Proceedings of the 2006 IEEE
International Conference on Information Acquisition
August 20 - 23, 2006, Weihai, Shandong, China

Solution to 0/1 Knapsack Problem Based on Improved Ant Colony Algorithm

Hanxiao SHI

Computer and Information Engineering College
Zhejiang Gongshang University
Hangzhou, China

hxshi@mail.zjgsu.edu.cn

Abstract **- Ant colony algorithms analogize the social behaviour of ant colonies, they are a class of meta-heuristics which are inspired from the behavior of real ants. It was applied successfully to the well-known traveling salesman problem and other hard combinational optimization problems. In order to apply it to the classical 0/1 knapsack problem, this paper compares the difference between the traveling salesman problem and the 0/1 knapsack problem and adapts the ant colony optimization (ACO) model to meet researches' purpose. At the same time, the parameters in ACO model are modified accordingly. The experiments based on improved ant colony algorithms show the robustness and the potential power of this kind of meta-heuristic algorithm.**

Index Terms - ant colony algorithm, knapsack problem, traveling salesman problem, meta-heuristic algorithm

I. INTRODUCTION

Knapsack problem occurs frequently in life. If there is a best approach to that problem be put forward, it must create highly economic value and decision function, such as goods dispatching problems, containers' loading problems and so on. This problem was proved to be NP-complete [1]; its original search space has 2^n possible values, therefore an exhaustive search would take $O(2^n)$ [2,3,4] time to find a solution in the worst case. Solving the knapsack problem can be seen as a way to study some large problems in number theory and, because of its exponential complexity, some public –key cryptosystem are based on it. Therefore, much effort has been made in order to find the techniques which could lead to practical algorithms with reasonable running times.

Greedy algorithm is an approximate algorithm while meta-heuristic algorithm can't ensure the result is an optimization answer. After making lots of research on combination optimization problems with ant colony algorithm, this author tries to expand its application area based on the existing mathematics model. This paper proposes an improved ant colony algorithm to solve the 0/1 knapsack problem. Through several times' experiments, the idea of that algorithm is proved to be feasible.

II. 0/1 KNAPSACK PROBLEM AND ANT COLONY SYSTEM

A. The Description of 0/1 Knapsack Problem

0/1 knapsack problem is referred to n articles with various value and weight, as well as partial articles are selected. To each article, there're two ways: selected or not. The total weight of selected articles can't overrun that of knapsack appointed boundary and should reach the maximum of total value. If the total weight of all the articles is less than that of the knapsack, then the problem will be extremely simple. And the benefit is equal to the total value of the whole articles. But actually, the knapsack's weight is always less than the total weight of the articles. 0/1 knapsack problem can be formally described as follows:

Suppose W is the total weight limitation of knapsack, a vector $M(w_1,w_2,\ldots\ldots,w_n)$ consists of n articles' weights, and another vector $V(v_1,v_2,\ldots\ldots,v_n)$ consists of their values, $W>0$, $w_i>0$, $v_i>0$, $(1 \le i \le n)$. To find another dimension-n vector $(x_1,x_2,\ldots,x_n)$, $x_i \in \{0,1\}$, $1 \le i \le n$, zero represents the article won't be selected, while one just the opposite. Thereout, 0/1 knapsack problem's requirement:

$$Max \sum_{i=1}^{n} v_i x_i$$

and satisfy two condition constraints downwards:

$$1.\ \sum_{i=1}^{n} w_i x_i \le W$$

$$2.\ x_i \in \{0;\ 1\} \quad 1 \le i \le n$$

Therefore, to the 0/1 knapsack problem, it gets the answer through making the decision sequence of the variable x_1, x_2, …, x_n. And to the variable x_i, the decision is that sets its value to zero or one.

