Ke-wei YANG, Li-Ning XING

School of Information System and Management, National University of Defence Technology, Changsha, 410073 P. R. China

The Learnable Ant Colony Optimization to Satellite Ground Station System Scheduling Problems

***Abstract**. The Learnable Ant Colony Optimization (LACO) is proposed to satellite ground station system scheduling problems. The LACO employs an integrated modelling idea which combines the ant colony model with the knowledge model. In order to improve the performance, LACO largely pursues the complementary advantages of ant colony model and knowledge model. Experimental results suggest that LACO is a feasible and effective approach for the satellite ground station system scheduling problem.*

***Streszczenie.** Zaproponowanie wykorzystanie algorytmu LACO (Learnable Ant Colony Optimization) do rozwiązywania problemu planowania działań naziemnej stacji satelitarnej. (**Uczący się algorytm mrówkowy do rozwiązywania problemu planowania działań naziemnej stacji satelitarnej**)*

Keywords: mission planning; ant colony optimization; knowledge; satellite ground station system
Słowa kluczowe: algorytm mrówkowy, satelitarna stacja naziemna

Introduction

The satellite ground station system scheduling problem is a resource optimization problem based on constraints (namely allocating ground stations and operation time for tasks). Its optimization goal can be described as: to complete the most tasks or to maximize the weight summation of tasks in given time (when the weighted tasks are considered). The difficulties of this problem can be summarized as: for a particular activity (task), the resource (ground station) remains available only for one or several time windows, and the planning process includes both the resource dispatching and the allocation of time windows.

The scholars usually adopted the artificial intelligence method to solve the satellite ground station system scheduling problem, such as the greedy algorithm [1], dynamic programming [2], heuristic search method [3], as well as constraint satisfaction method [4], etc. It is still a severe challenge facing us that how to solve the proposed problem more rapidly and effectively. For this purpose, the learnable ant colony optimization is proposed to satellite ground station system scheduling problems. The LCAO integrates Ant Colony Optimization (ACO) [5] with Guided Local Search (GLS) [6-8] to solve this problem.

Problem formulations

With the constraint of time window, this proposed problem takes into consideration both the weight of tasks and aerial conversion time, maximizes the weight summation of completed tasks, and then accomplishes the mission planning of multiple ground station system.

- There are m time windows on the ground stations $W=\{w_1,w_2,...,w_m\}$, the starting time and ending time of time window w_i are S_i and E_i respectively.
- There are n tasks $A=\{a_1,a_2,...,a_n\}$ that need to be completed, the required time and the weight of each task are respectively $D=\{d_1,d_2,...,d_n\}$ and $P=\{p_1,p_2,...,p_n\}$, the starting time and ending time of task i are s_i and e_i respectively.
- As to the decision variable $T=\{t_1,t_2,...,t_n\}$, if task i can be completed, then $t_i=1$; otherwise, $t_i=0$.
- There are totally l aerials on the ground stations, and the aerial conversion time (the adjustment time for aerials before the implementation of tasks) is $R=\{r_1,r_2,...,r_l\}$.
- The starting time of scheduling is T_s , the closing time of scheduling is T_e .

The model of the mission planning of satellite ground station system can be illustrated as follows,

$$F(T)=\max\left(\sum_{1\le j\le n} t_j p_j\right) \tag{1}$$

The constraint conditions include:

$$\sum_{k\in A_i} t_k\left(d_k+r_i\right)\le E_i-S_i,\ \ A_i\subseteq A,\ i\in[1,m] \tag{2}$$

$$s_j\ge S_i,\ \ j\in[1,n],\ i\in[1,m] \tag{3}$$

$$e_j\le E_i,\ \ j\in[1,n],\ i\in[1,m] \tag{4}$$

$$T_s\le s_j\le T_e,\quad T_s\le e_j\le T_e,\quad j\in[1,n] \tag{5}$$

The constraint (2) implies the sum of duration time of task completion and aerial conversion time cannot exceed the length of the time window, in which A_i indicates the set of arranged tasks in the i^{th} time window, r_i means the needed aerial conversion time at the completion of tasks in the i^{th} time window. The constraint (3) signifies that if task a_j is performed in the time window w_i , then the starting time of tasks must be after the starting time of corresponding time window. The constraint (4) means if task a_j is performed in time window w_i , then the ending time of tasks must be before the ending time of corresponding time window. Constraints (3) and (4) ensure that tasks have to be accomplished in the selected time window. The constraint (5) implies the starting and ending time of all tasks should be among the selected time period $[T_s,\ T_e]$.

