2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)

Ant Colony Optimization Parameters Control Based on Evolutionary Strength

Chen Chang
Army Engineering University
NanJing, China
c308051252@163.com

Jianjun Cao
The Sixty-third Research Institute, National University of Defense Technology
NanJing, China
jianjuncao@yeah.net

Nianfeng Weng
The Sixty-third Research Institute, National University of Defense Technology
NanJing, China
wengnf@hotmail.com

Guojun Lv
Army Engineering University
NanJing, China
11983286838@outlook.com

***Abstract*—Ant Colony Optimization (ACO) was developed to be a metaheuristic and widely used for combinatorial optimization problems. It is confirmed that the parameters setting is of vital importance for performance of ACO. This paper proposed a strategy to perform ACO Parameters Control based on Evolutionary Strength (ACOP_ES). First, we transform four parameters setting problem to decision making problem. Then, by learning the evolutionary strength curve, we set sum, peak, and kurtosis of evolutionary strength curve as the objective function, such that the continuous ant colony algorithm can be used to optimize this decision problem. Further, via characterizing an excellent evolutionary curve, a relatively optimal parameters setting can be found. Compared with existing methods, our proposed method focuses on the performance of the running process rather than the results only. To the best of our knowledge, we are among the first to introduce process performance evaluation of ACO to guide parameters control, and determine the relatively optimal parameters of ACO. At last, we apply our method to Multi-dimensional Knapsack Problem (MKP), the results look promising that parameters returned by ACOP_ES are better than the ones returned by other methods on datasets: WEISH*.DAT and WEING*.DAT.**

Keywords—ant colony optimization, parameter control, evolutionary strength

I. Introduction

Inspired by colonies of real ants, ACO is a classical heuristic global optimization algorithm proposed by Dorigo et al. [1]-[2]. It is a population-based constructive metaheuristic that exploits a form of past performance memory inspired by the foraging behavior of real ants. The behavior of the ACO algorithm highly relies on its parameters. Adaptation and parameter control are recurring themes in the field of bio-inspired algorithms like ACO.

The behavior of ACO depends strongly on the values given to parameters [3]. According to [1], significance weights for pheromone α, significance weights for heuristic information β, pheromone evaporation rate ρ and number of ants for each colony M are four important parameters that determine the performance of ACO. Pheromone is represented as numerical information that is modified iteratively to reflect the algorithm's search experience. Heuristic information can be derived from a problem instance to guide ants in the solution construction process [4].

MKP is one of the most intensively studied discrete programming problems [5]. Many practical problems can be formulated as MKP, such as the capital budgeting problem, allocating processors and databases in a distributed computer system, project selection and cargo loading, and cutting stock problem [6]. We use MKP as test problem to evaluate the performance of ACO with the parameters selected by ACOP_ES.

The rest of the paper is organized as follows: the next section reviews some recent works in the field of parameter adaptation. In Section III, the basics of evolutionary strength and parameter control strategy are briefly introduced. Then, we propose ACOP_ES and describe the implementation of this model in Section IV. In Section V, experiments setup and results on WEISH*.DAT and WEING*.DAT are discussed. Finally, we conclude this paper in the last section.

II. Related Work

The dependency of the performance of metaheuristics on the setting of their parameters is well known. In fact, finding appropriate setting for an algorithm's parameters is considered to be a nontrivial task and a significant amount of works has been devoted to it [7]. The approaches for solving this task can roughly be divided into offline and online methods.

A. *Offline Methods*

Offline tuning has the goal of finding appropriate settings for an algorithm's parameters before the algorithm is actually deployed. In [8]-[10], authors analyzed the relation between the parameters, and some combinations of parameter values were also presented. It is obvious that the proper combination of parameter values is problem-dependent, and the proper combination of parameter values for an instance may not be good for the other instances. Moreover, they may not be the optimal values for the system due to the intrinsically dynamic and adaptive characteristic of the ACO. In other words, parameters they proposed may not fit for the work at different stages of the search process. Uniform design is an important means of off-line parameter selection of ACO [11]-[12]. In this way, the parameter setting problem in the basic model of uniform design is described as a multi-factor and multi-level experimental design. As we can easily infer, for offline methods, there is no feedback during the run, but a proper combination of parameters must be adapted based on the search progress in practice.

B. *Online Methods*

Online tuning consists of the modification of an algorithm's parameters setting while solving a problem instance.

An alternative for online tuning is to use adaptive parameters setting. Some methods adapt the parameters based on a pre-scheduled number of iterations. Stützle et al.

2375-0197/19/$31.00 ©2019 IEEE
DOI 10.1109/ICTAI.2019.00069

examined two deterministic ant system variants: First, the number of ants used increases by some iterations until the value up to the number of variables. Second, the parameters of exploration decreases from 0.99 to 0.0 [13]. Similarly, Liu et al. take more parameters into consideration [14]. The same strategy was followed by Y. Lin et al. which adopt adaptive ρ and α for ACO, and update parameters if the current optimal solution is not improved after N iterations. During the early stage of the search, the value of α is small enough to allow extensive exploration of the search space, and the value of α increases over time to improve the local search ability of the algorithm [15]. These methods mentioned above adjust the parameters by a formula or an evaluation measure, and most could only predict the microarray data well, but lack stability on the other data.

