

Exploring the Vehicle Routing Problem with the Simulated Annealing Algorithm

Zijie Zeng

*Institute of SDU-ANU Joint Science College, Shandong University, Weihai, China
202200700063@mail.sdu.edu.cn*

Abstract: Vehicle routing problem has been widely concerned because of the development of transportation industry, and simulated annealing algorithm as a tool to solve it has also been deeply studied. This work introduces the background information of simulated annealing algorithm and the process of annealing in physics. In order to confirm the effectiveness of simulated annealing algorithm in solving the maximum or minimum value of a function, the author proves that the objective function simulated annealing algorithm almost surely escapes from the local optimal solution and converges to the global optimal solution by using the properties of discrete time Markov chain. Further, traveling salesman problem and two kinds of capacitated vehicle routing problem, which are special cases of vehicle routing problem, are solved by using the simulated annealing algorithm. It proves that simulated annealing algorithm, as a heuristic algorithm, has unique advantages in solving complex optimization problems. Finally, the author analyzes the strengths, disadvantages, and limitations of simulated annealing algorithm and then puts forward an improved measure combining simulated annealing algorithm and genetic algorithm to solve more complex vehicle routing problem, such as heterogeneous vehicle routing problems.

Keywords: Simulated annealing algorithm, Discrete time Markov chain, Vehicle routing problem, Travelling salesman problem

1. Introduction

With the rapid development of logistics industry in the world today, transportation planning has become an important issue of concern. Among them, vehicle routing problem is a typical problem, and people use many methods to try to solve this problem. For example, Fang proposes the vehicle routing problem (VRP-DS) considering unmanned aerial vehicles (UAVs), and an arc-based optimization model is constructed and solved by Dantzig-Wolfe decomposition design branch pricing algorithm [1]. Simulated annealing algorithm is one of the methods to solve the problem.

Nowadays, many people try to use the improved simulated annealing algorithm to solve different VRP. For example, İlhan mentions an improved simulated annealing algorithm with crossover operator to solve capacitated vehicle routing problem [2]. Similarly, a super-heuristic algorithm based on adaptive simulated annealing and reinforcement learning is proposed for electric vehicle routing problem (CEVRP) with capacity constraints [3]. Moreover, a two-stage brainstorming-simulated annealing algorithm (BSO-SA) was proposed for multi-vehicle routing problem (FSMRP-USTW) with single soft time window [4]. However, the existing research usually combines simulated annealing algorithm with other algorithms to solve optimization problems, which cannot fully show

the effectiveness of simulated annealing algorithm clearly and objectively. It means that there are few studies that only focus on simulated annealing algorithm. What is more, other researches on the algorithm principle of simulated annealing often lack examples to show the application of the algorithm. Therefore, there is a gap in the research field that connects the basic principle of simulated annealing with the basic application in vehicle routing problem.

To fills the gap, the paper studies both the theory and practical cases of discrete process Markov chain, and the research on simulated annealing algorithm is still lacking. Therefore, the research on vehicle routing problem based on simulated annealing algorithm fully combines theory with practice to help people better understand simulated annealing algorithm.

2. Simulated Annealing Algorithm

2.1. Background Information of Annealing

In physics, annealing is a method aimed at gathering metal particles and forming more stable crystals.

The annealing process is as follows. First, the metal heated to a higher temperature, and the particles of the metal perform Brownian motion within the range $S(x, r(T))$ which is positively related to the temperature ($T \downarrow, r(T) \downarrow$). Then, the extremely slow cooling process begins. As the cooling process is slow enough, cooling can be considered as a discrete process with particles moving once per cooling step. In each cooling process, the movement of particles can be regarded as randomly taking a potential position within the movement range $S(x, r(T))$. If the selected point has lower energy, the particle will definitely move to that point. Otherwise, the particle will move to that point with probability $P = e^{-\Delta E/KT}$. After that, the above process is repeated many times. Finally, almost all particles will gather at the minimum energy state when the temperature is sufficiently low.

