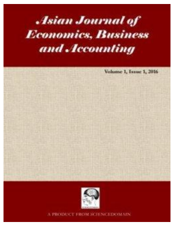


Asian Journal of Economics, Business and Accounting

6(2): 1-8, 2018; Article no.AJEBA.40350
ISSN: 2456-639X

Ant Colony Optimization for Storage Recombination Problem

Chia-Hui Cheng[1], Shih-Ming Pi[1*] and Kuo-Chen Li[1]

[1]Department of Information Management, Chung Yuan Christian University, Taiwan.

Authors' contributions

This work was carried out in collaboration between all authors. Author CHC designed the study, performed the system implementation, and wrote the first draft of the manuscript. Authors SMP and KCL managed the analyses of the study. Author SMP managed the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJEBA/2018/40350
Editor(s):
(1) Zsuzsanna Bacsi, Georgikon Faculty, University of Pannonia, Hungary.
Reviewers:
(1) Oyewola David, Federal University of Technology, Nigeria.
(2) Rajinder Tiwari, Jammu University, India.
Complete Peer review History: http://www.sciencedomain.org/review-history/23791

Original Research Article

Received 7th January 2018
Accepted 16th March 2018
Published 23rd March 2018

ABSTRACT

In a company, warehousing accounts for 20% of the operating costs and picking operations account for 50% of storage costs. Therefore, storage through the reorganization can be effectively reused these spaces, thereby affecting the follow-up operation of the warehouse staff picking operation time. In this study, after the rearrangement of storage spaces, the optimal spatial applications are sorted out. By constructing the mathematical programming model of problems and using the ant algorithm to solve the storage and the reconstruction path problem. Within a reasonable range of costs, a systematic solution method is developed to solve the "minimizing the moving distance" and find the optimal path by applying the mathematical model of the distance of storage and reorganization moving. There are two stages in this study, the first phase is based on the existing data, to calculate the ratio and the distance of the storage space, according to the calculation and reconstruction of storage after the reorganization of data to make an adjustment of the storage area of all storage status of the empty state of storage increased in the second stage, the storage location that needs to be stored and rebuilt is solved by the ant algorithm to find the shortest path. The simulation results suggest that this method is compared with the existing storage location reorganization mode. After the algorithm is actuated to (1) The

**Corresponding author: E-mail: happypi@gmail.com;*

method can increase the number of storage spaces and (2) the storage and reconstruction of the shorter moving distance and the results of this study can provide some reference for the warehouse staff.

Keywords: Ant colony optimization; storage recombination problem; path planning.

1. INTRODUCTION

Although warehousing is an integral part of the business, it costs about 20% of the total corporate entity's costs [1], which includes storage management costs, personnel costs, storage costs, depreciation charges, etc., Therefore, in order to reduce the cost of warehousing operations and improve operational efficiency, how to effectively use storage becomes an important management issues.

In the past studies, most of the topics it mainly discussed were storage location assignment and picking [2,3], while few studies focused on the storage location recombination. Storage recombination is moving goods into another storage bin expect more storage space is available. If the storage bin had stored some goods, the warehouse staff can't be quickly shelves, shelves must be unboxing. Due to unboxing operating hard, so placed pallets. It is influenced increase the difficulty of picking goods operations. Therefore, need to storage recombination to solve this problem. When many reserves are being reorganized, you have to plan for the best path to rebuild storage to reduce work time. There are many different combinations of paths. This problem is a matter of combinatorial optimization. In some cases, the computational time required to find the optimal solution is very large or not feasible. Based on the consideration of efficiency, many studies develop heuristic algorithms to solve the original problem. The solving characteristic of this heuristic algorithm usually needs logic to find the local minimum solution step by step. The solution obtained is not always the global minimum, but the approximate best solution, so this study hopes to optimize to improve the storage path reorganization of the problem. Analyze the storage recombination path, refer to the previous scholar's content [4], the results show that the ant colony optimization outperforms other nature-inspired algorithms are to plan the best path effective method.

Storage recombination can make more storage bin to empty, the goods can be placed directly on the storage, do not place the pallets. Picking goods can be taken out in the storage bin to reduce picking time. However, the cost of picking operations occupy the overall logistics costs as high as 50% [5].

