

A New Algorithm for TSP Based on Swarm Intelligence*

Xiaoxian He^{1,2}, Yunlong Zhu¹, Hechun Hu¹ and Ben Niu^{1,2}

¹Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang

²Graduate school of the Chinese Academy of Sciences, Beijing

{hexiaoxian, ylzhu}@sia.cn

Abstract - Inspired by the behavior of people, a new algorithm for the combinatorial optimization is proposed. This is a heuristic approach based on swarm intelligence, which is firstly introduced as the theoretical background in this paper. It is also a parallel algorithm, in which individuals of the swarm search the state space independently and simultaneously. When one encounters another in the process, they would communicate with each other, and utilize the more valuable experiences to improve their own fitness. A positive feedback mechanism is designed to avoid vibrations. Ten benchmarks of the TSPLIB are tested in the experiments. The results indicate that the algorithm can quickly converge to the optimal solution with quite low cost. Some conclusions about the algorithm are summarized finally.

Index Terms - Swarm intelligence. Combinatorial optimization. TSP. Route-exchange algorithm. Positive feedback.

I. INTRODUCTION

Traveling salesman problem (TSP) is probably the most well-known NP-hard combinatorial optimization problem [1][2]. The task of the traveling salesman problem is to find the shortest route for a traveling salesman to visit all the cities once and only once, and return to the starting city, which is also known as a Hamiltonian cycle. The problem can be described as follows.

Given a set of cities $C = \{c_1, c_2, \dots, c_n\}$, for each pair (c_i, c_j) , $i \neq j$, let $d(c_i, c_j)$ be the distance between city c_i and c_j . Solving the TSP entails finding a permutation π^* of the cities $(c_{\pi^*(1)}, \dots, c_{\pi^*(n)})$, such that

$$\sum_{i=1}^n d(c_{\pi^*(i)}, c_{\pi^*(i+1)}) \leq \sum_{i=1}^n d(c_{\pi(i)}, c_{\pi(i+1)}) \quad (1)$$
$$\forall \pi \neq \pi^*, (n+1) \equiv 1$$

In the symmetric TSP $d(c_i, c_j) = d(c_j, c_i), \forall i, j$, while in the asymmetric TSP this condition is not satisfied. In this work we consider the symmetric TSP.

Since the TSP is shown to be NP-complete by Karp [3], which means that we have little hope of computing exact solutions in polynomial time, researchers have proposed many heuristics for searching the optimal solution, such as simulated annealing(SA) [4], genetic algorithms(GA) [5][6] and ant colony optimizer(ACO) [7]. However, these methods performed quite poorly with this problem. They usually required high execution times, fell into the local-optimum, and were not easy to converge. The algorithm proposed in this paper endeavors to break through these difficulties. It is a

heuristic approach based on swarm intelligence. The rest of the paper is organized as follows: Section 2 gives a brief introduction to the swarm intelligence. Section 3 describes the route exchange algorithm. Section 4 tests it on the benchmarks of the TSP, and gives out the results. Finally, section 5 outlines some conclusions.

II. SWARM INTELLIGENCE

Even with today's ever-increasing computing power, there are still many types of problems that are very difficult to solve. Therefore, scientists have been returning to social insects that can be used for heuristics recently, such as ants, bees etc. Social insects are usually characterized by self-organization. Meanwhile, complex group behavior emerges from the interactions of individuals who exhibit simple behaviors by themselves. In social insects, there is no central control. Every individual is self-autonomous. They can only obtain local information, and interact with their geographical neighbors. They can also change the local environment or mark in the local environment to interact with the remote individuals indirectly, namely stigmergy. These features all characterize swarm intelligence. Generally speaking, swarm intelligence denotes that more complex intelligence emerges from the interactions of individuals who behave as a swarm [7][9].

Not only social insects but also animals, birds, fish, and even human beings can contact with each other to emerge complex behaviors. Human beings cooperate with each other, learn from each other, and get together to achieve the goals that cannot be achieved by scattering individuals. We have proof from stories of feral children raised by animals and only later in life introduced to the human world. In all cases, they were never able to adjust to languages, manners, or morals because in their early living they cannot communicate with the fellows.

