2023 International Symposium on Nonlinear Theory and Its Applications
NOLTA2023, September 26-29, 2023, Catania and Online

Performance investigation of chaotic search method for vehicle routing problems with drones

Kazuma Nakajima†, Yusuke Matsuzaki†, Takafumi Matsuura‡, and Takayuki Kimura‡

†Graduate School of Electronics, Information and Media Engineering,
Nippon Institute of Technology
‡Department of Data Science, Faculty of Advanced Engineering,
Nippon Institute of Technology
¶Department of Electrical, Electronics and Communication Engineering,
Faculty of Fundamental Engineering, Nippon Institute of Technology
Email: 2238010@stu.nit.ac.jp, 2228015@stu.nit.ac.jp, matsuura@nit.ac.jp, tkimura@nit.ac.jp

Abstract— Transport and logistics by unmanned aerial vehicles, commonly known as drones, have attracted attention in recent years for their potential to revolutionize the transport industry. Amazon was the first to use drones to deliver goods. Several distribution companies have since been working on similar services. To customer delivery route by using drones, first, Flying Sidekick Traveling Salesman Problems have been formulated as a problem that constructs delivery route by using a drone and a vehicle. In addition, vehicle routing problems with drone (VRPD), in which several numbers of drones and vehicles deliver goods to customers, has been proposed. Besides, an adaptive large-scale nearest neighbor search (ALNSVRPD), which shows effective solving performance for VRPD, has been proposed. As one of solving method to VRPD, in this study, we propose a new solution search method based on chaotic neurodynamics for VRPD and investigate its effectiveness. Experimental results show that the proposed method successfully improves the computation time as compared to the conventional solving method.

1. Introduction

With the recent development of technological progress, drones have attracted a lot of attention in the transport industry. The delivery of goods using drones was first introduced by Amazon, and since then several distribution companies, including DHL and Zookal, have been working on similar services [1]. Amazon plans to use prime air to deliver products from warehouses to customers and to move products between warehouses [2]. Although the potential for drone delivery is limited by distance, battery life, and payload, the use of drones and vehicles can have a significant impact on reducing delivery costs and time. Disadvantage of vehicle delivery is that the number of vehicles increases as the number of parcels increases. This disadvantages of vehicles are counteracted by the advantages offered by drones; the drones, which can safely grab and carry small loads, can assist drivers with deliveries, increasing the number of deliveries per hour without driving additional distances. A delivery planning problem using vehicles and drones was first proposed by Murray and Chu (2015) as the Flying Sidekick Traveling Salesman Problem (FSTSP) [3]. In this problem, one vehicle with one drone delivers goods to customers. The objective of FSTSP is to deliver parcels to all customers and minimize the total time took by the vehicle and the drone to leave and return to the depot. An extension problem to FSTSP is vehicle routing problems with drones [4]. The objective of this problem is to minimize the average delivery time of parcels.

In this study, we proposed a chaotic search method as a new solving method for VRPD. In the chaotic search, the firing of chaotic neurons determines a movement of a current solution to a neighborhood solution to optimize the problems. Previous studies [5, 6] confirmed the effectiveness of chaotic search for TSP or VRP. In this study, we evaluate the performance of chaotic search method for VRPD. In addition, we compared the proposed chaotic search method with an adaptive large neighborhood search method (ALNSVRPD) [7]. Numerical experiments show that the proposed method finds a solution in a shorter time than conventional method.

2. Vehicle routing problems with drones

A vehicle routing problem with drones (VRPD) is a problem that aims to minimize the time required to deliver parcels to all customers using vehicles and drones. In the VRPD, an objective function is to minimize the total cost including time and fuel and defined as follows [7].

$$\min \sum_{v \in V} \Big(\sum_{(i,j) \in A} c_{ij}^T l_{ij}^v + \sum_{z \in P} c_z^D y_z^v \Big), \quad z = (i, j, k), \tag{1}$$

$$c_{ij}^T = p f m d_{ij}, \tag{2}$$

$$l_{ij}^v = \begin{cases} 1 & (\text{if vehicle } v \text{ moves between } i \text{ and } j), \\ 0 & (\text{otherwise}), \end{cases} \tag{3}$$

