Proceedings of the 9th Annual World Conference of the
Society for Industrial and Systems Engineering,
2020 SISE Virtual Conference
September 17-18, 2020

# A Dynamic Approach to the Traveling Salesman Problem with Drones Using a Self-Organizing Map

**M Tairi, Y Wang, and SW Yoon**

State University of New York at Binghamton
Binghamton, NY, USA

Corresponding author's Email: mtairi1@binghamton.edu

**Autor Note:** Mohamed Tairi is a Graduate Research Associate at the Watson Institute of Systems Excellence at Binghamton University. Dr. Yong Wang is an Assistant Professor and Dr. Sang Won Yoon is an Associate Professor in the Department of Systems Science and Industrial Engineering at Binghamton University.

**Abstract:** This research proposes the use of a route-first cluster second heuristic using a Self-Organizing Map and a greedy algorithm to solve large instances of the traveling salesman problem with drones (TSP-D). The goal of the TSP-D is to efficiently route a drone equipped vehicle from a source depot to deliver packages to a set of customers and return to the source depot with minimal distance travelled. The TSP-D has been widely studied since first being introduced in 2015 and it has been shown to significantly reduce travel time in comparison to a pure TSP. Despite this, the majority of the current scientific literature involves the use of computationally demanding algorithms and techniques that require a considerable amount to time to execute and only focus on small scale problems. The current article aims to dynamically solve large instances of the TSP-D to near optimality using a SOM to generate a near optimal solution to the TSP, and a Greedy insertion heuristic to determine locally optimal insertion candidates for drone delivery. The  results show that the initial solution generated using the SOM was 7% higher than the optimal solution while the solution to the TSP-D was 39% less than the optimal TSP solution and was computed in 28.8 seconds.

*Keywords:* Traveling Salesman Problem with Drones, Self-Organizing Map, Greedy Heuristic, Final Mile Delivery

## 1. Introduction

In recent years, the application of unmanned aerial vehicles (UAV) in final mile transportation has been catching the attention of industry giants such as Amazon, Google, and DHL. UAVs, or drones, have proven to be very effective in a wide range of functions such as delivering aid and assessing structural damage caused by natural disasters, environmental preservation, and more recently, final mile transportation. Although the application of drones in final mile delivery is still heavily regulated by the Federal Aviation Administration (FAA), there have been many major breakthroughs made in this field during the last few years. For example, one year after sanctioning a small-scale UPS managed drone delivery system to deliver lab samples to a hospital in North Carolina, the FAA approved an expansion plan for the wide scale expansion of the medical delivery system. The expansion plan relaxed certain FAA regulations that constrained the drone delivery system, which included allowing for a drone fleet of unlimited size as well as allowing the drones to operate beyond visual line of sight. This was a major breakthrough in the field of drone managed transportation as no other corporation had attained this level of clearance from the FAA (Garcia, 2019). Thus, a future where final mile delivery can be accomplished via drone may be even sooner than expected.

Drone assisted delivery has the capability of significantly reducing transportation and causing a major shift in the way final mile logistics are handled. However, one significant drawback of drone assisted delivery is that drones have a limited battery life and a limited flight range. For example, the UPS Workhorse Horsefly drone has a limited flight range of about thirty minutes for a round trip. Additionally, the drone has a weight payload capacity of ten pounds (Burns, 2017). These constraints present a roadblock to the opportunity of drone managed delivery from a centralized depot as many packages can be expected to have a destination more than fifteen minutes from the distribution center. The solution to these constraints is to combine a parcel delivery drone with a specialized delivery vehicle that can make final mile deliveries to customers while simultaneously serving as a mobile depot for the drone. This scenario of drone-equipped vehicle used for final mile deliveries was popularized in 2015 and was initially dubbed the “Flying Sidekick Traveling Salesman Problem (FSTSP), but the more recent literature refer to this problem as the “Traveling Salesman Problem with Drones (TSP-D).