Swarm intelligence is a new way of treating systems, and it seems to be a powerful tool. The concept can be applied mathematically, and are proven effective. For example, GA, ACO and PSO [9] have been applied successfully. This algorithm described as follows is another attempt.

III. ROUTE-EXCHANGE ALGORITHM

The route-exchange algorithm is a heuristic approach inspired by the behaviors of people, in which every individual searches the state space independently and simultaneously. As far as people are concerned, if a number of them are asked to

* This work is supported by the National Natural Science Foundation, China (No. 70431003) and the National Basic Research Program, China (2002CB312204)

fulfill the same task independently, they would learn from each other to improve their own performances when encountering. This mechanism is applied to the route-exchange algorithm.

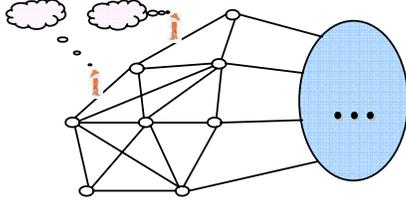


Fig.1 Two individuals encounter and exchange information with each other

Given a symmetric TSP of n nodes, m individuals are required to find a closed tour independently from different starting nodes with the same velocity v , where $m \leq n$. They could encounter the fellows somewhere in the process. Each of them has two storages: one is for the permutation of cities $(c_{\pi(1)}, \dots, c_{\pi(n)})$ visited by itself in the current repetition, another is for the best permutation $(c_{\pi^*(1)}, \dots, c_{\pi^*(q)})$ visited by itself or others ever encountered in the current repetition, where $q \leq n$. For the convenience of depiction, we denote the former with *route-storage-A*, RS_A for short, and the latter with RS_B . When two individuals encounter each other, they would compare the quality of the two routes visited in the current repetition with each other, put the better one into RS_B , and continue to tour. In the next repetition, the individual would prefer to RS_B of previous repetition and follow it first, then previous RS_A .

As for the k th individual, if he is at node i sometime, then the probability that he selects the node j as his next node is as follows:

$$p_{ij}^k = \begin{cases} \frac{(\eta_{ij})^\alpha \cdot \rho_B}{\sum_{r \in allowed_k} (\eta_{ir})^\alpha}, & l_{ij} \in RS_B_k \\ \frac{(\eta_{ij})^\alpha \cdot \rho_A}{\sum_{r \in allowed_k} (\eta_{ir})^\alpha}, & l_{ij} \in RS_A_k \cap l_{ij} \notin RS_B_k \\ \frac{(\eta_{ij})^\alpha \cdot \rho_C}{\sum_{r \in allowed_k} (\eta_{ir})^\alpha}, & else \end{cases} \quad (2)$$

where $\eta_{ij} = 1/d_{ij}$, d_{ij} is the distance between node i and j , l_{ij} denotes the edge (i, j) , $allowed_k$ denotes the set of all candidate nodes when the k th individual is at node i , ρ_A ,

ρ_B and ρ_C are selection parameters, $\rho_C < \rho_A < \rho_B$, and α is a control parameter.

Given that the k th individual reaches the node i at time T_k , and the l th individual reaches the same node at time T_l . If the inequality (3) is satisfied, then the two individuals are judged to have been encountered.

$$|T_k - T_l| \leq \frac{\bar{d}_{ij}}{\beta v}, \quad j \in Neighbor_i \quad (3)$$

where $Neighbor_i$ is the set of nodes connected directly with node i , and β is a control parameter.

In order to evaluate the quality of the route $(c_{\pi(1)}, \dots, c_{\pi(q)})$, we have G that

$$G = \left(\frac{1}{q}\right)^\delta \cdot \frac{\sum_{i=1}^{q-1} d_{i,i+1}}{q} \quad (4)$$

where δ is a control parameter. The smaller the value of G is, the better the quality of the route is.