It is worth noting that the core idea of simulated annealing algorithm comes from statistical thermodynamics, which is Metropolis criterion based on Boltzmann distribution. It drives and controls the operation of the algorithm [5]. It should be particularly emphasized that if $P = 0$, it will cause the particles to move only within a small interval and eventually aggregate at a local energy minimum, rather than necessarily at the global energy minimum. When $P > 0$, particles have the opportunity to move to entirely different locations, even though the energy of the state is higher, which ultimately leading to the global energy minimum state. As shown in Table 1, some symbols will be used in the following.

Table 1: Notations and symbols used in this paper.

Symbols	Descriptions
T_n	Temperature after n times of cooling
X_n	The state that is chosen after n times of cooling
$f(x)$	The objective function that is needed to find minimum value
$P_{n,ij}^{(1)}$	The probability from i to j at step 1 after n times of cooling
$A_{n,i,j}$	The event that a particle escapes from i to j at step n
S^*	The set of states that can make $f(x)$ get the minimum value

Then, the motion law of particles in this physical phenomenon can be abstracted. Energy can be regarded as any function that requires a minimum value, and the position of particles can be regarded as the independent variable of this function. Furthermore, the temperature after n times of slow

cooling can be recorded as T_n . Obviously, the process of annealing is a discrete time Markov chain. After n times of cooling, the value of the independent variable of the function can be recorded as X_n .

At this point, an abstract process of finding the maximum value of the function is created by imitating the movement of particles in the returned goods (see Figure 1). Just like the particle final gathering is at the minimum energy, when n is sufficiently large, the independent variable of the function can be at the minimum point of the function. Obviously, this process can not only find the minimum value of the function, but also can be used to find the maximum value of the function by reversing the Metropolis criterion.

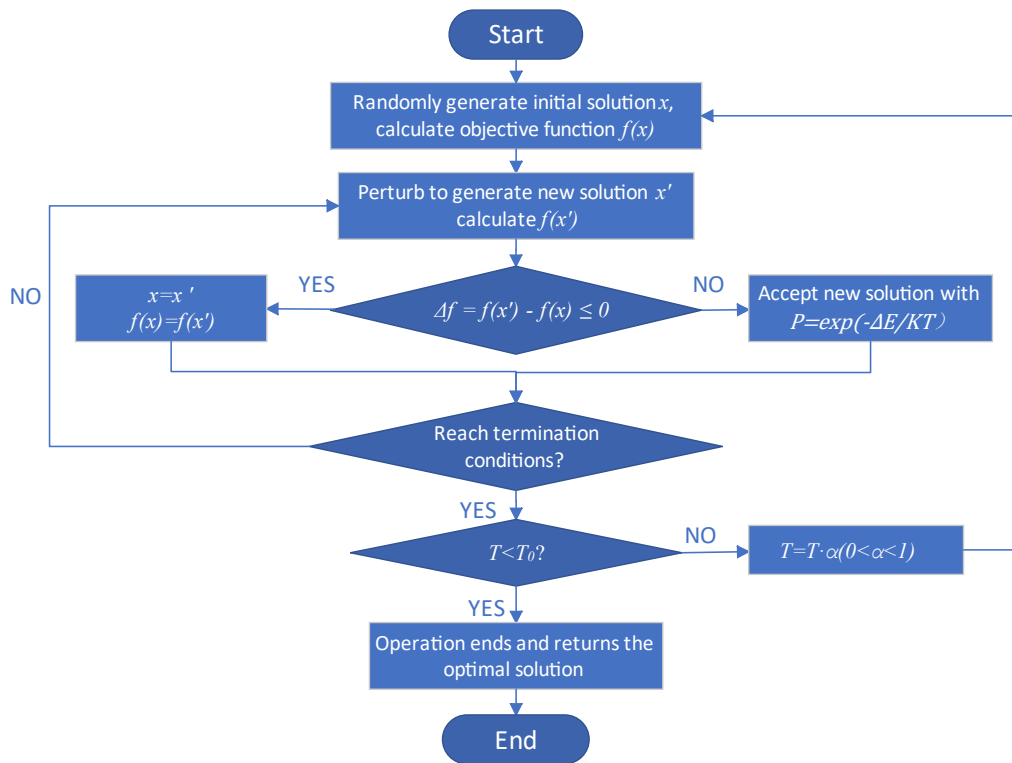


Figure 1: The process of simulated annealing algorithm.

