第29卷 第1期
2003年1月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 29, No. 1
Jan., 2003

A Nested Hybrid Ant Colony Algorithm for Hybrid Production Scheduling Problems[1)]

LI Yan-Jun[1] WU Tie-Jun[2]

[1](National Laboratory for Industrial Control Technology, Zhejiang University, Hangzhou 310027)
[2](Institute of Intelligent Systems & Decision Making, Zhejiang University, Hangzhou 310027)
(E-mail: yjlee@iipc.zju.edu.cn; tjwu@iipc.zju.edu.cn)

Abstract The validity of the ant colony algorithm has been demonstrated as a powerful tool to solve the optimization problems. This technique is used to solve difficult combinatorial optimization problems but is seldom used for continuous space search due to its biological background. A nested hybrid ant colony algorithm is proposed in this paper to solve the complicated production scheduling problem with hybrid variable structures, and a novel optimal path pheromone update algorithm is suggested to promote search efficiency. Computer simulation results show that the proposed method is more effective than genetic algorithms as a kind of evolutionary algorithms in solving such kind of difficult problems.

Key words Ant colony algorithm, hybrid production scheduling, pheromone update

1 Introduction

Hybrid production scheduling problems is a kind of problems with high degree of complexity. In production processes some production states change continuously with time and external inputs, while the other states switch interruptedly due to events occurrence (e. g., the completion of prearranged tasks) and the abrupt changes of system inputs. Those two kinds of process dynamics interact each other to realize flexible production of different products in the same production line. In comparison with generic production scheduling problems which have been proved to be combinatorial optimization problems with NP complexity, the hybrid production scheduling problems are more difficult to be solved in view of the fact that there exist different types of coupling variables in a system.

Ant colony algorithms are a kind of population-evolution based optimization approaches developed in recent years, and have been successfully used to solve combinatorial optimization problems with NP complexity[1]. Due to their short development history, their research in China[2~4] are mainly focused on the introduction of the principles of ant colony algorithms, and merely applied to the field of discrete optimization problems (e. g., TSP).

A nested ant colony algorithm is proposed in this paper to solve highly complex hybrid production scheduling problems. This algorithm realizes the optimization in integer-spaces, real-spaces, and the combinatorial optimization in discrete-sequence-spaces through a three-nested-layer architecture.

2 Problem statement

Suppose that there have L customer orders arrival in which the l-th order $O_l = (\Gamma_l, \Theta_l, T_l, \Psi_l)$, where the components of O_l are the specification, the quantity, the due date (or duration), and the punish rate for delivery delay of the ordered product, respectively. Let a production scheme, $P=(P_1, P_2, \cdots, P_m)$, be a list of production tasks as-

1) Supported by the National Hi-tech R & D Plan of P. R. China(9845-005)
Received September 3, 2001; in revised form May 17, 2002
收稿日期 2001-09-03; 收修改稿日期 2002-05-17

signed to the production line, with the subtask $P_i=(\gamma_i,\theta_i,\tau_i)$ consisting of the product specification, the quantity, and the starting time of P_i, respectively. The following constrains should be satisfied for each subtask:

$$\gamma_i \in \Gamma, \bigcup_{\forall i} \gamma_i = \Gamma \tag{1}$$

$$\sum_{P_i \in I_l} \theta_i = \Theta_l \in \Theta \tag{2}$$

where Γ and Θ are the set of all product specifications and the set of processing quantities, respectively; I_l is the set of the subtasks possessing the same specification Γ_l, $\aleph(I_l)=N_l$, i. e. , the order O_l will be processed by N_l subtasks. There are usually multiple system performance goals to be pursued in hybrid production scheduling problems, such as minimizing the total amount of punish charge due to delivery delay, making the total delivery just on due-date as possible, minimizing the total time and cost due to the production switched from one product specification to another, and so on. To the production scheme P, searching for the best subtask partitioning of the order, i. e. , N_l^*, $l=1,2,\cdots,L$, is an optimization subproblem in the integer-space Z_+; for the best allocation of the processing quantity for each subtask, i. e. , θ_i^*, $i=1,2,\cdots,m$, is an optimization subproblem in the real-space R_+; and for the best sequencing of the subtasks $P^*=(P_1^*,P_2^*\cdots,P_m^*)$ is a combinatorial optimization subproblem in the discrete sequence space S_m. These subproblems are, of course, coupled each other and thus cannot be simply solved independently.