We have designed a positive feedback mechanism to avoid vibrations. After each repetition, the individual who has the best results in the current repetition is required to save his experience, and take the same tour in the next repetition. In this way, the best individual can keep his capability of influence on others, and accelerate others to improve their fitness. The algorithm is described in Fig. 2.

IV. EXPERIMENTS AND RESULTS

The route-exchange algorithm was tested on some TSP instances defined in the TSPLIB [11]. The experiments were conducted on several Windows XP Pentium IV 1.60 GHz PCs with 256 Mbytes of memory. The results of the instances are summarized in Table 1. The first column stands for the names of the test instances, the second for exact optimal tour length for each problem given in the TSPLIB. It should be noticed that in the instances Ulysses22, Bayg29 and Att48, the optimal tour length is 75.67, 9074 and 33524 instead of 70.13, 1610 and 10628 respectively given in the TSPLIB corresponding to their tours. The third column stands for the best results we have obtained, the fourth to the sixth for the results in each run time, the seventh for the average results, and the eighth for the relative error (Err), respectively, where the relative error is calculated as

$$Err = \frac{Ave - Opt}{Opt} \times 100\% \quad (5)$$

The ninth column denotes the repetition times required, and the last denotes the number of individuals employed in the program.

```

Step 1 Initialize, put all individuals on different nodes randomly,  $NC=1$ 
Step 2 Take the nodes as the starting nodes, tour the cities by Formula (2) with velocity  $v$ 
Step 3 Repeat Step 2 for each individual; Goto Step 4
Step 4 For each node, calculate the time  $T$  that every individual reaches it, determine whether two individuals encounter
with each other according to Equality (3)
if yes, compare their routes with  $G$  in Formula (4) and put the better routes into  $RS\_B$ 
Step 5 Find the individual who has the best results in the current repetition, put his  $RS\_A$  into  $RS\_B$ 
Step 6  $NC=NC_{max}$ ?
if yes, stop the program and return the results
else,  $RS\_B$  is empty?
if yes, put the individuals on the first node of their  $RS\_A$ 
else put the individuals on the last node of  $RS\_B$ 
end
end
Goto Step 7
Step 7  $NC=NC+1$ , Goto Step 2

```

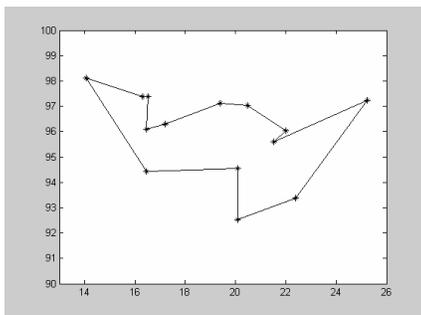
Fig.2 The route-exchange algorithm

Table 1 indicates that the proposed route-exchange algorithm can be used to solve symmetric TSP effectively. It can be seen that among 10 test problems, the maximum relative error is 10.11% (of the test problem Pr76) and the average relative error of all is easily calculated to be 4.39%. While the number of nodes increases, the repetition times required doesn't increase remarkably. In other words, the

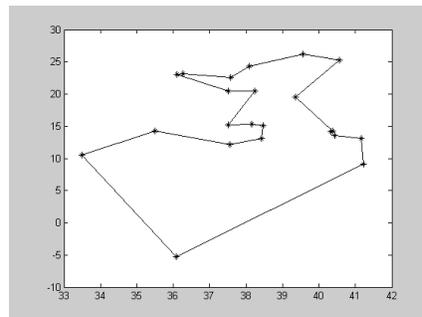
results can converge quickly to good solutions, and the complexity of the computation can be kept quite low. Fortunately, we have obtained better solutions than that given in the TSPLIB for Burma14. Fig. 3 illustrates the best tours of four instances we tested. Obviously the results of (a) and (b) are very well. As for (c) and (d), though not perfect, they look still very good.