2.2. Principle of Simulated Annealing Algorithm

The reason why people can find the maximum and minimum value of any function by using the method of annealing is as follows.

Assume that temperature $T_n = C / \ln(n + 1)$, where $n \in \mathbb{N}^+$, and C is constant. According to the background information, the probability of the Markov chain is as follows:

$$P_{n,ij}^{(1)} = \begin{cases} 1 & , f(i) \geq f(j) \\ \exp\left\{\frac{f(i) - f(j)}{T_n}\right\} & , f(i) < f(j) \end{cases} = \begin{cases} 1 & , f(i) \geq f(j) \\ (n+1)^{\frac{f(i)-f(j)}{C}} & , f(i) < f(j) \end{cases} \quad (1)$$

Assume that event $A_{n,i,j} = \{\text{Escape from } i \text{ to } j \text{ at step } n\}$. When i is a local optimum, which means that $\exists j \in S$ s. t. $f(i) < f(j)$. According to (1),

$$\sum_{n=1}^{\infty} P(A_n) = \sum_{n=1}^{\infty} (n+1)^{\frac{f(i)-f(j)}{c}} = \infty \quad (2)$$

Then, the following equation can be got because of Eq. (2):

$$P\left(\bigcap_{k=n}^N A_k^c\right) = \prod_{k=n}^N P(A_k^c) = \prod_{k=n}^N (1 - P(A_k)) = \exp\left\{\sum_{k=n}^N \ln(1 - P(A_k))\right\} \quad (3)$$

According to Taylor formula, $\ln(1 - x) = -\sum_{n=0}^{\infty} \frac{x^n}{n} \leq -x$, it is naturally found that

$$P\left(\bigcap_{k=n}^N A_k^c\right) = \exp\left\{\sum_{k=n}^N \ln(1 - P(A_k))\right\} \leq \exp\left\{\sum_{k=n}^N -P(A_k)\right\}. \quad (4)$$

This formula holds for any N , so it still holds when $N \rightarrow \infty$. Thus, it is inferred that

$$\lim_{N \rightarrow \infty} P\left(\bigcap_{k=n}^N A_k^c\right) \leq \lim_{N \rightarrow \infty} \exp\left\{\sum_{k=n}^N -P(A_k)\right\} = \exp\left\{-\sum_{k=n}^{\infty} P(A_k)\right\} = e^{-\infty} = 0 \quad (5)$$

Here, Eq. (2) is used in the penultimate step of Eq. (5). Hence,

$$P\left(\lim_{n \rightarrow \infty} \inf A_n^c\right) = P\left(\bigcup_{n=1}^{\infty} \bigcap_{k=n}^{\infty} A_k^c\right) \leq \sum_{n=1}^{\infty} P\left(\bigcap_{k=n}^{\infty} A_k^c\right) = \sum_{n=1}^{\infty} P\left(\lim_{N \rightarrow \infty} \bigcap_{k=n}^N A_k^c\right) \leq 0 \quad (6)$$

where the Eq. (5) is used in the last step of Eq. (6).