For adaptive methods, parameters are modified according to some rules which takes into account the search behavior of the ACO algorithm. Some measures are proposed to evaluate parameters behavior of ACO include average λ-branching, entropy-based measures for the pheromone, dispersion of solutions generated, or simply the quality of the solutions [16]-[18]. Later, some researches use the measures as a driver of adaptation to finish parameter control [19]-[21]. In [22], the ants construct their solutions and update the memory of the pheromone, in which the amount of rewards assigned depends on the way of feedback collected.

Some researches employed fuzzy logic control to adjust the parameters in ACO. In [23], three adaptive parameters control strategies based on fuzzy logic control are proposed, which adjust ρ, α and M respectively. In this method, only one parameter is allowed to be dynamically adjusted, the remaining parameters should be statically specified according to the parametric guidelines provided by the paper. Olivas F et al. proposed a dynamic parameter adaptation strategy for ACO based on interval type-2 fuzzy logic systems. The idea is able to apply this new ACO method with parameter adaptation to a wide variety of problems without the need of finding the best parameters for each particular problem [24]. These methods individually optimize the parameters without considering the combination of parameters, and it is difficult to determine a good combination of parameters.

III. Overview

A. Evolutionary Strength

Cao et al. proposed an evaluation methodology based on evolutionary strength for heuristic algorithms so as to assess the process performance of different heuristic algorithms[25], which took the subset problem for example, tested four different proposed algorithms, and evaluated the performance of four different algorithms through the trend of evolutionary strength. The details are shown below.

Take the MKP for example. Firstly, Tanimoto distance $D_{i(i\text{-}1)}$ is introduced to measure the difference between two feasible solutions at iteration i and i-1 when solving MKP by ACO.

$$D_{i(i-1)} = \frac{|R_i| + |R_{i-1}| - 2|R_i \cap R_{i-1}|}{|R_i| + |R_{i-1}| - |R_i \cap R_{i-1}|} \tag{1}$$

Where R_i and $R_{i\text{-}1}$ are feasible solutions of MKP found by algorithms, $|R_i|$ and $|R_{i\text{-}1}|$ are the cardinal number of R_i and $R_{i\text{-}1}$, $D_{i(i\text{-}1)} \in [0,1]$. $D_{i(i\text{-}1)}$=1 when R_i is equals to $R_{i\text{-}1}$, $D_{i(i\text{-}1)}$=0 when R_i is completely different from $R_{i\text{-}1}$.

Secondly, relative evolutionary range δ_i is defined as follows.

$$\delta_i = \frac{V_i - V_{i-1}}{V_b - V_{i-1}} = \begin{cases} 1 & i = 1 \\ 0 & V_b = V_{i-1} \\ \dfrac{V_i - V_{i-1}}{V_b - V_{i-1}} & others \end{cases} \tag{2}$$

Where V_b is the global optimal objective value, V_i is the objective value of iteration i.

Thirdly, evolutionary strength I_i at iteration i is introduced on the basis of above discussion.

$$I_i = \frac{\delta_i}{D_{i(i-1)}} = \begin{cases} 0 & D_{i(i-1)} = 0 \\ \dfrac{\delta_i}{D_{i(i-1)}} & others \end{cases} \tag{3}$$

From (3), the lager relative evolutionary range, the smaller difference of solutions, the better evolving of this iteration can be obtained. In other words, the larger evolutionary strength, the better evolving of iteration i is.

B. Parameter Control

From [25], we know that different variant of ACO has different evolutionary strength curves. Based on this knowledge, we find that ACO with different parameters combinations also performs different trend information about evolutionary strength. Fig.1 shows three typical evolutionary strength curves.

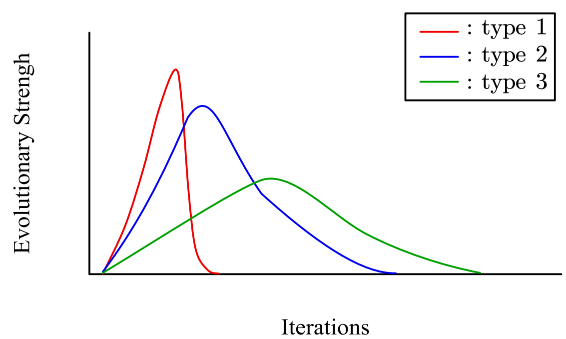

Fig. 1 Typical evolutionary strength curve

For type1, the rapid decline after reaching the peak of evolutionary strength indicates that this combination of parameters has fallen into a local optimal. For type3, it has a long time evolutionary, but after a long period of iterations, it still cannot converge and is not considered a good combination of parameters. For type2, this combination has more evolutionary iterations and stronger evolutionary strength. Obviously, performance of this type is significantly better. We use sum, peak and kurtosis of evolutionary strength curve as target value IN, PE, KU to guide ants to find the relatively optimal curve like type2. At the same time, relatively optimal parameters combination for ACO can be found.

Assume that N is number of total iterations. IN, PE, KU are defined as follows.

$$IN = \sum_{i=1}^{N} I_i$$

$$PE = \max_{0<i\leq N} I_i \tag{4}$$

$$KU = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{I_i-\overline{I}}{\sigma}\right)^4$$

Where I_i is the evolutionary strength of iteration i when solving MKP by ACO, $\overline{I}$ and σ are the mean value and the standard deviation of evolutionary strength of N iterations. IN is the sum of evolutionary strength of N iterations, PE is the peak of the evolutionary strength of N iterations, and KU is the kurtosis evolutionary strength of N iterations. Through the analysis of the curve, the bigger IN, PE, the smaller KU, the better curve we can get. The method proposed aims to find the relatively optimal evolutionary strength curve, and determines the relatively optimal parameters combination for ACO.