3 Nested ant colony algorithm

3.1 Basic ant colony algorithm

To present the algorithm developed in this paper more clearly, the main procedures of problem solving by basic ant colony algorithm[5~11] is briefly described in the following.

According to the characteristics of the problem to be solved, the solution space of the problem is transformed into a searching graph $G=(V,C)$, where $V=\{v_1,v_2,\cdots,v_m\}$ is the set of nodes in the graph G, and $C=\{c_1,c_2,\cdots,c_r\}$ is the set of oriented arcs among those nodes. A path, connecting an initiative node and a terminal node through a series of interim nodes by oriented arcs, is denoted by ω which corresponds to a feasible solution candidate. A colony of n ants are denoted by $A=\{a_1,\cdots,a_n\}$. In each searching period, an ant a_i chooses a path ω_i in the graph G randomly according to a predetermined path selection possibility. A searching period ends up in the algorithm when all the n ants finish their path seeking respectively. The path selection possibility $p_{i,j}(t)$ for a search from v_i to v_j in the searching period t is defined by

$$p_{i,j}(t) = \frac{\varphi_{i,j}^a(t)\cdot d_{i,j}^b}{\sum_{(i,k)\in S, k\notin U} \varphi_{i,k}^a(t)\cdot d_{i,k}^b} \tag{3}$$

where U is the partial route that has been searched by ants in the period t; $\varphi_{i,j}(t)$ and $d_{i,j}$ are the density of pheromone accumulated on the path segment (v_i,v_j) by ants in the period t and the cost of searching for that path segment, respectively; and $a,b>0$ are called the pheromone index and cost index, respectively. The pheromone $\varphi_{i,j}(t)$ will be updated at the end of each searching period in the way of

$$\varphi_{i,j}(t+1) = \lambda\varphi_{i,j}(t) + (1-\lambda)\Delta\varphi_{i,j}(t) \tag{4}$$

where $\lambda\in(0,1)$ is called the evaporation factor; $\Delta\varphi_{i,j}$ is a pheromone increment as a non-increasing function of objective values, e. g. ,

$$\Delta\varphi_{i,j} = \sum_{k=1}^{n}\psi(f_k) \Big/ \sum_{(i,j)\in S}\sum_{k=1}^{n}\psi(f_k) \tag{5}$$

where f_k is the objective value corresponding to the searching path ω_k of the ant a_k; and $\psi(\cdot)$ is a non-increasing function.

The basic ant colony algorithm described above cannot be used directly to solve the problem defined in Section 2 due to the fact that there are three different types of data structure involved in the solution space, and the integrated solution space cannot be mapped to the same searching graph. In addition, those three subproblems logically belong to different levels. In other words, the allocation of the processing quantity for each subtask is optimized only after the best subtask partitioning of the order is obtained; and the sequencing of the subtasks is optimized merely after the best subtask partitioning of the order and the allocation of the processing quantity for each subtask are determined. Therefore, a nested ant colony algorithm is proposed in this paper according to the features of the problem in such a way that different searching graphs are generated according to different types of optimization variables. On the other hand, the variables are optimized nestedly on the different logical levels by searching in different graphs.

3.2 Generation of searching graphs

1) To search for the best partitioning of subtasks in each order, a solution candidate N_l is represented by a binary number with L_1 bits, i.e.,

$$N_l \Leftrightarrow \{b_{L_1} b_{L_1-1} \cdots b_1\} \tag{6}$$

where L_1 satisfies $2^{L_1-1}-1 < \max(N_1, N_2, \cdots, N_L) \leqslant 2^{L_1}-1$. A searching graph $G_1=(V_1, C_1)$ is formed with the set of nodes

$$V_1 = \{v_s, v_{L_1}^0, v_{L_1-1}^0, \cdots, v_j^0, \cdots, v_1^0, v_{L_1}^1, v_{L_1-1}^1, \cdots, v_j^1, \cdots, v_1^1\} \tag{7}$$