In order not to contradict the nonnegativity of probability, $P\left(\lim_{n \rightarrow \infty} \inf A_n^c\right) = 0$ can be got easily from Eq. (6). Moreover, the probability of particles escaping from local optimal solution to better solution $P\left(\lim_{n \rightarrow \infty} \sup A_n\right)$ should be considered.

$$P\left(\lim_{n \rightarrow \infty} \sup A_n\right) = 1 - P\left(\left(\lim_{n \rightarrow \infty} \sup A_n\right)^c\right) = 1 - P\left(\lim_{n \rightarrow \infty} \inf A_n^c\right) = 1 \quad (7)$$

It means that the particle will almost surely escape from the local optimum. The set of states that can make the objective function $f(x)$ get the minimum value on the domain is named S^* . As the temperature decreases continuously, $T_n \rightarrow 0$ ($n \rightarrow \infty$), according to Eq. (1),

$$\lim_{n \rightarrow \infty} P_{n,ij}^{(1)} = \begin{cases} 1 & , f(i) \geq f(j) \\ 0 & , f(i) < f(j) \end{cases} = \begin{cases} 1 & , j \in S^* \\ 0 & , j \notin S^* \end{cases} \quad (8)$$

Overall, when n is large enough, the particle will almost surely escape from the local optimum and converge to the global optimum.

3. Vehicle Routing Problem

3.1. The Brief Introduction

Vehicle routing problem (VRP) is a classical graph theory problem. Given a directed graph, it is specified that the nodes in the graph are the depot or the destination, and several existing vehicles start from the depot, complete the specified cargo distribution tasks and return to the depot. Moreover, each destination can only be visited once by one car. Then, it is required to design a driving scheme to minimize the total distance of all vehicles. VRP involves many factors such as capacity constraints, time window, picking up goods, different vehicle types, and information dynamics, which makes it becomes a complex path planning problem [6].

Obviously, VRP is an NP-Hard Problem in computer science, which means that people cannot use regular methods to solve it such as linear programming. However, simulated annealing algorithm can be used to solve it. In the following, some of the same concepts have different names in different contexts. In order to prevent confusion of many concepts and language ambiguity, this paper will strictly follow each term mentioned in Table 2. The terms in the same row of the table have the same mathematical meaning, and the terms in different rows refer to different meanings.

Table 2: Key concepts in different contexts.

Annealing	Graph Model	TSP	CVRP
Particles		Travelling salesman	Vehicles
Energy	Function	Total distance	Total distance
Minimum energy	Optimal value	Shortest distance	Shortest distance
	Nodes	Cities	Depot/Customers
	Initial node	Starting city	Depot
Points	States	Paths	Routes
Starting point	Starting state	Initial path	Initial route
Potential positions	Feasible solutions	Feasible paths	Feasible routes
Convergence point	Optimal point	Optimal path	Optimal route

3.2. Traveling Salesman Problem

3.2.1. The Introduction

Travelling salesman problem (TSP) can be seen as a simple special case of VRP. The goal of TSP is to find the shortest path such that the salesman starts from a certain city, visits all other cities, and returns to the starting city, with each city being visited only once.

In other words, taking the starting node as depot, the passing city as customer node and the traveling salesman as the only vehicle, it is not difficult to find that the TSP is essential a VRP in which only one vehicle completes a cycle to traverse all nodes in a single day. For convenience, a typical example is shown in Figure 2.

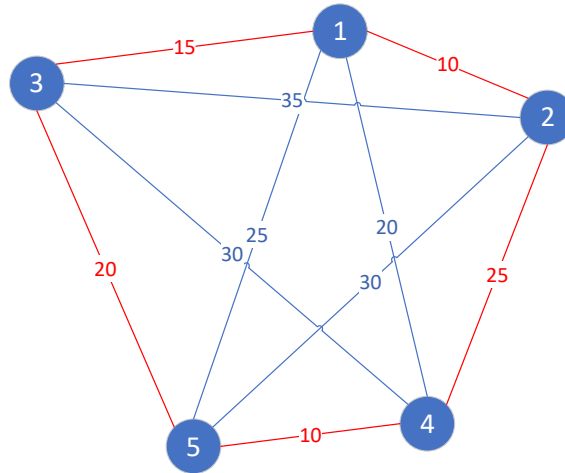


Figure 2: Example of TSP.