IV. Proposed Model

A. Objectives and Constraints

In parameters' control for ACO, we consider four important parameters α, β, ρ, M which determine the performance of ACO. The objective functions to be handled for our model are as follows.

$$\begin{gathered} \max\ IN(\alpha,\beta,\rho,M) \\ \max\ PE(\alpha,\beta,\rho,M) \\ \max\left(1-\max\left(KU(\alpha,\beta,\rho,M)\right)\right) \\ \text{s.t. } 0<\alpha<5,\ 0<\beta<5, \\ 0<\rho<0.5,\ 0<M<30 \end{gathered} \tag{5}$$

The objectives are illustrated as follows.

(1) The model hopes to obtain a combination of parameters, so that the evolutionary strength curve tends to be relatively optimal. The priority of three objectives functions are from high to low.

(2) Each parameter is associated with a range of values. The bounds of the ranges are set based on [26], it confirms that the relatively optimal parameters can be found with such a range.

(3) In order to facilitate the solution, three objectives are converted into one objective F which is the weighted sum of three objectives.

$$\begin{gathered} F = \max\left(\lambda_0 IN + \lambda_1 PE + \lambda_2\left(1-\max(KU)\right)\right) \\ \text{s.t. } \lambda_0,\lambda_1,\lambda_2>0,\ \lambda_0+\lambda_1+\lambda_2=1 \end{gathered} \tag{6}$$

B. ACOP_ES

In our model, four parameters setting problem is transformed to a decision making problem. The continuous ant colony algorithm is used to optimize this decision problem. Suppose that the total number of window translations is K_1, and for each group, the number of iterations is K_2. The details of procedure are explained as follows.

(1) window setting

Assuming that the windows in group $D(D=0,1,2,...K_1)$ of parameters α, β, ρ and M are $W_\alpha^D=[W_{\alpha L}^D, W_{\alpha R}^D]$, $W_\beta^D=[W_{\beta L}^D, W_{\beta R}^D]$, $W_\rho^D=[W_{\rho L}^D, W_{\rho R}^D]$ and $W_M^D=[W_{ML}^D, W_{MR}^D]$. W_{*L}^D and W_{*R}^D are the start and the end of window of parameter * in group D. Each window is divided into T portions, and the precision of each window is $\varDelta_\alpha$, $\varDelta_\beta$, $\varDelta\rho$, and $\varDelta_M$. At first, a uniform test is done to find a relatively optimal solution, and use this solution as the center of initial window $W_{\alpha C}^0$, $W_{\beta C}^0$, $W_{\rho C}^0$, *and* W_{MC}^0, such that the search is run on the basis of initial group. As in Fig. 2, the search window will move within its own range, and different groups have different combinations of search windows until no better solutions can be found.

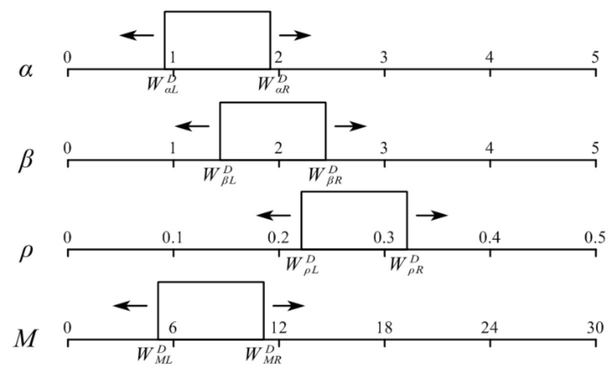

Fig. 2 Window setting of group D

Assume algorithm is running in the search window group D. As shown in Fig. 3, the four-variable continuous parameters optimization problem is translated into a decision making problem, and each parameter has T+1 grid points which pheromone is laid on the points. Such that the ant will choose grid point according to the pheromone and heuristic on it. Take Fig. 3 for example, the solution of this situation is $\alpha=1.1, \beta=1.9, \rho=0.25, M=8$.

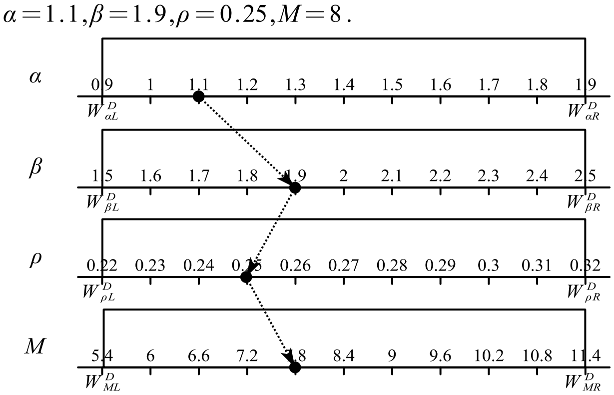

Fig. 3 Ant search of group D

For a partial solution being built by each ant at iteration $d(d=1,2,...,K_2)$, the probability $h_{y_k}(d)$ of selecting grid point y from T+1 points for k-th parameter as the next point is

$$h_{y_k}(d) = \frac{\tau_{y_k}^a(d-1)\eta_{y_k}^b}{\sum_{j=0}^{T}\tau_{j_k}^a(d-1)\eta_{j_k}^b} \quad \begin{array}{l} k=0,1,2,3 \\ ,j=0,1,2,...,T, \\ d=0,1,2,...K_2 \end{array} \tag{7}$$

