A hybrid ant colony optimisation algorithm for job shop problems and its convergence analysis

Yang Cao*

College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

and

Department of Digital Factory, Shenyang Institute of Automation, CAS, Shenyang, 110016, China

and

University of Chinese Academy of Sciences, Beijing, 100049, China

and

Key Laboratory of Networked Control, CAS, Shenyang, 110016, China

and

Faculty of Information and Control Engineering, Shenyang Jianzhu University, Shenyang, 110168, China

Email: caoyang@sia.cn

*Corresponding author

Haibo Shi

Department of Digital Factory, Shenyang Institute of Automation, CAS, Shenyang, 110016, China

Email: hbshi@sia.cn

Abstract: This paper presents a hybrid ant colony optimisation (HACO) algorithm for solving job shop problems. The criterion considered is the maximum completion time, the so-called makespan. The HACO algorithm improves the performance of intelligence optimisation algorithm, which adopts ant colony optimisation (ACO) algorithm to search in the global solution space, and tabu search (TS) algorithm is utilised as the local algorithm in each generation. The global asymptotic convergence of the hybrid algorithm is proved by Markov chain theory in the paper. By testing 13 hard benchmarks instance, the results demonstrate that the HACO algorithm is effective.

Keywords: ant colony optimisation algorithm; tabu search algorithm; convergence analysis; job shop problems; JSP.


Biographical notes: Yang Cao received his Bachelor degree from Shenyang Ligong University in 2001 and Master degree from Liaoning University in 2005, China. He is currently a PhD student in Northeastern University, China. His main research interest covers production planning and scheduling.

Haibo Shi received his PhD in Engineering from Harbin Institute of Technology, China in 2003. Currently, he is a Research Fellow and Director of doctor student of Shenyang Institute of Automation. His main research interests include production and operation management, modelling and simulation technology of manufacturing process, MES technology, digital equipment and intelligent system technology.
1 Introduction

The job shop problems (JSP) are the simplification model of actual problems, which are not only one of the most notorious and intractable NP complete problems, but also one of the most important scheduling problems that arise in situations where a set of activities that follow irregular flow patterns have to be performed by a set of scarce resources. It has important applications among others in manufacturing systems, information service facilities and assembly lines in use nowadays (Tasgetiren et al., 2006; Omkumar and Shahabudeen, 2009).

The ant colony optimisation (ACO) algorithm is a new competitive algorithm for JSP (Alberto et al., 1993), which has many advantages such as intelligence search, global optimisation, high suitability and the capability of easy to combine with a variety of heuristic algorithm and so on. However, this algorithm is slow and easy to fall into the local optimal solution. The tabu search (TS) algorithm is one of the better algorithms for solving JSP so far, which has a relative fast computed speed (Li et al., 2012). And TS algorithm can provide a better guidance than others in the whole search neighbourhood according to the result of study.

Based on the convergence theory of ACO algorithm and TS algorithm analysed by Markov chain theory, a hybrid algorithm of ant colony optimisation (HACO) and TS with global convergence is proposed in this paper. According to certain rules, the HACO algorithm is divided into two layers of search group. The first search group realises individuals of the group can be spread as far as possible to the locations in the solution space by ant pheromone concentration (namely explore). Then some ant individuals are selected to parallel TS local search, which ensure that the best possible solution of the individuals can be found in local space (namely search). So, HACO algorithm not only has search capability in a wide range, but also has local optimisation ability.

Under the guidance of the convergence theory, the optimums of the typical benchmarks problems were found in a relative short period of time by instance simulation, the optimisation effect was better than ACO algorithm and genetic algorithm (GA), and the convergence and efficiency of HACO algorithm were verified doubly both in experiment and theory.

