An Improved Shuffled Frog-Leaping Algorithm for Knapsack Problem

Zhoufang Li, Yan Zhou, and Peng Cheng

School of Information Science and Engineering
Henan University of Technology
ZhengZhou, China
lzhf1978@126.com

Abstract. Shuffled frog-leaping algorithm (SFLA) has long been considered as new evolutionary algorithm of group evolution, and has a high computing performance and excellent ability for global search. Knapsack problem is a typical NP-complete problem. For the discrete search space, this paper presents the improved SFLA (ISFLA), and solves the knapsack problem by using the algorithm. Experimental results show the feasibility and effectiveness of this method.

Keywords: shuffled frog-leaping algorithm, knapsack problem, optimization problem.

1 Introduction

Knapsack problem (KP) is a very typical NP-hard problem in computer science, which was first proposed and studied by Dantzing in the 1950s. There are many algorithms for solving the knapsack problem. Classical algorithms for KP are the branch and bound method (BABM), dynamic programming method, etc. However, most of such algorithms are over-reliance on the features of problem itself, the computational volume of the algorithm increasing by exponentially, and the algorithm needs more searching time with the expansion of the problem. Intelligent optimization problem for solving NP are the ant colony algorithm, greedy algorithm, etc. Such algorithms do not depend on the characteristics of the problem itself, and have strong global search ability. Related studies have shown that it can effectively improve the ability to search for the optimal solution by combining the intelligent optimization algorithm with the local heuristic searching algorithm.

Shuffled frog-leaping algorithm is a new intelligent optimization algorithm, it combines the advantages of meme algorithm based on genetic evolution and particle swarm algorithm based on group behavior. It has the following characteristics: simple in concept, few parameters, the calculation speed, global optimization ability, easy to implement, etc. and has been effectively used in practical engineering problems, such as resource allocation, job shop process arrangements, traveling salesman problem, 0/1 knapsack problem, etc. However, the basic leapfrog algorithm is easy to blend into local optimum, and thus, this paper improved the shuffled frog-leaping algorithm to

D. Jin and S. Lin (Eds.): Advances in CSIE, Vol. 2, AISC 169, pp. 299–303.
springerlink.com

solve combinatorial optimization problems such as knapsack problem. Experimental results show that the algorithm is effective in solving such problems.

2 The Mathematical Model of Knapsack Problem

Knapsack problem is a NP-complete problem about combinatorial optimization, which is usually divided into 0/1 knapsack problem, complete knapsack problem, multiple knapsack problem, mixed knapsack problem, the latter three kinds can be transformed into the first, therefore, the paper only discussed the 0/1 knapsack problem. The mathematical model of 0/1 knapsack problem can be described as:

$$\begin{cases} \max \sum_{i=0}^{n} x_i v_i \\ \sum_{i=0}^{n} x_i w_i \le C \quad (\mathrm{x}_i = 1 \quad or \quad 0, \quad i = 1, 2, ..., n) \end{cases} \tag{1}$$

where: n is the number of objects;

w_i is the weight of the *i-th* object(i=1,2…n);

v_i is the value of the *i-th* object;

x_i is the choice status of the *i-th* object; when the *i-th* object is selected into knapsack, defining variable x_i=1,otherwise x_i=0; C is the maximum capacity of knapsack.

3 The Basic Shuffled Frog-Leaping Algorithm

It generates P frogs randomly, each frog represents a solution of the problem, denoted by U_i, which is seen as the initial population. Then it calculates the fitness of all the frogs in the population, and arranges the frog according to the descending of fitness. Then dividing the frogs of the entire population into m sub-group of, each sub-group contains n frogs, so $P = m*n$. Allocation method: in accordance with the principle of equal remainder. That is, by order of the scheduled, the *1, 2, ..., n* frogs were assigned to the *1,2, ..., N* sub-groups separately, the *n+1* frog was assigned to the first sub-group, and so on, until all the frogs were allocated.

For each sub-group, setting U_B is the solution having the best fitness, U_W is the solution having the worst fitness, U_g is the solution having the best fitness in the global groups. Then, searching according to the local depth within each sub-group, and updating the local optimal solution, updating strategy is:

$$S = \begin{cases} \min\left\{\mathrm{int}(rand(U_B - U_W)), S_{\max}\right\}, U_B - U_W \ge 0 \\ \max\left\{\mathrm{int}(rand(U_B - U_W)), -S_{\max}\right\}, U_B - U_W \le 0 \end{cases} \tag{2}$$

$$U_q = U_w + S \tag{3}$$

where, S is the adjustment vector of individual frog, S_{max} is the largest step size that is allowed to change by the frog individual. *rand* is a random number between 0 and 1.

4 The Improved Shuffled Frog-Leaping Algorithm for KP

A frog is on behalf of a solution, which is expressed by the choice status vector of object, then frog $U=(x_1, x_2, \ldots, x_n)$, where, x_i is the choice status of the *i-th* object; when the *i-th* object is selected into knapsack, defining variable x_i=1,otherwise x_i=0; f (i), the fitness function of individual frog can be defined as:

$$f(i) = \sum_{i=1}^{n} x_i v_i \tag{4}$$

4.1 The Local Update Strategy of Frog

The purpose of implementing the local search in the frog sub-group is to search the local optimal solution in different search directions, after searching and iterating a certain number of iterations, making the local optimum in sub-group gradually tend to the global optimum individual.