ORCID iDs Kazuma Nakajima: 0009-0009-1999-3173, Yusuke Matsuzaki: 0009-0009-4559-0255, Takafumi Matsuura: 0000-0002-8007-0556, Takayuki Kimura: 0000-0003-0160-6146

$$c_z^D = c_{ij}^D + c_{jk}^D, \tag{4}$$

$$c_{ij}^D = \gamma c_{ij}^T, \tag{5}$$

$$y_z^v = \begin{cases} 1 & \text{(if the drone moves between customers } z\text{)}, \\ 0 & \text{(otherwise)}, \end{cases} \tag{6}$$

where c_{ij}^T is a transportation cost of vehicles between customers i and j, l_{ij}^v is the decision variable of vehicle v between customers i and j, z is a subset consisting of three consecutive customers; i as a drone departure point, j as a drone service customers, and k as a drone collection point, c_z^D is a drone transportation cost, c_{ij}^D is a drone transportation cost between customers i and j, y_z^v is a decision variable for drone delivery of vehicle v, V is a set of vehicles, A is a set of customers, P is a set of z, p is a fuel price, f is a fuel consumption, m converts *mile* to *km*, d_{ij} is a distance between customers i and j, and γ is a parameter that calculates the drone cost. The drone can only deliver to one customer per delivery, and its flight time is limited. In addition, both the vehicle and the drone have payload capacity limitations.

In this model, only one drone can be mounted on each vehicle. Figure 1 illustrates the prohibited movements of drones. In Fig. 1, squares indicate depots, circles indicate visiting customers, black circles indicate customers who cannot be delivered by drones because of their weights of parcels, and triangles indicate customers to be delivered by drones.

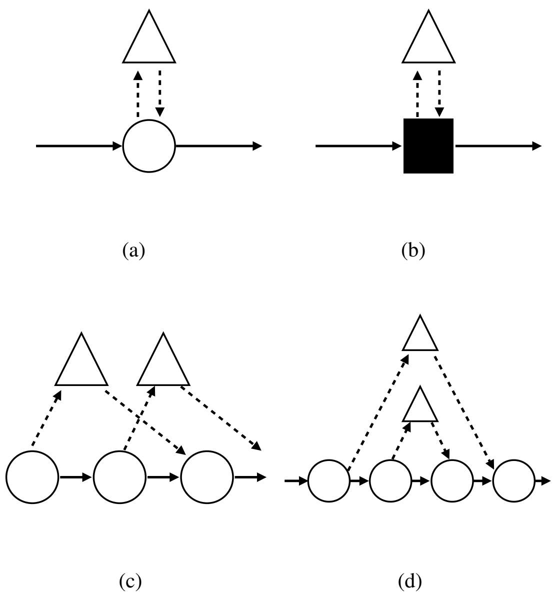

Figure 1: Prohibited drone movement

In Fig. 1(a), the drone cannot launch and arrive at the same customer because the vehicle has to wait for the arrival of drone and this increases delivery time. The drone is always launched from a vehicle. The movement of the drone from depot to depot is then prohibited as show in Fig. 1(b). In Fig. 1(c) and (d), two drones are launched to the next customer. These movements are then prohibited.

3. Neighborhood operation

Neighbor operations for improving solutions can obtain better solutions. As similar to Ref. [7], we used six neighborhood operations to search for a solution: two operations are destroy methods that remove customers from the current routes and four are repair methods that insert the removed customers back into the route.

3.1. Destroy method

In the destroy method, first, we determine the number of removing customer as follows [7]:

$$\beta = \min\{\max(c_{\text{low}}, \delta|C|), c_{\text{lim}}\}, \tag{7}$$

where c_{low} and c_{lim} are the absolute lower and upper bounds on the number of removing customers, δ is a ratio of removing customers, c_{low} is selected as a discrete random number between 1 and 3. In this study c_{lim} is set to 40, and δ is set to 0.15. If a removing customer is not the one where the drone launches or retrieves, the customer removed from the route. On the other hand, if a removing customer is the place where the drone launches or retrieves, we removed the other sortie customer delivered by the drone.

1. Random destroy

   The random destroy removes a randomly selected customer from the route.

2. Cluster destroy

   In the cluster destroy, one randomly selected customer is removed. Next, two customers that are close to the removed customer are selected, and one of them are removed.

3.2. Repair method

1. Greedy vehicle first sortie second repair method

   First, the Greedy vehicle first sortie second repair method inserts the removed customers into the routes with the smallest increase in costs. After inserting all removed customers, one customer from all customers is randomly selected. If the randomly selected customer is not place where a drone launches or retrieves, the customer is delivered by the drone if total costs decrease.

2. Nearby area vehicle first sortie second repair method

   First, this neighborhood operation randomly selects one of the removed customers. Next, if the insertion of this selected customer meets the time window constraint of the corresponding vehicle, this selected customer is inserted between the existing customers located within 5 miles of the selected removed customers. This process is repeated until no more customers are removed. Next, a customer is randomly selected from the current route, and a drone on the route can deliver the customer if possible. In this neighborhood operation, less than 10% of bad solutions are allowed to change the current solution. If there is no place for the removed customer inserts, we added a vehicle to visit this customer.