2. LITERATURE REVIEW

The Ant colony optimization (ACO) is developed according to the observation that real ants are capable of finding the shortest path from a food source to the nest without using visual cues [6]. To illustrate how the "real" ant colony searches for the shortest path, an example from [6] will be introduced for better comprehension. In Fig. 1(a), suppose A is the food source and E is the nest. The goal of the ants is to bring the food back to their nest. Obviously, the shorter paths have the advantage compared with the longer ones. Suppose that at time $t = 0$ there are 30 ants at point B (and 30 at point D). And at this moment there is no pheromone trail on any segments. So the ants randomly choose their path with equal probability. Therefore, on the average 15 ants from each node will go toward H and 15 toward C (Fig. 1(b)). At $t = 1$ the 30 new ants that come to B from A find a trail of intensity, 15 on the path that leads to H, laid by the 15 ants that went that way from B, and a trail of intensity 30 on the path to C, obtained as the sum of the trail laid by the 15 ants that went that way from B and by the 15 ants that reached B coming from D via C (Fig. 1(c)). The probability of choosing a path is therefore biased, so that the expected number of ants going toward C will be double of those going toward H: 20 versus 10, respectively. The same is true for the new 30 ants in D which come from E. This process continues until all of the ants will eventually choose the shortest path.

Given an n-city traveling salesman problem (TSP) with distances d_{ij}, the artificial ants are distributed to these n cities randomly. Each ant will choose the next to visit according to the pheromone trail remained on the paths just as mentioned in the above example. However, there are two main differences between artificial ants and real ants: (1) the artificial ants have "memory"; they can remember the cities they have visited and therefore they would not select

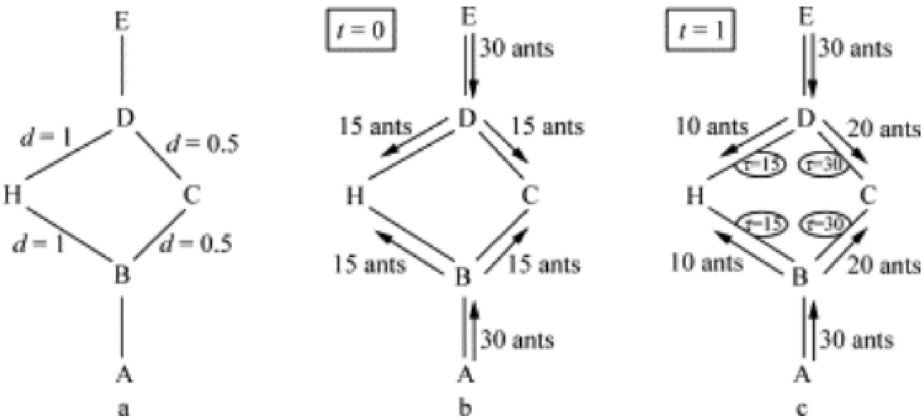


Fig. 1. An example with ant choose path

those cities again. (2) The artificial ants are not completely "blind"; they know the distances between two cities and prefer to choose the nearby cities from their positions. Therefore, the probability that city j is selected by ant k to be visited after city I could be written as follows:

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{s \in allowed_k} [\tau_{is}]^{\alpha} \cdot [\eta_{is}]^{\beta}} & j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In the beginning, l ants are placed in the n cities randomly. Then each ant decides the next city to be visited according to the probability p_{ij}^{k} was given by Eq. (1). After n iterations of this process, every ant completes a tour. Obviously, the ants with shorter tours should leave more pheromone than those with longer tours. Therefore, the trail levels are updated as on a tour each ant leaves pheromone quantity given by Q/L_k, where Q is a constant and L_k the length of its tour, respectively. On the other hand, the pheromone will evaporate as the time goes by. Then the updating rule of τ_{ij} could be written as follows:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k} \tag{3}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ travels on edge } (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Where t is the iteration counter, $\rho \in [0, 1]$ the parameter to regulate the reduction of τ_{ij}, $\Delta\tau_{ij}$ the total increase of trail level on the edge (i, j) and $\Delta\tau_{ij}^{k}$ the increase of trail level on the edge (i, j) caused by ant k, respectively. After the pheromone trail updating process, the next iteration $t + 1$ will start.

A hybrid approach of genetic algorithm (GA) and ant colony optimization (ACO) for the traveling salesman problem. By this way, GA can avoid its useful building blocks being frequently destroyed by genetic operations. Experiments on TSPLIB validated the building block learning capability of our approach [7].

