Intelligent Pipelining Method and Its Application in C-TSP Problem

Dingjia He and Hui Peng*

School of Software Engineering, Chengdu University of Information Technology, No.24 Block 1, Xuefu Road, Chengdu, China. Email: 1657546939@qq.com

Abstract. At present, researches on intelligent optimization problems such as China travel salesman problem (C-TSP) have obtained research results such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), Simulated Annealing Algorithm (SA), etc., but the single algorithm also exposes practical problems such as low efficiency and accuracy in the solving process. In order to improve the present situation effectively, this paper proposed an intelligent optimization pipeline method based on the idea of industrial production line flow. This method solved the C-TSP problem by combining the genetic simulated annealing (GA-SA), ant colony optimization simulated annealing (ACO-SA) algorithm pipeline. Experiments show that the efficiency of this method is improved by 4%-10%.

1. Introduction

"Pipeline" is a production mode that reduces coupling degree and improves work efficiency. In the early 20th century, Henry Ford created the first assembly line, which reduced production costs and increased production efficiency. Algorithm research for optimization problems such as C-TSP is still at a relatively inefficient level in the current algorithm field. Researchers have been trying to find new ways to solve problems, but progress in solving optimization problems has been slow due to limitations in basic science research as well as the computational power of a computer. In this paper, an intelligent optimization algorithm assembly line based on the idea of industrial production line assembly line is proposed, which integrates various intelligent optimization algorithms of independent operation into the algorithm assembly line for collaborative operation, in order to achieve the purpose of improving the solving efficiency and accuracy.

2. Industrial Production Line

The industrial production assembly line, which allows each person to focus on his or her area of expertise and continuously solve similar problems, massively reduces the problem of interpersonal dependency and allows each person to work in a fixed area. Take Ford's production line as an example, the assembly process is divided into 84 different steps, originally responsible for the entire production process of the car workers on the line is only responsible for one part of the production process, the car from the parts on line to the final assembly is completed on this line, this production mode will be the original 12 hours / car manufacturing speed, quickly increased to 1.5 hours / car speed, efficiency improved eight times.

The essence of the "pipeline " production mode is to design the product production process, split into coherent production steps, and then in a fully automated or semi-automatic way for each process of production, the relative independence between the various steps, only need to focus on their own work done and the interface between the external interface. Based on this idea, different algorithms are...
used to solve a certain part of the exact problem respectively, and the completion results of each part are combined to obtain the optimal solution of the problem.

3. Intelligent Optimization Method
With the development of emerging disciplines such as artificial intelligence and machine learning, intelligent optimization algorithms in related fields have also been studied and applied extensively, among which the more typical algorithms for solving optimization problems are: ant colony optimization, genetic algorithm, simulated annealing algorithm, particle swarm algorithm, differential evolution algorithm, A* algorithm, immune optimization algorithm, artificial fish swarm algorithm, etc. The ACO-SA and GA-SA algorithm pipelines proposed in this paper use the following three basic algorithms: ant colony optimization, genetic and simulated annealing.

3.1. ACO
The biology of the colony algorithm is that during foraging, ants release an ant-specific pheromone along the foraging path and enable the colony in that range to detect and influence their subsequent foraging path selection behavior. As the number of ants passing on a particular path increases, the concentration of pheromones on that path increases, so the probability of the ant choosing the path is higher. The behavioral characteristics of ants foraging for food are shown in Figure 1.

![Figure 1. Schematic diagram of ant foraging behavior](image)

The mathematical model of ant colony algorithm can be summarized by two mathematical formulas.

a) The probability formula of the ant choosing the next foraging site $j$ from foraging site $i$:

$$
P_{ij}^k(t) = \begin{cases} 
\frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & L_{ij} \in \text{Path} \\
0, & \text{otherwise}
\end{cases}
$$

In the (1), $\alpha$ is the pheromone the relative important degree, beta for stimulating factor relative important degree, $\eta_{ij}$ for stimulating factor, $\eta_{ij}=1/d_{ij}$, $d_{ij}$ is the distance from foraging point $i$ to foraging point $j$, $\tau_{ij}(t)$ for the pheromone concentration on the $L_{ij}$ at time $t$.

b) The formula of the change of pheromone concentration on $L_{ij}$ with time:

$$
\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{k} \Delta \tau_{ij}^k(t)
$$

In the (2) $\rho$ is the volatilization coefficient of pheromone, $\Delta \tau_{ij}^k(t)$ is the concentration of pheromone left on $L_{ij}$ in the foraging process of the KTH ant.