Learnable ant colony optimization

In the ant colony optimization, the artificial ant gradually constructs a feasible solution based on the state transition rules, and then introducing randomness into optimized results. Consequently, it is difficult to rapidly obtain the global optimal solution unless employing the local search to assist the ACO. Literature [9] integrates ACO with local search, which has been successfully adapted in combinational optimization problems.

A. Basic Framework

The LACO integrates ant colony model with knowledge model: the ant colony model searches the feasible domain according to the neighbour searching strategy, while the knowledge model discovers the knowledge from the previous iterations, and then guide the subsequent iterations of ant colony optimization using the knowledge. The basic framework of LACO is displayed as Fig. 1.

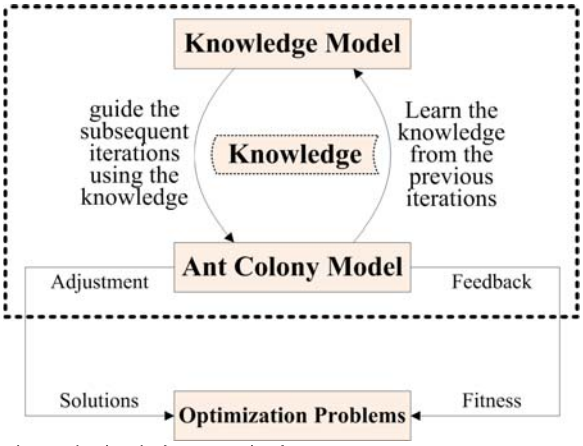

Fig. 1 The basic framework of LACO

B. Optimization Flow

The optimization flow of LACO is displayed as Fig. 2.

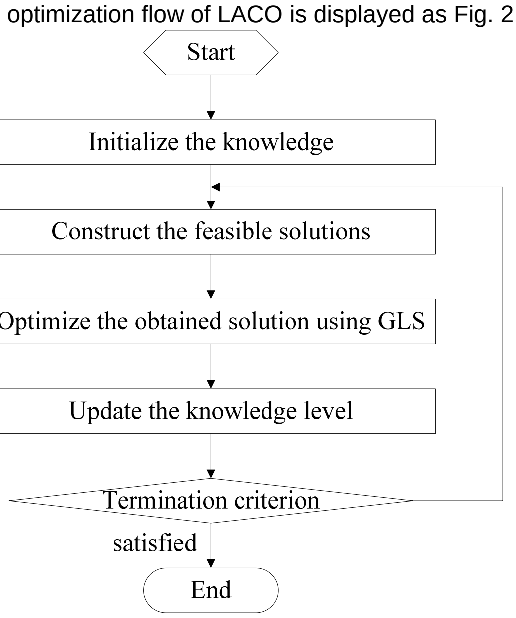

Fig. 2 The optimization flow of LACO

C. Initialization of Knowledge

1) Elite individual knowledge. After each iterative, LACO will select some elite individuals from the current population, and insert them into *the elite individual set. In this paper, all individuals in the elite individual set are called as elite individual knowledge. The elite individual knowledge can be summarized as follows,*

$$\left\langle EL_1, EL_2, \cdots, EL_{num} \right\rangle$$

where $EL_i = \left\{ X_i \middle| f\left(X_i\right) \right\}$ denotes the i^{th} elite individual, $f\left(X_i\right)$ denotes the fitness of individual X_i, num denotes the number of elite individuals. In the elite individual set, all the individuals are sorted with the depressed fitness order. The application mode of elite individual knowledge can be summarized as: keep the elite individuals, and modify the current individual based on elite individuals, that is, take the elite individual as the template, modify the substructure of the current individual, and obtain an improved individual.

