Research on Multi-Vehicle and Multi-Task Route Planning for Autonomous Delivery Robots in Parks

Lu Ke[1], Byung-Won Min[2*]
[1]Ph.D. Student, Division of Information and Communication Convergence Engineering, Mokwon University
[2]Professor, Division of Information and Communication Convergence Engineering, Mokwon University

공원 내 자율 배달 로봇을 위한 다중 차량 및 다중 작업 경로 계획 연구

노크[1], 민병원[2*]
[1]목원대학교 정보통신융합공학부 박사과정, [2]목원대학교 정보통신융합공학부 교수

Abstract In the context of multi-vehicle and multi-task logistics distribution within a park, traditional algorithms are often hindered by high computational complexity and slow convergence rates. Particle Swarm Optimization (PSO) has gained popularity in path planning for autonomous delivery vehicles due to its straightforward algorithmic principles, broad applicability, and comprehensive search capabilities. However, the conventional PSO is susceptible to premature convergence, leading to local optima. To address this, this study incorporates the Tent map into the PSO to enhance the algorithm's global search ability and prevent premature convergence. Benchmark function tests demonstrate that the improved Particle Swarm Optimization algorithm (TPSO), as proposed in this study, exhibits faster convergence and greater accuracy.In the instance verification section, X Park was selected as an example to construct a multi-vehicle and multi-task model for the logistics distribution within the park. The TPSO algorithm proposed in this paper was used to solve the model, and finally, the superiority of the TPSO algorithm was verified through comparative simulation.

Key Words : VRP;PSO ;Tent Map Chaos; Time Window Constraint; autonomous delivery vehicles

요 약 공원 내 다중 차량 및 다중 작업 물류 분배의 맥락에서 전통적인 알고리즘은 종종 높은 계산 복잡성과 느린 수렴 속도에 의해 제한된다. 입자 군집 최적화(PSO)는 그 간단한 알고리즘 원칙, 광범위한 적용 가능성, 그리고 포괄적인 검색 능력으로 인해 자율 배달 차량의 경로 계획에서 인기를 얻고 있다. 그러나 기존의 PSO는 조기 수렴에 취약하여 지역 최적해에 도달할 수 있다. 이를 해결하기 위해 본 연구는 PSO에 텐트 맵 혼돈(Tent map)을 도입하여 알고리즘의 전역 검색 능력을 향상시키고 조기 수렴을 방지하고자 하였다. 벤치마크 함수 테스트 결과, 이 글에서 제안한 개선된 입자 군집 최적화 알고리즘(TPSO)은 더 빠른 수렴과 높은 정확성을 보였다. 사례 검증 섹션에서는 X 공원을 예로 들어 공원 내 물류 분배를 위한 다중 차량 및 다중 작업 모델을 구축하였으며, 이 글에서 제안한 TPSO 알고리즘을 사용하여 모델을 해결하였고, 최종적으로 TPSO 알고리즘의 우수성을 비교 시뮬레이션을 통해 검증하였다.

주제어 : VRP, PSO, 텐트 맵 혼돈, 시간 창 제약, 자율 배달 차량

*교신저자 : 민병원(minfam@mokwon.ac.kr)
접수일 2024년 08월 19일 수정일 2024년 09월 25일 심사완료일 2024년 10월 09일

1. Introduction

With the ongoing development of society and the progress of technology, autonomous delivery vehicles, as a novel type of intelligent delivery tool, are increasingly capturing public attention. The parks, being specific areas, encounter numerous challenges and issues with their internal logistics distribution systems. The question of how to utilize multiple autonomous delivery vehicles for route planning and to collaboratively fulfill the logistics distribution tasks within the park, thereby enhancing delivery efficiency, has become one of the key focal points of current research.

The study of route planning for autonomous delivery vehicles is a complex and challenging topic. Unlike general route planning algorithms, the route planning algorithm for unmanned delivery vehicles in a park must consider not only the feasibility of obtaining a path from the starting point to the destination but also a comprehensive set of factors, including cost, safety, customer requirements for time nodes, and the payload capacity of the delivery vehicles [1-7] Reference [8] introduces a route planning method utilizing the Particle Swarm Optimization (PSO) algorithm, capable of generating an optimal logistics distribution plan by integrating considerations of transportation routes, plans, traffic congestion, transfer times, and waiting times. Addressing the "premature convergence" issue identified in Reference [9], an enhanced PSO algorithm has been developed for UAV route planning. This enhancement optimizes adaptive parameters and introduces global extreme value mutation and acceleration terms to balance global and local search efficiencies, thereby preventing premature convergence. Reference [10] develops a three-tier logistics model with multiple constraints for vehicle routing optimization, employing a Differential Evolution (DE) algorithm with refined mutation strategies to tackle logistics distribution problems with multiple objectives. Reference [11] presents a particle swarm algorithm with a priority strategy tailored to order delivery characteristics, designed to address the open vehicle routing problem with time windows and multiple vehicle types. Reference [12] applies a PSO algorithm integrated with Tent Map Chaos to optimize RFID network deployment. Reference [13] proposes an integer particle update method based on particle exchange principles to solve multi-task vehicle routing problems with time window constraints. Reference [14] introduces a reverse optimization technique that leverages historical route decisions of experts to formulate a cost matrix encapsulating expert knowledge, which is then utilized in the resolution of route planning models.