where v_s is the initiative node, and v_j^0 and v_j^1 stand for the two binary states of b_j, i.e., 0 or 1, respectively. The set of oriented arcs in G_1 is composed of

$$C_1 = \{(v_{L_1}^0, v_{L_1-1}^0), \cdots, (v_j^0, v_{j-1}^0), (v_j^0, v_{j-1}^1), (v_j^1, v_{j-1}^0), (v_j^1, v_{j-1}^1), \cdots, (v_2^1, v_1^1)\}$$

for $j=2,3,\cdots,L_1$. At the nodes v_j^0 and v_j^1, there exist and only exist the arcs pointing to the nodes v_{j-1}^0 and v_{j-1}^1.

2) To search for the best allocation of the processing quantity for each subtask, solution candidates are encoded as binary numbers with L_2 bits, where b_1 is the lowest bit and b_N is the highest bit in the string, and the value of L_2 is determined according to desired computational precision. The searching graph $G_2=(V_2, C_2)$ is composed of the set of nodes

$$V_2 = \{v_s, v_{L_2}^0, v_{L_2-1}^0, \cdots, v_j^0, \cdots, v_1^0, v_{L_2}^1, v_{L_2-1}^1, \cdots, v_j^1, \cdots, v_1^1\} \tag{8}$$

and the set of oriented arcs

$$C_2 = \{(v_{L_2}^0, v_{L_2-1}^0), \cdots, (v_j^0, v_{j-1}^0), (v_j^0, v_{j-1}^1), (v_j^1, v_{j-1}^0), (v_j^1, v_{j-1}^1), \cdots, (v_2^1, v_1^1)\}$$

for $j=2,3,\cdots,L_2$. The symbols in (8) have the same meaning as those in (7).

3) To search for the best sequencing of the subtasks, a searching graph $G_3=(V_3, C_3)$ is built up as a fully-interconnected oriented graph consisting of a set of m nodes, $V_3=\{v_1, v_2, \cdots, v_m\}$, where $v_i \in V_3$ represent the subtask P_i.

3.3 Nested searching procedures

For the sake of representing the proposed algorithm compactly, the hybrid production scheduling problem described in Section 2 is reformulated as

$$(P): \begin{cases} \min\limits_{(X_1, X_2, X_3) \in \Omega} f(X_1, X_2, X_3) \\ \text{s.t. } g(X_1, X_2, X_3) = 0 \end{cases}$$

where $X_1 \in Z_+$, $X_2 \in R_+$, and $X_3 \in S_m$, representing the number of subtasks in each customer order, the product quantity for each subtask, and the sequencing of the subtasks, respectively; $\Omega \subset Z_+ \times R_+ \times S_m$ is the integrated solution space, and f and g are the objective functions and the constraints, respectively. Then according to the characteristics of the problem (P), it can be resolved nestedly through a series of subproblem (P_i), $i=1, 2, 3$, where

$$(P_i):\begin{cases}\min\limits_{X_i} f(X_i,\overline{X}_j,\overline{X}_k) \\ \text{s.t. } g(X_i,\overline{X}_j,\overline{X}_k)=0\end{cases} \qquad i\neq j\neq k \text{ 且 } i,j,k\in\{1,2,3\}$$

for given $\overline{X}_j$ and $\overline{X}_k$. Correspondingly, the searching procedures for the whole integrated solution space Ω can be decomposed and conducted by three nested ant colony systems $\{A_i,G_i\}$, $i=1,2,3$ to solve the subproblems (P_1), (P_2), and (P_3).

As is shown in Fig. 1, the main searching procedures of the nested ant colony algorithm can be summaried as follows.

