Research on Intelligent Routing for Campus Library Deliveries Based on Simulated Annealing Algorithm

Peidong Xing #, Kaiyue Meng #, Na Min *

School of mathematics and statistics, Guangdong University of Technology, Guangzhou, China, 510006

* Corresponding Author Email: kelly2629891225@126.com

#These authors contributed equally.

Abstract. The intelligent delivery module for campus libraries is crucial in the construction of smart campuses. The use of delivery robots for efficient and convenient book delivery has increasingly become an important means to enhance the borrowing experience of teachers and students and to improve the operational efficiency of libraries. Based on the Simulated Annealing and Floyd algorithms, this paper first calculates the shortest distances from each building to the library. Using whether the robot passes through a building as the decision variable, a mathematical model is established with the objective of minimizing the total travel distance, while considering constraints such as robot capacity, book borrowing and returning demands, and the requirement that each building is served only once. Finally, simulation experiments are conducted to solve for the specific travel routes of the delivery robot using the SA algorithm, resulting in an optimal path diagram. Through simulation experiments, it verifies that the model has strong global search ability and ability to adapt to complex problems. At the same time, compared with other traditional delivery methods, the efficiency of the model has been significantly improved, showing the superior optimal solution approximation ability of the SA algorithm and the characteristics of jumping out of the local optimal for many times.

Keywords: VRPSPD, Campus Library Deliveries, Simulated Annealing, Floyd.

1. Introduction

As the economy and technology rapidly advance, the construction of smart campuses has become increasingly common in universities. In particular, intelligent delivery systems for campus libraries have emerged as an important means to enhance the borrowing experience for teachers and students and to improve the operational efficiency of libraries. However, with the increase in book borrowing volumes, traditional delivery methods have become inefficient and are unable to meet the growing demands. Delivery robots, as an emerging technology, offer a highly efficient and convenient solution for library book deliveries. Nevertheless, intelligent delivery systems in campus libraries face numerous challenges, the most critical of which is how to optimize delivery routes to improve efficiency and reduce operational costs.

Route optimization is closely related to planning and can be regarded as a variant of the Vehicle Routing Problem (VRP). The traditional VRP, first introduced in 1959, is a challenging combinatorial optimization problem and is known to be NP-hard [1]. The goal of VRP is to find the most efficient routes for a fleet of vehicles that start and end at a central depot. The primary objective is to serve a set of customers while minimizing total costs, subject to specific constraints. Typically, this cost is measured as a function of the total travel distance. Many variants of the classical VRP have been studied and classified in the relevant literature [2-6].

The Vehicle Routing Problem with Simultaneous Pick-up and Distribution (VRPSPD) is an extension of the classical VRP, first proposed by Min in the context of library book delivery and collection [7]. VRPSPD integrates pick-up demands into the delivery process, reducing the number of delivery vehicles compared to separate delivery and pick-up operations. This approach enhances customer satisfaction, increases vehicle load factors, reduces uncertain costs, and maximizes the utilization of logistics resources. As logistics demands grow, VRPSPD has become a more scientific and rational delivery method and an emerging research focus in academia.

Given the NP-hard nature of VRPSPD, its complexity grows exponentially with the number of customers. Therefore, scholars often employ metaheuristic algorithms to find optimal solutions. Ai et. al [8] proposed a formulation for VRPSPD and a Particle Swarm Optimization (PSO) algorithm to solve it. The primary objective of the model is to minimize routing costs, including fixed transportation costs and variable costs per unit distance. The results demonstrated the effectiveness of the proposed PSO method in solving VRPSPD. Fan et. al [9] investigated VRPSPD with time windows considering customer satisfaction. Their model aimed to minimize the total length of vehicle routes to reduce costs and maximize the sum of customer satisfaction to improve service quality. They used a nearest-neighbor method to provide an initial solution, which was improved by a tabu search algorithm, and tested the algorithm's performance. Sayyah et. al [10] proposed an Effective Ant Colony Optimization (EACO) algorithm, which includes insertion, exchange, and 2-Opt moves, to solve VRPSPD differently from the common Ant Colony Optimization (ACO).

