Application of Intelligence Algorithm on TSP

----Based on the Data the Cities in Guangdong

Liyao Yang, Kailian Lin, Shiyu Lin, Xiaofei Gao and Shiqi Ye

College of Information Science and Technology, Jinan University, Guangzhou, 510632, China
ojwlg@jnu.edu.cn

Abstract - In this paper, we applied four intelligent algorithms into solving the classic Traveling Salesman Problem (TSP). By drawing comparisons and analysis on the pros and cons of this four algorithms, TSP problem can be better solved. This paper begins with the discussion of the background and principles of four algorithms; and goes into the discussion on their performance when applied onto solving a practical problem of Guangdong province tourist rout planning project.

Index Terms - TSP, intelligence algorithm, tourist rout planning.

I . Background and Significance of Topics

Path planning problem, including Traveling Salesman Problem (TSP), Vehicle Path Planning problem (VRP), is a topic on which many domestic and foreign scholars have done a lot of researches. Previous research focus was mainly on finding the shortest path algorithm, such as Dijkstra's algorithm, Floyd-Warshall algorithm, etc. Starting in 1956 when Artificial Intelligence was formally put on the table as a newly born subject, many intelligent algorithms has been applied to TSP problem, such as Artificial Neural Networks, Ant Colony Optimization algorithm, Genetic Algorithm, Particle Swarm Optimization algorithm and Artificial Immune System, etc. Intelligent optimization algorithms provide a new train of thought. Although intelligent optimization algorithms cannot assure a global optimization result, it can get the result in a short time for solving large-scale problems.

In this paper, Dijkstra's algorithm and Floyd-Warshall algorithm are used to calculate the shortest distance between any two vertices in a given figure. On this basis, GA, SAA, ACA, PSO algorithms were applied to further solve the path selection.

II . Algorithm principles

A. Simulated Annealing Algorithm (SAA)

SAA was first put forward by Metropolis, et al. Its idea is based on the similarity between physical annealing process of the solid substance and general combinatorial optimization problem.

1) Algorithm process

There is the flow chart of SAA.

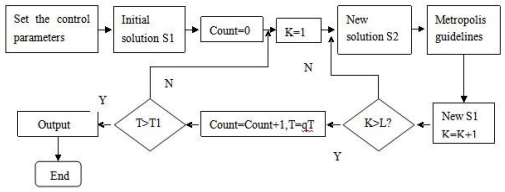

Fig . 1 SAA flow chart

2) Algorithm implementation

1. Initialization. Assigning control parameters and initial solution. Main control parameters are as cooling rate *q*, initial temperature *T0*, end temperature T_{end} and chain length *L*, where *L* is the number of iteration at each *T*. For the TSP problem having *n* locations, *n* cities is labeled by a sequence from 1 to *n* .

2. change into the new. Generate new solution of *S2* by operating transformation on current *S1*. Random number generator is used to generate a pair of numbers *(r1, r2)* which determines a change of position between this two cities, and neighborhood transformation method is further used to get the new path.

3. the Metropolis criterion. f(S) denotes the path length of the arrangement solution of S. Path difference *df = f(S2) – f(S1)*, by introducing into the Metropolis criterion,

$$P = \begin{cases} 1, df < 0 \\ EXP(-\frac{df}{T}), df \geq 0 \end{cases} \quad (1)$$

If *df* < 0, the new path is accepted with probability of *1*; Otherwise with a probability of exp *(-df/T)*.

4. cooling. Cooling with a cooling rate of *q*, and *T = qT*, if *T* is smaller than the end temperature, iteration of outputting the current state will be terminated; otherwise iterative process will continue.

B. Genetic Algorithm (GA)

1) Algorithm process

There is the flow chart of GA.

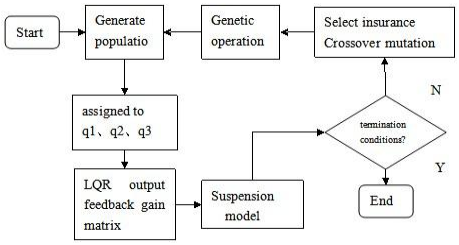

Fig .2 GA flow chart

2) Algorithm implementation

1. coding. As for an *N*-city TSP problem, chromosome is divided into n sections. Each section corresponds to a city out of a serial of cities from 1 to *N*. Any sequence of these N cities is a valid chromosome.

2.population initialization. After coding of chromosome,

an initial solution of population initialization is needed to play the part as the gene group of the ancestor of a certain species. First you need to decide the number of initial population which depends on the size of the city.

