2021 20th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

KTSP Solution Based on Workload Balance and Genetic

Zejun Hu
College of artificial intelligence
Guangdong mechanical & electrical polytechnic
Guangzhou, Guangdong 510515,China
Email:huzejun@sina.com

Guohua Yang
College of artificial intelligence
Guangdong mechanical & electrical polytechnic
Guangzhou, Guangdong 510515,China
Email: 41759307@qq.com

Abstract—**This paper focuses on the KTSP (K-person Traveling Salesman Problem) programming problem of workload balance. By adding a virtual central node, the KTSP problem is described as a TSP(Travelling Salesman Problem) problem with workload balance constraints. Then, 3-opt(3-optimization) is used for local optimization and genetic method is used to search the global optimal solution. Finally, the qualified solution is limited by workload balance constraints, and the workload balance algorithm is used to correct the results. The planning method in this paper is effective for the medium-sized workload balancing KTSP problem, and can obtain the ideal solution.**

Keywords-workload balance; TSP; KTSP; 3-opt; genetic; medium-sized

I. INTRODUCTION

KTSP(K-person Traveling Salesman Problem) is an extension of TSP(Travelling Salesman Problem) problem, which is used to solve some path planning problems in real life, and has a wide range of applications. For example, software companies should send a number of technicians to maintain the software systems purchased by customers all over the country, fast-food enterprises have delivery problems of takeout with several distributors, advertising companies' leaflet delivery, etc. These real application problems in life can be attributed to KTSP problems. TSP and KTSP are NP(Non-deterministic Polynomial) complete problems[1,2]. The optimal solution of the problem can be solved accurately by cutting plane algorithm, but the algorithm has the disadvantage of high time complexity, so it becomes very difficult to solve medium and large-scale problems[3,4,5]. Solving TSP problem with heuristic method is a kind of method widely studied at present, mainly including greedy Algorithm[6], genetic algorithm [7], simulated annealing algorithm[8],k-opt algorithm, LK algorithm, LKH algorithm[9-11],ant colony algorithm [12], particle swarm optimization algorithm[13], etc. These methods are mainly based on the idea of intelligent optimization, so they have fast search ability and can obtain high-quality solutions[14-15]. They are worthy of reference for KTSP planning with large scale. Generally, the KTSP problem basically does not consider the average workload between each salesman, so it is basically similar to the TSP planning method, and it is easy to find the optimal solution, but it is difficult to be applied to practice. The research content of this paper fully considers the workload balance of each salesman. Firstly, KTSP is transformed into TSP with workload balance problem, then hunting method and 3-opt(3-optimization) algorithm are used to search the local optimal solution, genetic mutation method is used to search the global optimal solution, and the workload balance algorithm is used to correct the results, which has high ability to solve the optimal solution, It is effective for medium-scale equilibrium KTSP problem.

II. BALANCED KTSP PROBLEM

For the general KTSP problem, given k salesmen and n cities, k salesmen start from city 0 and visit the other n-1 cities. Each city needs and only needs one salesman to visit once, and finally all return to the starting city 0. The goal is to plan the line with the shortest sum of all marketing paths. When studying practical problems, this paper requires that the workload of each salesman should be balanced, that is, the number of cities visited by each salesman should be the same, so as to eliminate the problem that the workload of individual salesmen is too small and the workload of individual salesmen is too large in the planning results, which forms a KTSP problem with workload balance. KTSP problem can be described in the form of graph theory: let V=($v_1,v_2,\ldots,v_n$)is a set of n points on the plane, G= (V, E) is a complete graph on vertex set V and edge set E. Let H be the set of k sub paths in graph G = (V, E) starting from city v_1 and returning to v_1 after cruising multiple cities, which is recorded as H = (H_1, H_2,..., H_k). Count (V) is used to represent the number of elements in vertex set V, and the constraints are (1), (2) and (3).

$$v_i \neq v_j, (v_i \in \mathrm{H}_a, v_j \in \mathrm{H}_b, i > 1, j > 1) \qquad (1)$$

$$\mathrm{V} = \cup\, \mathrm{V}_{Hi}\ ,(\mathrm{Hi} \in \mathrm{H}) \qquad (2)$$

$$\mathrm{Count}(\mathrm{V_{Hi}}) = \mathrm{Count}(\mathrm{V_{Hj}}), (\mathrm{Hi} \in H, \mathrm{Hj} \in H) \quad (3)$$

In other words, except for the starting point, there are no same vertices between the sub paths. Any vertex will be passed by the cruise path once and only once, and the number of vertices contained in any two sub paths is the same. The KTSP problem is to find the shortest k travel path H.