2) Component knowledge. Define "the probability of assigning given task in each time window" as the component knowledge on the solution space. The knowledge was recorded using matrix K with dimension $m \times n$, where m denotes the number of tasks, and n denotes the number of time windows. For each element K_{ij}, it denotes the probability of assigning task i in time window j. In this phase, all the elements in pheromone matrix K are initialized as λ_0.

D. Construction of Feasible Solutions

1)State transition rule. For each time window k, we select the next to-be-performed task a_i according to the following probability.

(6) $$\Pr(a_i, k, t) = \begin{cases} \dfrac{F(i,k)}{\sum\limits_{a_j \in allow(k,t)} F(j,k)} & a_i \in allow(k,t) \\ 0 & a_i \notin allow(k,t) \end{cases}$$

(7) $$F(i,k) = [K_{ik}]^a \times [\eta(i)]^b \times [\lambda(i)]^c$$

Here, $\Pr(a_i, k, t)$ means the probability of selecting task a_i in time window k at moment t; K_{ik} denotes the knowledge of allocating task a_i in time window k; $\lambda(i)$ denotes the heuristic value of processing time of task a_i, $\lambda(i) = 1/(d_i + r_i)$; $\eta(i)$ implies the heuristic value of priority of task a_i, $\eta(i) = p_i$; a, b, c denote the weight of different heuristic values respectively. In the construction process of feasible solutions, the traditional random state transition rule is replaced with the pseudo-random proportional rule.

2) The construction mechanism of feasible solutions. Assign the tasks that can be arranged in each time window in sequence (select arranged tasks in current time window from the set of tasks that can be arranged in accordance with state transition rule) until no tasks can be arranged in this time window, and repeat the above process until no tasks can be arranged in all the time windows.

E. Update the Knowledge Level

1) Local updating phase. After each iterative, the ant that obtained the optimal solution at this iterative have the ability to update current knowledge level using local updating rule, which is based on the optimal schedule to accomplish knowledge updating. If the task i is assigned to time window k, then

(8) $$K_{ik} = K_{ik} + Q_L$$

where Q_L denotes the increase level of knowledge in local updating phase.

2) Global updating phase. After each iterative, the ants that obtained the global optimal solution (the best solution from the starting to current iterative) at this iterative have the ability to update current knowledge level using global updating rule, which is based on the optimal schedule to accomplish knowledge updating. If the task i is assigned to time window k, then

(9) $$K_{ik} = K_{ik} + Q_G$$

where Q_G denotes the increase level of knowledge in global updating phase.

3) Knowledge evaporation phase. After each iterative, the current knowledge will be updated with the effect of knowledge evaporation rule. In order to reduce the chance of falling into local optimal, we constrain the knowledge level between $[\tau_{\min}, \tau_{\max}]$. The knowledge evaporation rule is

(10) $$K_{ik} = \min\{\tau_{\max}, \max\{\tau_{\min}, (1-\rho)K_{ik}\}\}$$

where $\rho(0 < \rho < 1)$ denotes the evaporation coefficient of knowledge.

F. Termination Criterion

Termination criterion mainly dominates the optimized process of LACO algorithm. Since the testing instances are generated randomly according to certain rules, and their optimal solutions is unknown. For this reason, we predefine the largest iteration times as the termination criterion.

Experimntal results

40 testing instances were produced to validate this proposed approach. The generation rules of testing instances are summarized as follows:

(1) Task number n values 100, 200, 300, 400 and 500.

(2) Time window number m values 4, 6, 8 and 10.

(3) The scheduling time is 0~1000 second.

(4) Consider two kinds of task weights: the weight of tasks is same, and the weight of tasks is evenly selected from $[1,50]$.

(5) The selection of task processing time. Firstly, calculate average processing time of all tasks $\bar{t}$ (see formula (11)), then select benchmark processing time $\bar{t_i}$ of every task (see Tab. I), and choose the required process time of tasks in each time window among $[0.8\bar{t_i}, 1.2\bar{t_i}]$.