2 Description of JSP

Variable description: $h, k = 1, 2, \ldots, m$ standing for the machine type; $i, j = 1, 2, \ldots, n$ is the type of the workpiece to be processed; $c_{jk}$ is the completion time when workpiece $j$ worked on machine $k$; where $M$ is a sufficiently positive number;

$$a_{hk} = \begin{cases} 1, & \text{If workpiece } i \text{ is processed on machine } h \text{ before machine } k; \\ 0, & \text{else;} \end{cases}$$

$$x_{jk} = \begin{cases} 1, & \text{If workpiece } i \text{ is processed before workpiece } j \text{ on machine } k; \\ 0, & \text{else;} \end{cases}$$

The JSP whose goal is to minimise the maximum of the flow time can be expressed as follow (Jarboui et al., 2009):

$$\min \max_{1 \leq i \leq m} c_{i} \quad \left(1\right)$$

s.t.

$$c_{ik} - t_{ik} + M \left(1 - a_{ik}\right) \geq c_{jh};$$

$$i = 1, 2, \ldots, n; h, k = 1, 2, \ldots, m \quad \text{(2)}$$

$$c_{jk} - c_{ik} + M \left(1 - x_{jk}\right) \geq t_{ik};$$

$$i, j = 1, 2, \ldots, n; k = 1, 2, \ldots, m \quad \text{(3)}$$

$$c_{ik} \geq 0;$$

$$i = 1, 2, \ldots, n; k = 1, 2, \ldots, m \quad \text{(4)}$$

$$x_{jk} = 0 \text{ or } 1;$$

$$i, j = 1, 2, \ldots, n; k = 1, 2, \ldots, m \quad \text{(5)}$$

Constraint(2) shows the working consequence of operations of each word pieces, which is determined by the technics process constraint conditions; and constraint (3) states the working consequence of machines on which various word pieces process.

The feasible solution of the above model composed of formulas is called scheduling. Moreover, the JSP is also subject to the following constraints. Each machine can process only one job at a time and each job can be processed by only one machine at a time. The sequence of operations of every job should be pre-defined. Machines are not subject to breakdowns and pre-emption is not allowed. Thus, the processing of an operation cannot be interrupted once started.

3 HACO algorithm for JSP

ACO algorithm is a new metaheuristic methodology, which is based on the indirect communication and cooperation of the phenomenon among ants, and has a theoretical foundation in probability theory. In each iteration, according to the state transition rules, the ant moves in the structure of
the solution, and structures the imperfect solutions (Christian and Michael, 2004; Ji et al., 2011). ACO algorithm procedure is as follows:

Step 1 Set a set of operations \( Q(I) = \{o_j | i = 1, \ldots, n; j = 1, \ldots, m\} \), and set a set of the first operations of jobs \( S(1) \).

Step 2 Set \( t = 1 \).

Step 3 Set \( o^* \) is an operation based on \( c(o^*) = \min\{c(o_j) | o_j \in S(t)\} \), \( m^* \) is the machine that performs the operation \( o^* \), and making sure \( o_{m^*} \) is an operation of set \( \{o_{m^*} | S(t); r(o_{m^*}) < c(o^*)\} \).

Step 4 Each ant decides the next operation \( Q(t + 1) = Q(t) \setminus \{o_{m^*}\} \) based on formula (6). Through removing the operation \( o_{m^*} \) and appending the next operation of job \( i \), we get a new set \( S(t + 1) \).

Step 5 Update the pheromone-based formula (8).

Step 6 If \( Q(t + 1) \) is no empty, then let \( t = t + 1 \), and go to Step 3. Otherwise ends.

We now detail the proposed ACO algorithm for solving the JSP to minimise makespan. We explain the solution representation, population initialisation, population update, local search procedure and neighbourhood structure.

3.1 Solution representation and population initialisation

One of the key issues when designing ACO algorithm lies in the solution representation where individuals bear the necessary information related to the problem domain at hand. This representation has been widely used in the literature for a variety of permutation JSP (Yong et al., 2013; Xu et al., 2013). Therefore, we also employ it in this study.