3. Closest insertion repair method

   This neighborhood operation inserts the removed customers using both vehicles and drones. First, one of the removed customers is randomly selected and inserted into the closest route. If customers cannot be inserted by this operation, the removed customers are inserted by using the Greedy vehicle first sortie second repair method.

4. Heavy insertion repair method

   First, this neighborhood operation inserts the removed customers where the drones cannot serve into the existing routes if the current objective value decreases. If the removed customers remain, they are inserted using the closest insertion repair method.

4. Chaotic search proposed Method

In the conventional chaotic search method for TSP [5] or VRP [6], each chaotic neuron corresponds to customers to which the neighborhood operations are applied. However, this neural coding is not suitable for using large numbers of neighborhood operations because of the increase in the number of neurons. Therefore, in this study, we construct the chaotic neural network in which each chaotic neuron corresponds to the destroy and repair operations. Figure 2 shows a chaotic neural network in which each neuron corresponds to each neighborhood operation.

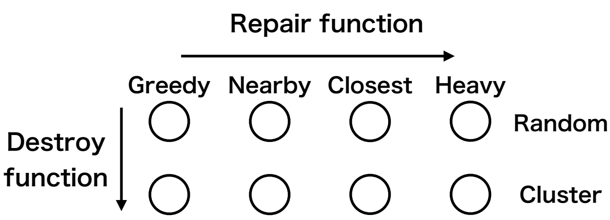

Figure 2: An example of chaotic neural network coding in this study

In our chaotic neural networks, the ijth neuron has a gain effect ξ_{ij}, a refractory effect ζ_{ij}, and a mutual connection η_{ij}. The gain effect encourages the firing of neurons. The refractory effect inhibits firing of neurons for a certain period of time if the neuron has fired. The mutual connection controls firing rates of neurons in the chaotic neural network. These effects are defined as follows:

$$\xi_{ij}(t+1) = \beta\Delta_{ij}(t), \tag{8}$$

$$\Delta_{ij}(t) = s_{\text{old}} - s_{\text{new}}, \tag{9}$$

$$\zeta_{ij}(t+1) = -\alpha \sum_{u=0}^{t} k_r^u x_{ij}(t-u) + \theta, \tag{10}$$

$$= k_r\zeta_{ij}(t) - \alpha x_{ij}(t) + (1-k_r)\theta, \tag{11}$$

$$\eta_{ij}(t+1) = -W \sum_{p=1}^{N}\sum_{q=1}^{M} x_{pq}(t) + W, \tag{12}$$

where $\beta > 0$ is a scaling parameter, $\Delta_{ij}(t)$ is a value of improvement for neighborhood solution, and s is objective function value. s_{old} is the best known solution, s_{new} is a solution obtained in the current search, $0 < k_r < 1$ is a damping factor of the refractoriness, $\alpha > 0$ is a scaling parameter, θ is a threshold value, W is a control parameter for the firing rate of neurons, N is the number of the destroy operations, and M is the number of the repair operations. Then, the output of the ijth neuron is defined as follows:

$$x_{ij}(t+1) = g\{\xi_{ij}(t+1) + \zeta_{ij}(t+1) + \eta_{ij}(t+1)\}, \tag{13}$$

where $g(y) = 1/(1+e^{-y/\epsilon})$ and ϵ is a small positive parameter. If the output value, $x_{ij}(t+1)$, is larger than 0.5, the neighborhood operation corresponding to the ijth neuron is performed.

5. Numerical experiment

We used the VRPD instance benchmark [7]. This benchmark is denoted a.b.c, where a is the number of customers, b is the number of grids, and c is an index of the problem. The duration of numerical simulations are set to 90 seconds. Table 1 shows the values of parameters used in the chaos search method for each benchmark.

Table 1: Parameter settings of chaos search

instance	ϵ	β	k_r	θ	α	W
50.10.1	0.001	0.7	0.9	0.04	0.09	0.007
100.10.1	0.001	0.8	0.4	0.01	0.01	0.007
150.10.1	0.002	0.8	0.5	0.03	0.04	0.007
200.10.1	0.002	0.9	0.8	0.02	0.02	0.001

Table 2 summarizes the experimental results, where O.average is an average value of objective function in Ref. [7]. Average is an average value of objective function over the best 10 times out of 30 trials by our method, a gap is an error rate of our method from the results of Ref. [7], O.bks is the best known solution [7], and bks is the best known solution obtained by our method.