(11) $$\bar{t} = (1000 \times m)/n$$

(6) The selection of available time window for tasks. Firstly, produce one random number RN among $[3,5]$, and select RN ones from all time windows for the given task.

(7) The length of each time window was randomly produced among $[800,1000]$.

(8) Generate testing instances of various scales as per n and m.

The parameter setting of LACO is displayed in Tab. II. This approach was implemented using Matlab, and solve these testing instances with computer equipped with Pentium IV, CPU 2.4 GHz and storage 2G.

Table 1. Probabilities of the benchmark processing time

Probabilities	0.1	0.2	0.4	0.3
$\bar{t_i}$	$[1, 0.5\bar{t}]$	$[0.5\bar{t}, \bar{t}]$	$[\bar{t}, 1.5\bar{t}]$	$[1.5\bar{t}, 2\bar{t}]$

The experimental results of solving 40 testing instances by our approach are as summarized in Tab. III. Each instance was solved 20 times, and the average results were recorded. The time complexity of our approach was analysed in Fig. 3. For the different situations, with the increase in the number of tasks, the computing time is increasing almost in linear. Simultaneously, the calculation time of the most complex instance is no more than 1800 seconds (30 minutes), so the proposed algorithm is acceptable in terms of calculation time consumption.

Table 2. Parameters of the experiment

Symbol	Value	Remark
$AntSize$	10	Number of ants
a	3	The weight of pheromone heuristic value
b	5	The weight of pheromone heuristic value
c	2	The weight of process time heuristic value
τ_0	0.10	Initialized level of pheromone
$\tau_{\max}$	1.0	Upper limit of pheromone level
$\tau_{\min}$	0.01	Lower limit of pheromone level
ρ	0.02	Coefficient of pheromone evaporation
Q_L	0.02	Increment level of pheromone in local updating phase
Q_G	0.10	Increment level of pheromone in global updating phase
Max_Iter	20	The maximum iterations

Table 3. Time complexity of the experiment using LACO

SN	n-m	Result	Time	SN	n-m	Result	Time
1	100-4	2582	14.1	21	100-4	82	16.1
2	200-4	5037	86.4	22	200-4	180	89.1
3	300-4	7315	274.1	23	300-4	255	336.1
4	400-4	10450	599.2	24	400-4	362	674.6
5	500-4	12392	1174.2	25	500-4	488	1357.9
6	100-6	2498	16.5	26	100-6	88	16.5
7	200-6	5279	95	27	200-6	171	107.1
8	300-6	7413	303.2	28	300-6	263	319.6
9	400-6	9771	678.1	29	400-6	350	1012.3
10	500-6	12542	1319.2	30	500-6	431	1603.5
11	100-8	2526	18	31	100-8	80	17
12	200-8	4839	102.9	32	200-8	162	109.4
13	300-8	7297	288.2	33	300-8	248	317.9
14	400-8	10373	653.3	34	400-8	346	825.6
15	500-8	12967	1322.6	35	500-8	441	1806.7
16	100-10	2319	18.3	36	100-10	73	19.3
17	200-10	5157	108.3	37	200-10	162	110.8
18	300-10	7011	305.4	38	300-10	247	328.7
19	400-10	10251	698.6	39	400-10	346	976.1
20	500-10	12479	1301.5	40	500-10	442	1761.3

Note: “SN” represents the number of instances; “n-m” means the scale of instances; “Result” implies the sum of priority values after calculation; “Time” shows calculation time (unit: s). In the first 20 instances, the weight of every task is selected between 1 and 50; in the latter 20 tests, the weight of each task is 1.

More experimental conclusions are summarized as follows. Tasks with the higher priority level have been performed whereas only part of those with lower priority level performed in the circumstances of limited time window resources. Most tasks have been accomplished in optimal (suboptimal) time windows (the least or minor time consumed in task performance). Also, each time window has the high utilization rate and balanced load (total time consumed in task performance). The above conclusions suggest that LACO is a viable and effective approach for this scheduling problem.