Although significant progress has been made in the study of VRPSPD, its application in specific scenarios such as campus libraries still faces many challenges. For example, delivery tasks in campus libraries need to consider both book returns and borrowings simultaneously, and delivery robots may face various constraints during operation, such as capacity and battery limitations. Additionally, the complexity of the campus environment increases the difficulty of route planning. Therefore, optimizing delivery routes to improve efficiency while meeting practical needs is an important direction for current research.

This paper focuses on the intelligent delivery route planning problem for campus libraries. First, the Floyd algorithm is used to calculate the actual shortest distances between buildings. Then, an optimization model is established with the decision variable being whether a building is visited, and the objective function being the minimization of the total travel distance of the delivery robot. The model considers constraints such as robot capacity, book borrowing and returning demands of each building, and the requirement that each building can only be served once. The Simulated Annealing algorithm is employed to solve for the specific routes of the robot, resulting in an optimal route map.

2. Description of the Application Method

2.1. The basic functions of Floyd Algorithm

The Floyd algorithm is a classic dynamic programming approach used to compute the shortest paths between all pairs of vertices in a graph. It is particularly well-suited for dense graphs and can handle edges with negative weights. The core idea of the Floyd algorithm is to dynamically update the shortest paths between all vertex pairs by progressively introducing intermediate vertices.

The Floyd algorithm consists of the following three main steps, which are:

(1) Initialize the distance matrix.

Input the adjacency matrix $dist$, where $dist[i][j]$ represents the direct distance between vertices i and j. If there is no direct edge between i and j, set $dist[i][j]$ to infinity (typically represented by a very large number). Initialize the distance from each vertex to itself as 0, i.e., $dist[i][j]=0$.

(2) Dynamic programming update.

For each vertex k (as an intermediate vertex), update the shortest paths between all vertex pairs (i,j). The specific update formula is

$$dist[i][j]=\min\left(dist[i][j], dist[i][k]+dist[k][j]\right) \tag{1}$$

Here, $dist[i][k]+dist[k][j]$ represents the path length from vertex i to vertex j via vertex k.

(3) Output the results.

The final $dist$ matrix contains the shortest path lengths between all pairs of vertices in the graph.

2.2. The introduction to Simulated Annealing Algorithm

Simulated Annealing is inspired by the annealing process in metallurgy. It starts from a high initial temperature and gradually cools down, using probabilistic jumps to randomly search for the global optimum in the solution space. Unlike other optimization algorithms, SA probabilistically accepts worse solutions to escape local optima and converge to the global optimum. It is simple, robust, and widely applicable, making it effective for finding global optima. The flowchart of the algorithm is shown in Figure 1.