3. fitness function. Fitness value of each chromosome fitness is calculated. Suppose *k1 | k2 | k3 |... | ki |... | kn |* is a chromosome coded by a serial of integers, D_{kikj} is used to denote the distance between city *ki* and *kj*. The fitness of this chromosome can be described as:

$$fitness = \frac{1}{\sum_{i=1}^{n-1} D_{kikj} + D_{knk1}} \qquad (2)$$

4. selection operation. Parts of the chromosomes from the old group will be selected to form a new group at a certain probability. The probability of selection is related to its fitness. Larger fitness value of the chromosome, higher probability to be selected.

5. crossover operation. The parent generation of the crossover operation is determined by partial hybridization mapping. Divide the parent generation into pairs, and repeat the following process in each pair :

Generate two random numbers *r1* and *r2* , cross-swap the data in between. After cross-wrapping, delete duplicate numbers. Conflict can be eliminated by partial mapping, namely by mapping using the correspondent relationship of the intermediate part.

6. mutation. Generate two random numbers *r1* and *r2* out of 1 to *n* which determine the two positions for swapping.

7. evolutionary reversal operation. Operate crossover and mutation on each individual variation. By substituting into the fitness function, its fitness can be evaluated and for those who has a larger fitness will be chose to operate the next-generation crossover, mutation and evolutionary reversal operation. Iterative process: determine whether it satisfies the given MAXGEN (maximum genetic algebra). If not, jump to fitness calculation; Otherwise, terminate the genetic operation.

C. Ant Colony Algorithm (ACA)

Ant Colony Algorithm was first proposed by M.Dorigo et al. in the 1990s. It is a new simulated evolutionary algorithm which simulates realistically the nature foraging behavior of ant groups. M.Dorigo et al. got satisfied result from using it to solve TSP problem.

1) Algorithm process

There is the flow chart of ACA.

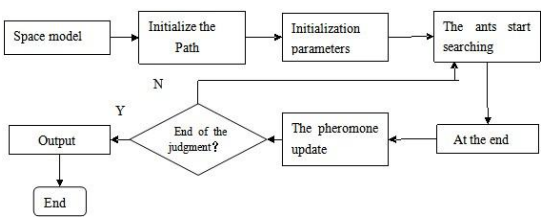

Fig .3 ACA flow chart

2) Algorithm implementation

1. Parameter initialization

2. Construct the solution space. Placed each ant on different starting point. Calculate the next city to be visited for each ant *k (k-1, 2, …, m)*

3. Update the pheromone intensity. Calculate the length of path L_k each ant *k* traversed. Record the optimal solution (the shortest path) in the current iteration process. Update the pheromone intensity of the path for each pair of two connected cities.

Ant cycle system model:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/d_{ij}, & \text{ant } k \text{ visitscity}_j \text{ from city}_i \\ 0, & \text{others} \end{cases} \qquad (3)$$

4. Termination condition. If *iter<iter_max*, then let *iter=iter+1* and clear the table recording ant traversing path.

D. Hybrid Particle Swarm Optimization (Hybrid PSO)

Hybrid PSO is improved standard particle swarm optimization algorithm. Instead of updating the position of the particle by tracking the extreme value used in conventional particle swarm algorithm, hybrid PSO introduces in the crossover and mutation from genetic algorithm, searches for optimal solution by means of crossover on extremes of individual and group, as well as particles variation of their own.

1) Algorithm process

There is the flow chart of Hybrid PSO.

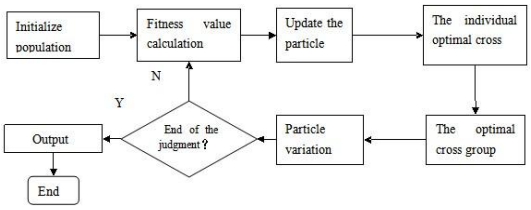

Fig . 4 Hybrid PSO flow chart

2)Algorithm implementation

1. number the particles. In Hybrid PSO , every city is a particle. Particles are numbered by integers.

2. calculate the fitness. The fitness represents the length of path traversed. The formula is:

$$Fitness(\mathrm{i}) = \sum_{i,j=1}^{n} path_{i,j} \qquad (4)$$

where *n* denotes the number of cities, $path_{i,j}$ denotes the distance between $City_i$ and $City_j$.

