Taming a 0-1 knapsack problem with monkey algorithm

Xiaoping Xu1,*, Li Xu1, Xiao Zhang1, Daqing Zhang1, Yi Luo1, Long Liu2 and Fucai Qian2

1School of Sciences, Xi’an University of Technology, Xi’an, China
2School of Automation and Information Engineering, Xi’an University of Technology, Xi’an, China

*Corresponding author e-mail: xuxp@xaut.edu.cn

Abstract. Now, the 0-1 knapsack problem has a wide range of applications in real life, but it is a typical NP hard problem. thereby, solving the 0-1 knapsack problem has been widely concerned by many researchers. This paper puts forward an improved monkey algorithm where uniform distribution’s Kent chaotic map is adopted as the initial feasible solution of the algorithm, and the descending factor is used as the step size in the climb process of the algorithm. Finally, in numerical simulation, a 0-1 knapsack problem is solved using the presented method and other existing methods to show that the given method is effective.

1. Introduction

In the 1950s, Dantzig came up with the knapsack problems for the first time [1], and then it was widely concerned. Moreover, in practical applications, many industrial and financial problems can be described by knapsack problems [2], for example, the idealized cutting problem, the choice of prizes for customers after winning the prize, construction industry, intelligent editing and transcribing of CD tracks, optimized cutting of strip steel, printing and typesetting tasks, business, combinatorics, computational complexity theory, cryptography, applied mathematics, cargo loading, capital investment, financial portfolio, material cutting, object packing, etc. As we all know, in real problems, with the increasing scale of the problems, the complexity is also increasing. Consequently, the 0-1 knapsack problem is regarded as a NP hard problem studied most in recent years [2]. Accordingly, knapsack problem has become a hot topic among many scholars. That is to say, solving 0-1 knapsack problem is an important problem at present [3].

In 2008, Zhao and Tang monkey put forward monkey algorithm for the first time based on the daily mountain-climbing behaviors of the monkeys [4]. Moreover, compared other intelligent algorithm, monkey algorithm has many merits, such as better convergence, better stability, better accuracy, better search capability and so on. Thereby, many complex practical engineering problems have the monkey algorithm’s shadow of being widely used, for example, detection technology [5], vehicle energy management [6], rigid Frame bridge [7] and others. In the manuscript, an improved monkey algorithm which combines uniform distribution’s Kent chaotic map and descending factor is proposed to solve the 0-1 knapsack problem. Finally, simulation results show that the given method is valid in numerical simulation.
2. 0-1 knapsack problem

The number of items to be set is $n$, the volume (weight) and price of item $i$ are $w_i$ and $p_i$ respectively, the maximum capacity of a backpack can be $V$, now some items are chosen and put them in the backpack. Accordingly, in order to satisfy the knapsack constraint, It is necessary to consider how to choose the goods to maximize the total value of the goods in the backpack, and the problem is called the following 0-1 knapsack problem.

$$\max f = PX = \sum_{i=1}^{n} p_i x_i \quad (1)$$

subject to

$$LX = \sum_{i=1}^{n} l_i x_i \leq V \quad (2)$$

where, $P=(p_1,p_2,...,p_n)$ represents the value vector of an item , $L=(l_1,l_2,...,l_n)$ denotes the corresponding volume (weight) vector of the item, $X=(x_1,x_2,...,x_n)$ expresses the vector of the solution , Eq. (1) represents the objective function, $x_i$ is the decision variable in Eq. (2).

3. Monkey algorithm

3.1. Basic monkey algorithm

Basic monkey algorithm mainly includes the representation and initialization of the solution, the process of climbing, the process of watching, and the process of jumping. Its specific operations are found in Reference [4].

3.2. Improved monkey algorithm

When basic monkey algorithm were applied, we found that the algorithm initialization and climbing process of the algorithm to optimize performance is more obvious, so here initialization and climbing process in the algorithm are improved to improve the algorithm accuracy, which is as follows.

(1) Improvement of solution initialization

Setting initialization position of monkey often would have a certain impact on the optimal results. Generally, the initial position distribution is more uniform, the better result is obtained. The Lyapunov index is a measure of the divergence rate of uncertainty over time, the divergence rate is greater, the distribution is more homogeneous. Here, in order to make the initial position distribution is more uniform, the initial position of monkey is constructed by Kent chaotic map, and it’s Lyapunov index is 0.696 [8]. Moreover, in the Kent chaotic mapping, when the parameter $\varepsilon=0.4$, $a_j$ is the uniform distribution of $(0,1)$. Therefore, Kent chaotic mapping has the characteristics of ergodicity and repeatability, and the specific function is generated by the following Eq. (3).

$$a_{j+1} = \begin{cases} 
\frac{a_j}{\varepsilon} & 0 < a_j \leq \varepsilon \\
1 - \frac{a_j}{1-\varepsilon} & \varepsilon < a_j \leq 1 
\end{cases} \quad (3)$$

where, $a_j$ is a chaotic variable, $\varepsilon_j \in [0,1]$, $j=1,2,...,n$, $\varepsilon$ is a parameter, when $\varepsilon=0.4$, the $a_j$ is a uniform distribution in $(0,1)$, $x_{ij}$ can be generated by the following Eq. (4).

$$x_{ij} = x_{\min} + h_j (x_{\max} - x_{\min}) \quad (4)$$

where, $x_{ij}$ is the specific position of the monkey $i$ in the dimension $j$, $[x_{\min},x_{\max}]$ is the range of $x_{ij}$.