E_{ij} represents the length of the edge from vertex i to vertex j; If the edge from vertex i to vertex j is included in the cruise path, it is recorded as δ_{ij} = 1, otherwise δ_{ij} =0.

The objective function for finding the minimum k-cycle path length can be expressed as (4) and (5).

$$f(V) = min \sum_{i\in V, j\in V} \delta_{ij} E_{ij} \qquad (4)$$

$$\text{among,}\ \delta_{ij} = \begin{cases} 1 & Salesman\ passes\ i,\ \ j \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

III. ALGORITHM DESCRIPTION:

Since only the local optimal solution can be obtained by optimizing the sub path alone, we can form all the sub paths

2473-3636/21/$31.00 ©2021 IEEE
DOI 10.1109/DCABES52998.2021.00044

into a whole path and then optimize the whole path, so as to obtain the global optimal solution. Considering that each salesman starts from the starting city and finally returns to the starting city, and the sub paths are connected to form a closed loop that passes through the starting city many times, we can virtual k-1 vertices that completely coincide with the starting city, and set the distance between the K coincident starting cities as infinity. In this way, we can solve the KTSP path problem through global optimization. Let the set of all k starting cities be recorded as C, then the distance between cities can be expressed as (6):

$$d_{ij} = \begin{cases} \infty, & i \in C, j \in C \\ \sqrt{(x_i - x_j)^2} + \sqrt{(y_i - y_j)^2} & (other) \end{cases} \tag{6}$$

Through the nearest neighbor algorithm, an initial feasible solution of workload balance is found and saved as the current optimal solution.

A. global planning using genetic mutation algorithm

Input: optimalSolution:the current optimal solution
Output: optimalSolution:the new optimal solution
1) while true do
2) testSolution ← optimalSolution;
3) testSolution ← geneticAlgorithm(testSolution)[7]
4) testSolution ← huntingAlgorithm(testSolution)
5) testSolution ← 3-optAlgorithm(testSolution)[10]
6) if testSolution is worse than optimalSolution then
7) exit
8) end
9) testSolution ← workBalanceAlgorithm(testSolution)
10) if testSolution is better than optimalSolution then
11) optimalSolution ← testSolution
12) end
13) end

B. local optimization algorithm

In the loop sequence, if a point in the line is captured by the edge close to it, the position of the point changes, which shortens the length of the whole line. We call this algorithm hunting algorithm. As shown in Fig. 1, point E in line AB is too close to line CD, and point E is captured by line CD, thus shortening the length of the whole line.

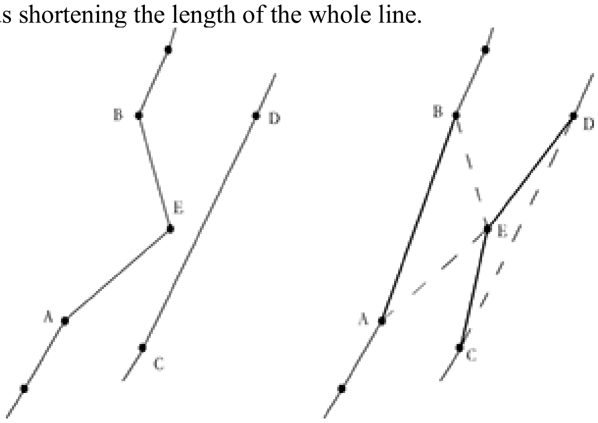

Figure 1. hunting method

In the process of local optimization, the 3-opt algorithm[10] is adopted. The algorithm uses three new edges to replace the three edges on the current loop to obtain a new loop. If the new loop solution after exchange is better, the result will be adopted, otherwise the attempted loop solution will be abandoned. As shown in Fig. 2, it shows that CD edge, EF edge and AB edge are replaced by the other three edges.