The advantages of ACO algorithm are good parallelism, cooperation and robustness, and fast convergence in the later stage. The disadvantages are that the algorithm has a long initialization time, many parameters, and the quality of the solution is greatly affected by the parameters, which can
easily stuck into local optimization. The quality of the solution and the time to obtain it are also affected due to the distribution of the pheromones, the mode of volatilization and the random choice of the ant's forward direction.

3.2. GA
The biological principle of genetic algorithm is that biological genes will select, cross and mutate genes according to the Darwinian principle of "survival of the fittest" in the process of genetic evolution, so that the next generation will have the best genes.

The advantage of genetic algorithms is that the algorithm flow is very simple and detailed, does not require the optimization function to have the conditions such as continuous, leadable and single peak, and has a strong robustness, which is an efficient, parallel and global search algorithm. The disadvantages are premature convergence due to loss of population diversity, long search time, poor local search ability, etc. The implementation process is shown in Figure 2.

3.3. SA
The principle of the simulated annealing algorithm is that the internal particles of a solid become disordered as the temperature increases, the internal energy increases, and as the temperature slowly decreases the particles gradually become ordered, reaching equilibrium at each temperature, and finally reaching the base state at room temperature, where the internal energy reaches a minimum state. The algorithmic steps of the simulated annealing algorithm are shown in Figure 3.
4. The Algorithm Assembly Line
In simulated annealing algorithm, the randomly generated initial solution has a certain influence on the algorithm results. If the initial solution is relatively ideal, the obtained result will also be ideal, but if the initial solution is not ideal, the quality of the result will be affected. Based on the fact that the results of simulated annealing algorithm will be affected by the initial solution, the solution process is abstracted into the production process of industrial assembly line, and the steps of generating the initial solution randomly are completed by ant colony algorithm and genetic algorithm. Ant colony algorithm and genetic algorithm transfer the solution obtained through certain optimization calculation to simulated annealing algorithm for initial solution, and the solution of c-tsp problem is finally completed by the cooperation of ant colony algorithm, genetic algorithm and simulated annealing algorithm. The steps of the algorithm assembly line are shown in Figure 4.

![Figure 4. Schematic diagram of algorithm assembly line flow model](image)

5. Experimentation

5.1. C-TSP Problem
The c-tsp problem is a classic operational research problem and also a classic np-hard problem. It can be described as: there are n cities of 1, 2..., n, the distance from city i to city j is $d_{ij}$. A travel agent starts from a certain city, passes through each city only once, and finally returns to the city of departure. Question: how should a travel agent choose a route so that he can travel the shortest distance? The question is based on China's 2020 administrative divisions, with 34 major cities.

5.2. Experimental Data Processing and Experimental Environment
The c-tsp experiment collected the accurate geodetic coordinates of international airports in 34 major cities in China through bai du map coordinate pick-up system. For example, the accurate geodetic coordinates of Cheng du Shuang Liu international airport are 103.958189 degrees east longitude and 30.565774 degrees north latitude. The experiment was conducted in python3.7. In addition, through a total of 2,500 experiments of ant colony algorithm, genetic algorithm, simulated annealing algorithm, ACA_SA algorithm pipeline and GA_SA algorithm pipeline, the shortest distance of C-TSP problem in 34 cities is 17154km. The shortest path is shown in figure 5:

![Figure 5. C-TSP problem shortest path](image)
5.3. GA-SA Algorithm Pipeline Experiment

The experimental method of GA-SA algorithm pipeline is as follows: on the GA-SA algorithm pipeline, the simulated annealing algorithm continues to optimize the results obtained by the genetic algorithm as the initial solution to obtain the optimal solution. The experiment was carried out by changing the number of iterations of the genetic algorithm, and the genetic algorithm was iterated for 5, 10, 20, 40 generations... The solutions obtained after 640 generations are first solved by simulated annealing algorithm in turn, and the genetic algorithm is tested 50 times per iteration. Under the same experimental data and experimental environment, the probability of 17154km obtained by GA-SA algorithm pipeline, genetic algorithm and simulated annealing algorithm is shown in figure 6:

![Figure 6. Comparison of GA-SA algorithm pipeline experiment results](image)

The experimental results show that after 40 iterations of the genetic algorithm, the probability of 17154km is 56%, which is 10% higher than the simulated annealing algorithm and 54% higher than the genetic algorithm.