In Table 2, the proposed method obtains a value close to that of the conventional method [7] with an execution time of 90 seconds. As a result, the proposed method constructs close solutions in less time than the conventional method, whose execution time was 300 seconds [7]. The reason for getting better solutions in a less time is that the proposed method effectively uses firing and non-firing neurons to search for a solution. We believe that this effective neuron firing successfully leads to good solutions in the solution space.

Table 2: Experimental results

instance	O.average	average	gap(%)	O.bks	bks
50.10.1	5.86134	5.87688	0.26	5.86134	5.87617
100.10.1	6.89015	7.02259	1.92	6.85741	7.00385
150.10.1	8.93509	9.20537	3.02	8.79027	9.07597
200.10.1	10.40499	10.5221	1.12	10.09452	10.3322

Tables 3 and 4 show the results for averaged values of objective function if we changed the computation time for each problem. The numbers in parentheses indicate the error rate compared to Ref. [7].

Table 3: Average of 30 solutions by our proposed method

instance	30sec	60sec	90sec	120sec	300sec
50.10.1	5.88701	5.88475	5.88347	5.87939	5.87821
	(0.43)	(0.39)	(0.37)	(0.30)	(0.28)
100.10.1	7.13983	7.1086	7.10124	7.0852	7.06185
	(3.62)	(3.17)	(3.06)	(2.83)	(2.49)
150.10.1	9.52087	9.42754	9.34568	9.42528	9.33632
	(6.55)	(5.51)	(4.59)	(5.48)	(4.49)
200.10.1	10.773	10.7135	10.6628	10.6437	10.5769
	(3.53)	(2.96)	(2.47)	(2.29)	(1.65)

Table 4: Average of top 10 solutions by our proposed method

instance	30sec	60sec	90sec	120sec	300sec
50.10.1	5.8783	5.87724	5.87688	5.87697	5.87655
	(0.28)	(0.27)	(0.26)	(0.26)	(0.25)
100.10.1	7.06403	7.0397	7.02259	7.02867	7.00423
	(2.52)	(2.17)	(1.92)	(2.01)	(1.65)
150.10.1	9.29544	9.25481	9.20537	9.23348	9.21103
	(4.03)	(3.57)	(3.02)	(3.33)	(3.08)
200.10.1	10.6784	10.5985	10.5221	10.5142	10.412
	(2.62)	(1.85)	(1.12)	(1.04)	(0.06)

In Tables 3 and 4, the proposed method shows similar values to the ALNSVRPD with an error rate of less than 4% for the problems except for 150.10.1 problem. Especially, the proposed method shows closer values of objective functions for top 10 solutions than that by 30 solutions. In addition, the computation time for getting is also shorter than that of ALNSVRPD. These results confirm that good solutions can be obtained in about 90 seconds, confirming the effectiveness of chaos search.

6. Conclusion

In this study, we evaluated the performance of a chaotic search method for VRPD. As a result, we confirmed the effectiveness of the proposed method with regard to the computation time. Our method obtained reasonable solutions with the shorter time than the conventional method. In the future works, we plan to change neuron coding in the chaotic search method from neighborhood operations to customer operations to increase solution diversity and improve solution search performance. In addition, we will continue to analyze the reasons for getting good results achieved by the chaotic method. The research of T. M and T. K was partially supported by JSPS KAKENHI Grant Numbers JP22K04602, JP23K04274.

References

[1] French,S,2015. Drone delivery is already here and it works. https://www.marketwatch.com/story/drone-delivery-is-already-here-and-it-works-2015-11-30. 2023/06/23.

[2] Wang,D,2016. The Economics of Drone Delivery. https://www.flexport.com/blog/drone-delivery-economics/. 2023/06/23.

[3] C. C. Murray and A. G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, Vol.54, pp.86–109, 2015.

[4] R. Daknama and E. Kraus. Vehicle routing with drones. *arXiv preprint arXiv:1705.06431*, 2017.

[5] M. Hasegawa, T. Ikeguchi, and K. Aihara. Solving large scale traveling salesman problems by chaotic neurodynamics. *Neural Networks*, Vol.15(2), pp.271–283, 2002.

[6] T. Kimura, T. Hoshino and T. Ikeguchi. *A New Diversification Method to Solve Vehicle Routing Problems Using Chaotic Dynamics*, pp.409–412. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[7] D. Sacramento, D. Pisinger, and S. Ropke. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, Vol.102, pp.289–315, 2019.