In conclusion, to tackle the challenges of uneven initial population distribution and premature convergence commonly faced by the Particle Swarm Optimization (PSO) algorithm in route planning, this study introduces an enhanced Tent-Enhanced Particle Swarm Optimization (TPSO) algorithm tailored for multi-vehicle and multi-task routing in parks. Simulations conducted in Matlab 2023a have confirmed that the proposed TPSO algorithm is both efficient and practical for multi-vehicle and multi-task route planning within park delivery systems.

2. Problem Description and Model Construction

2.1 Problem Description

The problem is specifically described as a plurality of autonomous delivery vehicles of the same model starting from the same place, providing delivery services to distribution points at different locations in a certain order according to a pre-set time, and returning to the starting point after completing the delivery task

of each distribution point.

Figure 1 depicts the autonomous delivery vehicles and their operational environment within the park. Typically, the working environment for these vehicles is a relatively enclosed park area. The vehicles are equipped with a fixed number of storage compartments of varying sizes on both sides of the vehicle body, which are utilized for storing items intended for delivery.

[Fig. 1] Autonomous Delivery Vehicles and Their Delivery Environment

To effectively translate the problem into a model, this paper makes the following assumptions:

(1) The study focuses on the route optimization problem from a single park logistics center to multiple locations within the same park.

(2) The demand requirements, time window constraints, and geographical locations of all delivery points are known and predefined.

(3) The demand at each delivery point can be serviced by a single autonomous delivery vehicle.

(4) The park logistics distribution center has a fleet of autonomous delivery vehicles, all of which are of the same model and carrying capacity.

(5) All delivery vehicles depart from and return to the park logistics distribution center after completing their delivery tasks, with no pick-up tasks involved during the middle of the route.

(6) There are accessible paths between all delivery points, ensuring that vehicles can navigate to each location as required.

2.2 Symbols and decision variables

$z = \{z_0, z_1, \cdots z_n\}$--represent the set of the logistics sorting center and distribution points within the park, where the logistics sorting center serves n distribution points.

Q--The maximum load capacity of the delivery vehicle;

q_i--The demand at distribution point i $(i = \{1,2,\cdots,n\})$;

d_{ij}--The distance between distribution point i and distribution point j; $(i,j = \{1,2,\cdots,n\})$;

v_m--The vehicle's travel speed during time period m;

t_{ijm}--The travel time of the vehicle from delivery point i to delivery point j during time period m;

f_k--The fixed cost of the k-th vehicle. $(k = \{1,2,\cdots,k\})$;

c_{ij}^k--The cost incurred per kilometer by the k-th vehicle when traveling from delivery point i to delivery point j;

t_0^k--The departure time of the vehicle from the park's logistics sorting center;

t_i^k--The arrival time of vehicle k at delivery point i;

t_{fi}-- The time required to complete the unloading task at distribution station i;

y_{ki}--if the task at delivery point i is completed by vehicle k then $y_{ki} = 1$,otherwise $y_{ki} = 0$;

x_{ijk}--If vehicle k travels from delivery point i to delivery point j then $x_{ijk} = 1$, otherwise $x_{ijk} = 0$.

R_k--if the k-th vehicle at the park's logistics sorting center is utilized then $R_k = 1$,otherwise $R_k = 0$.

2.3 Objective Function Design

(1) Fixed Costs of Vehicles

The fixed cost of vehicles is typically a constant that encompasses the depreciation of the vehicle, insurance costs, and other fixed expenses associated with the vehicle's operation. This cost is only related to the number of vehicles actually used in the delivery process. Therefore, the fixed cost C_1 can be expressed as:

$$C_1 = \sum_{k=1}^{k} f_k R_k \qquad (1.1)$$

In this context, f_k represents the fixed cost generated by the k-th vehicle. R_k indicates whether the k-th delivery vehicle is used, where $R_k = 1$ signifies that the k-th vehicle is utilized.