B. Ant System and Ant Colony System

Because of being a hard combinational optimization problem, 0/1 knapsack problem hasn't been completely solved so far. At present, the popular solution in this field is the application of artificial intelligence or the bionic meta-heuristic algorithm. With quite a long time's research on computer algorithm, the author considered the effective approach to the problem was to set up a suitable mathematic model. In view of its supereminent performance on solving TSP problem and successful application on a series of complicated combinational optimization problems, the ant system and ant colony system have been applied successfully in other applications such as the quadratic assignment problem [5], data mining [6], space-planning [7], job-shop scheduling and graph coloring [8]. The author attempted to get the answer

1-4244-0529-7/06/$20.00 ©2006 IEEE

to 0/1 knapsack problem through trying the expansion and improvement of ant colony algorithm.

Ant colony algorithm stems from that real ants can find the shortest path from a good source to nest. Ants are moving on a straight line that connects a food source to their nest. It is well know that the primary means for ants to form and maintain the line is a pheromone trail. Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction rich in pheromone. This elementary behaviour of real ants can be used to explain how they can find the shortest path that reconnects a broken line after the sudden appearance of an unexpected obstacle has interrupted the initial path. In fact, once the obstacle has appeared, those ants which are just in front of the obstacle cannot continue to follow the pheromone trail and therefore they have to choose between turning right or left. In this situation we can expect half the ants to choose to turn right and the other half to turn left. A very similar situation can be found on the other side of the obstacle. It is interesting to note that those ants which choose, by chance, the shortest path around the obstacle will more rapidly reconstitute the interrupted pheromone trail compared to those who choose the longer path. Thus, the shorter path will receive a greater amount of pheromone per time unit and in turn a larger number of ants will choose the shorter path. Due to this positive feedback (autocatalytic) process, all the ants will rapidly choose the shorter path. The most interesting aspect of this autocatalytic process is that finding the shortest path around the obstacle seems to be an emergent property of the interaction between the obstacle shape and ants distributed behaviour: although all ants move at approximately the same speed and deposit a pheromone trail at approximately the same rate, it is a fact that it takes longer to contour obstacles on their longer side than on their shorter side which makes the pheromone trail accumulate quicker on the shorter side. It is the ants preference for high pheromone trail levels which makes this accumulation still quicker on the shorter path. It will be shown that how a similar process can be put to work in a simulated world inhabited by artificial ants that try to solve the traveling salesman problem (TSP) [9].

The operation of ant system can be illustrated by the classical TSP. A traveling salesman problem is seeking for a round route covering all cities with minimal total distance. Suppose there are n cities and m ants. The entire algorithm is started with initial pheromone intensity set to τ_0 on all edges. In every subsequent ant system cycle, or called episode, each ant begins its tour from a randomly selected starting city and is required to visit every city once and only once. The experience gained in this phase is then used to update the pheromone intensity on all edges.

The algorithm of the ant system for TSP is depicted as follows [10,11]:

Step 1: Randomly select the initial city for each ant. The initial pheromone level between any two cities is set to be a small positive constant. Set the cycle counter to be 0.

Step 2: Calculate the transition probability from city r to city s for the kth ant as

$$P_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)]\cdot[\eta(r,s)]^\beta}{\sum_{u\in J_k(r)}[\tau(r,u)]\cdot[\eta(r,u)]^\beta} & if\ s\in J_k(r), \\ 0 & otherwise, \end{cases} \quad (1)$$

where r is the current city, s is the next city, $\tau(r,s)$ is the pheromone level between city r and city s, $\eta(r,s)=1/\delta(r,s)$ the inverse of the distance $\delta(r,s)$ between city r and city s, $J_k(r)$ is the set of cities that emain to be visited by the kth ant positioned on city r, and β is a parameter which determines the relative importance of pheromone level versus distance. Select the next visited city s for the kth ant with the probability $P_k(r,s)$. Repeat Step 2 for each ant until the ants have toured all cities.