3.interlace operation. Different from Genetic Algorithm, in Hybrid PSO updating is achieved by crossover of individual extremes and group extremes, where integer crossover method is used. Firstly, two crossover positions are determined, then crossover operation is applied between individual value and individual extreme, or between individual extreme and group extreme. Adjust when the newly

generated individual contains duplicated position and substitute the duplicated city with those that are not included by the individual.

4. Mutation. Mutation is an operation of interchange of two intra-individuals. Same as the first step of the SAA, two neighborhood transformation method is used. In the third and fourth step, only when the fitness of the new particle is better than the old one, updating process of the new particle is triggered.

III. The Result Analysis of the intelligence algorithms

A. The Result Analysis of SAA

Using a SAA coded by MATLAB, the result of optimization rout of cities in Guangdong is shown in Fig. 5.

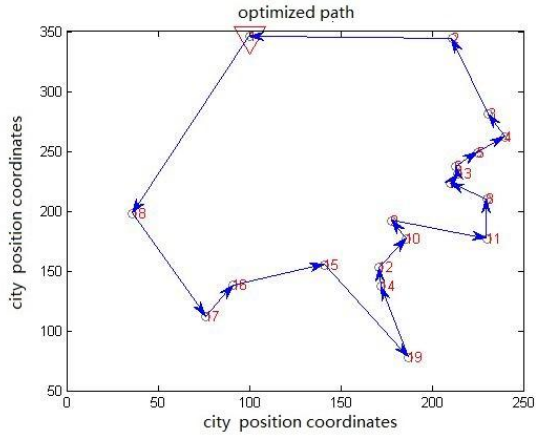

Fig. 5 Path after optimization using SAA

The optimal path is:

1—>18—>17—>16—>15—>19—>14—>12—>10—>9—>11—>8—>7—>13—>6—>5—>4—>3—>2—>1

Total length of the path is: 911.7853

Fig. 6 shows the procedure of optimization iteration.

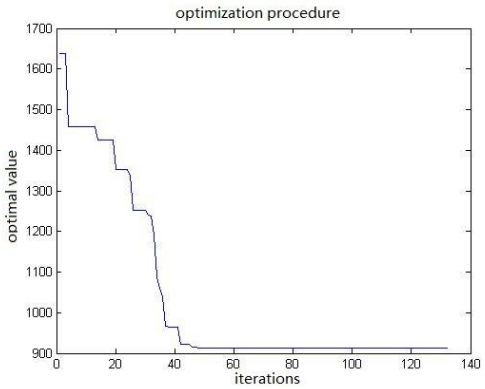

Fig. 6 The evolutionary process of SAA

As shown in Fig. 6, the path length has been greatly shortened to a stable stage as the number of iteration reaches 60. In other words, an optimal solution is generated.

B. The Result Analysis of GA

GA implemented by MATLAB the optimized path with Guangzhou as the start point is shown in Fig. 7.

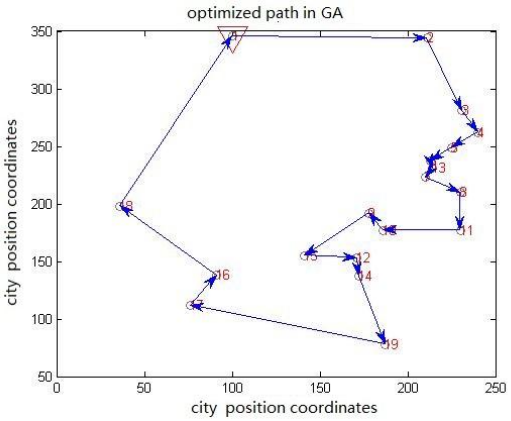

Fig. 7 Path after optimization using GA

Initial population:

1—>6—>13—>17—>8—>2—>j5—>11—>4—>10—>12—>7—>14—>18—>19—>15—>3—>16—>9—>1

Total path length: 2281.7394

The optimal path is:

1—>2—>3—>4—>5—>6—>13—>7—>8—>11—>10—>9—>15—>12—>14—>19—>17—>16—>18—>1

Total path length: 915.888

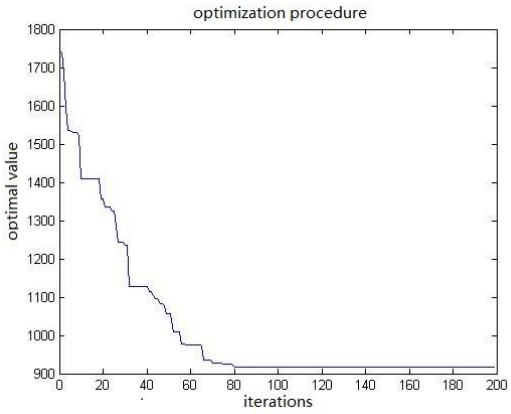

Fig. 8 The evolutionary of GA

As shown in Fig. 8, the path length sharply drops down from 2281.739 and reaches the lowest point of 915.888 at the 50th iteration .