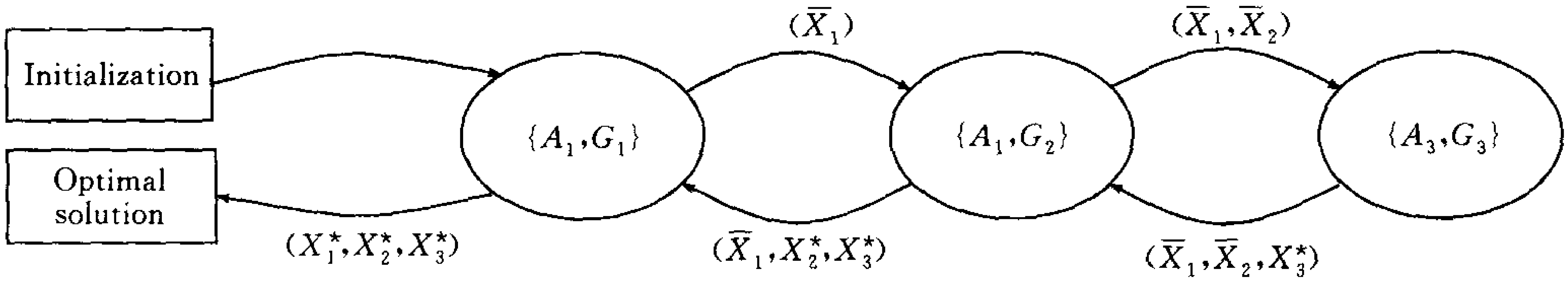

Fig. 1 Nested solving procedures

1) Initialization.

2) For the ant colony system $\{A_i,G_i\}$, $i=1,2,3$, proceed the following operations.

• Generate the searching graph G_i on the basis of $\overline{X}_1,\cdots,\overline{X}_{i-1}$ except for $i=1$. In the case of $i=1$, construct the searching graph G_1 according to (7).

• Dispatch an ant to search for a solution candidate $\overline{X}_i$ of subproblem (P_i) on G_i.

• If $i=3$, compute the objective value of problem (P_3) based on $\overline{X}_3$ directly. Otherwise, initiate the ant colony system $\{A_{i+1},G_{i+1}\}$ to search for the best solution of problem (P_i) and its corresponding objective value with the best solution $(\overline{X}_1,\ X_2^*,\ X_3^*)$ when $i=1$ or $(\overline{X}_1,\overline{X}_2,X_3^*)$ when $i=2$.

• Send next ant to search on G_i if there exist ants not having worked yet. Otherwise, find a best solution candidate of subproblem (P_i) in this searching period and update the pheromone of each trail in G_i.

• Start the next searching period if a termination criterion has not been satisfied. Otherwise return the optimal solution (X_1^*,X_2^*,X_3^*) of problem (P) and the corresponding objective value when $i=1$, or return to the ant colony system $\{A_{i-1},G_{i-1}\}$ with the optimal solution X_i^* and its corresponding objective value of problem (P_i) when $i=2$, 3.

3) The algorithm is terminated.

The nested ant colony searching procedures described above have an important feature, i. e. , when the outer system $\{A_i,G_i\}$ is searching for optimal solution of the subproblem P_i, the inner system $\{A_{i+1},G_{i+1}\}$ can also look for the optimal solution of the subproblem P_{i+1} simultaneously. This manifests the essential parallel characteristics of ant colony algorithms as multi-agent systems[6]. The parallelism of the algorithm is very useful when a large-scale problem is solved using multiple processors.

3.4 Pheromone update strategy

In the basic ant colony algorithm described in Section 3.1, the pheromone is updated according to (4) and (5). The drawback of this update strategy is that, for all the paths the ants have searched, some of them may correspond to worse solutions, in this case, the amount of pheromone increases due to the pheromone increment. However, if the best path corresponding to the optimal solution has not been searched, the amount of pheromone on this path will decrease due to the effect of evaporation. Thus in the next searching period, the nodes in the best path may be searched with a small probability. This may lead to incorrect searching guide and a large amount of inefficient search. According to the analysis above, a new pheromone update method is proposed as follows.

Suppose $\omega^*(t)$ is the best path obtained in the t-th searching period, and the corresponding objective function satisfies

$$f^*(t) < f^*(t-1) \tag{9}$$

where $f^*(t)$ is the best objective value in searching period t according to path $\omega^*(t)$. It for each segment $(v_i, v_j) \in \omega_k^*(t)$, the pheromone increment is updated according to

$$\Delta\varphi_{i,j}(t) = \psi(f^*(t)) \tag{10}$$

otherwise $\Delta\varphi_{i,j}(t)=0$. In this algorithm, only the path segments included in the best path are assigned the pheromone increment at the end of each searching period. In this way, the pheromone can be used to guide the searching of ants correctly in the next searching period, to eliminate the influence of the pheromone update in the worse paths on these path segments. It avoids large amount of invalid searching and promote the searching efficiency greatly.