Step 3: Update the pheromone level between cities as

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \sum_{k=1}^{m}\Delta\tau_k(r,s), \quad (2)$$

$$\Delta\tau_k(r,s) = \begin{cases} \dfrac{1}{L_k} & if\ (r,s)\in route\ done\ by\ ant\ k, \\ 0 & otherwise, \end{cases} \quad (3)$$

$\Delta\tau_k(r,s)$ is the pheromone level laid down between cities r and s by the kth ant, L_k is the length of the route visited by the kth ant, m is the number of ants and $0 < \alpha < 1$ is a pheromone decay parameter.

Step 4: Increment cycle counter. Move the ants to the originally selected cities and continue Steps 2-4 until the behavior stagnates or the maximum number of cycles has reached, where a stagnation is indicated when all ants take the same route.

From Eq. (1) it is clear ant system (AS) needs a high level of computation to find the next visited city for each ant. In order to improve the search efficiency, the ant colony system (ACS) was proposed [11]. ACS is based on AS but updates the pheromone level before moving to the next city (local updating rule) and updating the pheromone level for the shortest route only after completing the route for each ant (global updating rule) as

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s), \quad (4)$$

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1} & if\ (r,s)\in global\ best\ route, \\ 0 & otherwise, \end{cases} \quad (5)$$

where L_{gb} is the length of the shortest route and α is a pheromone decay parameter.

III. An Improved Ant Colony Model for 0/1 Knapsack Problem

TSP is the problem of finding a shortest closed tour which visits all the cities in a given set. An ant has left a pheromone trail as heuristic information on the route connecting two cities. So did the length of route itself. S is a set of cities, |S|=n. At time t, it is acquired that the partial solution of ant k is $tabu_k(t)=\{y_1,y_2,\ldots\ldots,y_j\}$, and y_i is the latest element that joins the set of $tabu_k$. During the process of finding next element y_p, it is mainly in terms of the probability P_{jp} between element y_j and y_p. And let y_p with the maximum probability join $tabu_k$. Next time, ant k starts at y_{j+1} and continues to find an element y_q which makes probability $P_{j+1,q}$ to be maximum among the rest elements. And continue like this, when the ant reaches objective condition, there will be the expression $|tabu_k|=n$. Thereout, this is a sequent process for the ant to choose next city continually. And the sequence of the elements in $tabu_k$ embodies the route which the ant walked. The state of step j is illustrated in Fig.1.