To guarantee an initial population with a certain level of quality and diversity, a common trend is to construct a few good individuals by effective heuristics and to produce smaller. So the optimal solution did not improve after cycling times, \( \rho \) value is reduced, but not less than the minimum.

The operations \( o_i \) and \( o_j \) are relevant operations processing on the same machine. According to the JSP that \( n \) jobs process in \( m \) machines, the operations of \( n \) jobs constitute the relevant operation set on machine \( k \). The \( n \times n \) matrix indicates the relevant operation set on machine \( k \). The \( n \times n \) matrix indicates the relevant operation set on machine \( k \). The relevant operation set on machine \( k \) indicates pheromone value (Merkle and Middendorf, 2000). As shown in formula (7).

\[
\mathbf{3} = \begin{bmatrix}
0 & \tau_{12} & \tau_{13} & \tau_{1n} \\
\tau_{21} & 0 & \tau_{23} & \tau_{2n} \\
\tau_{31} & \tau_{32} & 0 & \tau_{3n} \\
\tau_{n1} & \tau_{n2} & \tau_{n3} & 0 \\
\end{bmatrix}
\] (7)

where \( \tau_{ij} \) is the pheromone value that \( o_i \) is processed before \( o_j \), \( \tau_j \) is the pheromone value that \( o_j \) is processed before \( o_i \).

After the process, every ant completes a tour. The global pheromone trail updating process is only to update the pheromone of the best total path. Meanwhile, the concentration of pheromone trail should be restricted, making the limits to overcome the disadvantages of prematurity and the tendency of falling into local optimisation. The updating rule of \( \tau_{ij} \) could be calculated as following:

\[
\tau_{ij} = \tau_{ij} + \rho \times (\delta(o_i, o_j) - \tau_{ij})
\] (8)

\[
\delta(o_i, o_j) = \begin{cases} 
\tau_{max} & \text{If } O_i \text{ is the front process to } O_j \text{ in } S_{bh} \\
\tau_{min} & \text{If } O_j \text{ is the front process to } O_i \text{ in } S_{bh}
\end{cases}
\]

After updating the pheromone, we check the value of pheromone whether it is beyond the mark or not. If it is more than the maximum, the value of pheromone will be equal to the maximum. Otherwise, if it is less than the minimum, the value of pheromone will be equal to the minimum. As the concentration of pheromone increased, the volatile is faster. The volatile coefficient of pheromone \( \rho \) must meet demand via self-adaptive behaviour. Reducing \( \rho \) value, the paths were chosen by the possibility of getting smaller. So the optimal solution did not improve after cycling times, \( \rho \) value is reduced, but not less than the minimum.

\[
\rho_{(i)} = \begin{cases} 
\varepsilon \times \rho_{(i-1)} & \varepsilon \times \rho_{(i-1)} > \rho_{min} \\
\rho_{min} & \text{otherwise}
\end{cases}
\] (9)

where \( \rho_{min} \) is the minimum of \( \rho \) value, \( \varepsilon \) is self-adaptive coefficient.

After several generations, the pheromone is close to \( \tau_{max} \) or \( \tau_{min} \). The solutions of ACO algorithm may be premature convergence, and the algorithm traps in stagnation behaviour. So the convergence factor \( cf \) is calculated after each iteration. If \( cf > 0.99 \), the pheromone of algorithm is soft reset to the initial value. \( cf \) could be calculated as follows:

\[
\text{cf} = 2 \left( 1 - \frac{\sum_{\tau_{ij} \leq \tau} \max \{\tau_{max} - \tau_{ij}, \tau_{ij} - \tau_{min}\}}{\tau \left( \tau_{max} - \tau_{min} \right)} - 0.5 \right) 
\] (10)
3.3 The new neighbourhood structure

The feasible solution of JSP is usually denoted by the Gantt figure. The $6 \times 6$ problem solution Gantt figure is illustrated in Figure 1. In the figure, x-axis denotes the process time, y-axis denotes the machines, and every rectangular block marked $((i,j))$ denotes the operation $j$ of task $i$.

