Optimizing Particle Swarm Optimization to Solve Knapsack Problem

Yanbing Liang, Linlin Liu, Dayong Wang, and Ruijuan Wu

Hebei Polytechnic University, Tangshan 063000, China
snoopy6408665@163.com

Abstract. Knapsack problem, a typical problem of combinatorial optimization in operational research, has broad applied foregrounds. This paper applies particle swarm optimization to solve discrete 0/1 knapsack problem. However, traditional particle swarm optimization has nonnegligible disadvantages: all the parameters in the formula affect the abilities of local searching and global searching greatly, which is liable to converge too early and fall into the situation of local optimum. This paper modifies traditional particle swarm optimization, and makes the position of particle which achieves global optimization reinitializated. Through analyzing the final result, the paper has proven that the improved algorithm could improve searching ability of particle swarm, avoid converging too early and solve 0/1 knapsack problem more effectively.

Keywords: particle swarm optimization, local optimum, global optimum, fitness.

1 0-1 Knapsack Problem

Knapsack problem is a typical problem of combinatorial optimization in operational research, having broad application foreground, such as resource allocation problem, goods shipment problem, project selection problem and so on. Knapsack problem belongs to NP hard complete problem [1, 2, 3], at Present the methods of solving optimization problem are accurate method (such as dynamic programming, recursion method, backtracking, branch and bound algorithm and so on), approximation algorithm (such as greedy method, Lagrange method and so on) and intelligent optimization algorithm (such as simulated annealing algorithm, genetic algorithm, ant colony algorithm and so on) [4, 5]. Accurate method can obtain exact solution, but time complexity is 2^n, and there is an exponential relation between time complexity and goods number. Approximation algorithm and intelligent optimization algorithm don't always obtain exact solution, but they can obtain better of approximate solution and time complexity is lower [6, 7, 8].

1.1 Problem Description

There are N goods and a knapsack that capacity is B. a_i is the cost of the ith good, b_i is the value of the ith good. Then solve which goods are let into the knapsack to

R. Zhu et al. (Eds.): ICICA 2010, Part I, CCIS 105, pp. 437–443, 2010.

make the cost total of the goods no more than capacity of the knapsack and get the maximum total of the value.

1.2 Problem Analysis

0/1 knapsack problem is the most basic knapsack problem, it includes the most basic idea of design state and equation in the knapsack problem, in addition, other kinds of knapsack problem can be converted into 0/1 knapsack problem to solve.

Expressions are as follows:

$$\max\sum_{i=1}^{n} a_i * x_i \quad \text{s.t} \quad \sum_{i=1}^{n} b_i * x_i <= B\ ,\ x_i = 0,1(\mathrm{i} = 1,2,3\cdots \mathrm{n})\ ,$$

where x_i is 0-1 decision variable. $x_i = 1$ expresses to put the ith good into knapsack, otherwise $x_i = 0$.

2 Particle Swarm Optimization and Mathematical Derivation

Particle swarm optimization was developed in 1995 by J. Kennedy and R. C. Eberhart, etc. It is a kind of evolutionary computation technology, and it comes from the simulation about a social model simplified. Swarm coming from particle swarm accords with five basic principles of swarm intelligence proposed by M. M. Millonas when he develops and applies to the model of artificial life. Particle is a eclectic selection, because the members of population need to be described as no quality and no volume ones, they need to be described speed state and acceleration state [9, 10].

PSO is for graceful but unpredictable movement of graphical simulation bird group at first. But through observing the animal social behavior, they find that it provides an advantage of evolution for social sharing of information in the group, so they develop an algorithm which is based on it. Through adding adjacent velocity matching and considering multi-dimensional search, according to acceleration of distance, the initial version of PSO is formed. Then inertia weight w is introduced to control development and exploration better, the standard version of PSO is formed.

2.1 Principle

PSO is based on the swarm, it makes individual in the swarm move to a good region. But it doesn't use evolution operator to individual, each individual is regarded as a no volume particle(point) in the D-dimensional search space and flies by a definite speed in the search space, the speed can be adjusted dynamically according to its own flight experience and flight experience of other particles. The ith particle is pressed as $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, the best position undergone (having the best fitness value) is $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, it is called p_{best} also. Index sign of the best position where all particles of the swarm have undergone is g , that is P_g , it is called g_{best} also. The speed of the ith particle is $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$. For each generation, the Dth dimensional changes according to the following equations:

$$v_i(t+1) = w * v_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (p_g - x_i(t)) .$$

$$x_i(t+1) = x_i(t) + v_i(t+1) .$$

In which w is inertia weight, c_1 and c_2 are acceleration constants, making $c_1 = c_2 = 2$ usually, $rand()$ and $Rand()$ are two random values changing on the range of $[0,1]$.

In addition, the velocity of the particle is limited by a maximum velocity $v_{\max}$. If accelerating the particle v_{id} causes that the velocity of some dimension exceeds the maximum velocity of the dimension $v_{\max,d}$, the velocity of some dimension is limited the maximum velocity of the dimension $v_{\max,d}$.