4 Simulation study

The nested ant colony algorithm proposed in this paper has been implemented to solve several hybrid production scheduling problems[12] in the real world. Suppose there are four customer orders, $O_1=\{$Spec #1, 3000 tons, 36 days, 0.1 yuan$\}$, $O_2=\{$Spec #2, 4500 tons, 20 days, 0.3 yuan$\}$, $O_3=\{$Spec #3, 1000 tons, 3 days, 0.2 yuan$\}$ and for all the products $O_4=\{$Spec #4, 2500 tons, 10 days, 0.25 yuan$\}$. The processing rate is 300 tons/day. The rinse costs for the production switching from product 1 to the other three products is 2.5 tons each time, from the last three products to product 1 is 5 tons each time, and between each pair of products among products 2,3 and 4 is 3.5 tons each time.

Three performance goals are considered as follows:

$$\min_P \sum_{l=1}^{4} \alpha_l * (\Theta_l - \hat{\Theta}_l) \tag{11}$$

$$\min_P \sum_{l=1}^{4} |T_l - \hat{T}_l| \tag{12}$$

$$\min_P \sum_{i,j\in\{1,2,3,4\}} \lambda_{ij} \cdot \eta_{ij} \tag{13}$$

where $\hat{\Theta}_l$ is the quantity of order O_l completed before its due date, α_l is the punish rate for the quantity of order O_l delivered after the due date, $\hat{T}_l$ is the real delivery time of order O_l, and λ_{ij} and η_{ij} are the rinse quantity and the number of production switching from order O_i to order O_j, respectively. Formula (11) is considered as the first level goal, while (12) and (13) are taken into account as the second level goals. In order to evaluate these multiple objectives for solution candidates in the nested ant colony algorithm, an objective ranking method developed by the authors of this paper in the multi-objective genetic algorithm[12] is adopted.

The nested ant colony algorithm (denoted by Alg. 1) and the multi-objective genetic algorithm[12] (denoted by Alg. 2) are used to solve the same problem for comparison. The parameters used in Alg. 1 are set as follows: $n=10, \lambda=0.5, a=0.5, b=0, L_1=3, L_2=8$, $\alpha_1=0.1$, $\alpha_2=0.3$, $\alpha_3=0.2$, $\alpha_4=0.25$, and the number of searching periods=20. Accordingly, in Alg. 2, it is set that the population size=10, the encoding length=8, the crossover probability=0.9, the mutation probability=0.3, the number of evolution periods=20. The termination conditions of the both algorithms are set up as the predetermined number of iterations or evolutionary generations.

In view of the probabilistic characteristics of these methods, 100 rounds of computer simulations are conducted for each algorithm. The statistical results are shown in Table 1 where $f_{i,opt}$, $\bar{f}_i$, and σ_i, for $i=1,2$, are the best objective value, the mean value and the standard deviation of the i-th level goal in the 100 rounds of independent problem solving, respectively. The comparisons of the first level and the second level objective values obtained in the 100 rounds of problem solving by Alg. 1 and Alg. 2 are shown in Fig. 2 and Fig. 3, respectively.

Table 1 Statistic analysis of two methods

	$f_{1,opt}$	$f_{2,opt}$	$\bar{f}_1$	$\bar{f}_2$	σ_1	σ_2
Alg. 1	5038.6	17.82	5462.2	23.0070	175.8199	2.4138
Alg. 2	5181.0	24.49	5946.2	32.1970	372.4956	3.4625