$$p \in \{j+1, j+2, ..., n\},\ q \in \{j+2, j+3, ..., n\}$$

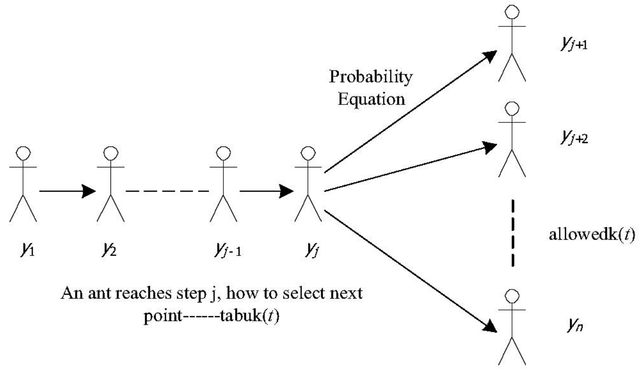

Fig. 1 An ant colony model for TSP

But there is no conception of path in 0/1 knapsack problem. Through the research on TSP problem, it is found that with a trifle modification on the original model, the ant colony algorithm will be suitable for 0/1 knapsack problem. In other words, the quiddity of TSP problem is that the more pheromones in a route, the more probability an ant will select it. And the conclusion is that the route must be shorter. It can be applied this central idea to the 0/1 knapsack problem: the more pheromones assemble in article, the more probability the article will be selected. In this paper, S is defined as the set of articles, $|S|=n$. At time t, it is acquired that the partial solution of ant k is $tabu_k(t)=\{y_1,y_2,...y_j\}$, and y_j is latest article that join the set of $tabu_k$. During the process of searching next article, which is different from TSP, It must be calculated the total weight of all the articles in $tabu_k$ firstly, and then calculate the surplus capacity of knapsack. According to the surplus capacity of knapsack and definite strategy, by probability (8), article y_p with the maximum probability is put into $tabu_k$ in succession until it is beyond the capacity of the knapsack. When algorithm reached the objective condition, there's an expression $|tabu_k| \le n$. Through comparison, it could be seen that the ant search is sequent in TSP, but there is no such character in 0/1 knapsack problem. See Fig.2 in an ant colony model for 0/1 knapsack problem.

$$p \in \{j+1, j+2, ...n\}$$

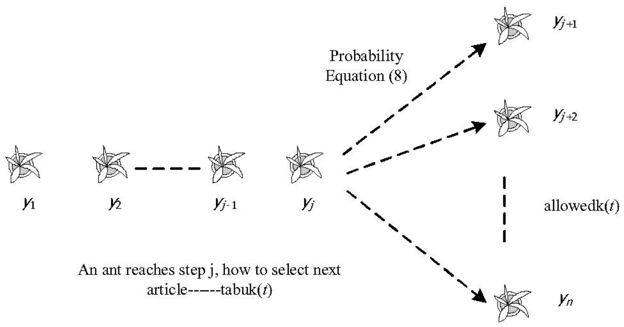

Fig. 2 An ant colony model for 0/1 knapsack problem

At time t, the pheromone of article i is $\tau_i(t)$, and $\Delta\tau_i(t)$ acts as the pheromone which are accumulated by article i after time t. At initialization, initial pheromone of every article is defined to be constant C and at time t+1, the pheromone of article i concluded by the underneath expression:

$$\tau_i(t+1) = \rho\tau_i(t) + \Delta\tau_i(t) \quad (6)$$

ρ denotes the permanence of pheromone. 1-ρ can be perceived as the degree of attenuation of pheromone in one article: ($0 \le \rho < 1$).

In addition:

$$\Delta\tau_i(t) = \sum_{k=1}^{m} \Delta\tau_i^k(t) \quad (7)$$

Among others things, $\Delta\tau_i^k(t)$ $=w_i$*G($Weight_k$), G($Weight_k$) =Q/$Weight_k$, which expressions are defined as the pheromone function that ant k left behind on article i. Q is a constant, $Weight_k$ is the weight that ant k loads. $Weight_k=\sum_{i=1}^{|tabu_k|} w_i$.

Objective function of each ant is composed of two parts: One is above-mentioned $Weight_k$, the other is the total value ($Value_k$) of all the articles in $tabu_k$. $Value_k=\sum_{i=1}^{|tabu_k|} v_i$

According to the conditions of the ant in article y_j to select next article y_p, $tabu_k(t)=\{y_1,y_2,......,y_j\}$ is defined as ant k acquired partial solution at time t. The underneath expression could be inferred:

$allowed_k(t) \subseteq S- tabu_k(t)$

$y_p \in allowed_k(t)$.

The probability equation of next article that ant k selects is as follows:

$$P_{y_p}^k = \frac{[\tau_{y_p}(t)]^\alpha [\eta_{y_p}(t)]^\beta}{\sum_{j \in allowed_k(t)} [\eta_j]^\beta [\tau_j(t)]^\alpha}$$

$$(p \in \{j+1, j+2, ..., n\}). \quad (8)$$

Among other things, the more value of this heuristic factor η_y, the more probability article y_p will be selected initially. The value of η_{y_p} is referred to the ratio of the value of selected article and its weight and article in $tabu_k$ satisfy the bound of 0/1 knapsack. Let α and β respectively represents information accumulated during an ant's movement and the different function of meta-heuristic factor when an ant selects various article. As to the value of parameters Q, C, α , β , ρ , they could be identified the optimization combination by experimentation.