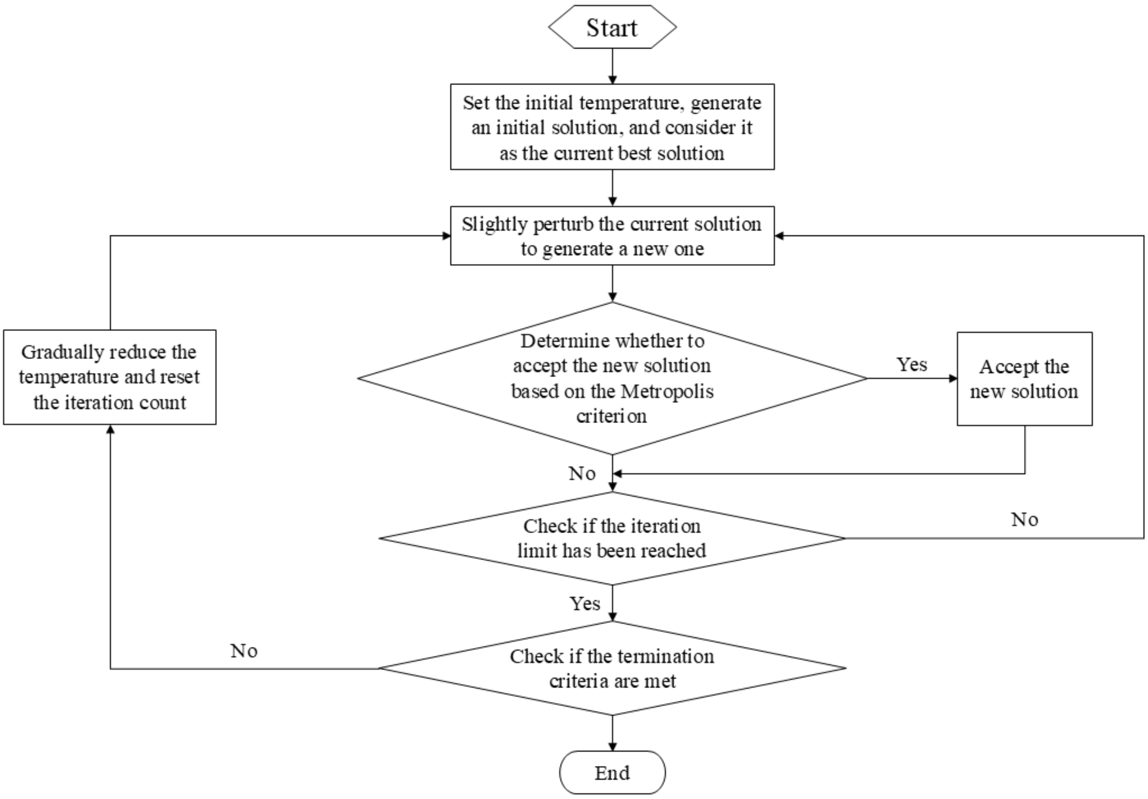

Figure 1. SA flowchart

The Metropolis criterion is the foundation of the SA. It aims to escape local optima and achieve global optimality by probabilistically accepting new states rather than following a deterministic rule. Suppose the current state is A, and the system transitions to a new state B based on a certain metric (e.g., gradient descent or energy from the previous section), resulting in a change in system energy from $f(A)$ to $f(B)$. The acceptance of the new solution depends on the following probability:

$$P_t = \begin{cases} 1, f(A) < f(B) \\ e^{-(f(A)-f(B))\times C_t}, f(A) \geq f(B) \end{cases} \tag{2}$$

where $f(A)$ denotes the objective function mentioned earlier.

SA can be divided into the following five steps:

(1) Randomly generate an initial solution x and compute the initial objective function value $f(x)$.

(2) Perturb the current solution x to generate a new solution x' using methods such as swapping, shifting, or inversion, and then compute the objective function value $f(x')$ for the new solution.

(3) Calculate the difference $\Delta f = f(x') - f(x)$. If $\Delta f < 0$, replace the current solution x and its objective function value $f(x)$ with the new solution x' and $f(x')$. Otherwise, generate a random number between 0 and 1. If this number is less than the acceptance probability P_t at the current temperature, accept the new solution x'; otherwise, retain the original solution x and $f(x)$.

(4) Compare the current iteration count n with the maximum iteration count $n_{\max}$ at the current temperature. If $n < n_{\max}$, repeat the perturbation and acceptance process. If $n \geq n_{\max}$, record the best solution x and $f(x)$ at the current temperature, store them, and update the historical best solution $x_{\max}$ and $f(x)_{\max}$. Then, decrease the temperature.

(5) Compare the current temperature T with the stopping temperature T_{end}. If $T \leq T_{end}$, terminate the process and use the current solution x and $f(x)$ as the initial solution for the next run. Otherwise, continue perturbing and accepting new solutions.

3. Results

3.1. The establishment of simulation model

The campus library delivery path problem can be described as follows: A delivery robot starts from the library and provides pick-up and delivery services to its service area, required to return to the distribution center after satisfying the book borrowing and returning needs of each building, and the robot must serve all buildings. This paper establishes an optimization model with the decision variable being whether to pass through a certain building, the objective function being the minimization of the total distance traveled by the delivery robot, and the constraints including the robot's capacity, the book borrowing and returning needs of each building, and the requirement that each building can only be served once.

Additionally, this paper makes the following assumptions: The delivery robot requires no time to collect and distribute books at each building or to load books at the library; the robot maintains a constant average speed during travel; the robot has sufficient battery life to complete its tasks without needing to recharge; all borrowed and returned books are unique, eliminating the need to consider lending books collected from one building to another; and the delivery robot serves each building only once.