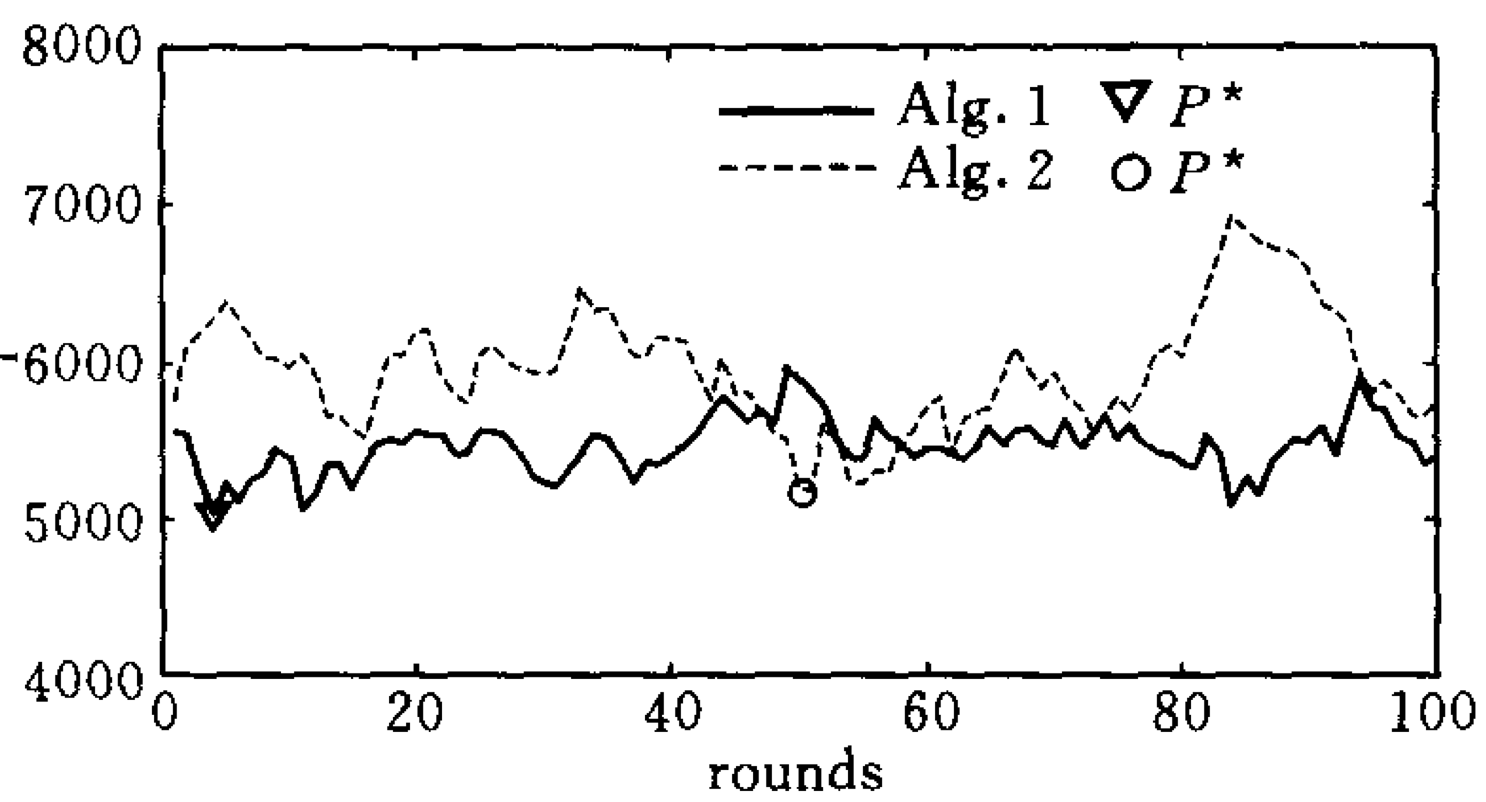

Fig. 2 Comparison of the first level objective values

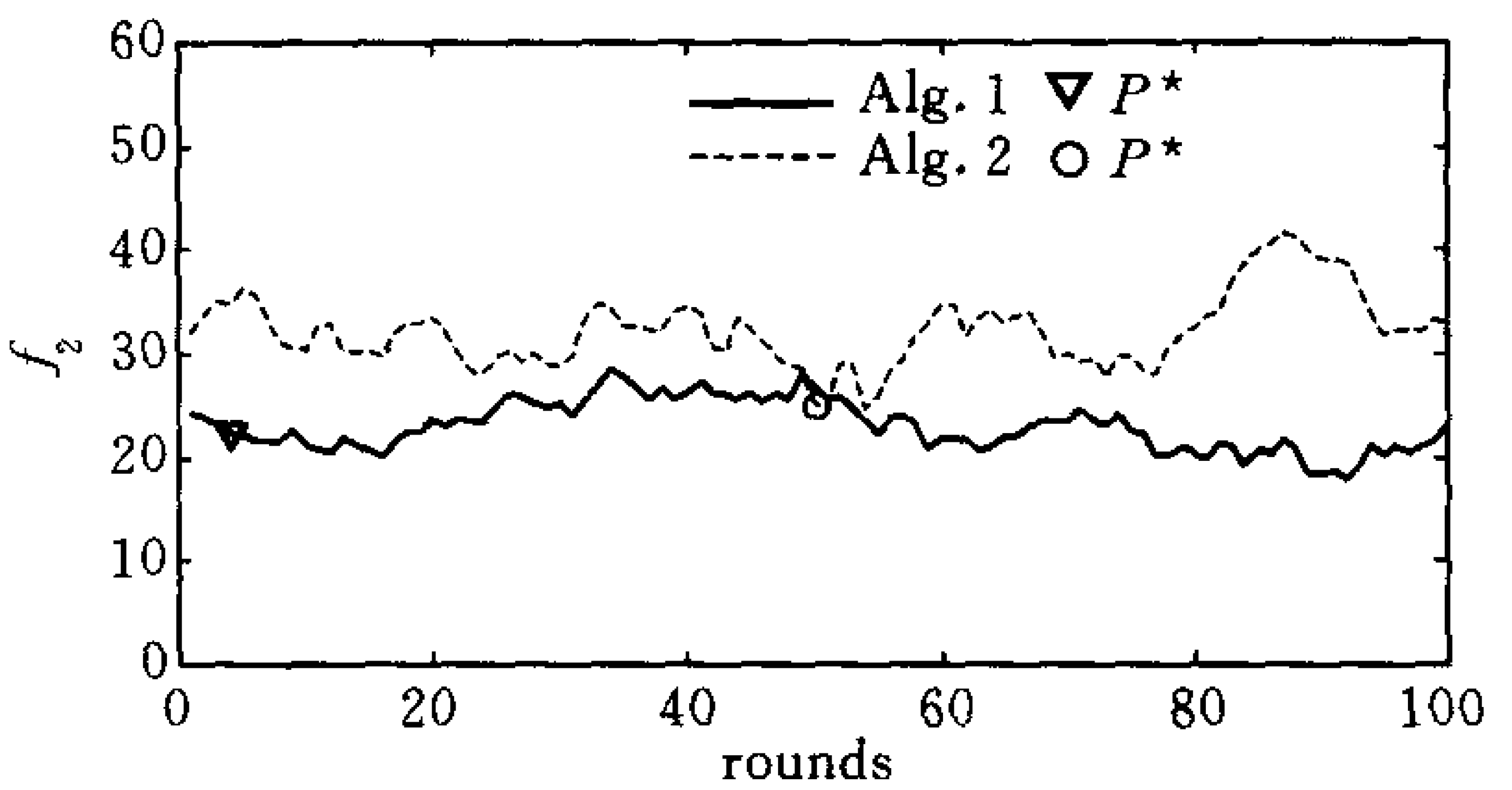

Fig. 3 Comparison of the second level objective values

Simulation results indicate that all the statistical indexes of Alg. 1 are better than those of Alg. 2. For example, in comparison with Alg. 2, the best objective values of the two levels of goals in Alg. 1 are decreased by 2.75% and 27.24%, respectively; the mean values of the 100 objectives are decreased by 8.14% and 28.54%, respectively; and the standard deviations of the objective values are decreased by 52.80% and 30.29%, respectively.