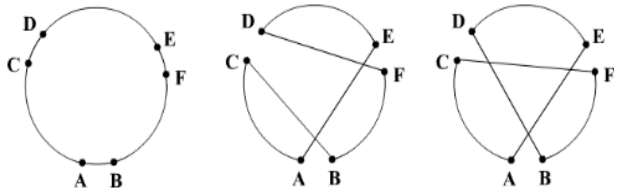

Figure 2. 3-opt algorithm

C. workload balancing algorithm

According to the results of the previous planning, the workload of each salesman may be different. In order to average the workload of each salesman, the following workload balancing algorithm is used:

Input: *cruisePath* with unbalanced workload
Output: *cruisePath* with balanced workload
1) for each *node* in *cruisePath* do
2) if *node* = *centralNode* then
3) connect the front and rear nodes of *centralNode*
4) delete *centralNode*;
5) end
6) end
7) start a new *subPath*;
8) for each node in *cruisePath* do
10) add the *node* to the *subPath*
11) *nodesNumber* ← the nodes number of the *subPath*
12) if *nodesNumber* = *workload* then
13) add a *centralNode* after the *node*
14) start a new *subPath*
15) end
16) end

After the workload balancing algorithm, if the total length of the planned path is less than the current optimal solution, a better solution is obtained, and the better solution is saved as a new optimal solution.

Taking VS2010 as the development tool, the author programmed and implemented the algorithm described in this paper. From the test results, the algorithm is effective and the effect is ideal. This algorithm can find the global optimal solution for medium-sized balanced KTSP path planning. The algorithm considers the workload balance of each salesman globally, so it can better balance various factors in path planning. At the same time, the heuristic algorithm 3-opt algorithm with the most outstanding performance is used to obtain the local optimal solution, search the global optimal solution through genetic mutation, and distribute the same workload through workload balance.

IV. ALGORITHM OPERATION EFFECT

When Huang Xiyue studied the non load unbalanced KTSP problem [1], he adopted the standard data set tsplbi95. (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/). He studied the example eil51 of TSPLIB95, but the number of urban nodes is relatively small, only 51. The workload balanced KTSP studied in this paper is more difficult than the KTSP problem, but in order to further study the operation effect of the algorithm and deal with more cities, this paper selects another larger example eil101 in the

TSPLIB95 dataset Test and research. For the solution of KTSP problem, in order to facilitate comparison with TSP algorithm, we set the simplified condition that the demand of each vertex is 1 and the workload of each salesman is the same. Eil101 in the data set is a complete graph of 101 cities. This paper sets the starting point at the central 101 node and tests the lines when the number of salesman is k = 2, 4 and 5 Road planning. Compared with reference [1] and reference [2], the balanced workload KTSP algorithm studied in this paper has the advantages of larger data processing scale and better effect.

For eil101 data set, the planning algorithm is run separately for 10 times, 30000 iterations are executed each time, and the planning results are recorded. The worst solution, optimal solution and average value are shown in Table I. It can be seen that in a certain range, with the increase of the number k of salesmen, the value of the optimal solution also increases.

TABLE I. RESULTS OF EIL101 OF 101 NODES RUNNING 10 TIMES IN EACH CASE WHEN K TAKES DIFFERENT VALUES

K	Workload of sub Tour	Optimal solution	Worst solution	average value
2	50	656.8	672.5	660.4
4	25	714.7	796.7	751.8
5	20	735.6	847.1	790.2

During the planning process, record the change process of the optimal solution, as shown in Fig.3. The evolution curve shows that the optimal value decreases rapidly at the initial stage of optimization. With the increase of optimization times, the decline of the optimal value slows down and finally tends to the level. It can also be noted that the descent curve is the steepest when k = 2 and the most gentle when k = 5. The results show that in a certain range, with the increase of the number k of salesmen, the planning becomes more and more difficult.