(2) Transportation Costs of Vehicles

The transportation cost of vehicles primarily refers to the fuel consumption cost, which is directly proportional to the distance traveled by the vehicle. The transportation cost C_2 incurred during the delivery process can be expressed as:

$$C_2 = \sum_{k=1}^{K}\sum_{i=1}^{N} x_{ijk} v_m t_{ijm} c_{ij}^k \qquad (1.2)$$

In this context, c_{ij}^k represents the cost per kilometer incurred by the k-th vehicle when traveling from delivery point i to delivery point j; d_{ij} denotes the distance from delivery point i to delivery point j. x_{ijk} indicates the travel status of vehicle k, and when $k=1$, it signifies that the k-th vehicle is traveling from delivery point i to delivery point j.

(3) Late Penalty Costs

The late penalty cost is an additional cost incurred when an autonomous delivery vehicle k does not arrive at the delivery point i within the time specified by the customer. This cost is represented by tp.

(4) Overload Penalty Costs

The overload penalty cost refers to the additional cost incurred when an autonomous delivery vehicle k exceeds its rated carrying capacity. This cost is imposed as a penalty for overloading and is represented by op.

2.4 Model Construction

Incorporating the analysis above, this paper aims to minimize the sum of vehicle fixed costs c_1, vehicle transportation costs c_2, late penalty costs tp, and overload penalty costs op as the objective function of this study, as shown in Equation (3). The model for the distribution route from the park logistics sorting center is constructed as follows:

$$C = c_1 + c_2 + tp + op \qquad (1.3)$$

$s.t.$

$$\sum_{i=1}^{N} y_{ki} = 1, \forall k \in K \qquad (1.4)$$

$$\sum_{i=1}^{N} x_{ijk} = \sum_{i=1}^{N} x_{jik}, \forall i,j \in N, k \in K \qquad (1.5)$$

$$\sum_{i=1}^{N} y_{ki} q_i \le Q_{\max}, \forall k \in K \qquad (1.6)$$

$$\sum_{k\in K}\sum_{j\in N} x_{ijk} = 1, \forall i \in N \qquad (1.7)$$

$$\sum_{k\in K}\sum_{j=0}^{N} x_{0jk} = \sum_{k\in K}\sum_{j=0}^{N} x_{ji0} = 1 \qquad (1.8)$$

$$t_j^k = t_i^k + t_{fi} + t_{ijm} \qquad (1.9)$$

$$R_k \in \{0,1\}, x_{ijk} \in \{0,1\}, y_{ki} \in \{0,1\}, \forall i,j \in N, k \in K \qquad (2.0)$$

Equation (1.4) indicates that each distribution point can only be served by one vehicle, and all distribution points must be served. Equation (1.5) ensures the continuity of the vehicle's route,

meaning that after a vehicle arrives at a certain distribution point i, it must depart from that distribution point j. Equation (1.6) states that the total demand of stores on each delivery route must not exceed the vehicle's maximum load capacity. Equation (1.7) specifies that each distribution point can only be served once. Equation (1.8) stipulates that vehicles leaving the park's logistics sorting center must return to the park's logistics sorting center after completing their delivery tasks, ensuring the closure of the entire route. Equation (1.9) indicates that the time a vehicle arrives at the next distribution point is the sum of the time the vehicle arrives at the previous distribution point, the service time at the previous distribution point, and the travel time from the previous distribution point to the next distribution point, ensuring continuity in the vehicle's travel. Equation (2.0) defines the range of decision variables.

3. Algorithm design

The core of the basic Particle Swarm Optimization algorithm is to update the velocity and position of particles based on individual best and global best values. The specific algorithm model involves setting m particles in a D -dimensional search space, where the value of m can be adjusted according to actual conditions. In this space, the entire particle swarm is represented by $X=\{x_1,x_2,x_3,\cdots x_m\}$, where each particle i, during the t-th iteration of its own velocity and position, is represented by the D -dimensional position vector$x_i(t)=(x_{i1},x_{i2},\cdots x_{id})$ and the flying speed $v_i(t)=(v_{i1},v_{i2},\cdots v_{id})$[15]. A fitness function is constructed according to the actual situation to calculate the fitness value of the current population X_i, and it is compared with the current position of the particle to determine whether the position where the particle is located is optimal. From this, the PSO algorithm generates two types of extreme values: one is the best solution found by the particle itself, i.e., the individual best, represented by $p_i=(p_{i1},p_{i2},p_{i3}\cdots p_{id})$ .The other is the best solution found by the entire swarm, i.e., the global best, represented by $p_g=(p_{g1},p_{g2},p_{g3}\cdots p_{gd})$. Therefore, when looking for these two types of extreme values, the standard particle swarm updates its velocity and position according to the following formulas:

$$v_{id}(t)=w*v_{id}(t-1)+c_1r_1(p_{id}(t-1)-x_{id}(t-1))+c_2r_2(p_{gd}(t-1)-x_{id}(t-1)) \quad (2.1)$$

$$x_{id}(t)=x_{id}(t-1)+v_{id}(t) \quad (2.2)$$

A key characteristic of the PSO algorithm is that particles guide each other toward the currently discovered optimal positions through information sharing. This means that when a particle finds an excellent solution, other particles will adjust their flight paths to explore the area around that solution. This strategy helps the particle swarm to focus on areas that seem more likely to contain the optimal solution. However, this rapid convergence towards the optimal solution can also pose some problems. If one or more particles in the swarm discover an optimal solution that is actually a local optimum rather than a global optimum, the swarm may mistakenly take this local optimum as the target, causing all particles to converge on this local optimal position. When all particles are concentrated near this local optimum, the algorithm loses the ability to explore other areas of the solution space and cannot continue to search for potentially better solutions. At this point, the algorithm becomes trapped in a local optimum, leading to the phenomenon of premature convergence.

The pseudocode of the PSO algorithm is as

follows:

```
randomly generate an initial population
repeat
for each particle i
if f(xi)〉 f(pi) then pi=xi;
Pg=max {pi} ;
update velocity;
update position;
end
until temmination criterion is met;
```

To overcome the issue of Particle Swarm Optimization (PSO) algorithms easily getting trapped in local optima, this study introduces chaos theory into the PSO algorithm. Chaos, a form of random motion within nonlinear systems, is characterized by extreme sensitivity to initial conditions, ergodicity, and quasi-randomness. It enables chaotic motion to traverse every point in the phase space without repetition over a sufficiently long period, which can be leveraged for searching the optimal solution. By incorporating chaotic concepts into the PSO algorithm and optimizing based on these three characteristics of chaotic motion, the algorithm is enhanced to escape local optima, expand the search range, and maintain the diversity of the swarm. However, different chaotic mapping methods significantly influence the optimization process. The Logistic map is frequently cited in literature, but comparisons indicate that the Tent map offers better uniformity in traversal and faster iteration speed than the Logistic map. Through rigorous reasoning, it is concluded that the Tent map meets the prerequisites for a chaotic sequence in optimization algorithms. The algorithm employs the Tent map to generate a chaotic sequence, with the formula as follows:

$$x_{n+1} = \begin{cases} 2x_n, x_n \in [0,0.5] \\ 2(1-x_n), x_n \in [0.5,1] \end{cases} \quad (2.3)$$

This paper proposes a Tent map-based chaotic Particle Swarm Optimization algorithm, utilizing the population fitness variance as a criterion for premature convergence. When the algorithm exhibits signs of premature convergence, the chaotic Tent map is introduced into the basic PSO algorithm. By altering the update strategy of individual particle positions, the particle swarm is guided to undergo a certain number of chaotic updates, enabling the algorithm to escape local extrema and thereby enhancing the global optimization capability of the PSO algorithm.

The specific computational process of the algorithm is as follows:

(1) Initialize parameters, including the maximum number of iterations $iera_{\max}$, the number of particles m, acceleration constants c_1 and c_2, maximum inertia weight $w_{\max}$ and minimum value $w_{\min}$, maximum velocity $v_{id}(t)$, and position boundary $X_{\max}$.

(2) Chaotic initialization. Utilize the Tent chaotic map to generate m particle positions, with initial velocities set to $v_{id}(0) = 0.1X_{\max}$.

(3) Evaluate initial particles. Assess each particle according to the fitness function$f(x)$, and set the individual best for the particle swarm as $pbest_i(0)$ for the current position, and the global particle position corresponding to the optimal swarm fitness as $gbest_i(0)$.

(4) Begin iteration, determine whether the algorithm meets the convergence condition, i.e., whether it has reached a certain precision or the maximum number of iterations? If satisfied, proceed to step 9; otherwise, continue to the next step.

(5) Update the particle's velocity $v_{id}(t)$ and position $x_{id}(t)$, and perform boundary processing.

(6) Calculate the fitness value based on the new particle and update $pbest_i(0)$ and $gbest_i(0)$ according to the new fitness value.