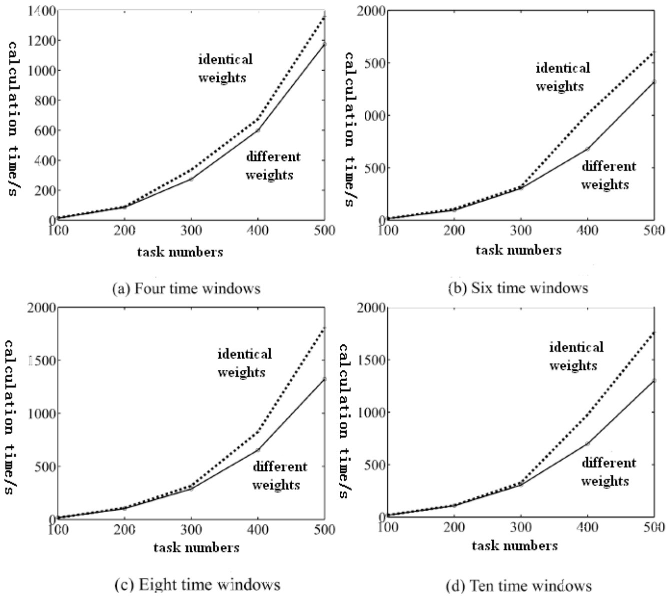

Fig. 3 Analysis of time complexity

Conclusions

A learnable ant colony optimization is designed to the satellite ground station system scheduling problem. It applies an integrated modelling idea which combines ant colony model with knowledge model. Experimental results suggest that LACO is an effective approach for the satellite ground station system scheduling problem.

The future research directions can be summarized as follows. Further consider the constraints of the satellite ground station system scheduling problem, for example, logical constraints among tasks. Solve the satellite ground station system dynamic scheduling problem, such as, randomly add or delete tasks in the process of scheduling.

Acknowledgments: This research was funded by the National Science Foundation of China (Grant No. 70901074, 71001104, 71031007).

REFERENCES

[1] Liu Y, Chen YW, Tan YJ. "Mission planning method of the satellite ground station based on the greedy algorithm," *Systems Engineering and Electronics*, 2003, 25(10): 1239-1241

[2] Liu Y, Chen YW, Tan YJ. "The method of mission planning of the ground station of satellite based on dynamic programming," *Chinese Space Science and Technology*, 2005, 25(1): 47-50.

[3] Liu Y, Chen YW, Tan YJ. "Modeling and solution of a multi-resources dynamic scheduling problem with time windows," *Systems Engineering*, 2004, 22(6): 8-11.

[4] He RJ, Tan YJ. "Apply constraint satisfaction to optimal allocation of satellite ground station resource," *Computer Engineering and Applications*, 2004, 40(18): 229-232

[5] Dorigo M, Stutzle T. "Ant Colony Optimization". Cambridge, USA: MIT Press, 2004

[6] Voudouris C, Tsang E. "Guided local search and its application to the traveling salesman problem," *European Journal of Operational Research*, 1999, 113(2): 469-499

[7] Tsang E, Voudouris C. "Fast local search and guided local search and their application to British Telecom's workforce scheduling problem.," Operations Research Letters, 1997, 20(3): 119-127

[8] Li JF, Tan YJ. "The guided local search method for a class of work-force scheduling problems," *Systems Engineering*, 2004, 22(9): 87-91.

[9] Blum C. "Beam-ACO-hybridizing ant colony optimization with beam search: an application to open shop scheduling," *Computers and Operations Research*, 2005, 32(6):1565-1591

***Authors**: Ke-wei YANG, Li-ning XING are with the school of Information System and Management, National University of Defense Technology, Changsha, P. R. China. E-mail: kayyang27@nudt.edu.cn, xln_2002@126.com*

The correspondence address is: Ke-wei Yang, School of Information System and Management, National University of Defense Technology, Changsha, P. R. China. 410073

e-mail:kayyang27@nudt.edu.cn, xln_2002@126.com