Without loss of generality, the state 1 can be seen as the starting state. There are $4! = 24$ possible outcomes which are shown in Table 3. Obviously, paths 1-2-4-5-3-1 and 1-3-5-4-2-1 have the shortest distance which is 80. However, if the number of the states is larger, the problem will be much harder than before. For example, there are 100 cities, and the number of possible outcomes will be much larger than before, which is $99! \approx 9.3326 \times 10^{115}$. It means that it is almost impossible for people to find all the possible outcomes to solve the problem. Then, people come up with the simulated annealing algorithm to solve the problem.

Table 3: All outcomes of the example of TSP.

Path	Distance	Path	Distance	Path	Distance
1-2-3-4-5-1	110	1-3-4-2-5-1	125	1-4-5-2-3-1	110
1-2-3-5-4-1	95	1-3-4-5-2-1	95	1-4-5-3-2-1	95
1-2-4-3-5-1	110	1-3-5-2-4-1	110	1-5-2-3-4-1	140
1-2-4-5-3-1	80	1-3-5-4-2-1	80	1-5-2-4-3-1	125
1-2-5-3-4-1	110	1-4-2-3-5-1	125	1-5-3-2-4-1	125
1-2-5-4-3-1	95	1-4-2-5-3-1	110	1-5-3-4-2-1	110
1-3-2-4-5-1	110	1-4-3-2-5-1	140	1-5-4-2-3-1	110
1-3-2-5-4-1	110	1-4-3-5-2-1	110	1-5-4-3-2-1	110

3.2.2. Solving Travelling Salesman Problem

Now, it is for people to use simulated annealing algorithm to solve the example problem of TSP. The simulated annealing algorithm consists of two phases which are initial phase and improvement phase [7]. In initial phase, each feasible path can be regarded as a state and set the total distance as the objective function, and a full permutation of five cities can be randomly selected as the initial state. In improvement phase, randomly exchange two cities as a new state, and decide whether to change the path according to the Metropolis criterion. Repeat the above method, write the Python code and run the code 20 times.

Python is chosen because it is an ideal programming language for Metaheuristic Optimization, as well as academic study and professional practice [8]. In particular, the Python code in this paper uses

proportional series $T_n = \alpha \cdot T_{n-1}$ instead of $T_n = \frac{c}{\ln(n+1)}$. It is stated here that if $T_n \rightarrow 0$ is guaranteed, the global optimum will also be found [9]. The programming results are shown in Table 4.

Table 4: The result of 20 code runs.

Number	Path	Distance	Number	Path	Distance
1	[3, 1, 2, 4, 5]	80	11	[4, 2, 1, 3, 5]	80
2	[4, 5, 3, 1, 2]	80	12	[4, 5, 3, 1, 2]	80
3	[3, 1, 2, 4, 5]	80	13	[1, 2, 4, 5, 3]	80
4	[3, 1, 2, 4, 5]	80	14	[5, 3, 1, 2, 4]	80
5	[3, 1, 2, 4, 5]	80	15	[4, 5, 3, 1, 2]	80
6	[5, 4, 2, 1, 3]	80	16	[2, 1, 3, 5, 4]	80
7	[4, 5, 3, 1, 2]	80	17	[5, 3, 1, 2, 4]	80
8	[4, 2, 1, 3, 5]	80	18	[5, 4, 2, 1, 3]	80
9	[1, 2, 4, 5, 3]	80	19	[1, 2, 4, 5, 3]	80
10	[4, 2, 1, 3, 5]	80	20	[5, 3, 1, 2, 4]	80

Each run of the code may yield different optimal paths, but the optimal distance remains 80, which matches the correct answer. This indicates that the optimal states are not unique, and the simulated annealing algorithm can stably confine the particle's movement to these optimal states. Furthermore, since the initial path is randomly generated but does not affect the shortest distance calculation. This precisely reflects the memoryless property of the Markov Chain.