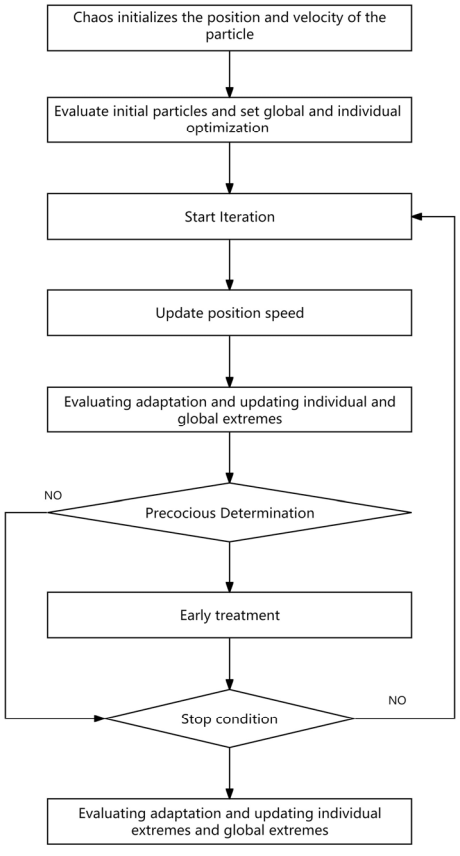
(7) Determine if the particle has reached a state of premature convergence. The swarm's fitness variation is used to judge whether

the particle has become prematurely convergent. If the conditions for premature convergence are satisfied, perform the premature treatment as outlined in step 8; otherwise, proceed to step 4.

(8) Execute the premature treatment phase, perform Tent chaotic mutation on the individual best of the swarm particles $gbest_i(0)$, within the decision variable space, according to formula (13), iteratively generate a chaotic sequence of the same quantity as the size of the particle's dimensions, repeat multiple times, compare the fitness value after adding perturbation, and update the global optimal position and global optimal solution based on the best. Return to step 4.

(9) Output the optimal result and its optimal fitness value.

Based on the aforementioned steps, a flowchart of the improved algorithm based on the Tent map can be drawn as shown in Figure 2.



[Fig. 2] Flowchart of the Improved Particle Swarm Algorithm Based on the Tent Map

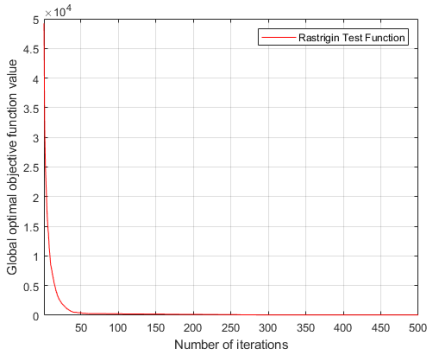
The improved PSO algorithm presented in this paper builds upon the foundation of the standard PSO algorithm. It assesses the degree of aggregation within the particle swarm by monitoring the variance of the swarm's fitness. Based on this assessment, the algorithm decides whether to introduce a chaotic search via the Tent map. This intervention is designed to enable the particle swarm to break free from local minima during the later iterations, thereby enhancing its global search capabilities. The ultimate goal is to expedite the discovery of the global optimum solution to the problem at hand.

The pseudocode for introducing Tent chaos for perturbation is as follows:

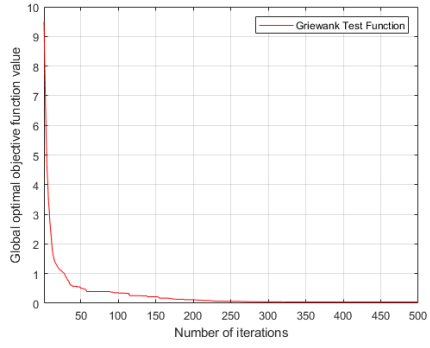
```
Tent map initialization function
function population = Tent_int(cusnum , ···)
    Initialize positions and velocities
    population = zeros(4, cusnum);
    x = rand(1, cusnum);
    for i = 1 to cusnum
        if x(i) 〈 0.5
            x(i) = 2 * x(i);
        else
            x(i) = 2 * (1 - x(i));
        end
    end
```

This pseudocode outlines the initialization process using the Tent map, where xis initialized randomly and then transformed according to the Tent map rules, which depend on the value of $x(i)$ being less than 0.5.

Common test functions such as the Rastrigin function and the Griewank function are used to evaluate the aforementioned improved PSO algorithm. From Figures 3 and 4, it can be observed that the enhanced particle swarm algorithm exhibits favorable convergence characteristics.