TABLE 1
SUMMARY OF THE EXPERIMENTS RESULTS

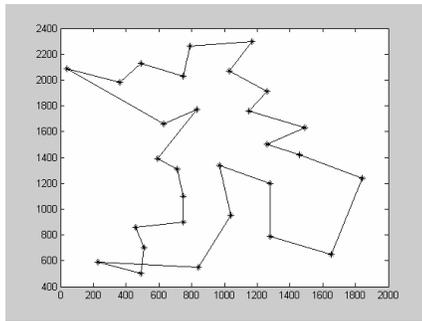
Problem	Opt	Best Result	Run1	Run2	Run3	Average	Err(%)	NC_{max}	Individuals
Burma14	33.23	31.88	33.23	31.88	33.21	32.77	-1.38	10	10
Ulysses22	75.67	75.67	76.06	76.63	75.67	76.12	0.59	30	20
Bayg29	9074	9336	9399	9336	9433	9389	3.47	30	20
Bays29	2020	2097	2097	2097	2107	2067	2.33	30	28
Swiss42	1273	1303	1325	1329	1303	1319	3.61	30	30
Att48	33524	34707	35073	34707	36458	35413	5.63	30	40
Eil51	426	451	451	454	459	455	6.81	30	40
Berlin52	7542	7870	7870	7918	7931	7900	4.75	30	40
St70	675	719	719	734	735	729	8.00	30	60
Pr76	108160	118210	118210	119450	119600	119090	10.11	30	70



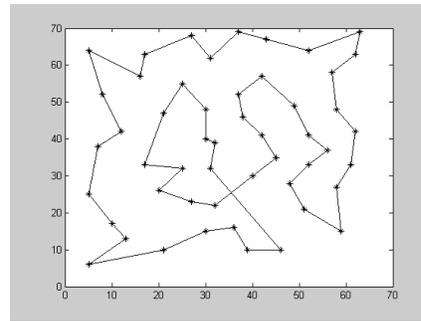
(a) Burma14, length=31.88



(b) Ulysses22, length=75.67



(c) Bays29, length=2097



(d) Eil51, length=451

Fig.3 The best tours we obtained in the experiments

V. CONCLUSIONS

In the algorithm, individuals achieve their goals by exchanging information with each other. In this way, the better routes have more chances to be taken. Because the exchange of route information is the most important idea in the algorithm, it is called *route-exchange algorithm*. As can be seen in the experiments and results, this algorithm can converge quickly to a good solution with quite low computing complexity. The stagnancy in local optimum, which is generally existent as a defect in the analogous other algorithms, can be effectively avoided. Also, the positive feedback mechanism we designed plays a very important role in this algorithm, which helps to attract other individuals to the global optimum step by step. Because this is an entirely new algorithm, there are still many works to do in the future.

REFERENCES

[1] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy, D. B. Shmoys, *The traveling salesman problem: a guided tour of combinatorial optimization*, John Wiley and Sons, New York, 1985

[2] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer Verlag, Berlin, Germany, 1994

[3] R. M. Karp, "Reducibility among combinatorial problems," *Complexity of computer computations*. Plenum press, New York, 1972: 85-103

[4] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing," *Science*, 1983, 220: 671-680

[5] J. Grefenstette, R. Gopal, B. Rosimaita, D. Van Gucht, "Genetic algorithms for the traveling salesman problem," *Proceedings of the International Conference on Genetics Algorithms and their Applications*, 1985: 160-168

[6] G. A. Croes, "A method for solving traveling salesman problems," *Operations Research*, 1958, 6:791-812

[7] Eric Bonabeau, Marco Dorigo, Guy Theraulaz, *Swarm intelligence-from natural to artificial system*. Oxford University Press, New York, 1999

[8] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995: 1942-1948

[9] J. Kennedy, R. C. Eberhart, Y. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, San Francisco, 2001

[10] Robert G. Reynolds, B. Peng, "Cultural algorithms: modeling of how cultures learn to solve problems," *Proceedings of the 16th IEEE International Conference on Tools with Artificial intelligence*. Boca Raton, Florida, 2004: 166-172

[11] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>