where $\tau_{y_k}(d\text{-}1)$, $\eta_{y_k}(d\text{-}1)$ is the pheromone and heuristic of grid point y in for k-th parameter at iteration d-1. a and b are the importance factor of pheromone and heuristic for our model. The heuristic η_{y_k} is computed by

$$\begin{aligned} &\eta_{y_0} = |W_{\alpha L}^D + y\Delta_\alpha - W_{\alpha C}^0| \\ &\eta_{y_1} = |W_{\beta L}^D + y\Delta_\beta - W_{\beta C}^0| \\ &\eta_{y_2} = |W_{\rho R}^0 - (W_{\rho L}^D + y\Delta_\rho)|\frac{\Delta_\alpha}{\Delta\rho}. \\ &\eta_{y_2} = |W_{MR}^0 - (W_{ML}^D + y\Delta_M)|\frac{\Delta_M}{\Delta\rho} \\ &\Delta_\alpha = \Delta_\beta = 0.1,\ \Delta\rho = 0.01,\ \Delta_M = 0.6 \end{aligned} \tag{8}$$

From (8), each ant will select grid points which are farther away from the window center, that is, each window update has a farther translation to quickly reach the relatively optimal area. At the same time, the heuristic information of different types of variables is balanced by the grid width ratio. Moreover, the heuristic information has dynamic adaptability with window translation. With its farther away from the initial grid point of the window, the heuristic information is used relatively weak.

When a colonies' search finished, pheromone update strategy is given as follows.

$$\tau_{y_k}(d) = \begin{cases} (1-c)\tau_{y_k}(d-1) + \frac{F_{d-1}}{Q}, & y_k \in U_{d-1} \\ (1-c)\tau_{y_k}(d-1), & others \end{cases} \tag{9}$$

Where c is pheromone evaporation factor, $F_{d\text{-}1}$ is the optimal objective value of iteration d-1, Q is a parameter of the model which is decided by the special issues, and $U_{d\text{-}1}$ is the set of grid points which does correspond with the relatively optimal solution found in iteration d-1.

(2) Window translation

After searching of group D with K_2 iterations, the windows center of group D+1 will be translated to optimal solution found by window groups from 1 to D. In order to make the window not exceed the definition domain of the four parameters. Take α for example, the center of window α in group D+1 is controlled by

$$W_{\alpha C}^{D+1} = \begin{cases} W_{\alpha C}^{D+1} & 0.5 \leq W_{\alpha C}^{D+1} \leq 4.5 \\ 0.5 & W_{\alpha C}^{D+1} < 0.5 \\ 4.5 & W_{\alpha C}^{D+1} > 4.5 \end{cases}. \tag{10}$$

Other window centers of β, ρ, M are controlled similar to the way of α. If the window center does not translate for K_3 groups, output the optimal solution found so far as the parameters selected by ACOP_ES.

C. The Outline of ACOP_ES

The ACOP_ES is illustrated as follows:

INPUT

a: Significance weights for pheromone
b: Significance weights for heuristic
c: Pheromone evaporation rate
e:Number of ants for each colony
Q: Constant determined by specific issue
K_3: Stop condition for window translation (outer loop)
K_2: The number of iterations in each window group (inner loop)
Δ_α, Δ_β, Δ_ρ, Δ_M : Precision of each window

OUTPUT

α, β, ρ, M :parameters selected by ACOP_ES

INITIALIZATION

η_{y_k}: Heuristic information matrix
$W_\alpha^0, W_\beta^0, W_\rho^0, W_M^0$:Search window of group 0

LOOPING PROCEDURE

While (stopping criteria of outer loop K_3 not met) {
Initialize pheromone matrix $\tau_{y_k}(0)$
  While (stopping criteria of inner loop K_2 not met)
    For each ant
      Gradually select grid point of parameter based on Equations (7)
      Use parameters selected as the parameters of ACO to solve the MKP and calculate the objective function value based on Equations (6)
      Compare and update the best parameter combination for this iteration
    End For
    Update pheromone matrix based on Equations (9)
    Compare and update the best parameters of this group
` **End while**
Search window translation
Compare and update the global best parameters selected
End while
Output the global best solution as parameters selected by ACOP_ES

The general scheme of ACOP_ES is shown in Fig. 4. By extending the parameter application and parameters to be controlled, ACOP_ES can be widely used by other methods for parameters control.