[Fig. 3] Rastrigin Test Results



[Fig. 4] Griewank Test Results

4. Example Solution

4.1 Basic Data

This paper selected X Park as a case study for research, where deliveries are made from the park's logistics center to various locations within the park according to agreed-upon times. The logistics center possesses several autonomous delivery vehicles, all of which are of the same model and have the same load capacity. All delivery vehicles depart from the park's logistics sorting center and must return there after completing their delivery tasks, with no pick-up tasks involved during the journey.This includes route details, demand volumes, delivery time information, and the locations of the delivery points. The data has been streamlined as follows, with comprehensive details provided in Table 1.

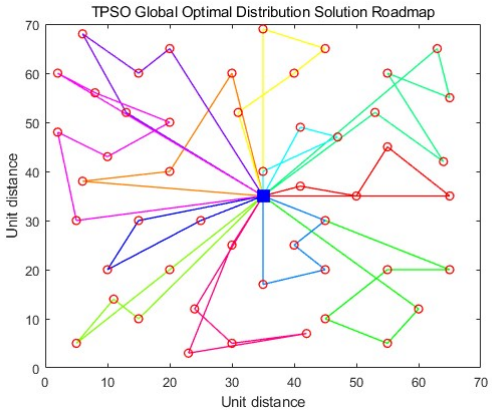
〈Table 1〉 Base data

Station Number	X	Y	Demand	ReadyTime	DueTime	Service Time
0	35	35	0	0	230	0
1	41	49	10	161	171	10
2	35	17	7	50	60	10
3	55	45	13	116	126	10
4	55	20	19	149	159	10
5	15	30	26	34	44	10
6	25	30	3	99	109	10
7	20	50	5	81	91	10
8	10	43	9	95	105	10
9	55	60	16	97	107	10
10	30	60	16	124	134	10
11	20	65	12	67	77	10
12	50	35	19	63	73	10
13	30	25	23	159	169	10
14	15	10	20	32	42	10
15	30	5	8	61	71	10
16	10	20	19	75	85	10
17	5	30	2	157	167	10
18	20	40	12	87	97	10
19	15	60	17	76	86	10
20	45	65	9	126	136	10
21	45	20	11	62	72	10
22	45	10	18	97	107	10
23	55	5	29	68	78	10
24	65	35	3	153	163	10
25	65	20	6	172	182	10
26	45	30	17	132	142	10
27	35	40	16	37	47	10
28	41	37	16	39	49	10
29	64	42	9	63	73	10
30	40	60	21	71	81	10
31	31	52	27	50	60	10
32	35	69	23	141	151	10
33	53	52	11	37	47	10
34	65	55	14	117	127	10
35	63	65	8	143	153	10
36	2	60	5	41	51	10
37	20	20	8	134	144	10
38	5	5	16	83	93	10
39	60	12	31	44	54	10
40	40	25	9	85	95	10
41	42	7	5	97	107	10
42	24	12	5	31	41	10
43	23	3	7	132	142	10
44	11	14	18	69	79	10
45	6	38	16	32	42	10
46	2	48	1	117	127	10
47	8	56	27	51	61	10
48	13	52	36	165	175	10
49	6	68	30	108	118	10
50	47	47	13	124	134	10
51	49	58	10	88	98	10

4.2 Example Solution Results

The experimental algorithm was simulated in Matlab2023a, and the computer operating environment was Windows 10 platform , 2.9 GHz CPU 6-core i5 processor, 16 GB memory. The TPSO algorithm

proposed in this paper was programmed in MATLAB2023a and the global optimal particle was decoded to obtain the following global optimal path distribution solution, as shown in Figure 5.



[Fig. 5] TPSO global optimal distribution solution route map

In Figure 5, the X and Y axes represent the simplified coordinates of the distribution points within the park. Each circle represents a distribution point, and lines of different colors represent different delivery vehicles. Circles connected by lines of the same color indicate all the distribution points visited by one vehicle.