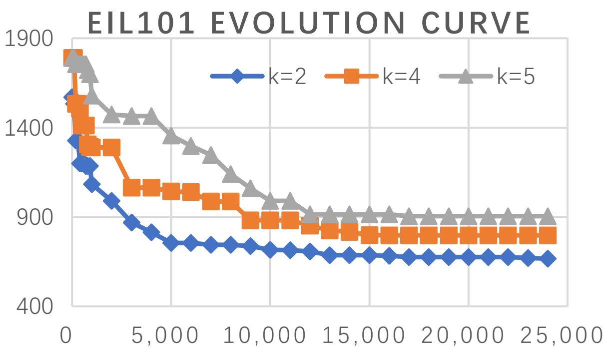

Figure 3. Evolution curve of eil101 at k = 2, 4 and 5

Using this algorithm to solve the eil101 problem, when k = 2, the optimal route is shown in Fig. 4, and the length value of the optimal solution is 656.9. When k = 4, the optimal route is shown in Fig. 5, and the optimal solution is 714.7. When k = 5, the solution result is shown in Fig. 6, and the optimal solution is 735.6. The details of the specific city access sequence when k takes different values are shown in Table II.

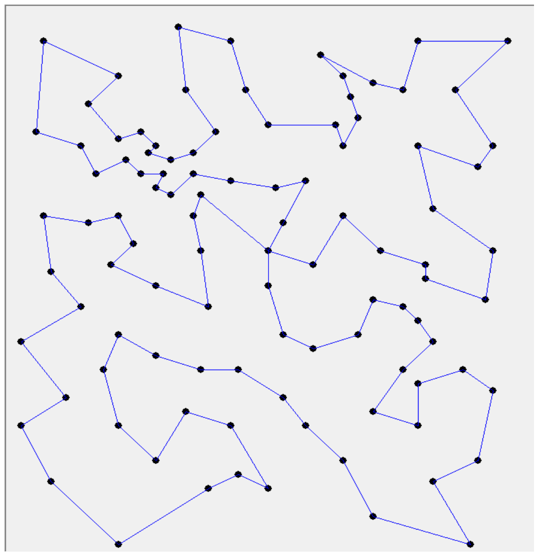

Figure 4. TSPLIB95 example eil101, when the number of operators k = 2, the optimal solution path

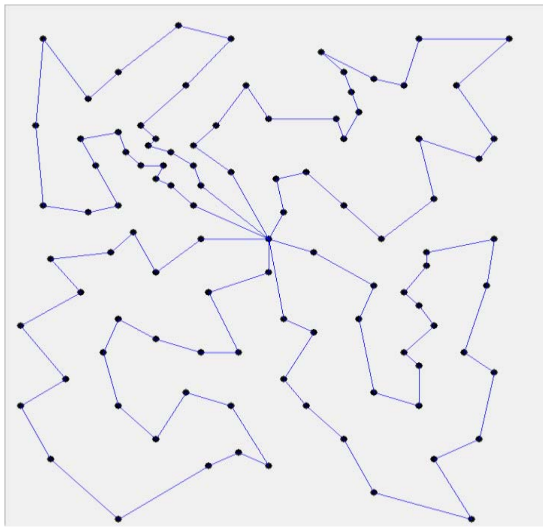

Figure 5. TSPLIB95 example eil101, optimal solution path when the number of operators k = 4

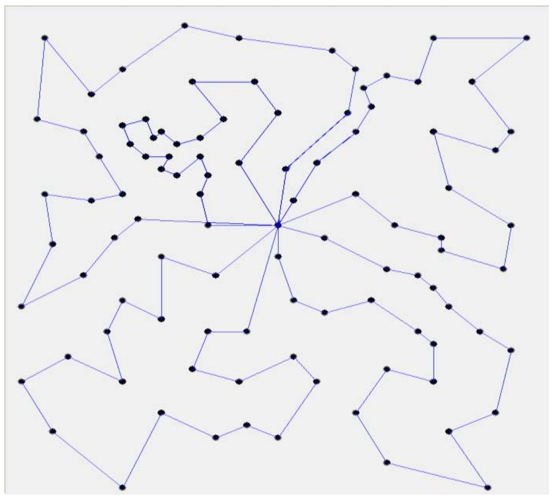

Figure 6. TSPLIB95 example eil101, when the number of operators k = 5, the optimal solution path