3.3. Capacitated Vehicle Routing Problem

3.3.1. The Introduction

Capacitated vehicle routing problem (CVRP) is also a special case of VRP. The problem studied by CVRP is that several vehicles start from depot, deliver the goods to several customers according to the demand of customers, and then return to depot. Each vehicle can only go back and forth once. However, each vehicle has the same capacity limit, which requires that the sum of the demand of each customer passing by each vehicle should not be greater than the capacity of the vehicles. Therefore, it is necessary to arrange the routes of different vehicles reasonably to ensure that each vehicle can complete all orders without overloading. Here is an example shown in Table 5.

Table 5: Example of CVRP.

Customers	Coordinates	Demands	Customers	Coordinates	Demands
Depot (0)	(50, 50)	0	5	(12, 60)	21
1	(34, 67)	12	6	(90, 75)	7
2	(22, 45)	18	7	(40, 25)	14
3	(78, 31)	9	8	(55, 10)	16
4	(65, 88)	15	9	(85, 50)	20
Capacity: 50					

In Figure 3, the author uses (nodes, demands) to represent customers and demands.

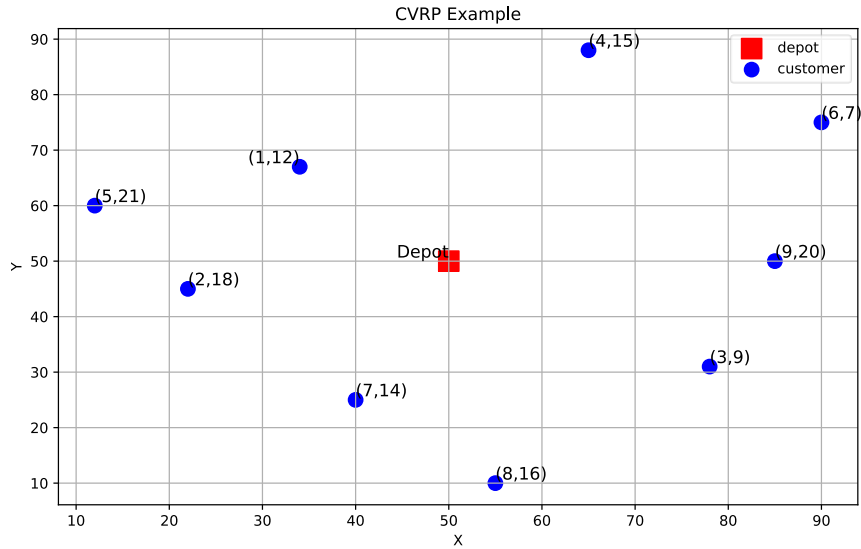


Figure 3: Example of CVRP.

The example has too many outcomes to list all the possible outcomes, which means that people cannot solve the problem by using their own hands without the help of a computer.

3.3.2. Solving Capacitated Vehicle Routing Problem

For TSP, any full permutation starting with the starting node is a feasible route. However, this is not the case for CVRP. The difficulty in implementing simulated annealing algorithm is how to generate a route that is not overloaded.

A regular method that can be called "partition method" can be used to generate feasible routes. The steps of the "partition method" are as follows:

- Generate a full permutation of customers: $0 \rightarrow 2 \rightarrow 5 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 1 \rightarrow 0$
- Sum up the demand of each customers in order: $0 \rightarrow 2 \rightarrow 5 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 1 \rightarrow 0$. Here, the Demands = $0 + 18 + 21 + 20 = 59 > 50$.
- If the demand exceeds the capacity at a customer node, a depot node will be inserted in front of the customer node: $0 \rightarrow 2 \rightarrow 5 \rightarrow 0 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 1 \rightarrow 0$
- Restart to sum up demands from this node: $0 \rightarrow 2 \rightarrow 5 \rightarrow 0, 0 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 1 \rightarrow 0$. Here, Demands = $0 + 20 + 7 + 15 + 9 = 51 > 50$.
- Repeat the above steps until the route ends: $0 \rightarrow 2 \rightarrow 5 \rightarrow 0 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 0 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 0 \rightarrow 1 \rightarrow 0$.