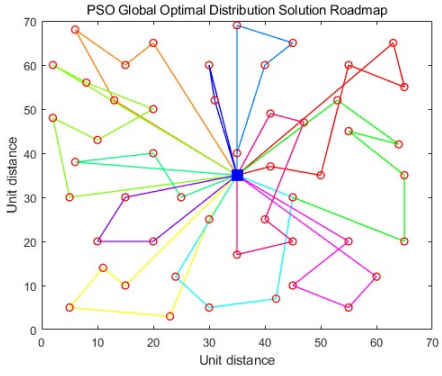
The optimal delivery plans obtained through simulation calculations for the autonomous delivery vehicles are as follows:

Delivery Route 1: 0->28->12->3->24->0.
Delivery Route 2: 0->45->18->10->0.
Delivery Route 3: 0->31->30->20->32->0.
Delivery Route 4: 0->14->44->38->37->0.
Delivery Route 5: 0->39->23->22->4->25->0.
Delivery Route 6: 0->33->29->9->34->35->0.
Delivery Route 7: 0->27->50->1->0.
Delivery Route 8: 0->2->21->40->26->0.
Delivery Route 9: 0->5->16->6->0.
Delivery Route 10: 0->11->19->49->48->0.
Delivery Route 11: 0->36->47->7->8->46->17->0.
Delivery Route 12: 0->42->15->41->43->13->0.

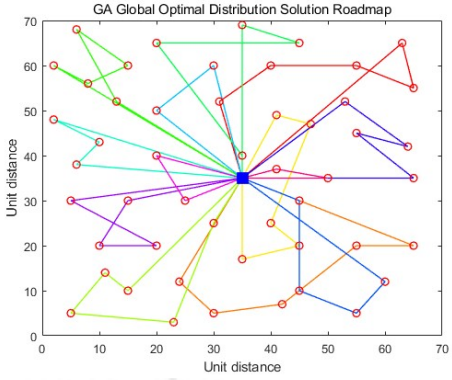
In the implementation of the Tent-Enhanced Particle Swarm Optimization (TPSO) algorithm, the parameter settings were as follows: the penalty function coefficient for capacity constraints was set to 10; for time window constraints, it was 100; the number of particles was 50; the number of iterations was 100; and the parameter for the chaotic map was 0.5. The resulting global optimal delivery plan yielded a total cost of the global optimal solution of 1053.9134, with 12 vehicles used, a total travel distance of 1053.9134, and no violations in terms of constraint paths or customer constraints.

4.3 Comparative Analysisclear

Figures 6 and 7 display the global optimal delivery plans obtained through 100 generations of iteration using traditional Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) in Matlab simulations, respectively.

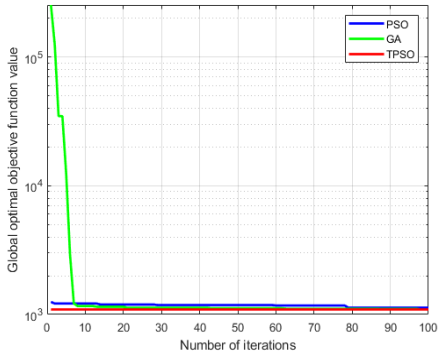


[Fig. 6] PSO global optimal distribution solution route map



[Fig. 7] GA global optimal distribution solution route map

The traditional PSO algorithm, after 100 generations of data iteration and convergence, obtained a total cost of 1057.8177 for the global optimal delivery plan. The GA algorithm, after iteration and convergence, obtained a total cost of 1093.1074 for the global optimal delivery plan.



[Fig. 8] Relationship between Iteration Counts and Optimal Objective Function Values for TPSO, GA, and PSO Algorithms.

In Figure 8, it can be observed that the TPSO algorithm has a significantly faster convergence rate compared to the Particle Swarm Optimization and Genetic Algorithm. In terms of the total cost of the optimal solutions obtained, TPSO outperforms the comparative sample algorithms.

〈Table 2〉 Cost Comparison of the Three Algorithms

Algorithm	Total Cost	Vehicles	Distance	Routes Violating Constraints
TPSO	1053.9134	12	1053.9134	0
PSO	1057.8177	12	1057.8177	0
GA	1093.1074	13	1093.1074	0

From Table 2, it can be observed that when comparing the TPSO algorithm presented in this paper with the PSO and GA algorithms, the TPSO outperforms the traditional Particle Swarm Optimization and Genetic Algorithm in terms of total cost of the optimal solution, number of vehicles used, and total distance traveled. The TPSO demonstrates superior performance in solving the multi-vehicle multi-task routing problem in the park delivery.