Definition 1. Giving a frog's status vector U, the switching sequence $C(i,j)$ is defined:

$$C(i,j) = \begin{cases} 1, \text{ when i = j and } U_i \text{ changed} \\ 0, \text{ when i} \neq \text{j and } U_i = U_j \\ 2, \text{ when i} \neq \text{j and } U_i \neq U_j \end{cases} \tag{5}$$

where, U_i said the state of object i becomes from the selected to the cancel state, or in turn; $U_i= U_j$, object i and object j exchange places, that object i and object j are selected or deselected at the same time. $U_i \neq U_j$, object i is selected or canceled, or in turn. Then the new vector of switching operation is: $U' = U + C(i, j)$

Definition 2. Selecting any two vectors U_i and U_j of frog from the group, D, the distance from U_i to U_j is all exchange sequences that U_i is adjusted to U_j.

$$D(U_i, U_j) = \{C(i_1, j_1) \rightarrow C(i_2, j_2) \rightarrow \ldots \rightarrow C(i_m, j_m)\} \tag{6}$$

where, m is the number of adjusting.

Based on the above definition, the update strategy of the individual frog is defined as follows:

$$l = \min\left\{\text{int}[rand \times \left|D_{B,W}\right|], l_{\max}\right\} \tag{7}$$

$$s = \left\{D(U_B, U_W)\right\} \tag{8}$$

$$U_q = U_w + S \tag{9}$$

where, l is the number of switching sequence $D(U_B,U_W)$ for updating U_W; l_{max} is the maximum number of switching sequence allowed to be selected; S is the switching sequence required for updating U_W.

4.2 The Global Information Exchange Strategy

During the execution of the basic shuffled frog-leaping algorithm, the operation of updating the feasible solution was is executed repeatedly, it is usually to meet the situation that updating fail, the basic shuffled frog-leaping algorithm updates the feasible solution randomly, but the random method often falls into local optimum or reduces the rate of convergence of the algorithm.

Obviously, the key that overcoming the shortcomings of basic SFLA in evolution is: it is necessary to keep the impact of local and global best information on the frog jump, but also pay attention to the exchange of information between individual frogs. In this paper, first two jumping methods in basic SFLA are improved as follows:

$$Pn = PX + r1*(Pg - Xp1\ (t)) + r2*(PW - Xp2\ (t)) \tag{10}$$

$$\mathbf{P_n = P_b + r3*(P_g - Xp3\ (t))} \tag{11}$$

where, $Xp1(t)$, $Xp2(t)$, $Xp3(t)$ are any three different individuals which are different from X. Meanwhile, we remove the sorting operation according to the fitness value of frog individual from basic SFLA, and appropriately limit the third frog jump. Thus, we get an efficient modified SFLA based on the improvements of above. In the modified algorithm, the frog individual in the subgroup generates a new individual (the first jump)by using formula(10), if the new individual is better than its parent entity then replacing the parent individual. Otherwise re-generating a new individual (the frog jump again) by using (11). If better than the parent , then replacing it. Or when $r_4 \leq FS$ (the pre-vector, its components are $0.2 \leq FSi \leq 0.4$), generating a new individual (the third frog jump) randomly and replacing parent entity.

The new update strategy will enhance the diversity of population and the search through of the worst individual in the iterative process, which can ensure communities' evolving continually, help improving the convergence speed and avoid falling into local optimum, and then expect algorithm both can converge to the nearby of optimal solution quickly and can approximate accuracy, improved the performance of the shuffled frog-leaping algorithm.

5 Simulation Experiment

Two classical 0/1 knapsack problem instances were used in the paper, example 1 was taken from the literature_[4], example 2 was taken from the literature_[5]. The comparison algorithm used in the paper was branch and bound method for 0/1 knapsack problem. Under the same experimental conditions, two instances of simulation experiments were conducted 20 times, the average statistical results were shown in Table 1.

Table 1. Comparison of two algorithms

instance	The set of solutions	Capacity of value		Average running time (ISFLA/BABM)
		ISFLA	BABM	
1	101110101111101	1042/878	1037/878	1.83/2.35
2	1101110101101001101110010001101000101 1	3025/989	3025/989	6.73/8.21

6 Conclusion

The shuffled frog-leaping algorithm is a kind of search algorithm with random intelligence and global search capability, this paper improved shuffled frog-leaping algorithm and solved the 0/1 knapsack problem by using the algorithm. Experiments show that the improved algorithm has better feasibility and effectiveness in solving 0/1 knapsack problem.

References

1. Eusuff, M.M., Lansey, K.E.: Optimization of water distribution network design using the shuffled frog leaping algorithm. Water Resource Planning and Management 129(3), 210–225 (2003)
2. Li, Y.-H., Zhou, J.-Z., Yang, J.-J.: An improved shuffled frog-leaping algorithm based on the selection strategy of threshold. Computer Engineering and Applications 43(35), 19–21 (2007)
3. Luo, X.-H., Yang, Y., Li, X.: Improved shuffled frog-leaping algorithm for TSP. Journal of Communication 30(7), 130–135 (2009)
4. He, Y.-C.: Greedy genetic algorithm and its application for KP. Computer Engineering and Design 28(11), 19–22 (2007)
5. Wu, Z.-H.: Algorithm Design and Analysis, pp. 251–252. Higher Education Press, Beijing (1993)