With the increase of the number of salespeople and K value, the demand for computing power increases, and the length of the minimum cruise path solution tends to increase. In particular, when k = 1, it is an ordinary TSP problem.

TABLE II. COMPARISON OF OPTIMAL PATHS OF EXAMPLE EIL101

K	optimal value	Access path order
2	656.8	101,94,6,89,52,18,83,60,5,84,17,45,8,46,47,36,49,64,63,90,32,10,62,11,19,48,82,7,88,31,70,30,20,66,65,71,35,34,78,81,9,51,33,79,3,77,76,50,1,69,27, 101,53,40,58,13,95,96,99,59,93,85,61,16,86,38,14,44,91,100,37,98,92,97,87,42,43,15,57,2,73,21,72,74,22,41,75,56,23,67,39,25,55,4,54,24,29,68,80,12,26,28,101
4	714.7	101,69 ,1 ,70 ,30 ,20 ,66 ,65 ,71 ,35 ,34 ,78 ,29 ,24 ,80 ,68 ,77 ,3 ,79 ,33 ,81 ,9 ,51 ,50 ,76 ,28 , 101 ,53 ,58 ,40 ,26 ,12 ,54 ,4 ,55 ,25 ,39 ,67 ,23 ,56 ,75 ,41 ,22 ,74 ,72 ,21 ,73 ,2 ,57 ,87 ,97 ,13 , 101 ,94 ,95 ,92 ,98 ,37 ,100 ,42 ,15 ,43 ,14 ,44 ,38 ,86 ,17 ,84 ,5 ,61 ,16 ,91 ,85 ,93 ,59 ,99 ,96 ,6 , 101 ,89 ,18 ,60 ,83 ,45 ,8 ,46 ,47 ,36 ,49 ,64 ,63 ,90 ,32 ,10 ,62 ,11 ,19 ,48 ,82 ,7 ,88 ,31 ,52 ,27 ,101
5	735.6	101,52 ,18 ,7 ,82 ,48 ,19 ,47 ,36 ,49 ,64 ,11 ,63 ,90 ,32 ,30 ,70 ,10 ,62 ,88 ,31 , 101 ,28 ,76 ,77 ,3 ,79 ,78 ,34 ,35 ,71 ,65 ,66 ,20 ,51 ,9 ,81 ,33 ,50 ,1 ,69 ,27 , 101 ,58 ,73 ,22 ,41 ,15 ,43 ,14 ,44 ,38 ,86 ,16 ,61,5 ,84 ,17 ,45 ,46 ,8 ,83 ,60 , 101 ,89 ,6 ,94 ,95 ,96 ,99 ,59 ,93 ,85 ,91 ,100 ,98 ,37 , 92 ,97 ,87 ,42 ,57 ,2 ,13 , 101 ,53 ,40 ,21 ,72 ,74 ,75 ,56 ,23 ,67 ,39 ,25 ,55 ,4 ,54 ,24 ,29 ,68 ,80 ,12 ,26 ,101

V. ANALYSIS OF EXPERIMENTAL RESULTS

A large number of experiments show that the optimal solution of the model studied in this paper has some characteristics, which is basically distributed in a flower shape around the distribution center. If the nodes are sparse, the patrol paths of each salesman will not intersect. If the nodes are dense, the petals may intersect near the distribution center. Each petal may contain another petal under certain circumstances. In the same petal, there is no path crossing. The study of such optimal solution characteristics has guiding significance for the further development of high-performance algorithms.