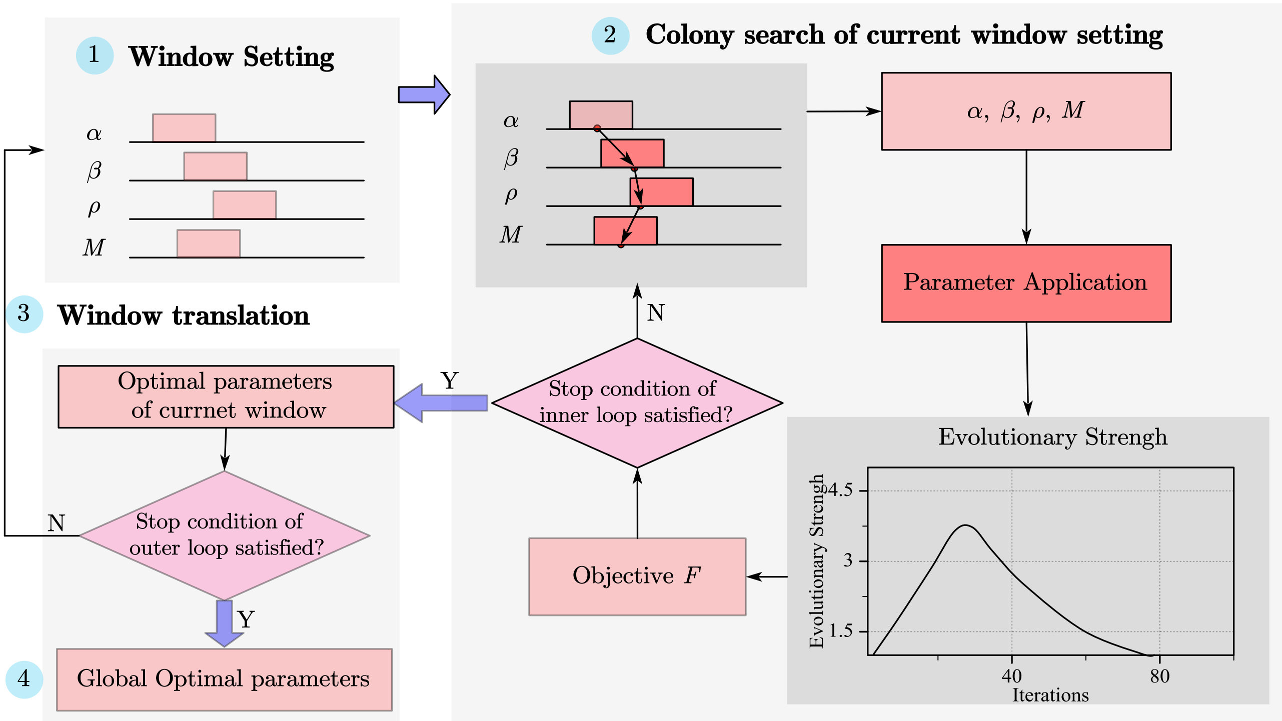

Fig. 4 The general scheme of ACOP_ES

V. EXPERIMENTS AND ANALYSIS

A. Experiment Design

The goal of this section is to evaluate the proposed model ACOP_ES against the state-of-the-art methods including uniform design [12], FAACO [15], and PSACO [22]. For uniform design, we use a $U_{15}(15^4)$ uniform table (Table I) which contains 15 groups of parameters setting. For FAACO, the parameters used are the same with [15]. What's more, to make a fair comparison, we set the initial parameters of FAACO are the same with the initial window centers of ACOP_ES too. For PSACO, the rewards given to the parameters during the run are based on the best-so-far solution in colony-level rather than ant-level. For all methods, the bounds of all parameters to be selected are between the constant values defined in (5).

TABLE I UNIFORM DESIGN TABLE $U_{15}(15^4)$

Group	α	β	ρ	M
Group 1	2.45	4.19	0.303	20
Group 2	4.19	2.45	0.129	20
Group 3	1.58	1.29	0.129	20
Group 4	3.32	3.32	0.5	20
Group 5	3.03	5	0.187	20
Group 6	5	1.87	0.332	20
Group 7	1	2.74	0.245	20
Group 8	4.48	4.77	0.448	20
Group 9	1.87	3.61	0.1	20
Group 10	3.61	1	0.274	20
Group 11	2.16	2.16	0.477	20
Group 12	4.77	3.9	0.216	20
Group 13	1.29	4.48	0.361	20
Group 14	2.74	1.58	0.158	20
Group 15	3.9	3.03	0.39	20

The experiments have been carried out on the same computer (a Pentium IV 3.6GHz CPU and 4GB RAM). The programming language is matlab. A collection of MKPs named WEISH*.DAT and WEING*.DAT are from [27]-[28]. ACO stops after 300 generations when solving MKPs for all four methods. It should be noted that MKPs are considered examples to prove the validity of the model, which is also applicable to other problems.

B. Experiment Results and Analysis

(1) ACOP_ES performance analysis

Table II shows the parameters selected by ACOP_ES for all six datasets.

TABLE II PARAMETERS SELECTED BY ACOP_ES

Dataset	Parameter Selected
WEISH1	[0.7,2.4,0.01,4]
WEISH2	[1.2,1.8,0.01,10]
WEISH3	[0.3,4.7,0.03,6]
WEING1	[0.2,1.8,0.02,10]
WEING2	[2.2,4.8,0.13,3]
WEING3	[0.1,2.1,0.02,12]

To demonstrate the performance of ACO with parameters selected by ACOP_ES, Fig. 5 shows the evolutionary strength curve and convergence curve when solving MKPs using the parameters selected in Table II. Obviously, the evolutionary strength curve is consistent with expectation from [25], it can be speculated that the parameters makes the ACO have an excellent performance. First, during the operation of the ACO, the better solution is continuously found, and when it reaches the peak, it still has a large evolutionary strength, and does not fall into the local optimal. Second, during the iterative process, the ACO maintains a high evolutionary strength, and has a relatively superior convergence speed. For WEING2, an abnormal curve we got because it has an unusual high evolutionary strength at the beginning of the iteration. For this situation, we can get a great parameters combination too.