Since $\varepsilon_j$ is uniformly distributed in $(0,1)$, $x_{ij}$ is distributed uniformly using Eq. (4) in $[x_{\min},x_{\max}]$. The initial position is uniformly distributed, which facilitates the monkey to search carefully next step, which largely avoids escaping the optimal value. Thus, the search time is saved and the amount of calculation is reduced. Namely, the optimal position is quickly found. Therefore, it is beneficial to enhance the precision of the whole algorithm, and also accelerates the convergence speed of the algorithm.
(2) Improvement of climbing process

In the climbing process of basic monkey algorithm, climbing step of the algorithm often decides the accuracy of search and CPU time, climbing step value in a certain range is smaller, the search precision of the algorithm is higher, the corresponding CPU needs more time, and vice versa. Since the Sigmoid function is continuous, derivable, bounded, smooth and strictly monotone, it is a class of incentive functions [9]. It can map variables to the interval [0,1] and the variables become smaller in the interval. Therefore, the introduction of the Sigmoid function in the process can get the gradual acceleration, which can get a balance between the accuracy of the solution and the search speed. Here, the diminishing factor constructed is as follows.

\[
d_{k+1} = \frac{1}{1 + e^{\frac{2\ln100 - \ln n_{\text{Max}}}{n_{\text{Max}} \cdot K} \cdot \ln n_{\text{Max}}}} \cdot a_K
\]

where, \( K \) represents the present iteration number, \( n_{\text{Max}} \) is the iteration maximum number, and \( a_K \) is the climbing step of the \( K \)th iteration.

4. Solving 0-1 knapsack problem by improved monkey algorithm

When the 0-1 knapsack problem is solved by the above mentioned modified monkey algorithm, Eq. (1) is regarded as the target function to find the maximum value of Eq. (1). In the solution, their corresponding relation is as follows. The position of each monkey is equivalent to a set of feasible solutions of Eq. (1). The population size of monkey corresponds to the number of feasible solution. The dimension corresponds to the quantity of the item. In this paper, binary encoding is adopted, and the following steps are executed:

Step 1: Set algorithm’s parameters, for example, \( M, n_{\text{Max}} \) and other parameters.

Step 2: Binary vectors \( X_i=(x_{i1}, x_{i2}, ..., x_{in}) \) generated using Eqs. (3) and (4) are as the initial position of all monkeys. And the object function values of all monkeys are calculated to find better position.

Step 3: Based on Eq. (5), the climbing step of the climbing process is transformed from fixed value to alterable step to perform climbing process. Therefore, new binary vectors \( Y_i=(y_{i1}, y_{i2}, ..., y_{in}) \) are generated, moreover, \( f(Y_i) \) are computed. If \( f(Y_i)>f(X_i) \), the present location of monkey \( i \) is updated, that is, \( X_i=Y_i \). Until the process continues to the preset maximum number \( N_c \).

Step 4: After performing the watching process, new binary vectors \( Y_i=(y_{i1}, y_{i2}, ..., y_{in}) \) are generate in the field of view \([x_{ij}-b, x_{ij}+b]\), \( f(Y_i) \) are calculated. If \( f(Y_i)>f(X_i) \), the current position of monkey \( i \) is updated, i.e., \( X_i=Y_i \).

Step 5: After performing the jumping plication, new binary vectors \( Y_i=(y_{i1}, y_{i2}, ..., y_{in}) \) are generate, \( f(Y_i) \) are calculated. If \( f(Y_i)>f(X_i) \), the present location of monkey \( i \) is updated, i.e., \( X_i=Y_i \).

Step 6: Repeat the Step3 - Step5 until the \( n_{\text{Max}} \)is scheduled. The algorithm ends, output optimal individual and optimal values.

5. Numerical simulation

To illustrate the feasibility of the given improved monkey algorithm (IMA), a numerical example are given below. Moreover, differential evolution algorithm (DEA) [7], particle swarm optimization algorithm (PSO) [8], genetic algorithm (GA) [9] and basic monkey algorithm (BMA) [4] are also used to solve the problem.