After considering the workload balance, the difficulty of solving KTSP is greatly increased. The convergence speed of the method given in this paper is still slow for the large-scale balanced KTSP problem. Through the analysis, we can know that the problem is that when planning to obtain the better value, it does not necessarily meet the conditions of workload balance. This two-step planning method has limitations. The next research can combine the solution of the better value with workload balance to improve the efficiency of the algorithm.

VI. CONCLUSION

In this paper, genetic algorithm, 3-opt algorithm, hunting algorithm and balanced workload algorithm are used to solve the workload balanced KTSP problem. Firstly, KTSP is transformed into TSP problem through virtual central node, and the local optimal solution is obtained by using 3-opt algorithm and hunting algorithm. Then, the result is corrected by workload balancing algorithm, and finally the optimal solution is searched globally by genetic algorithm. The practical results show that the algorithm is effective in solving the medium and large-scale equilibrium KTSP problem.

REFERENCES

[1] HUANG Xi-yue 1 , HU Xiao-bing, Application of Ant Colony Algorithm in K-person Traveling Salesman Problem, Journal of computer simulation, 2004(12),pp.162-164

[2] WENG Miao-feng LI Wen-yu, Applying Chaotic Neural Network Based on Annealing Strategy to KTSP Problem, Journal of East China Shipbuilding Institute(Natural Science Edition) 2003(08),pp.42-46

[3] Manfred Padberg,Giovanni Rinaldi,A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems,SIAM REV, Mar 1991

[4] Kahng, A.B.. (1989). Traveling salesman heuristics and embedding dimension in the Hopfield model. 513 - 520 vol.1. 10.1109/IJCNN.1989.118627.

[5] Naddef D. (2007) Polyhedral Theory and Branch-and-Cut Algorithms for the Symmetric TSP. In: Gutin G., Punnen A.P. (eds) The Traveling Salesman Problem and Its Variations. Combinatorial Optimization, vol 12. Springer, Boston, MA.

[6] G.Gutin and A.Yeo.The Greedy Algorithm for the Symmetric TSP.Algorithmic Oper. Res., Vol.2, 2007, pp.33--36. // Greedy

[7] P. Merz and B. Freisleben, "Genetic local search for the TSP: new results," Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), 1997, pp. 159-164, doi: 10.1109/ICEC.1997.592288.

[8] Xiutang Geng, Zhihua Chen, Wei Yang, Deqian Shi, Kai Zhao,Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search,Applied Soft Computing,Volume 11, Issue 4,2011,Pages 3680-3689

[9] G. A. Croes, A Method for Solving Traveling-Salesman Problems,Operations Research, INFORMS, vol. 6(6), 1958(12),pp.791-812.

[10] K. Helsgaun,General k-opt submoves for the Lin-Kernighan TSP heuristic.Mathematical Programming Computation,1(2-3):119-163 (2009).

[11] Tinós R., Helsgaun K., Whitley D. (2018) Efficient Recombination in the Lin-Kernighan-Helsgaun Traveling Salesman Heuristic. In: Auger A., Fonseca C., Lourenço N., Machado P., Paquete L., Whitley D. (eds) Parallel Problem Solving from Nature – PPSN XV. PPSN 2018. Lecture Notes in Computer Science, vol 11101. Springer, Cham. https://doi.org/10.1007/978-3-319-99253-2_8

[12] S Li, S Cai, L Li, R Sun, G Yuan,CAAS: a novel collective action-based ant system algorithm for solving TSP problem, Soft Computing,2020

[13] Wang Qingxi;Guo Xiaobo,Particle swarm optimization algorithm based on Levy flight, Application Research of Computers, 2016,33(9):2588-2591

[14] V Gabrel, AR Mahjoub, R Taktak, E Uchoa,The Multiple Steiner TSP with order constraints: complexity and optimization algorithms, December 2020,Soft Computing 24(3)

[15] UshaMohan,SivaramakrishnanRamani,SounakaMishrac,A 4-approximation algorithm for the TSP-Path satisfying a biased triangle inequality, Discrete Applied Mathematics Volume 271,1 December 2019, Pages 108-118