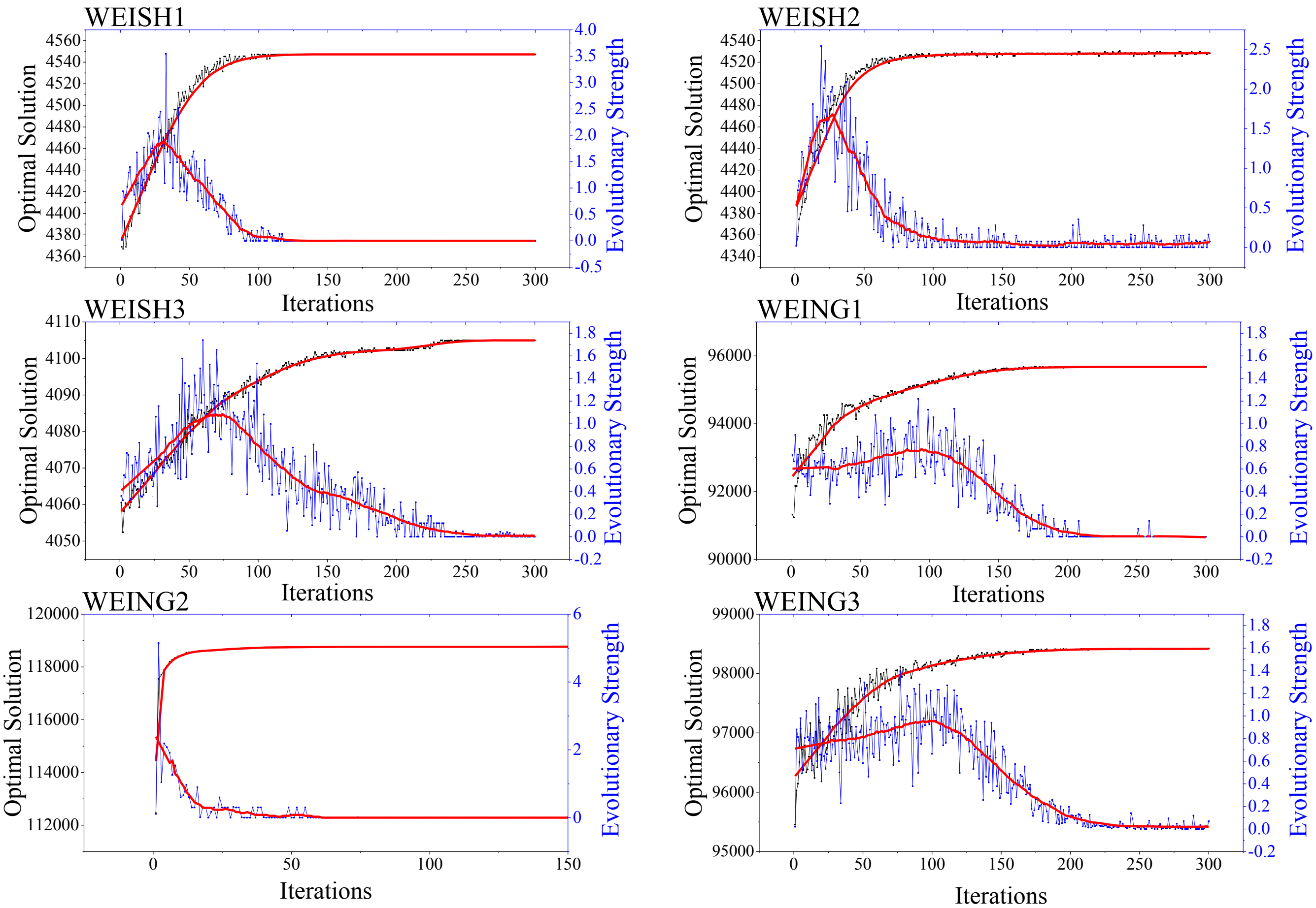

Fig. 5 Evolutionary Strength Curve and Convergence Curve

To demonstrate the superiority of ACOP_ES. The parameters selected by ACOP_ES in Table II are compared with the state-of-art methods. The results for all methods are reported as Relative Percentage Deviation (*RPD*) when solving MKPs. *RPD* is calculated as follows.

$$RPD = \frac{\textit{the best known} - \textit{the result}}{\textit{the best known}} \times 100\% \tag{11}$$

Note that *the best known* means the optimal value of the problem, and *the result* means the result returned by ACO (parameters of ACO are based on state-of-art approaches or ACOP_ES). We test the problem for 50 times independently, and report the mean and standard deviation of *RPD* of 50 times on six datasets as in Fig. 6.

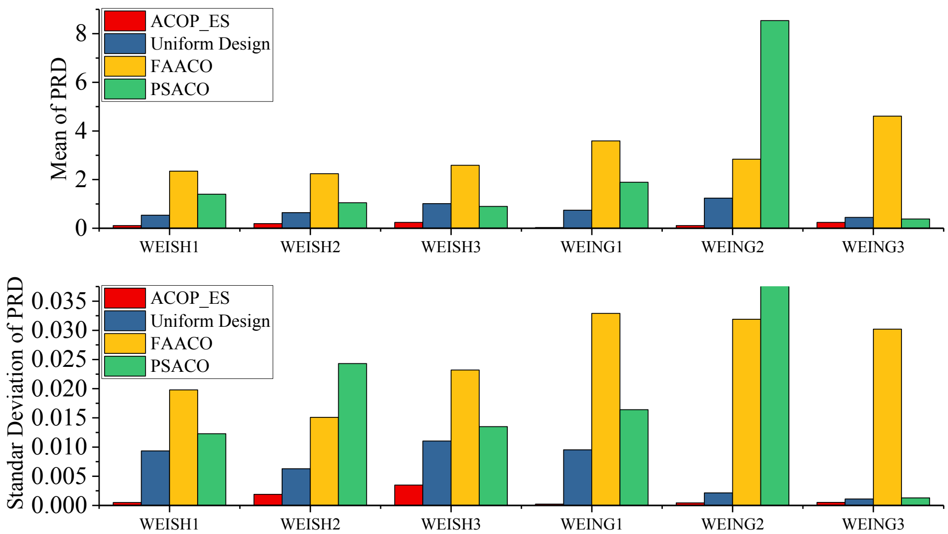

Fig. 6 Comparison results

Obviously, ACOP_ES outperforms all state-of-art approaches on all datasets. Compared with other three typical parameters control approaches, the parameters returned by ACO have better performance in solving MKPs. We analysis the reasons as follows.