5 Conclusions

A nested ant colony algorithm is proposed in this paper to tackle hybrid production scheduling problems. Computer simulation results manifest its predominant performance in such kind of front-end problems in the fields of production scheduling and demonstrate its energetic vitality in solving similar complex optimization problems with hybrid data structures. Besides, to apply this approach in the field of production scheduling there still exist many aspects to be further investigated in the future, e. g., how to accelerate its computation speed, which is the common problem faced by biological-heuristic-based algorithms, such as genetic algorithms, immune algorithms, and ant colony algorithms. Because many complex optimization problems with hybrid data structures, such as departure and landing scheduling of airplanes, intelligent transportation scheduling, communication network routing, task assignment of multi-robots, and graph generation and partitioning, continuously emerge in recent developments of science and technologies, the research of hybrid ant colony algorithm has prosperous application perspectives.

References

1 Stützle T, Hoos H H. Max-min ant system. *Future Generation Computer Systems*, 2000, **16**(8): 889~914

2 Zhang Ji-Hui, Gao Qi-Sheng, Xu Xin-He. A self-adaptive ant colony algorithm. *Control Theory and Applications*, 2000, **17**(1): 1~3(in Chinese)

3 Zhang Ji-Hui, Xu Xin-He. A new evolutionary algorithm—Ant colony algorithm. *Systems Engineering-Theory & Practice*, 1999, **3**(3): 84~87(in Chinese)

4 Wu Qing-Hong, Zhang Ji-Hui, Xu Xin-He. An ant colony algorithm with mutation features. *Journal of Computer Research & Development*, 1999, **36**(10): 1240~1245(in Chinese)

5 Gutjahr W J. A graph-based ant system and its convergence. *Future Generation Computer Systems*, 2000, **16**(8): 873~888

6 Dorigo M, Bonabeau E, Theraulaz G. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 2000, **16**(8): 851~871

7 Hertz A, Kobler D. A framework for the description of evolutionary algorithms. *European Journal of Operational*

Research, 2000, **126**(1): 1～12

8 Bilchev G A, Parmee I C. The ant colony metaphor for searching continuous design spaces. *Lecture Notes in Computer Science*, 1995, **993**: 25～39

9 Song Y H, Chou C S, Stonham T J. Combined heat and power economic dispatch by improved ant colony search algorithm. *Electric Power Systems Research*, 1999, **52** (2): 115～121

10 Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics*, 1996, **26**(1): 28～41

11 Preux Ph, Talbi E-G. Towards hybrid evolutionary algorithms. *International Transactions in Operational Research*, 1999, **6**(6): 557～570

12 LI Yan-Jun, WU Tie-Jun. A novel parallel multi-objective genetic algorithm and its application in process scheduling. In: Proceedings of the 3th World Congress on Intelligent Control and Automation, Hefei: Press of University of Science and Technology of China, 2000. 525～528(in Chinese)

LI Yan-Jun Received her Ph. D. degree in Control Science and Engineering from Zhejiang University in 2001, and is now a post-doctoral fellow at Zhejiang University. Her research interests include computation intelligence, optimization and control of hybrid systems, and neural networks. She is a member of the IEEE.

WU Tie-Jun Professor and head of Institute of Intelligent Systems & Decision Making, Zhejiang University. His research interests include intelligent control of large-scale systems, nonlinear control, control and decision of hybrid dynamic systems and its applications in complex systems. He is a member of the IEEE.

求解混杂生产调度问题的嵌套混合蚁群算法

李艳君[1]　　吴铁军[2]

[1](浙江大学工业控制技术国家重点实验室　杭州　310027)

[2](浙江大学智能系统与决策研究所　杭州　310027)

(E-mail: yjlee@iipc.zju.edu.cn; tjwu@iipc.zju.edu.cn)

摘　要　蚁群算法作为解决优化问题的有力工具,它的有效性已经得到了证明.由于其生物学背景,基本蚁群算法被设计来求解复杂的排序类型组合优化问题,在连续空间优化问题的求解方面研究很少.本文提出一种嵌套混合蚁群算法,用于解决具有混杂变量类型的复杂生产调度问题,在一种新的最佳路径信息素更新算法的基础上,提高了搜索效率.计算机仿真结果表明,本文提出的方法在求解此类问题上性能优于另一种基于进化计算的有效方法——遗传算法.

关键词　蚁群算法,混杂生产调度,信息素更新

中图分类号　TP202