IV. ALGORITHM DESCRIPTION

The detail steps of algorithm can be stated as follows:

Step 1: Initialization. Set g to be zero (g represents iteration times or evolution algebra), τ_i and $\Delta\tau_i$ is initialized, and place m ants on n articles and put each ant's present article to each current solution set.

Step 2: To each ant k (k=1 ,….., m), it can select next article j based on above-mentioned probability (8).And then let article j join each current solution set $tabu_k$.

Step 3: According to $tabu_k$, we could calculate each ant's value of objective function. If the value cannot reach the requirement that satisfies knapsack's capacity, turn to step two and record the best solution at present.

Step 4: According to pheromone, update equation (6) and pheromone intensity of all the articles. To each article i, it is set to as follows:

$\Delta\tau_i$=0, g=g+1

Step 5: If g is less than scheduled iteration time and has no degenerate action, turn to step two.

Step 6: Output the best solution.

Here implies thus a non-proved hypothesis, that is, after each iteration, the solution set is always optimized precedent one.

V. EXPERIMENTAL RESULTS AND PERFORMANCE STUDY

Above-mentioned algorithm is carried out on Turbo C. It is used the idea of best priority search method in artificial intelligence during programming realization. According to current and objective situation, it is found that it couldn't get the best optimization solution if it only chooses heuristic factor η rearrange the table of $allowed_k$. Then under the bound of knapsack's capacity, it has to make dynamic adjustment for each ant's current solution set. And completed heuristic factor should be [η $tabu_k(t)$] .Parameters ρ is evaluated 0.7 and Q is a random positive integer that is more than ten but less than one hundred, $\alpha\in$ (0, 5), $\beta\in$ [0, 5].

The first experiment's data are as follows:

n = 10, W= 269, M= {95, 4, 60, 32, 23, 72, 80, 62, 65, 46}, V= {55, 10, 47, 5, 4, 50, 8, 61, 85, 87}

After four times' iteration, the result is basically stabile. The result is illustrated in Fig. 3.

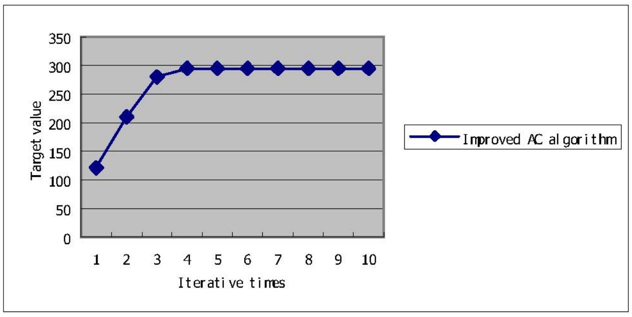

Fig. 3 An result of the first experiment (n=10)

The second experiment's data are as follows:

n = 20, W = 878,

M = {92, 4, 43, 83, 84, 68, 92, 82, 6, 44, 32, 18, 56, 83, 25, 96, 70, 48, 14, 58},

V = {44, 46, 90, 72, 91, 40, 75, 35, 8, 54, 78, 40, 77, 15, 61, 17, 75, 29, 75, 63}

After nine times' iteration, the result is basically stabile. The result is illustrated in Fig. 4.

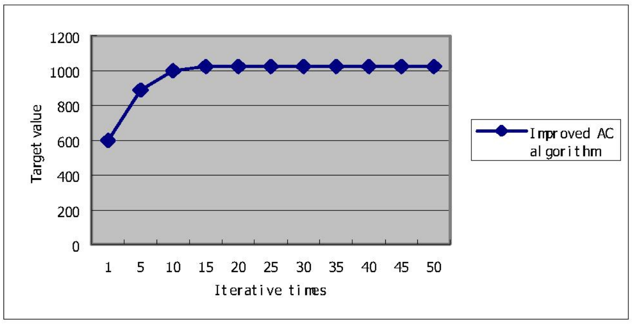

Fig. 4 An result of the second experiment (n=20)

Based on above results, some conclusions can be made. Firstly, the iteration times is in direct proportion to the size of n. Secondly, if the question is very complicated, it needs more iteration times for getting better target value.