Uniform design can get a roughly suitable set of parameters, but the search space of the uniform design is limited, so it is difficult to get the optimal combination of parameters.

For FAACO， it can be seen that the dynamic adjustment control method fails to have a great impact on the experimental results, and even play an opposite role for some datasets. In fact, it is not easy to start from a small value and adjust to a better parameters combination to achieve a good experimental effect during the running process. This point is also given experimental proof in [29].

PSACO appends the direct results of the operation to the various parameters and then selects parameters. This kind of parameters selection, which only consider the results without considering the process, and is difficult to select the real optimal combination of parameters. Differently, ACOP_ES fully consider the performance of the parameters applied to specific problems, evaluate them from the perspective of the process rather than results, and feedback the evaluation of the process to finish the parameters control, so as to get better combination of parameters.

(2) Sensitivity Analysis

In order to show the influence of the model's own parameters setting on the ACO parameters selection results. According to the idea of uniform design, we adopt uniform design table $U_{15}(15^4)$ and selected 15 groups of ``uniform" and ``tidy" possible combinations of parameters to perform experiments on dataset WEISH1. It should be pointed out that based on the idea of uniform design, we believe that the experiment results contain the performance of the worst and best case of ACOP_ES. In fact, users can get better performance with simple debugging like the method of uniform design, which is considered to be the most basic attempt and adjustment of parameters when using optimization algorithms. As in Fig. 7, the parameters selected by ACOP_ES outperform other approaches in most situations. Some parameters are considered to be very inappropriate parameters. In other words, ACOP_ES can have better performance as long as the user has a proper understanding of ACO. Also, we encourage users to make simple attempts on the parameters of the model itself as the parameter setting of our model is not sensitive.

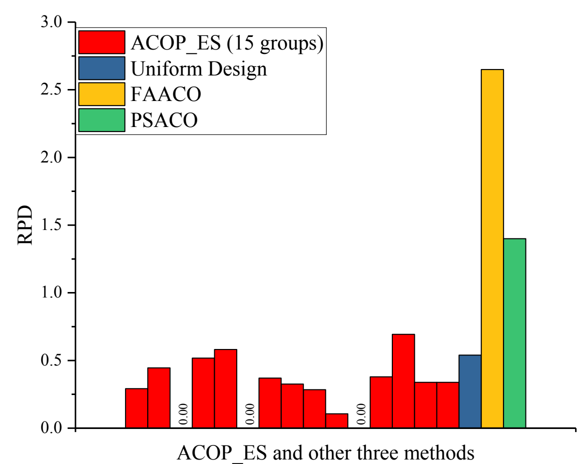

Fig. 7 Sensitivity of ACOP_ES on WEISH1.DAT

At the same time, under different combinations of parameters, the convergence process of the model is as in Fig. 8. Due to space limitation, we only show two examples group1 and group2, others have the same trend that the model achieves convergence within 20 iterations, and different parameters combinations have no significant influence on the convergence speed of the model. On the other hand, the results also show that the algorithm converges fast and have high efficiency.

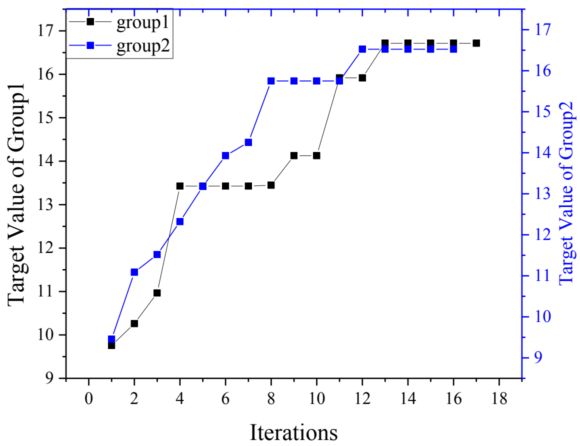

Fig. 8 Convergence Process of ACOP_ES on WEISH1.DAT

VI. Conclusions

In this paper, a parameters control model ACOP_ES for ACO was proposed, and applied it to MKPs. Obviously, a powerful evaluation of algorithm can be used as excellent feedback to perform parameters control. ACOP_ES was evaluated through comparison with offline method uniform design and online methods FAACO and PSACO on WEISH*.DAT and WEING*.DAT. The experiment results show that the parameters returned by ACOP_ES were more excellent than other baseline methods.

For future studies, we intend to find more powerful modeling approaches to describe evolutionary strength curve to find the parameters for ACO with faster convergence.

Ackonwledgement

This work is supported by National Science Foundation of China (Grant No.61371196), the China Postdoctoral Science Foundation (Grant No.2009046425), the China Postdoctoral Science Foundation Funded Project (Grant No.201003797).

References

[1] A. Colorni, "Distributed Optimization by Ant Colonies," European Conference on Artificial Life, The MIT Press, 1992.