The critical path is that the longest path without time intervals among operations in an available schedule (Pan et al., 2010). A solution always has many critical paths. There are two critical paths in Figure 1, (4,1) (3,2) (5,1) (5,2) (4,2) (3,3) (3,4) (3,5) (6,4) (5,5) (5,6) and (4,1) (3,2) (5,1) (5,2) (4,2) (4,3) (5,3) (3,4) (3,5) (6,4) (5,5) (5,6).

Furthermore, the exchangeable neighbours in the critical path are considered as a set for neighbour selection. The local search algorithm TS is based on a certain neighbourhood structure in the paper. That is, a move is defined by the interchange of two successive tasks, where one of tasks is the first or last task in a block that belongs to a critical path. In the first block only the last two tasks, and symmetrically in the last block of the critical path only the first two tasks are swapped. A block is a set of consecutive operations in a critical path in one machine. For example, operation (3,5) (6,4) (5,5) form the block respectively, and we get neighbour (6,4) (3,5) (5,5) or (3,5) (5,5) (6,4) after exchange.

In critical path, the operation $o$ is called critical operation as soon as the operation satisfies the formula (11).

$$S_i(0) = E_i(M_p(0)) = E_i(J_p(0))$$

where $J_p(0)$ is the previous operation of operation 0 in the same job, $M_p(0)$ is the previous operation of operation 0 processed in the same machine, $S_i(0)$ and $E_i(0)$ are the start time and the end time of the operation 0 respectively.

Operation (3,4) is a critical operation by analysis of Figure 1, because $S_t(3,4) = \text{MAX}(E_t(5,3), E_t(3,3))$. We can deduce that a schedule is obtained before the critical operation (3,4) cannot reduce the makespan.

3.4 Local search

In order to make better search results and higher efficiency, the local search algorithm is TS algorithm. So far, TS algorithm running speed is faster, and it may provide a better induct within the whole searching field compared with other algorithms. To improve the efficiency of the local searching, we modify the TS algorithm (Song et al., 2008). A kind of new neighbour is designed based on neighbour exchanging of critical operation, TS algorithm reduces invalid neighbour exchange, and improves the quality of searching solutions. If the set described above is null, it indicates that the modified cost of this solution is too large (or is already optimal), then stop the current search with TS.

The method of TS algorithm choosing neighbours based on critical operation is as follows:

Step 1 According to Figure 1, find the last critical operation $o$.

Step 2 Identify the critical operation $o$. If the operation $o$ is not empty, then go to Step 3. Otherwise go to Step 4.

Step 3 The exchangeable neighbours between the operation $o$ and the last operation in the critical path are viewed as a set for neighbour selection.

Step 4 The exchangeable neighbours in the critical path are considered as a set for neighbour selection.

Step 5 If the set for neighbour selection is null, or TS algorithm search process runs for certain times, then ends.

---

**Figure 1** $6 \times 6$ problem solution Gantt figure
3.5 HACO algorithm design

Recently, the theorem of no free lunch (NFL) is proposed for evaluating optimisation algorithms by Wolpert and Macready of Stanford University. It is shown that there is not a single solution that adapts to all problems effectively. The hybrid algorithm is an effective method, which enlarges the application domain and improves their performance (Song and Shi, 2012; Zhao et al., 2013). Each ant randomly generates solution in ACO algorithm, not necessarily optimal solution or suboptimal solution. Its chief consequence is that the pheromone is updated in random, and the pheromone of the best paths may be not updated. The ant optimum will be over a long period of time, and the local optimal solution will be gained. Because TS algorithm has ability to jump out of the neighbourhood, the ant will not get into local optimal solution.