Based on the problem description and assumptions, the following mathematical model is established:

$$\min S = \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} {x_{ij}}^{k} \tag{3}$$

$$\sum_{i=2}^{n} \sum_{j=2}^{n} {x_{ij}}^{k} = 1 \tag{4}$$

$$\begin{gathered} \sum_{j=2}^{n} {x_{1j}}^{k} = 1, \forall k = 2, \cdots, m \\ \sum_{i=2}^{n} x_{i1}^{k} = 1, \forall k = 2, \cdots, m \end{gathered} \tag{5}$$

$$d^k = \sum_{j=2}^{n} d_j x_{ij}^k, \forall i = 1, \cdots, n; \forall k = 1, \cdots, m$$

$$r^k = \sum_{j=2}^{n} r_j x_{ij}^k, \forall i = 1, \cdots, n; \forall k = 1, \cdots, m \tag{6}$$

$$d^k + r^k \leq Q, \forall k = 1, \cdots, m$$

$$\sum_{k=2}^{m} d^k = D$$

$$\sum_{k=1}^{m} r^k = R \tag{7}$$

$$x_{ij}{}^k \in \{0, 1\}, \forall i = 1, \cdots, n; \forall j = 1, \cdots, n; \forall k = 1, \cdots, m \tag{8}$$

The objective function (3) represents the minimum total distance traveled, (4) indicates that each building must be serviced and only once, (5) indicates that the path of each robot must start at the starting point and ends at the end point, (6) represents the robot capacity limit, (7) represents the need to meet the borrowing and returning needs, and (8) is the constraint of (0-1).

3.2. Simulation experiments

This paper selects University M as a case study to validate the developed route planning model and the SA algorithm. The library of University M serves as the distribution center for surrounding buildings, with delivery robots completing daily book pick-up and delivery services. The paper solves for the book borrowing and returning demands of each building on a specific day. Figure 2 illustrates the campus buildings and traffic routes, with Building No. 1 representing the university library. The borrowing and returning demands for each building are listed in Table 1.

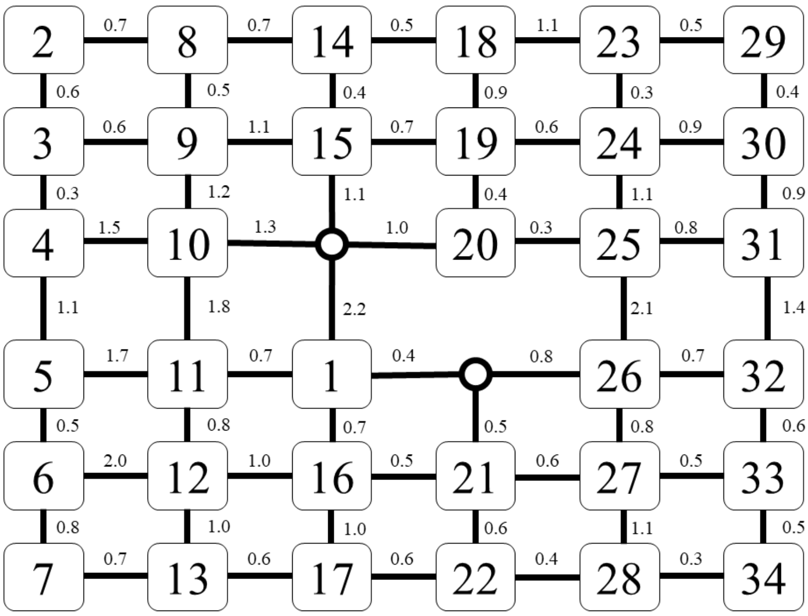

Figure 2. Campus building and path diagram

Table 1. Number of books returned and borrowed at each building

Building	Returned	Borrowed	Building	Returned	Borrowed
2	0	0	19	4	0
3	3	0	20	2	3
4	4	1	21	5	0
5	2	0	22	6	0
6	5	2	23	5	2
7	6	0	24	3	0
8	6	2	25	8	0
9	5	0	26	3	0
10	2	0	27	6	2
11	4	0	28	2	0
12	3	1	29	3	0
13	6	0	30	4	0
14	4	3	31	2	2
15	3	0	32	1	0
16	6	0	33	5	0
17	2	0	34	0	0
18	7	2			

Figure 2 shows the connections and actual distances between each building. Counting the connections between any two buildings is impractical. In reality, we only need to focus on the shortest distances between any two buildings. Using the Floyd algorithm, we can obtain the actual shortest distances between all buildings. Table 2 lists the shortest distances from the library (Building No. 1) to each building.