[2] M. Dorigo, G Di Caro, New Ideas in Optimization. UK: McGraw-Hill, 1999.

[3] D. Favaretto, E. Moretti, and P. Pellegrini, "On the explorative behavior of MAX–MIN ant system," Spring Berlin Heidelberg, 2009.

[4] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based search for combinatorial optimization: A critical survey," Annals of Operations Research, vol. 31(1–4), pp.373–395, 2004

[5] M. Dorigo, D. Caro G, LM. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5(3), pp.137–72.

[6] M. Kong, P. Tian, and Y. Kao, "A new ant colony optimization algorithm for the multidimensional Knapsack problem," Computers and Operations Research, vol. 35(8), pp. 2672-2683, 2005.

[7] T. Stützle, M. López-Ibáñez, and P. Pellegrini, et al. "Parameter Adaptation in Ant Colony Optimization," Autonomous Search. Springer Berlin Heidelberg, 2011.

[8] H. Duan, G. Ma, S. Liu, "Experimental study of the adjustable parameters in basic ant colony optimization algorithm," IEEE Congress on Evolutionary Computation, pp. 149-156, 2007.

[9] A. C. Zecchin et al., "Parametric study for an ant algorithm applied to water distribution system optimization," IEEE Transactions on Evolutionary Computation, vol. 9(2), pp. 175-191, 2005.

[10] X. Y. Luo, F. Yu, J. Zhang, "Study of parametric relation in ant colony optimization approach to traveling salesman problem," vol. 4115(10), pp. 22-32, 2005.

[11] K. Zhang, H. F. Ling, "Parameter turning of ant colony algorithm based on uniform design and chaos theory." Computer Engineering, vol. 38(14), pp. 141-143, 2014.

[12] Y. Q. Huang, C. Y. Liang, and X. D. Zhang, "Parameter Establishment of an Ant System Based on Uniform Design," Control & Decision, vol. 21(1), pp. 93-96, 2006.

[13] M. Lopez-Ibanez, T. Stützle, "Automatically improving the anytime behaviour of optimization algorithms," Eur. J. Oper. Res, vol. 235(3), pp. 569-582, 2014.

[14] X. Liu, C. Yang and X. Liu, "Optimization of vehicle routing problem based on max-min ant system with parameter adaptation," 7th IEEE International Conference on Computational Intelligence and Security. Hainan, China, pp. 305-307, 2011.

[15] L. Yuan, C. A. Yuan, and D. S. Huang, "FAACOSE: A Fast Adaptive Ant Colony Optimization Algorithm for Detecting SNP Epistasis," Complexity, pp. 1-10, 2017

[16] L. M. Gambardella, M. Dorigo, "Ant-Q: A reinforcement learning approach to the traveling salesman problem,". the Twelfth International Conference on Machine Learning, CA, pp. 252–260, 1995.

[17] S. Colas, N. Monmarche, P Gaucher, and M. Slimane, "Artificial ants for the optimization of virtual keyboard arrangement for disabled people," Artificial Evolution 8th International Conference, Germany, pp. 87–99, 2008.

[18] P. Pellegrini, D. Favaretto, and E. Moretti, "Exploration in stochastic algorithms: An application on MAX–MIN Ant System," Nature Inspired Cooperative Strategies for Optimizatio, Studies in Computational Intelligence, Germany, vol. 236, pp. 1–13, 2009.

[19] Z. Cai, H. Huang, Y. Qin, and X. Ma, "Ant colony optimization based on adaptive volatility rate of pheromone trail," Communications, Network and System Sciences, vol. 2(8), pp. 792–796, 2009.

[20] C. Amir, A. Badr, and I. Farag, "A fuzzy logic controller for ant algorithms," Computing and Information Systems, vol. 11(2), pp. 26–34, 2007.

[21] Z. Hao, H. Huang, Y. Qin, and R. Cai, "An ACO algorithm with adaptive volatility rate of pheromone trail," the 7th International Conference, Proceedings, Part IV, Lecture Notes in Computer Science, Germany, vol. 4490, pp. 1167–1170, 2007.

[22] S. Rafid et al., "Nature-inspired Parameter Controllers for ACO-based Reactive Search," Applied Sciences, Engineering and Technology, vol. 10(1), pp. 09-117, 2015.

[23] H. E. Chu, Y. Zhang et al, "Multiple parameter control for ant colony optimization applied to feature selection problem," Neural Computing & Applications, vol. 26(7), pp. 1693-1708, 2015.

[24] F. Olivas, F. Valdez, O. Castillo, et al., "Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems," Applied Soft Computing, pp. 74-87, 2015.

[25] J. J. Cao, et al., "Process performance assessment for ant colony optimization using evolutionary strength," Pla University of Science & Technology, vol. 14(1), 2013.

[26] M. Dorigo, T. Stützle, "Ant Colony Optimization: Overview and Recent Advances," New York, pp. 227-263, 2010.

[27] S. Wei, "A branch and bound method for the multi constraint zero one knapsack problem," Opl. Res. Soc, pp. 369-378, 1979.

[28] H. M. Weingartner, D. N. Ness, "Methods for the solution of the multi-dimensional 0/1 knapsack problem," Operations Research, vol. 15, pp. 83-103, 1967.

[29] P. Pellegrini, T. Stützle, and M. Birattari, "A critical analysis of parameter adaptation in ant colony optimization," Swarm Intell, vol. 6, pp. 23-48, 2012.