Considering about the different search mechanism between ACO algorithm and TS algorithm, HACO algorithm is proposed in the paper. According to the regularity of objective function, the optimisation must be in conditional probability satisfy the formula (12) as for each integer \( n \in T \) and any \( s_0, s_1, s_2, \ldots, s_{n+1} \in S \).

\[
P(X_{n+1} = s_{n+1} \mid X_0 = s_0, X_1 = s_1, \ldots, X_n = s_n) = P(X_{n+1} = s_{n+1} \mid X_n = s_n) \tag{12}
\]

Definition 2 (Transition probability matrix): The conditional probability \( p_{ij} = P(X_{n+1} = j \mid X_n = i) \) is called transition probability of Markov chain \( \{X_n, n \in T\} \), where \( i, j \in S \) is. The matrix \( \{\mathbf{P}_{ij}; i, j = 1, \ldots, k\} \) is called \( k \times k \) transition probability matrix.

Since the generation of a solution has business only with its previous solution, but not with the prevenient solution in the hybrid algorithm mixed by ACO algorithm and TS algorithm as well. From the definition of Markov chain, we can know that the corresponding solution sequence of the HACO algorithm is a Markov chain.

Definition 3: Given a \( n \times n \)-moment probability matrix \( P \), its ergodic coefficient is defined as follows:

\[
\tau(P) = \frac{1}{2} \max_{i,j} \sum_{k=1}^{k} | p_{ik} - p_{jk} | \tag{13}
\]

Lemma 1: If \( P, Q \) are probability matrices, then

\[
\tau(PQ) \leq \tau(P) \tau(Q) \leq \tau(P). \tag{14}
\]

Lemma 2: Non-stationary Markov chain is weakly ergodic if and only if there is a strictly increasing sequence of positive integers \( \{k_i\}, i = 0, 1, 2, \ldots \) satisfying

\[
\sum_{j=0}^{\infty} \left[ 1 - \tau(P(k_i, k_j)) \right] = \infty \tag{15}
\]

Lemma 3: Markov chain is strongly ergodic if it is weakly ergodic and there is row vector that makes \( \pi(m) = \pi(m)P(m) \), \( \|\pi(m)\| = 1 \) and \( \sum_{m=0}^{\infty} \| \pi(m) - \pi(m+1) \| < \infty \) for any integer \( m \). Moreover, if \( e^* = \lim_{n \to \infty} \pi(n) \), then

\[
\lim_{n \to \infty} \| g(m, n) - e^* \| = 0 \text{ for any integer } m.
\]
In TS algorithm, the one-step transition probability from population \( L \) to population \( K \) is \( a_{iLK} (n) \). Let \( a_0 = \min_{L \in \Lambda} \min_{K \in S_n (L)} g_{LK} \), when \( K \in S_n (L) \), then \( a_{iLK} (n) \geq \omega c_{iLK} = \omega_1 \); When \( K \notin S_n (L) \), then \( a_{iLK} (n) \geq \mu \). Let \( \omega = \min (a_1, a_2, \ldots) \), then \( a_{iLK} (n) \geq \omega \). If \( z \) individuals are chosen to be operated by TS algorithm, then \((a_{iLK} (n)) = \omega^z \), where \( z \) is a positive number.

**Theorem 1:** The solution sequence of HACO algorithm is strongly ergodic if feasible solution \( X_n \) satisfies \( \lim_{n \to \infty} P_k (X_n) = \infty \), where \( D = \max_{L \in \Lambda} \max_{K \in S_n (L)} | f(K) - f(L) | \).

That is to say that the HACO algorithm will converge to the set of global optimal value if the condition above is hold.

**Proof:**

1. **Weakly ergodic property:**

   Let \( \omega = \min_{L \in \Lambda} \min_{K \in S_n (L)} g_{LK} \), then \( \forall L \in \Lambda, K \in N (L) \),
   \[
   a_{iLK} (X_n) \geq \omega \exp \left( -D / X_n \right), \quad h = 0, 1, \ldots \quad (16)
   \]

   Therefore, there must be an integer \( k_0 < \infty \) satisfying
   \[
   a_{iLK} (T_{k_0}) \geq \omega \exp \left( -D / X_n \right), \quad h = (k_0 - 1) r \quad (17)
   \]
   for all \( L \in (\Lambda - \Lambda_{n}) \).