Table 2. The shortest distances from the library to each building

Building	Distance	Building	Distance	Building	Distance
2	4.4	13	2.3	24	4.2
3	3.8	14	3.7	25	3.3
4	3.5	15	3.3	26	1.2
5	2.4	16	0.7	27	1.2
6	2.9	17	1.7	28	1.9
7	3.0	18	4.2	29	4.6
8	4.2	19	3.6	30	4.2
9	3.7	20	3.2	31	3.3
10	2.5	21	0.9	32	1.9
11	0.7	22	1.5	33	1.7
12	1.5	23	4.5	34	2.2

After obtaining the shortest distances between buildings, we use the established mathematical model combined with the SA algorithm to solve for the shortest delivery path for the delivery robot. This paper uses MATLAB 2023a for programming and solving. Based on the proposed algorithm structure, the initial temperature is set to $T_0 = 100$, the termination temperature to $T_{end} = 0.1$, and the annealing factor to $\alpha = 0.99$. Figures 3 and 4 show the iterative optimization convergence of the SA algorithm and the optimal travel path for the vehicle, respectively. The delivery paths for the delivery robot are listed in Table 3.

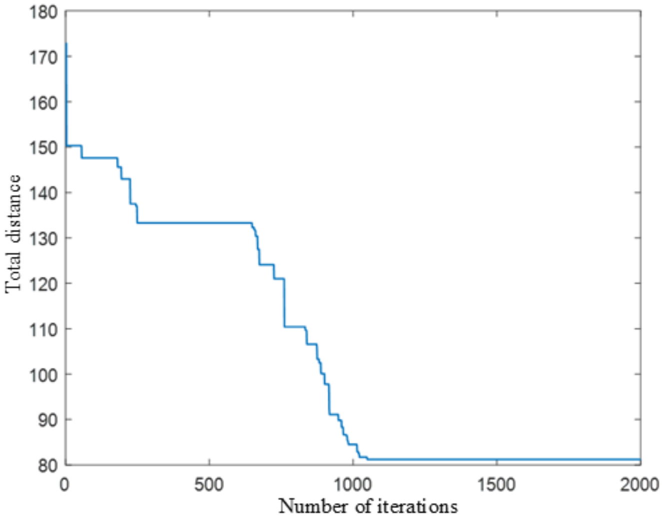

Figure 3. Iteration graph of global optimal solution

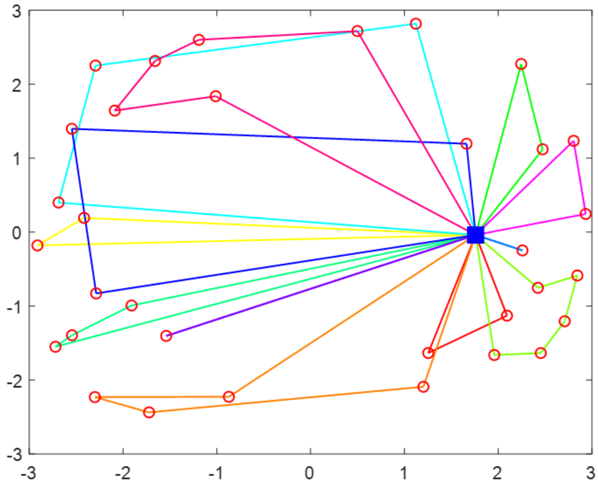

Figure 4. Optimal path planning diagram

Table 3. The shortest route planning for robot book delivery

Route	Specific route	Distance
1	1→27→26→1	3.2
2	1→32→30→29→31→1	9.2
3	1→11→18→1	8.4
4	1→16→1	1.4
5	1→7→12→1	6.2
6	1→23→24→20→1	9.0
7	1→21→22→28→34→33→1	4.4
8	1→14→2→6→1	10.5
9	1→11→8→19→1	9.6
10	1→25→1	6.6
11	1→13→17→1	4.6
12	1→5→4→3→9→10→1	8.1