Example: the quantity of items \( n=100 \), the capacity of knapsack \( V=2010 \), the value of items \( P=[68, 101, 125, 159, 65, 146, 28, 92, 143, 37, 5, 154, 183, 117, 96, 127, 139, 113, 100, 95, 12, 134, 65, 112, 69, 45, 158, 60, 142, 179, 36, 43, 107, 143, 49, 6, 130, 151, 102, 149, 24, 155, 41, 177, 109, 40, 124, 139, 83, 142, 116, 59, 131, 107, 187, 146, 73, 30, 174, 13, 91, 37, 168, 175, 53, 151, 125, 31, 192, 138, 88, 184, 110, 159, 189, 147, 31, 169, 192, 56, 160, 138, 84, 42, 151, 37, 21, 22, 200, 85, 135, 200, 139, 189, 68, 94, 84, 22, 18, 115] \), the weight of items \( W=[42, 35, 70, 79, 63, 6, 82, 62, 96, 28, 92, 3, 93, 22, 19, 48, 72, 70, 68, 36, 4, 23, 74, 42, 54, 48, 63, 38, 38, 24, 30, 17, 91, 89, 41, 65, 47, 91, 71, 7, 94, 30, 85, 57, 67, 32, 45, 27, 38, 19, 30, 34, 40, 5, 78, 74, 22, 25, 71, 78, 98, 87, 62, 56, 56, 32, 51, 42, 67, 8, 8] \).
In the simulation, when the above problems are solved by IMA, $M=20$, $n_{\text{Max}}=500$, climbing step size $a=0.1$, climbing step accuracy $1\times10^{-6}$, the iterations number of climbing $N_c=20$, visual field length $b=0.5$, the iterations number of watching $N_w=3$, and jumping interval $[c,d]=[-1,1]$. Based on the above five algorithms, Example is solved fifty times respectively, and their optimal values, the worst values, the average values and the variances are showed in Table 1, respectively.

| Algorithms | Optimal values | Worst values | Average values | Variances   |
|------------|----------------|--------------|----------------|-------------|
| DEA        | 5809           | 4818         | 5.2455e+003    | 6.9042e+004 |
| PSO        | 7391           | 5499         | 6.9499e+003    | 1.6198e+005 |
| GA         | 6701           | 4906         | 5.9956e+003    | 1.5427e+005 |
| BMA        | 7034           | 6176         | 6.6096e+003    | 3.4405e+004 |
| IMA        | 7968           | 7765         | 7.8904e+003    | 1.4831e+003 |

From Table 1, we can see that the solving accuracy of IMA is relatively high, and the variance is also relatively small. Namely, it shows that the proposed IMA has higher feasibility and stability to solve the 0-1 knapsack problem.

6. Conclusions
When using monkey algorithm solves 0-1 knapsack problem, an improved monkey algorithm combining uniform distribution’s Kent chaotic map and descending factor in the basic monkey algorithm is proposed to avoid detours. Simulation results express that the 0-1 knapsack problem is settled via the given IMA, and the expected results are achieved.

7. Acknowledgments
This work was financially supported by National Natural Science Foundation of China (grant nos. 61773016 and 61673318), Innovative Training Grogram for College Student of Shaanxi Province (grant no. S201910700025) and Innovation Capability Support Program of Shaanxi (Program No. S2020-ZC-PT-0023).

References
[1] G.B. Dantzig, Discrete variable extremum problems, Oper. Res. 5 (1957) 266–277.
[2] M. Ralph, H. Martin, Hiding information and signatures in trapdoor knapsacks, IEEE Trans. Inform. Theory 24 (1978) 525-530.
[3] V. Srinivasan, G. Varghese, Fast address lookups using controlled prefix expansion, ACM Trans. Comput. Syst. 17 (1999) 1-40.
[4] R.Q. Zhao, W.S. Tang, Monkey algorithm for global numerical optimization, J. Uncertain Syst. 2 (2008) 164–175.
[5] J. Zhang, Y. Zhang, J. Sun, Intrusion detection technology based on monkey algorithm, Comput. Engin. 37 (2011) 131-133.
[6] C. Shen, T. Gao, Hybrid electric vehicle energy management strategy based on monkey algorithm, Comput. Engin. Appl. 50 (2014) 9-13.
[7] X. Li, F. Ni, W. Sun, S. Cao, Research on optimal sensor placement of continuous rigid Frame bridge based on monkey algorithm, Highway 61 (2016) 65-69.
[8] Z. Chen, Z. Qian, “Kent chaos mapping application in the digital fountain codes,” in Proceedings of the 30th Chinese Control Conference, Yantai, China, 2011, pp. 4371-4376.
[9] X. Zhang, X. Huang, W. Zhong, L. Zhang, FPGA implementation of Sigmoid function and its derivation function, J. Fujian Normal Univ. 27 (2011) 62-65.
[10] Y. Jing, Differential evolution algorithm for knapsack problem, Brilliance 7 (2011) 92-93.
[11] C. Zhao, Y. Ji, Application of particle swarm optimization in 0-1 knapsack problem, Micro. Develop. 15(2005) 23-25.
[12] W. Chen, Y. Lin, L. Wang, Application of genetic algorithm in solving knapsack problem, J. Guyuan Tea. Coll. 24 (2003) 31-34.