C. The Result Analysis of ACA

ACA implemented by MATLAB, figures out the optimized path with Guangzhou as the start point is $City_{14}$ .is shown in Fig . 9

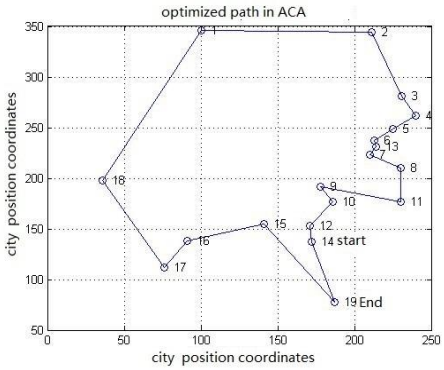

Fig . 9 Path after optimization using ACA

The shortest length of path : 911.7853
The optimal path is:

14→12→10→9→11→8→7→13→6→5→4→3→2→1→18→17→16→15→19→14

Fig. 10 shows the procedure of optimization iteration. The minimum and average length of path decrease as the number of iteration goes up. When the number of iteration reaches 112, the minimum length of path no longer changes with the iterative process, which indicates an optimal path has been found.

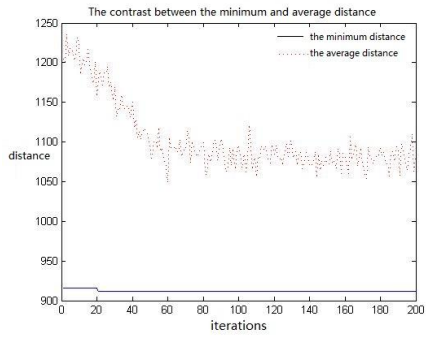

Fig.10 Shortest length of path and average length of path after each iteration

D. The Result Analysis of Hybrid PSO

Optimal path is calculated by the Hybrid PSO in MATLAB and the best particle adaptation degree are shown in Figure 11 and Figure 12 respectively.

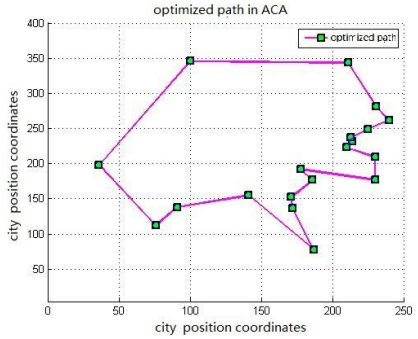

Fig . 11 Path after optimization using Hybrid PSO

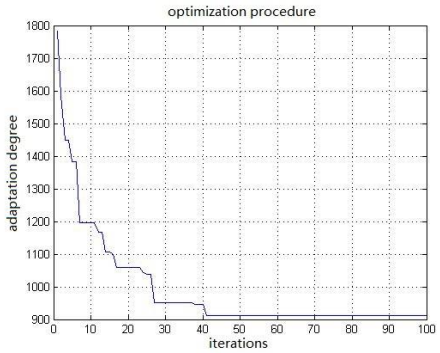

Fig. 12 Change of fitness

From Fig. 12 indicates that the fitness becomes stable after 40 iterations. The optimal solution can be gained worked out by Hybrid PSO with a relative high speed.

IV . .Algorithm Comparison

This paper simulates the position and calculates an optimal path with algorithms of GA, SAA, ACA, PSO respectively .

Simulation results proved that SAA can the optimal solution with a certain size condition of data compared with other algorithms; ACA and PSO can approach an optimal solution at the cost of increasing population scales as well as the maximum number of iteration. GA can achieve optimization goal when the population scale is small, otherwise, only approximate solutions when solving massive problems.

V . Acknowledgment

This work was supported by the Guangdong Undergraduate Training Programs for Innovation and Entrepreneurship (No.1055912014). It is instructed and supervised under Professor Shiqi YE.

References

[1] M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
[2] H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
[3] M. King and B. Zhu, "Gaming strategies," in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
[4] B. Simpson, et al, "Title of paper goes here if known," unpublished.
[5] J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.