For formula 1, the first part is the previous act inertia of the particle, the second part is the part of "cognition", it expresses the particle's own thinking, the third part is the part of "society", and it presses information sharing and mutual cooperation between the particles.

The part of "cognition" can be explained by the effect rule of Thorndike, that is one random behavior strengthened can appear more in the future. This behavior is "cognition", it is assumed that the acquirement of correct knowledge is strengthened; the particle will be encouraged to reduce the error according to this model assumption.

The part of "society" can be explained by replacement intensifying of Bandura. According to the expectation of this theory, when the observer observes a model strengthening some behavior, the probability that carries out the behavior is increased. Namely the particle's own cognition will be imitated by the other particles.

PSO is assumed with the following psychology: individual remembers ones own belief usually and considers colleagues' belief simultaneously in the process of searching uniform cognition. When it perceives colleagues' belief better, it can adjust adaptively.

2.2 The Process of Standard PSO

The process of standard PSO is as follows:

a). Initialize a swarm of particles(swarm size is m), including random position and random velocity;

b). Evaluate the fitness of each particle;

c). To each particle, compare the best global position undergone p_{best} with its fitness value, if it's better, it's the best current position p_{best}.

d). To each particle, compare the best position undergone g_{best} with its fitness value, if it's better, the index sign of g_{best} will be set anew.

e). Change the velocity and the position of each particle according to equation 1;

f). If it doesn't reach the termination condition (the condition is enough good fitness value or reaching a presupposition maximum algebra G), return to b).

2.3 Algorithm Parameters

PSO parameters conclude: swarm size is m, inertia weight is w, acceleration constants are c_1 and c_2, maximum velocity is $v_{\max}$, maximum algebra is $G_{\max}$.

$v_{\max}$ determines the resolution of the region between current position and the best position. If $v_{\max}$ is too high, the particle can fly over the best value, if $v_{\max}$ is too low, the particle can't make enough exploration, $v_{\max}$ falls into the local optimum. The limitation has three objectives: preventing calculation from exceeding; realizing artificial learning and attitude change; determining research size of the problem space.

w makes the particle keep motion inertia, extend the trend of searching space and suffice to explore the new region.

c_1 and c_2 represent the weight of accelerating term that each particle is pushed into the positions of p_{best} and g_{best}. The low value allows the particle linger besides target regional before the particle is pulled back, the high one leads the particle to dash to the target regional or pass over it suddenly.

3 Algorithm Process

3.1 Process of PSO

The process of PSO is as follows:

Step1: Initialize a swarm of particles (swarm size is m), set random position and random velocity of each particle on the allowable range randomly, the position of each particle determines randomly according to $x_{ij}(0)=\begin{cases}0 & rand(0,1)<0.5\\ 1 & rand(0,1)>=0.5\end{cases}$, the velocity of each particle generates randomly according to $v_{ij}(0)=v_{\min}+rand(0,1)(v_{\max}-v_{\min})$, $v_{\min}$ is the minimum of velocity, $v_{\max}$ is the maximum of velocity.

Step2: Evaluate the fitness of each particle

$$f(x_i)=\sum_{j=1}^{n}a_j x_{ij}(t)-Q\left|\min\left\{0, B-\sum_{j=1}^{n}b_j x_{ij}(t)\right\}\right|,$$

in which Q is a sufficiently large positive number, and calculate objective function of each particle;

Step3: To each particle, compare the best global position undergone p_{best} with its fitness value, if it's better than p_{best}, it's the best current position p_{best}.

Step4: To each particle, compare the best position undergone g_{best} with its fitness value, if it's better than g_{best}, it's the best swarm position ,the index sign of g_{best} will be set anew.