This study proposes a rank-based ant colony optimization (ACO) method with a rank-based nonlinear selective pressure function and a modified Q-learning method to enhance the convergence characteristics of original ACO [6], which defines the probability of exploring a city to be visited by ants with a random proportional rule. This probability distribution of the random proportional rule, which is similar to that of the stochastic universal sampling method generally applied to the selection operation in genetic algorithms, is good for exploring favorable paths in small traveling salesman problems (TSPs), but inefficient at exploring such paths in large TSPs. Therefore, this study presents the rank-based nonlinear selection pressure function, based on ranking of [τ (r, z)][η (r, z)]β, to improve the performance of the state transition rule of the original ACO, as well as a modified Q-learning method to solve reinforcement learning problems efficiently [8].

3. RESEARCH METHODS

3.1 Problem Description

The issue of recombination of storage mainly focuses on the current situation of the distribution of stored goods on the shelf, and the recombination of diversified operations to exchange inventory on the stored positions, with the goal of 0% usage rate. Due to manual storage repositioning operation, consider the storage height and reorganization moving distance. This study not only discusses the method of emptying the reserve, but also adds each stored position, and then uses the ant algorithm to calculate the shortest path of the reforming operation. Based on this algorithm, empty storage spaces can be added to minimize the movement distance, to increase warehouse operators more flexibility in storage operations, improve warehouse operation efficiency.

3.2 Mathematical Models

Parameters are as follows:

Area_{ijk}	: the same storage area.
Seat_{ijk}	: the same storage shelf.
H_{ijk}	: need to storage recombine.
SH_{ijk}	: storage bin had more space.
SS_{ijk}	: the i shelf, the j layer, the k storage bin available space
SU_{ijk}	: Whether the storage bin is empty.
SV_{ijk}	: This storage bin has been used to save the storage space.
SR_{ijk}	: The remaining space of the storage bin.
SRt_{ijk}	: The amount of storage bin capacity ratio.
T_{ijkc}	: Calculate the number of exchanges on the c whether to move.
c	: exchange times.
Dist(i,j,k)	: the distance between the storage bin and the recombine storage bin.
$Cs(i,j,k)$	: the coordinates of the center point between the storage bin and the recombine storage bin.
M	: Ant Quantity
α	: the relative importance of the trail.
β	: the relative importance of the visibility.
P	: trail persistence.
Q	: a constant related to the quantity of trail laid by ants.

The objective function: "Minimize moving distance".

Minimize

$$\sum_{ijk}\sum_{c}\sum_{Dist} T_{ijke}$$

Subject to:

$$Area_{ijk} = \begin{cases} 1 & \text{If the same storage area} \\ 0 & \text{Otherwise,} \end{cases} \tag{5}$$

$$Seat_{ijk} = \begin{cases} 1 & \text{If the same storage shelf} \\ 0 & \text{Otherwise,} \end{cases} \tag{6}$$

$$H_{ijk} = \begin{cases} 1 & \mathrm{j} > 2 \text{If layer is greater than 2} \\ 0 & \mathrm{j} \le 2 \ \text{Otherwise,} \end{cases} \tag{7}$$

$$SH_{ijk} = \begin{cases} 1 & \text{If storage bin have space can be to store} \\ 0 & \text{Otherwise,} \end{cases} \tag{8}$$

$$SU_{ijk} = \begin{cases} 1 & \mathrm{SRt_{ijk}} > 0\% \text{ If storage bin is not empty} \\ 0 & \text{Otherwise,} \end{cases} \tag{9}$$

$SRt_{ijk} = \frac{SV_{ijk}}{SS_{ijk}} \times 100\%$ "Storage bin capacity ratio. (10)

In *c* times the number of exchange can't exceed the SS_{ijk}(storage bin available space).

$$\sum_{i,j,k} T_{ijke}, \forall \mathrm{SS}_{ijk} \tag{11}$$

Other Calculate formula:

Calculate the distance between the storage bin and the recombine storage bin, and plus multiply the difficulty of different floors.

$$\mathrm{Dist}(i,j,k) = \sqrt{\left(x_{ijk1} - x_{ijk2}\right)^2 + \left(y_{ijk1} - y_{ijk2}\right)^2 + \left(z_{ijk1} - z_{ijk2}\right)^2} + \left(y_{ijk1} - y_{ijk2}\right) \times 1.1 \tag{12}$$

Calculate coordinates of the center point between the storage bin and the recombine storage bin.

