 210046, China  
shanglin@nju.edu.cn

ABSTRACT

Particle Swarm Optimization (PSO) has shown its advantages not only in dealing with continuous optimization problems, but also in dealing with discrete optimization problems. Binary Particle Swarm Optimization (BPSO), the discrete version of PSO, has been widely applied to many areas. Although there are some variations aiming to improve BPSO’s performance, none of them has been proven to be a promising alternative. In this paper, we propose a novel binary particle swarm optimization called Fitness Proportionate Selection Based Binary Particle Swarm Optimization (FPSBPSO). We test FPSBPSO’s performance in function optimization problems and multidimension knapsack problems. Experimental results show that FPSBPSO can find better optima than BPSO and a variation of BPSO.

Categories and Subject Descriptors

D.3.3 [Programming Languages]: Language Constructs and Features—abstract data types, polymorphism

General Terms

Algorithms, Performance, Reliability, Experimentation

Keywords

Particle Swarm Optimization, Discrete Optimization, Fitness Proportionate Selection

1. INTRODUCTION

Particle Swarm Optimization (PSO) [2] was originally designed to solve continuous optimization problems. In order to deal with discrete problems, Kennedy and Eberhart proposed binary particle swarm optimization (BPSO) [3] in 1997. BPSO has attracted much attention, however, most researchers focused more on the application of BPSO rather than the analysis of BPSO. Although there are several improved versions of BPSO, none of them has become a real alternative to BPSO. The original BPSO model is used in most publications reporting the application of PSO in dealing with discrete problems.

There are several problems with BPSO. Engelbrecht argues that maybe weight is useless in BPSO [1]. Khanesar discussed the memory problem and parameter selection problem of BPSO in [4].

Our goal is to improve PSO’s performance in dealing with discrete problems with a better interpretation than BPSO. A new way to calculate velocities based on fitness values is proposed and based on that we will propose a new binary version of PSO named FPSBPSO.

2. NEW METHOD

We adopt fitness proportionate selection to update a particle’s position component, and that is why we call the new model FPSBPSO. In the following content, when we refer to involved particles in the following content, we actually mean the three particles $p_i$, $p_i$’s personal best particle $p_i^{pb}$ and $p_i$’s global best particle $p_i^{gb}$. The newly designed update formulas are as follows.

$$v_{id}^{t+1} = \begin{cases} mr, & \text{if } n_0 == 0 \\ 1 - mr, & \text{if } n_1 == 0 \\ \frac{f_1}{f_1 + f_0}, & \text{otherwise.} \end{cases} \quad (1)$$

$$x_{id}^{t+1} = \begin{cases} 0, & \text{rand < } v_{id}^{t+1} \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

Where $n_0$ is the number of involved particles with $x_{id} = 0$, while $n_1$ is the number of involved particles with $x_{id} = 1$. $f_1$ and $f_0$ are computed as follows: First, involved particles are divided into two sets $S_1$ and $S_0$ based on whether they select 1 or 0 at the $d_{th}$ bit. Then, $f_1$ and $f_0$ are calculated by averaging the fitness values of particles in $S_1$ and $S_0$ respectively. $v_{id}^{t+1}$ is the probability of setting $x_{id}^{t+1}$ to 0. The higher $f_1$ is, the higher probability $x_{id}^{t+1}$ will be 0. It is obvious that the formulas given above are for minimization problems and can easily be modified to deal with maximization problems.

In this new model, the update formula for velocities is very different from that of the BPSO. Parameters commonly used in PSO models like inertia weight $V_{max}$ are eliminated. A new parameter $mr$ is added and it is the only one free parameter.
3. EXPERIMENTS

In order to evaluate FPSBPSO’s performance, we tested FPSBPSO on two kinds of problems: function optimization problem and Multidimension knapsack problem (MKP). We compare FPSBPSO’s results with the basic BPSO’s results and NBPSO’s results [4]. The fitness function we used in MKP was proposed by Khuri et al. in [5].

Experimental results are shown in six tables. From Table 1 to Table 5 are results obtained on function optimization problems while Table 7 gives results on MKP. Table 6 gives the MKP description.

4. CONCLUSIONS

In this paper, we redefined the velocity component and proposed a new BPSO model. The new model is very simple with only one free parameter. Experimental results show that the new model is superior to BPSO, especially when the dimensionality of the optimization problem is high.

5. ACKNOWLEDGMENTS

We really appreciate National Natural Science Foundation projects (No.61170180, 61035003).

REFERENCES