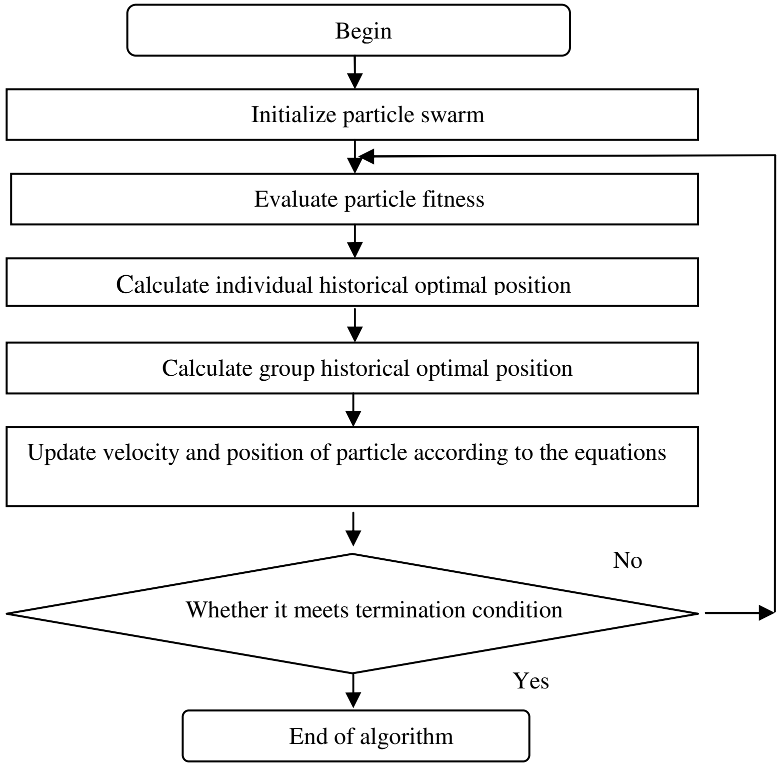

Fig. 1. The flow chart of PSO

Step5: Change the velocity and the position of each particle according to equation 1 and equation2, according to the iteration formulas:

$$v_i(t+1) = w * v_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (p_g - x_i(t)), \qquad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1). \qquad (2)$$

Step6: Check termination condition (the condition is enough good fitness value or reaching the maximum iterations, or that optimal solution changes no longer),If it meets the above condition , stop iteration; otherwise return to Step2.

The flow chart of PSO is shown as in Fig.1.

3.2 Operating Result Analysis

The program operates 50 times independently, we compare with optimal solution obtained by exact solution.

The result obtained by exact solution is 3103, we obtain that the close range is (97.132%,99.679%) with comparing optimal solution obtained by particle swarm optimization and exact solution.

Table 1. Optimized value

optimal value	3062	3067	3069	3033	3050
goods weight in knapsack	1000	999	1000	1000	1000
optimal value	3067	3060	3077	3038	3049
goods weight in knapsack	999	999	998	1000	1000
optimal value	3081	3078	3021	3062	3083
goods weight in knapsack	1000	999	1000	1000	1000
optimal value	3070	3040	3018	3054	3065
goods weight in knapsack	1000	1000	999	999	1000
optimal value	3081	3079	3014	3093	3041
goods weight in knapsack	999	999	998	1000	1000
optimal value	3078	3066	3053	3073	3036
goods weight in knapsack	1000	998	1000	1000	1000
optimal value	3052	3063	3056	3050	3069
goods weight in knapsack	999	1000	1000	1000	998
optimal value	3062	3067	3069	3033	3050
goods weight in knapsack	1000	999	1000	1000	1000
optimal value	3067	3060	3077	3038	3049
goods weight in knapsack	999	999	998	1000	1000
optimal value	3081	3078	3021	3062	3083
goods weight in knapsack	1000	999	1000	1000	1000

4 Conclusions

Particle swarm optimization obtains enlightenment from social behavior of bird group, it is a novel intelligent optimization algorithm, its implementation is simple and its effect is good. This paper applies particle swarm optimization to solve 0/1 knapsack problem, and elucidates the realization process of the algorithm. We improve basic particle swarm optimization for accelerating search ability of particle swarm. When the position of some particle equates the best position of the swarm, we make initialization assignment anew for the position of the particle and make the operation that the new particle replaces the particle adapting weak value to prevent the algorithm from falling into local optimization. Results show that the algorithm can solve 0/1 knapsack problem effectively.

References

1. Li, J., Fang, P., Zhou, M.: A Hybrid Genetic Algorithm about Solving Knapsack Problem. Journal of Nanchang Institute of Aeronautical Technology 12(3), 31–35 (1998)
2. Shen, X.J., Wang, W.W., Zheng, B.J., Li, Y.X.: Based on Improved Optimizing Particle Swarm Optimization to Solve 0-1 Knapsack Problem. Computer Engineering 32(18), 23–24 (2006)

3. http://baike.baidu.com/view/1531379.htm?fr=ala0_1
4. Liu, Q.D., Wang, L.: Research and Application on the Intelligent Particle Swarm Optimization 5 (2005)
5. Zeng, J.C., Jie, J., Cui, Z.H.: Particle Swarm Optimization. Science and Technology Press (2004)
6. Wang, X.D.: Computer Algorithm Design and Analysis. Electronic Industry Press (2001)
7. Yang, W., Li, Q.Q.: Survey on Particle Swarm Optimization Algorithm. Engineering Science 5 (2004)
8. Hembecker, F., Lopes, H.S.: Particle Swarm Optimization for the Multidimensional Knapsack Problem. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 358–365. Springer, Heidelberg (2007)
9. Ma, H.M., Ye, C.M., Zhang, S.: Binary improved particle swarm optimization algorithm for knapsack problem. Journal of University of Shanghai for Science and Technology 1 (2006)
10. Zhao, C.X., Ji, Y.M.: Particle Swarm Optimization for 0/1 Knapsack Problem. Microcomputer Development 15(10) (2005)