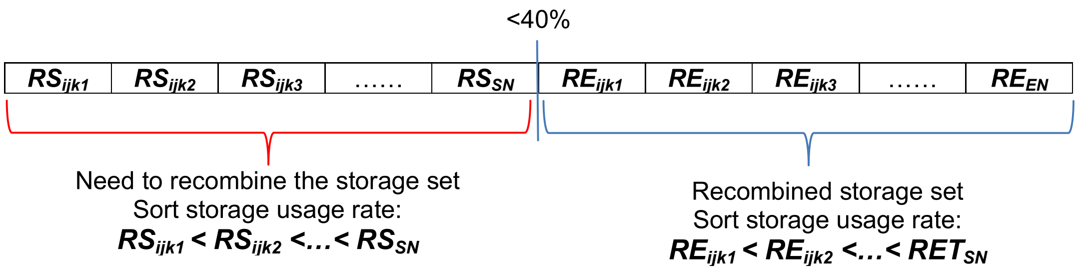
$$\mathrm{Cs(k, i+j)} = \left(\frac{\mathrm{x_{k1} - x_{k2}}}{2}, \frac{\mathrm{y_{i+j1} - y_{i+j2}}}{2}\right) \tag{13}$$

3.3 Procedure

Step 1: Data input.

Step 2: Calculate the usage amount of storage.

Step 3: If$SH_{ijk} = 1$and $SU_{ijk} = 1$, sort of storage bin usage rate according to Eq. (10).



Step 4: Calculate the distance between the storage bin and the recombine storage bin according to Eq. (12).

Step 5: Calculate coordinates of the center point between the storage bin and the recombine storage bin according to Eq. (13).

Step 6: Establish each node coordinates.

Step 7: Coding an ant colony optimization program (Fig 2) for storage bin recombination path planning

Step 8: Produce a proposal for storage recombination

```
Algorithm ACO
   Input:
      Establish each node coordinates
   ACO Parameters:
      MaxIt  % Maximum Number of Iterations
      nAnt   % Number of Ants (Population Size)
      Q=1;
      tau0=10*Q/(nVar*mean(model.D(:)));        % Initial Phromone
      alpha       % Phromone Exponential Weight
      beta        % Heuristic Exponential Weight
      rho         % Evaporation Rate
   Initialization:
      eta=1./model.D;              % Heuristic Information Matrix
      tau=tau0*ones(nVar,nVar);    % Phromone Matrix
      BestCost=zeros(MaxIt,1);     % Array to Hold Best Cost Values
   ACO Main Loop:
     for it=1 to
       % Move Ants
        for k=1:nAnt
           for l=2:nVar
              RouletteWheelSelection()
           end
           ant(k).Cost=CostFunction();
           if ant(k).Cost<BestSol.Cost
              BestSol=ant(k);
   Update Phromones:
      for k=1:nAnt
        for l=1:nVar
          tau(i,j)=tau(i,j)+Q/ant(k).Cost;
      tau=(1-rho)*tau;  % Evaporation
      BestCost(it)=BestSol.Cost;  % Store Best Cost
   Results
     Store Best Cost
```

Fig. 2. Ant colony optimization source code

4. RESULTS AND DISCUSSION

4.1 Result Analysis

Data input and sort by calculate the usage amount of storage. And get a reorganization of the storage combination (Fig. 3).

Calculate the distance between the storage bin and the recombine storage bin according to Eq. (12). And get the shortest distance is C01-661->C01-663, its distance = 2. Then calculate the center position according to Eq. (13): X = (3+1) / 2 *10 = 20, Y = (66+66)/2 = 66.

In this study, we carry out to get the storage coordinate table (Table 1).

Calculations using the Matlab R2012a coding Aco program to plan to the path. According to [9], one of the main ideas introduced by MAX – MINAnt System, the utilization of pheromone trail limits to prevent premature convergence, can also be applied in a different way. And under α∈ (0,0.5,1,2), β∈ (0.1,2,3,4,5), ρ∈ (0.1,0.2,0.3,0.5,0.7,0.9) using the ant algorithm to solve the optimization problem, it is found that is best [9]. Through the simulation results of different parameters, it is found that the lowest path cost is α = 1, β = 5, ρ = 0.1, M = 150~200. The lowest path cost is calculated to be 205.442, so this parameter is defined as the parameter value of the ant algorithm in this study.