The time complexity of this algebra is O(g*m*n^2). Here g refers to iteration number or evolution algebra and generally isn't more than ten. This is mainly because in the process of implementation, it increases the time of internal cycle which prominently decreases that of external cycle. In this way, ideal effect is reached. Generally, it is defined that the number of ants (m) is equal to that of articles (n).

VI. CONCLUSION

At first, ants' pheromone on each article is a constant. When calculating probability heuristic factor η plays a fundamental role. At this moment, according to the optimization idea, it's easy to fall into partial optimization. But actually one or some selections don't meet our purpose. Through artificial interference of $tabu_k$, the right articles join instead of improper ones. To one of the ants, its pheromone is distributed to the selected articles by proportion. In view of a group behavior, to certain specifically article, its pheromone is the sum of the pheromones that ants left behind. It is considered as the study experience of ants that accumulates on the article. Additional, how much pheromone on an article embodies selection probability of the article. Heuristic factor η is a static data. According to probability equation, pheromone on articles will play an important role in next time's iteration. With more iteration times, the article with more pheromone will have more probability to be selected during this time's ant search. And it exactly proves the positive feedback mechanism of ant colony system.

Meanwhile, we need to emphasize that heuristic factor η overcomes the drawback of slow convergence of ant colony algorithm at initial stage of evolution. This algorithm saves lots of search time which represents in iteration times. It's an advantage. But the disadvantage is the adjustice of current solution set. So when use ant colony system to solve 0/1 knapsack problem, we still need to spend great time and energy on research and experiment on heuristic factor η. And more careful work also needs to perform. In addition, the influence of the number of ants (m), the parameters α, β in probability (8) on iteration times as well as the discussion of the condition of terminate algorithm and so on, we will discussion on other papers.

REFERENCES

[1] Garey M. R., Johnson D. S., "Computers and intractability: a guide to the theory of NP-Completeness," W. H. Freeman and Co, San Francisco, 1979, pp. 261-262.

[2] Shamir A., "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem," IEEE Trans. Inform. Theory, vol. 30, no. 5, pp. 699-704, 1984.

[3] Chor B., Rivest R. L., "A knapsack type public key cryptosystem based on arithmetic in finite fields," IEEE Trans. Inform. Theory, vol. 34, no. 5, pp. 901-909, 1988.

[4] Laih C. S., Lee J. Y., Harn L., et al., "Linearly shift knapsack public-key cryptosystem," IEEE J. Selected Areas Commun, vol. 7, no. 4, pp. 534-539, 1989.

[5] V. Maniezzo, A. Colorni, "The ant system applied to the quadratic assignment problem," IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 5, pp. 769-778, 1999.

[6] R.S. Parpinelli, H.S. Lopes, A.A. Freitas, "Data mining with an ant colony optimization algorithm," IEEE Transactions on Evolutionary Computation, vol. 6, no.4, pp. 321-332, 2002.

[7] J.A. Bland, "Space-planning by ant colony optimization," International Journal of Computer Applications in Technology, vol. 12, no. 6, pp. 320-328, 1999.

[8] M. Dorigo, G.D. Caro, L.M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, no. 2, pp. 137-172, 1999.

[9] Zhang J.D., Gao Q.S., Xu X.H., "A self-adaptive ant colony algorithm," Control Theory And Applications, vol. 17, no. 1, pp. 1-4, 2000.

[10]M. Dorigo, V. Maniezzo, A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics--Part B: Cybernetics, vol. 26, no.1, pp. 29-41, 1996.

[11]J.M. Dorigo, L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53-66, 1997.