3.3. Results analysis

As shown in Figure 3, the convergence curve of the SA exhibits a high slope, demonstrating its superior ability to approach the global optimum and its capacity to escape local optima multiple times. This is primarily attributed to the algorithm's global search capability, which allows it to probabilistically accept inferior solutions, thereby facilitating the escape from local optima and convergence towards the global optimum. Figure 4 illustrates that the distribution of customer points along each delivery route is relatively dispersed, indicating the algorithm's strong global search ability and adaptability to complex problems. According to Table 3, the minimum total travel distance for robot book delivery is calculated to be 81.1, which represents a significant improvement in efficiency compared to traditional delivery methods. This confirms the effectiveness and advantages of the proposed intelligent delivery route planning method for campus libraries based on the SA algorithm.

However, the current model has limitations in comprehensively considering all aspects of robot delivery. For instance, allowing multiple visits to each building could potentially reduce the total travel distance. Future work could incorporate predictive models to forecast book borrowing and returning demands in advance, enabling preemptive route planning and optimization of the number of books carried by the robot. Additionally, the model's applicability can be extended to other domains such as emergency material delivery, smart city logistics, and intelligent medical supply distribution, where efficient delivery is crucial.

4. Conclusions

This paper constructs a model for the campus library delivery route planning problem, with the optimization goal of minimizing the total travel distance. To achieve the shortest path and improve delivery efficiency, the Floyd algorithm is used to determine the shortest distances between buildings, while the Simulated Annealing algorithm is employed to solve for the shortest delivery route. Focusing on the actual needs of 33 buildings at University M, this paper applies the SA algorithm to solve the problem and identifies the shortest delivery route and the minimum total travel distance. The paper assumes that all deliveries can be completed in one attempt; however, in reality, unsuccessful deliveries may also impact overall costs. Additionally, the need for delivery robots to return for recharging during the delivery process remains an issue to be addressed. Therefore, future research could further explore the impact of delivery failure rates and insufficient battery levels of delivery robots on overall delivery costs.

References

[1] Dantzig G.B., Ramser J.H. The Truck Dispatching Problem [J]. Management Science, 1959, 6: 80-91.

[2] Eksioglu B., Vural A.V., Reisman A. The vehicle routing problem: A taxonomic review [J]. Computers and Industrial Engineering, 2009, 57(4): 1472-1483.

[3] Lin C., Choy K.L., Ho G.T.S., Chung S.H., Lam, H.Y. Survey of Green Vehicle Routing Problem: Past and future trends [J]. Expert Systems with Applications, 2014, 41(4): 1118-1138.

[4] Braekers K., Ramaekers K., Nieuwenhuyse I.V. The vehicle routing problem: State of the art classification and review [J]. Computers and Industrial Engineering, 2016, 99: 300-313.

[5] Kucukoglu I., Dewil R., Cattrysse D. The electric vehicle routing problem and its variations: A literature review [J]. Computers and Industrial Engineering, 2021, 161: 107650.

[6] Konstantakopoulos G., Gayialis S., Kechagias E. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification [J]. Operational Research, 2022, 22: 2033-2062.

[7] Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points [J]. Transportation Research Part A: General, 1989, 23(5): 377-386.

[8] Ai T.J., Kachitvichyanukul V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery [J]. Computers and Operations Research, 2009, 36(5): 1693-1702.

[9] Fan J. The Vehicle Routing Problem with Simultaneous Pickup and Delivery Based on Customer Satisfaction [J]. Procedia Engineering, 2011, 15: 5284-5289.

[10] Sayyah M., Larki H., Yousefikhoshbakht M. Solving the vehicle routing problem with simultaneous pickup and delivery by an effective ant colony optimization [J]. Industrial Engineering and Management, 2016, 3: 15-38.