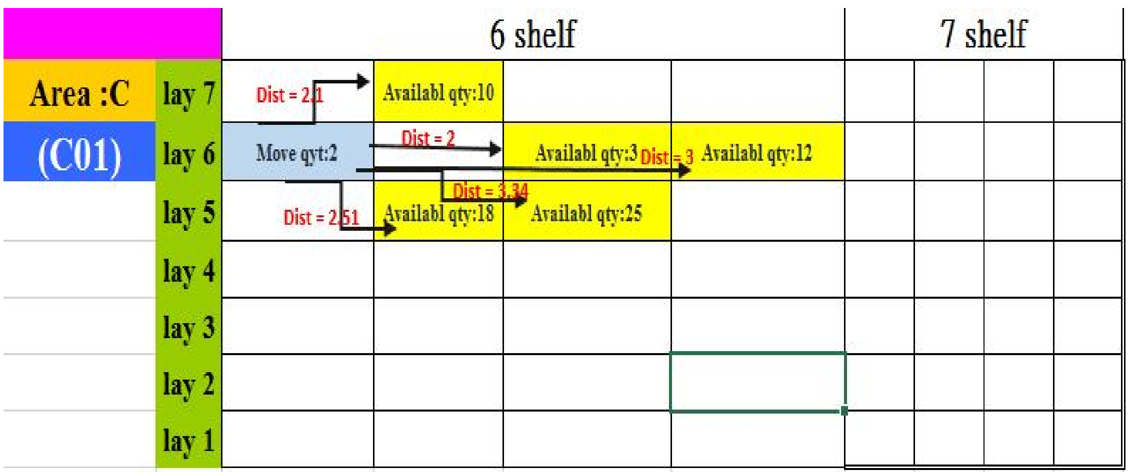


Fig. 3. Matching results diagram

Table 1. Storage recombination coordinate

Note	X	Y	Note	X	Y	Note	X	Y
1	20	66	15	15	53	29	35	63
2	25	62	16	15	31.5	30	15	45
3	35	56	17	15	35	31	30	47
4	25	76	18	10	35.5	32	30	42
5	15	73	19	20	75	33	35	45.5
6	30	66	20	10	51.5	34	20	64
7	25	46	21	30	31	35	20	71.5
8	15	62	22	25	35.5	36	20	44
9	25	43	23	20	42	37	30	61
10	25	57	24	35	53.5	38	30	74.5
11	15	41	25	15	71	39	20	33.5
12	30	30.5	26	35	73	40	40	52.5
13	20	76.5	27	15	74			
14	20	75.5	28	35	43.5			

Table 2. Parameter setting simulation test results

α	*β*	*ρ*	Path cost	*α*	*β*	*ρ*	Path cost
1	2	0.1	211.509	1	3	0.1	212.391
1	2	0.2	217.300	1	3	0.2	206.947
1	2	0.3	208.765	1	3	0.3	213.483
1	2	0.5	209.036	1	3	0.5	216.867
1	2	0.7	208.940	1	3	0.7	209.048
1	2	0.9	216.151	1	3	0.9	211.653
α	*β*	*ρ*	Path cost	*α*	*β*	*ρ*	Path cost
1	4	0.1	210.503	1	5	0.1	206.178
1	4	0.2	214.034	1	5	0.2	208.121
1	4	0.3	212.789	1	5	0.3	215.045
1	4	0.5	214.977	1	5	0.5	214.639
1	4	0.7	209.774	1	5	0.7	211.767
1	4	0.9	220.981	1	5	0.9	217.869

Ant qty (*M*).	5	10	25	50	100	150	200
Path cost	209.178	209.582	206.178	206.860	206.335	205.442	205.756