   Any node in \( \Lambda \) will be transferred to the centre \( I_c \) after \( r \) steps at most, then for any \( L \in \Lambda \), when \( h \geq k_0 r > (k_0 - 1) r \),
   \[
   a_{iLK} (T_{k_0}, T_{k_0}) \geq \sum_{i=1}^{b-1} \left[ \omega \exp \left( -D / X_i \right) \right] \geq \omega^r \exp \left( -rD / X_{k_0 - 1} \right) \quad (18)
   \]

   According to Lemma 1,
   \[
   \tau \left( P \left( X_{k_0 - 1}, X_{k_0} \right) \right) \leq \tau \left( A \left( X_{k_0 - 1}, X_{k_0} \right) \right) \leq \tau (A) \quad (19)
   \]
   for all \( k \geq k_0 \).

   Because the node will certainly reach the centre \( I_c \) of the graph by conversion \( A \left( X_{k_0 - 1}, X_{k_0} \right) \), therefore
   \[
   \min_{i,j} \sum_{k=1}^{n} \min \left( a_{i,k}, a_{j,k} \right) \geq \min_{i,j} \left( a_{i,k}, a_{j,k} \right) \quad (20)
   \]
   Since one-step transition probability is bigger than the product of \( r \)-step transition probability, then
   \[
   \min_{i,j} \left( a_{i,k}, a_{j,k} \right) \geq \omega^r \exp \left( -rD / T_{k_0 - 1} \right) \quad (21)
   \]
   for all \( i, j \in L \).

   Then the following formula can be obtained
   \[
   1 - \min_{i,j} \left( a_{i,k}, a_{j,k} \right) \leq 1 - \omega^r \exp \left( -rD / X_{k_0 - 1} \right) \quad (22)
   \]

   In summary, when \( k \geq k_0 \),
   \[
   \tau \left( P \left( X_{k_0 - 1}, X_{k_0} \right) \right) \leq 1 - \omega^r \exp \left( -rD / X_{k_0 - 1} \right) \quad (23)
   \]

   Then
   \[
   \sum_{k=k_0}^{\infty} \left( \tau \left( P \left( X_{k_0 - 1}, X_{k_0} \right) \right) \right) \geq \omega^r \sum_{k=k_0}^{\infty} \exp \left( -rD / X_{k_0 - 1} \right) = \infty \quad (24)
   \]

   Let \( k_i = (i-1)r \), then
   \[
   \sum_{i=0}^{\infty} \left( 1 - \tau \left( P \left( k_i, k_{i+1} \right) \right) \right) \geq \infty \quad (25)
   \]

   Therefore, the Markov chain of HACO algorithm is weakly ergodic.

2. **Strongly ergodic property:**

   When sampling of Metropolis is stable, the stationary distribution could be obtained at various values. It can be concluded that in the hybrid algorithm, the ACO algorithm provides the initial solution to TS algorithm; and it will not affect the stationary distribution as the frequency in sampling of TS algorithm is abundant. According to the characteristic of normal number, \( \sum_{m=0}^{\infty} \pi \left( m + \pi (m+1) \right) < \infty \) can be shown. And Markov chain of the solutions to the HACO algorithm is strongly ergodic from Lemma 3.

In summary, the HACO algorithm will converge to the set of global optimal value by probability 1. The conclusion is proved to be true.

### 5 Experiment and discussion

According to the above analysis, the global asymptotic convergence of HACO algorithm can be guaranteed theoretically. However, the proof is based on perfect operation situations such as sufficiently large tabu list, infinite time and so on. Considering about the reality of computer limitations and the limited time, we just take the convergence theory as the guidance in the specific computational experiments; some relaxations are made in accordance with the actual conditions on aspects of tabu length, search steps and so on. Therefore the solutions of some problems we obtained can just go nearly to rather than reach the optimal solution.
The proposed hybrid algorithm for JSP mentioned above is written in C programming language and conducted on an Intel Pentium IV 2.4 GHz PC with 4G memory. By simulations with the 13 typical benchmark problems (namely FT10, LA02, LA19, LA21, LA24, LA25, LA27, LA29, LA36, LA37, LA38, LA39, LA40), the makespans of HACO algorithm with ACO algorithm and GA can be compared and listed in Table 1.