5.4. ACO-SA Algorithm Pipeline Experiment

The experimental method of ACO-SA algorithm pipeline is as follows: on the ACO-SA algorithm pipeline, simulated annealing algorithm continues to optimize the results of ant colony algorithm as the initial solution, and obtains the optimal solution. The experiment was carried out by changing the number of iterations of ant colony algorithm. The algorithm was iterated for 1 generation, 2 generations, 4 generations, 8 generations... The solutions obtained after 512 generations are first solved by simulated annealing algorithm, and 100 experiments are carried out under each iteration number of ant colony algorithm. Under the same experimental data and experimental environment, the probability of 17154km obtained by ACO-SA algorithm pipeline, ant colony optimization and simulated annealing algorithm is shown in figure 7:

![Figure 7. Comparison of ACO-SA algorithm pipeline experiment results](image)

The experimental results show that after 256 iterations of ant colony algorithm, the probability of 17154km is 60%, which is 4% higher than that of simulated annealing algorithm and 60% higher than that of ant colony algorithm. The average optimal distance obtained by ACO-SA algorithm is 7km.
higher than that obtained by simulated annealing algorithm and 548km higher than that obtained by ant colony algorithm.

5.5. Analysis of Experimental Results
The experimental results are as follows: the probability of 17154km obtained by two algorithm pipelines is higher than that obtained by a single algorithm. From the algorithm flow of simulated annealing algorithm, it can be seen that the initial solution of simulated annealing algorithm is a randomly generated initial solution, and the resulting initial solution will affect the quality of the optimal solution to some extent. The simulated annealing algorithm on the two algorithm pipeline takes the solution obtained by the ant colony algorithm and the genetic algorithm as the initial solution, so as to reduce the initial influence on the optimal solution and improve the probability of the optimal solution obtained by the algorithm pipeline.

6. Conclusion
The experiments show that it is feasible to introduce the idea of industrial assembly line into the research field of algorithm based on the existing algorithm, and it can improve the solution accuracy of c-tsp and other optimization problems. We also hope that more people can dig the value of relevant algorithms on the basis of existing algorithms.

7. Acknowledgements
We would like to thank all reviewers for very useful comments and suggestions. This work was in part supported by the Sichuan Science and Technology Program under Grant 2019YJ0356, and in part supported by the Scientific Research Foundation of Education Department of Sichuan Province under Grant 18ZB0117.

8. References
[1] Mooney, M., Wilmot, B., & Mweeney, S 2012 The GA and the GWAS: Using Genetic Algorithms to Search for Multilocus Associations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9, pp899-910.
[2] Jayakumar, V., & Raju, R 2011 A multi-objective genetic algorithm approach to the probabilistic manufacturing cell formation problem. *South African Journal of Industrial Engineering*, 22, pp199-212.
[3] Ozdamar, L., & Pedamallu, C. S 2010 New Simulated Annealing Algorithms for Constrained Optimization. *Asia-Pacific Journal of Operational Research*, 27, pp347-367.
[4] Jian, C., Wang, Y., Tao, M., & Zhang, M 2013 Time - Constrained Workflow Scheduling In Cloud Environment Using Simulation Annealing Algorithm. *Journal of Engineering Science and Technology Review*, 6, pp33-37.
[5] Vrajitoru, D 1998 Crossover improvement for the genetic algorithm in information retrieval. *Information Processing and Management*, 34, pp405-415.
[6] Barra, S., Dendouga, A., Kouda, S., & Bouguechal, N 2013 Contribution to the analysis and modeling of the non-ideal effects of pipelined ADCs using MATLAB. *Journal of Circuits, Systems, and Computers*, 22, pp 1250085.1-1250085.14
[7] Hao, W., & Hongfu, Z 2012 Using genetic/simulated annealing algorithm to solve disassembly sequence planning. *Journal of Systems Engineering and Electronics*, 20, pp906-912.
[8] Leno, I. J., Sankar, S. S., & Ponnambalam, S. G 2013 An elitist strategy genetic algorithm using simulated annealing algorithm as a local search for facility layout design. *The International Journal of Advanced Manufacturing Technology*, 84, pp787-799.
[9] Li, X., & Wei, X 2008 An Improved Genetic Algorithm-Simulated Annealing Hybrid Algorithm for the Optimization of Multiple Reservoirs. *Water Resources Management*, 22, pp1031-1049.
[10] Behnamian, J., Zandieh, M., & Ghomi, S. M 2009 Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. *Expert Systems With Applications*, 36, pp9637-9644.
[11] Wang, J., Gao, X., & Zhu, Y 2011 Solving algorithm for TA optimization model based on ACO-SA. *Journal of Systems Engineering and Electronics*, **22**, pp628-639.

[12] Zaman, T., Paul, S. K., & Azeem, A 2012 Sustainable operator assignment in an assembly line using genetic algorithm. *International Journal of Production Research*, **50**, pp5077-5084.