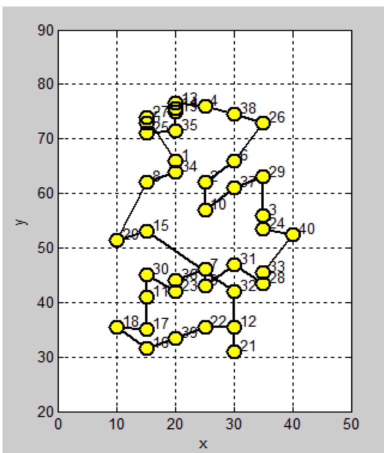


Fig. 4. Best path

The best storage recombination path is :29 → 37 → 2 → 10 → 24 → 3 → 40 → 33 → 28 → 32 → 9 → 7 → 31 → 12 → 21 → 22 → 39 → 16 → 17 → 18 → 11 → 30 → 36 → 23 → 15 → 20 → 8 → 34 → 1 → 35 → 25 → 5 → 27 → 14 → 19 → 13 → 4 → 38 → 26 → 6。

4.2 The Impact of Picking Goods Time Analysis

The goods save storage bin can be reduced picking times by 2 minutes on storage of pallets. Because storage of pallets needs to find the demand for the batch number in the box by box. Taking C01 as an example, an increase of 40 empty slots saves 40 × 2 = 80 minutes of picking time. Storage recombination need to process time is 10(Picking up) × 40 (empty storage bin) +173.37 (total distance) + 10(Putting in) × 40(empty storage bin) ÷ 60(sec) = 22.89 minutes. This algorithm can reduce the storage area C01 warehouse operation time 56.11 minutes. The average reduction of 36 minutes.

Table 3. Warehousing personnel to reduce operating time

Area	Total storage bin	Increase the empty space qty.	Distance	picking times(min.)	Storage recombination time(min)	Reduce time(min)
C01	196	40	173.37	80	22.89	57.11
C02	200	25	114.37	50	14.41	35.59
C03	200	18	84.67	36	10.41	25.59
C04	200	17	87.66	34	9.96	24.04
C05	200	32	166.44	64	18.77	45.23
C06	200	27	125.78	54	15.60	38.40
C07	200	27	148.37	54	15.97	38.03
C08	224	21	87.43	42	11.96	30.04
Total				414	119.97	36.75(avg.)

5. CONCLUSION

Applying the algorithm to solve the problem of storage and recombination path, we develop a set of storage reformation method that is most similar to the optimal solution within the acceptable cost range. The conclusions of this study are as follows:

1. The parameter combinations α = 1, β = 5, ρ = 0.1, M = 150~200(α is the concentration of pheromones, β is the absolute distance parameter, ρ is the pheromone evaporation coefficient, M is the ant quantity) can be found in the search solution to find the best solution.
2. For the follow-up, the operation can be put on the shelf, not placed pallets, picking operation to reduce the time, about 36 minutes can be reduced / storage area.

In this paper, we choose the ant colony optimization to solve the path optimization problem of storage recombination. Ant colony optimization is a good tool to find out the desired solution by the continuous search of the ant colony. In the past ten years, the ant algorithm has made considerable progress in solving the path vehicle planning TSP. Therefore, this paper suggests that other evolutionary algorithms may be used to solve such storage path reorganization path planning problems in order to compare the results and efficiency.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Kearney E, Kearney A. Excellence in logistics 2004. European Logistics Association, Brussels; 2004.
2. Gademann A., van Den Berg JP, van der Hoff HH. An order batching algorithm for wave picking in a parallel-aisle warehouse. IIE Transactions. 2001;33(5):385-398.
3. Linn R, Wysk R. An analysis of control strategies for an automated storage/retrieval system. INFOR: Information Systems and Operational Research. 1987;25(1):66-83.
4. Dorigo M, Gambardella, LM. Ant colony system: a cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation. 1997;1(1):53-66.
5. Hwang H, BAEK WJ, Lee MK. Clustering algorithms for order picking in an automated storage and retrieval system. The International Journal of Production Research. 1988;26(2):189-201.
6. Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 1996;26(1):29-41.
7. Gong D, Ruan X. A hybrid approach of GA and ACO for TSP. In Intelligent Control and Automation, WCICA 2004. Fifth World Congress. 2004;3:2068-2072.
8. Liu JL. Rank-based ant colony optimization applied to dynamic traveling salesman problems. Engineering Optimization. 2005; 37(8):831-847.
9. Stützle T, Hoos HH. MAX–MIN ant system. Future Generation Computer Systems. 2000;16(8):889-914.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history/23791