In HACO algorithm, the population number is set to $|O| * 70\%$ and maximum of iterative generations is set to $|O| * 200\%$. We choose the unfeasible solution of ACO algorithm by probability $P_r$ ($P_r = 20\%$) as the initial solution of TS algorithm, after 50 search steps, algorithm will end if it could not find a better solution in TS algorithm. In ACO algorithm, we let population number $|O| * 200\%$, and evolution generation number of population is $|O| * 200\%$. The swarm size of GA is set to $|O| * 70\%$ and the maximum of iterative generations is set to $|O| * 70\%$, where the crossover probability is 0.85, and the mutation probability is 0.05.

Because HACO algorithm search domain is limited in active schedule, the algorithm is more possible to obtain the best solution. Table 1 shows that we can find the optimums of problem FT10, LA02, LA19, LA36 and LA38 when we apply HACO algorithm to solve the 13 benchmark problems. The makespans of problems mostly are better than that of ACO and GA.

The relative error percentage (PRE) of ten times experiment shows the approximation between optimisation and standard optimisation (PRE = $(\text{Avg} - C^{*}) * 100\%/C^{*}$). The average relative error percentage (APRE) of the average value of ten times experiment of HACO algorithm result is 7.32, which is 3.06\% respectively smaller than ACO, but is higher than GA by 1.99\%.

The experimental result shows that HACO algorithm has the ability of composition and colony, and improves the solution. At the same time, parallel TS algorithm is proposed to overcome shortcomings of signal TS algorithm, in which the result depends on the executed number of the algorithm and initial solution. It is indicated that HACO algorithm contributes to better search the best solution, when ACO algorithm’s results is better. Thus the overall search capability of the algorithm is improved, which make the algorithm get closer to the optimum solution.

### Table 1  
Optimal values and averages of ten times experiments

<table>
<thead>
<tr>
<th>Problem</th>
<th>c**</th>
<th>HACO</th>
<th>ACO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c*</td>
<td>Avg</td>
<td>Time/S</td>
<td>c*</td>
</tr>
<tr>
<td>FT10(10 × 10)</td>
<td>930</td>
<td>930</td>
<td>942.3</td>
<td>15.78</td>
</tr>
<tr>
<td>LA02(10 × 5)</td>
<td>655</td>
<td>655</td>
<td>655.0</td>
<td>2.92</td>
</tr>
<tr>
<td>LA19(10 × 10)</td>
<td>842</td>
<td>842</td>
<td>850.1</td>
<td>25.12</td>
</tr>
<tr>
<td>LA21(15 × 10)</td>
<td>1,046</td>
<td>1,060</td>
<td>1,110.2</td>
<td>89.11</td>
</tr>
<tr>
<td>LA24(15 × 10)</td>
<td>935</td>
<td>956</td>
<td>1,025.3</td>
<td>230.16</td>
</tr>
<tr>
<td>LA25(15 × 10)</td>
<td>977</td>
<td>1,012</td>
<td>1,022.7</td>
<td>200.32</td>
</tr>
<tr>
<td>LA27(20 × 10)</td>
<td>1,235</td>
<td>1,247</td>
<td>1,321.3</td>
<td>600.15</td>
</tr>
<tr>
<td>LA29(20 × 10)</td>
<td>1,152</td>
<td>1,301</td>
<td>1,405.1</td>
<td>500.27</td>
</tr>
<tr>
<td>LA36(15 × 15)</td>
<td>1,268</td>
<td>1,268</td>
<td>1,309.7</td>
<td>357.9</td>
</tr>
<tr>
<td>LA37(15 × 15)</td>
<td>1,397</td>
<td>1,528</td>
<td>1,537.4</td>
<td>600.86</td>
</tr>
<tr>
<td>LA38(15 × 15)</td>
<td>1,196</td>
<td>1,196</td>
<td>1,308.7</td>
<td>438.5</td>
</tr>
<tr>
<td>LA39(15 × 15)</td>
<td>1,233</td>
<td>1,284</td>
<td>1,409.5</td>
<td>476.7</td>
</tr>
<tr>
<td>LA40(15 × 15)</td>
<td>1,222</td>
<td>1,276</td>
<td>1,307.7</td>
<td>513.2</td>
</tr>
</tbody>
</table>