5. Conclusion

This paper explores the path planning for multi-vehicle and multi-task autonomous delivery vehicles in parks using an improved Particle Swarm Optimization algorithm. The study introduces the Tent map into the basic PSO algorithm and integrates considerations for time and load costs during the computation process. By applying the Tent chaotic mapping with a certain probability when local optima may occur, the algorithm can escape from these suboptimal solutions. The TPSO algorithm effectively combines the global traversal, randomness, and regularity of the Tent map with the multidimensional space optimization capability of the PSO algorithm, enhancing the algorithm's resistance to premature convergence and improving its global search ability, as well as accelerating the speed of iterative convergence. Comparative simulation experiments with several commonly used path planning algorithms have fully demonstrated these advantages, indicating that the TPSO algorithm can effectively address the issues of basic PSO algorithms falling into local optima and slow convergence when solving route planning problems for autonomous delivery vehicles in parks. It should be noted that there is still much room for improvement in this research, especially in dealing with local path planning, which requires further refinement.

REFERENCES

[1] W. Hejing and W. Lina. "A review of robot path planning algorithms". Journal of Guilin University of Technology.https://kns.cnki.net/kcms/detail//45.1375.N.20221213.1104.001.html

[2] S. Ruitong, Yuan Qingni and Y. Junhui, et al. "Dynamic path planning based on improved particle swarm optimization and dynamic window method ". Journal of Small and Micro Computer Systems, Vol44, No.8, pp.1707-1712., 2023.

[3] Z. Juntao, L. Xiaochuan and L. Junmi. "Application of improved whale optimization algorithm in robot path

planning". Journal of Northeastern University (Natural Science Edition), Vol44, No.8, pp. 1065-1071, 2023.

[4] W. Ziqiang, H. Xiaoguang and L. Xiaoxiao, et al. "A review of global path planning algorithms for mobile robots ". Computer Science, Vol48, No.10, pp. 19-29, 2021.

[5] P. Wu, T. Li and G. Song, "UCAV Path Planning Based on Improved Chaotic Particle Swarm Optimization," 2020 Chinese Automation Congress (CAC), Shanghai, China, pp. 1069-1073, 2020.

[6] Y. Duan, N. Chen, L. Chang, Y. Ni, S. V. N. S. Kumar and P. Zhang, "CAPSO: Chaos Adaptive Particle Swarm Optimization Algorithm," in IEEE Access, vol. 10, pp. 29393-29405, 2022.

[7] L.Guo-ming, L.Jun-hua. "Stochastic vehicle routing problem based on hybrid tabu search algorithm". Control and Decision Vol36, No.9, pp. 2161-2169, 2021.

[8] Zh. Yuxin. "Research on optimization of exhibition logistics distribution service based on mixed time windows". Beijing Jiaotong University, 2023.

[9] Zh.Shu and Tang Miao. "Improved PSO algorithm and its application in UAV path planning. Computer Systems Applications" Vol23, No.3 pp. 330-337, 2023,

[10] Zh. Haifei. "Research on vehicle routing optimization for urban logistics distribution under multiple constraints". Jiangnan University, 2023.

[11] G. Zijian and Chu Liangyong . "The open multi-vehicle routing problem considering order priority and time window" . Science Technology and Engineering , Vol24, No.6 pp. 2521-2529 , 2024.

[12] W.Lei. "Research on RFID network deployment optimization based on improved particle swarm algorithm". Jiangnan University, 2014.

[13] M. Xuan , P. Peng and L. Qing."Improved Particle Swarm Optimization for Vehicle Routing Problem with time windows".Computer Engineering and Applications , Vol45, No.7 pp.200-202 , 2009.

[14] C.Yuyi and Chen Lu. "Inverse optimization method for vehicle path planning problem ". Journal of Shanghai Jiaotong University, Vol56, No.1 pp. 81-88, 2022.

[15] L.JianQiang, C. JunChuang,S. Tao,Zh. QingLing and L.QiuZhen. "Multitask-based assisted evolutionary algorithm for vehicle routing problems incomplex logistics distribution scenarios". Acta Automatica Sinica, Vol50, No.3 pp. 544-559, 2024.

[16] J. Dethloff "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up". ORSpektrum, Vol23, No.1 pp. 79-96, 2001

노 크(Lu Ke) [정회원]

- June 2000, PLA Information Engineering University, Computer Applications (Bachelor of Engineering)
- May 2008, China University of Mining and Technology (Beijing), computer engineering technology field (Master of Engineering)
- September 2022-Present, has been pursuing Ph.D in Intelligent Fusion in IT at Mokwon University (student)

〈관심분야〉

Computer Networks, Path Planning

민 병 원(Byung-Won Min) [정회원]

- He received M.S. degree in computer software from Chungang University, Seoul, Korea in 2005.
- He received Ph.D. degree in the dept. of Information and Communication Engineering, Mokwon University, Daejeon, Korea, in 2010.
- He is currently a professor of Mokwon University since 2010.

〈관심분야〉

digital communication systems, Big Data