The route is divided into multiple sections by the depot node, and each section of the route is transported by different vehicles. This ensures that the vehicle will not be overloaded on the specified section, which means that the method can be used to generate the initial route and other feasible routes. Considering the example problem and carrying out simulated annealing algorithm, the following conclusions can be drawn (see Figure 4). The best paths of four vehicles are respectively $0 \rightarrow 2 \rightarrow 5 \rightarrow 0$, $0 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 0$, $0 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 0$, $0 \rightarrow 1 \rightarrow 0$. The shortest distance of the problem is 375.10.

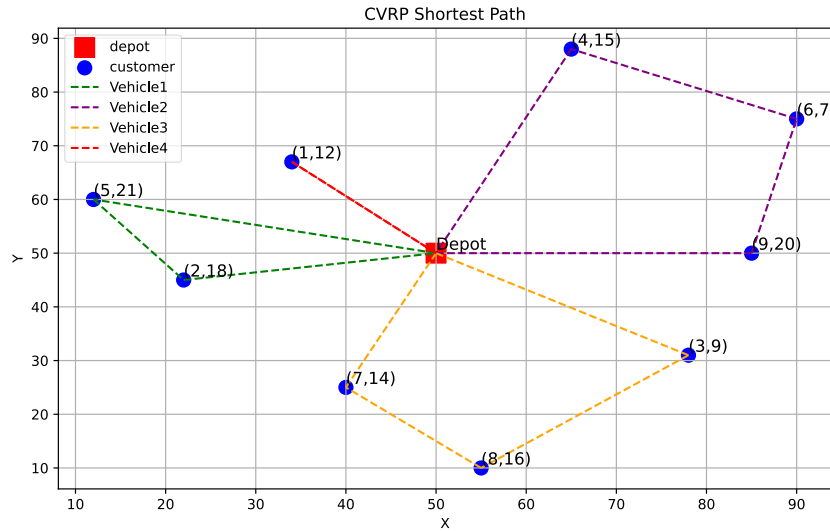


Figure 4: The result of the CVRP.

After running the code for many times, the route may be different in driving direction and vehicle order, but the shortest path is the same every time, which shows that the application of simulated annealing algorithm converges to the same route in this problem. This proves again that the process of simulated annealing is convergent as a discrete-time Markov chain.

3.3.3. Solving the Problem with Limited Number of Vehicles

If conditions are added to a CVRP problem, the number of vehicles should not exceed K . Feasible routes should be reduced, and on the basis of ordinary CVRP, the routes which need more than K vehicles should be eliminated.

In order to solve this problem, the author added a step after the "partition method", that is, to check whether the number of inserted nodes exceeds $K - 1$ through code. When there are too many inserted nodes, the route will be considered infeasible and cannot participate in the simulated annealing algorithm.

Considering the example problem again and $K \leq 3$, the author write the Python code. The result of the code is shown in Figure 5. Obviously, the best paths of three vehicles are respectively $0 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 0$, $0 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 0$, $0 \rightarrow 6 \rightarrow 9 \rightarrow 3 \rightarrow 0$, and the shortest distance of the problem is 384.30.

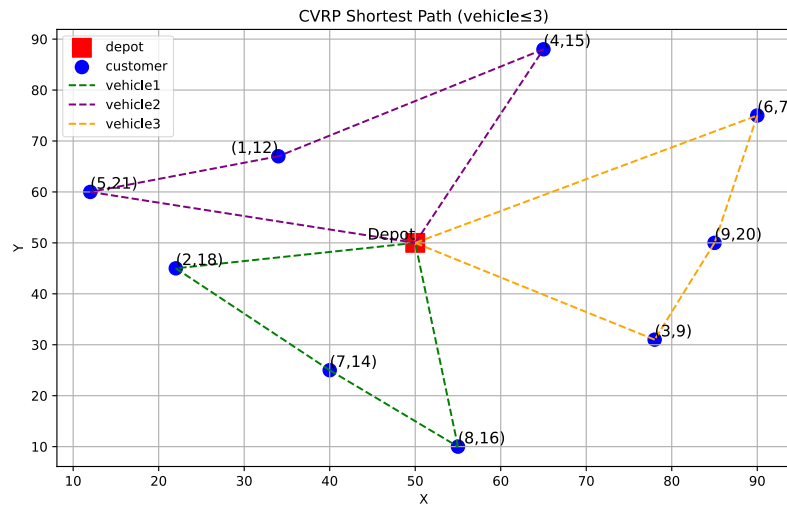


Figure 5: The result of the CVRP with limit numbers of vehicles.