### Table 2  
PRE of ten times experiments

<table>
<thead>
<tr>
<th>Problem</th>
<th>HACO(PRE)</th>
<th>ACO(PRE)</th>
<th>GA(PRE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT10(10 × 10)</td>
<td>1.31</td>
<td>3.05</td>
<td>3.55</td>
</tr>
<tr>
<td>LA02(10 × 5)</td>
<td>0.00</td>
<td>0.00</td>
<td>4.16</td>
</tr>
<tr>
<td>LA19(10 × 10)</td>
<td>0.95</td>
<td>1.75</td>
<td>0.00</td>
</tr>
<tr>
<td>LA21(15 × 10)</td>
<td>5.78</td>
<td>14.88</td>
<td>2.10</td>
</tr>
<tr>
<td>LA24(15 × 10)</td>
<td>8.81</td>
<td>19.69</td>
<td>1.50</td>
</tr>
<tr>
<td>LA25(15 × 10)</td>
<td>4.47</td>
<td>16.38</td>
<td>5.07</td>
</tr>
<tr>
<td>LA27(20 × 10)</td>
<td>6.53</td>
<td>15.80</td>
<td>16.2</td>
</tr>
<tr>
<td>LA29(20 × 10)</td>
<td>18.1</td>
<td>17.45</td>
<td>15.51</td>
</tr>
<tr>
<td>LA36(15 × 15)</td>
<td>3.18</td>
<td>10.16</td>
<td>4.60</td>
</tr>
<tr>
<td>LA37(15 × 15)</td>
<td>9.13</td>
<td>9.46</td>
<td>3.16</td>
</tr>
<tr>
<td>LA38(15 × 15)</td>
<td>8.61</td>
<td>8.22</td>
<td>4.60</td>
</tr>
<tr>
<td>LA39(15 × 15)</td>
<td>12.5</td>
<td>7.40</td>
<td>1.16</td>
</tr>
<tr>
<td>LA40(15 × 15)</td>
<td>6.55</td>
<td>6.37</td>
<td>5.27</td>
</tr>
<tr>
<td>APRE</td>
<td>7.32</td>
<td>10.38</td>
<td>5.33</td>
</tr>
</tbody>
</table>

### 6 Conclusions

In the paper, the hybrid algorithm of ACO and TS is proposed for solving JSP. We present HACO algorithm with global asymptotic convergence by Markov chain theory. The algorithm has made full use of the large scale random search capability and the social cooperation of ACO algorithm. At the same time, some solutions generated by ACO algorithm were searched parallelly by TS algorithm. We apply the above convergence theory to computational experiment and find the optimum of problem FT10, LA02,
A hybrid ant colony optimisation algorithm for job shop problems and its convergence analysis

LA19, LA36 and LA38 in a short period, and improve convergence compared with ACO and GA. Thus the overall search capability of the algorithm is improved, which has demonstrated the effectiveness for solving JSP by HACO algorithm both in theory and practice.

Acknowledgements

This work was supported by Science and Technology specific Project for Strategic Emerging Industry, China (No. Y1A518H502) and Liaoning Province Doctor Starting Foundation, China (No. 20131126).

References