Compared with Sec. 3.2.2, it can be seen that the limitation of vehicle number will greatly affect the optimal route. Obviously, most of time, the more vehicles a logistics company have, the shorter the best path is. It means that the result of the code accords with common sense of life. To sum up, people can also solve CVRP by using the simulated annealing algorithm easily even though the number of vehicles is limited.

4. Conclusion

From the code results of the example problem in the article, the simulated annealing algorithm shows good convergence in the results, which shows the effectiveness of the simulated annealing algorithm. Simulated annealing has a wide range of applications and can be fully applied in the field of logistics and transportation programming. The reasons are as follows. First, this paper selects NP-Hard problems, which cannot be solved by traditional optimization methods such as linear programming, can be solved simply by simulated annealing algorithm. Moreover, from the structure of simulated annealing algorithm, it is obvious that simulated annealing algorithm has no requirement for the form of objective function of optimization problem, which greatly improves its application scope.

However, by comparing traveling salesman problem with capacitated vehicle routing problem, people know that the set of feasible solutions will change when the constraint conditions change, and it becomes difficult to find feasible solutions because there is no general method, which makes the application and popularization of the algorithm difficult. Therefore, the author thinks that simulated annealing algorithm can be combined with genetic algorithm to solve the problem that it is difficult to express constraint equations in mathematical language and form a new and more powerful heuristic algorithm to solve more complicated optimization problems.

References

- [1] Fang, K., An, Y., Zhu, N., & Huang, D. (2025). *Vehicle routing problem considering drone stations*. *Journal of Management Sciences in China*, 28(1), 61-76.
- [2] İlhan, İ. (2021). *An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem*. *Swarm and Evolutionary Computation*, 64, 100911.
- [3] Rodríguez-Esparza, E., Masegosa, A. D., Oliva, D., & Onieva, E. (2024). *A new Hyper-heuristic based on Adaptive Simulated Annealing and Reinforcement Learning for the Capacitated Electric Vehicle Routing Problem*. *Expert Systems with Applications*, 252, 124197.

- [4] Liang, X., Yang, J., & Xiang, Z. (2024). *A two-stage BSO-SA algorithm for solving the multi-vehicle routing problem with unilateral soft time windows*. *Journal of Wuhan University of Technology (Transportation Science & Engineering)*, 48(1), 19-24.
- [5] Li, Y., Xiang, Z., & Xia, J. (2019). *Dynamic system model and convergence analysis of simulated annealing algorithm*. *Chinese Journal of Computers*, 42(6), 1161-1173.
- [6] Zhang, Z., Sun, H., Lu, J., & Li, X. (2024). *A review of logistics vehicle routing optimization problems*. *Logistics Technology*, 47(21), 89-92.
- [7] Normasari, N. M. E., Yu, V. F., Bachtiyar, C., & Sukoyo. (2019). *A simulated annealing heuristic for the capacitated green vehicle routing problem*. *Mathematical Problems in Engineering*, 2019.
- [8] Aurachman, R., Baskara, D. B., & Habibie, J. (2021). *Vehicle routing problem with simulated annealing using python programming*. *IOP Conference Series: Materials Science and Engineering*, 1010, 012010.
- [9] Jin, L. X., Tang, H. W., Li, B., Ji, M. J., & Zhu, X. Z. (2005). *A class of continuous function simulated annealing algorithm and its convergence analysis*. *Mathematical Numerica Sinica*, (1), 19-30.