## 2. Literature Review

Literature regarding the TSP-D has been rapidly growing since first being introduced by Murray & Chu in 2015 (Murray & Chu, 2015). This initial study dubbed the problem as the "Flying Sidekick Traveling Salesman Problem" (FSTSP) and it studied the potential time savings that can be achieved by a single truck and drone combination. This study also introduced the parallel drone scheduling travelling salesman problem, which is a scenario where the distribution center is close enough to the customers that drones can be scheduled in parallel with a vehicle to make final mile deliveries from a central depot. Despite the study being the first of its kind to solve the TSP-D, most studies follow the same process for determining a solution to the TSP-D. The solution process is typically a two-step process which first involves solving the underlying TSP problem using standard heuristic algorithms, followed by an additional algorithm to determine the ideal customer nodes to be serviced by the drone. Although this aspect of Murray & Chu's research has remained relatively the same in the more recent literature, one way in which the literature has evolved since then is by introducing scenarios where multiple vehicles may be paired with multiple drones (Karak & Abdelghany, 2019; Kitjacharoenchai, et al., 2019).

Two additional research articles of interest are those that utilized a greedy algorithm to minimize either time or cost (Sacramento, et al., 2019; Olivares et al., 2015). The article that aimed to minimize cost used three different approaches. One approach was based on the approach taken in the study by Murray and Chu, one was using a greedy algorithm, and the last was using CPLEX to find the optimal solution. Under the same time window, the greedy algorithm was able to generate optimal solutions when iterated ten times, and the generated costs were 20%-53% lower than the approach taken in the study by Murray & Chu (Olivares, et al., 2015). Another article used the greedy algorithm to reduce the total time required to complete the route, and one unique aspect to this study was that the authors also incorporated the time associated with drone launch and recovery, and it also provided a sensitivity analysis based on drone speeds (Sacramento, et al., 2019). Despite the successful results of both articles, one issue that neither article accounts for is the case of a large TSP-D instance with 100+ nodes. Thus, the present study also aims to utilize a greedy algorithm paired with a SOM to solve the TSP-D, with an additional exception to also test the solution on a TSP-D instance with over 100 customer nodes.

Table 1. Literature review for the TSP-D

| Author (Year) | Vehicles | Drones | Objective | Algorithm |
|---|---|---|---|---|
| Murray and Chu (2015) | Single | Single | Time | C&W, NN, Sweep |
| Karak and Abdelghany (2019) | Multiple | Multiple | Cost | Hybrid C&W, VDH, DDH |
| Kitjacharoenchaia and Ventrescaa (2019) | Multiple | Multiple | Time | Adaptive Insertion Heuristic |
| Sacremento, Pisinger, Ropke (2019) | Multiple | Single | Cost | ALNS |
| Olivaresa, Cordovaa, Derpicha (2015) | None | Multiple | Time | GA |
| Liu (2019) | None | Multiple | Safety | MIP/CPLEX |
| Crisana and Nechita (2019) | Single | Single | Time | Greedy |
| Haa, Devillea, Phamb, Hac (2018) | Single | Single | Cost | Greedy |
| De Freitas and Vaz Penna (2018) | Single | Single | Time | RVND |
| Poikonena and Golden (2020) | Single | Single | Time | Route, Transform, Shortest Path |

## 3. Methodology

The model proposed in this study uses a self-organizing map (SOM) to determine an initial solution to the underlying TSP. Next, a greedy algorithm is used to determine ideal customers nodes to be serviced by the drone in order to reduce the total distance traveled by the vehicle. The data used for this study was retrieved from a National TSP library offered by the University of Waterloo. Since the data was formatted as a ".tsp" file, the first step was to convert the data to a workable file type. This was done in Python, using the Pandas and NumPy libraries to convert the file to a Python data frame. Next, since the TSP data is given in terms of x and y coordinates, the data was normalized in order to train the SOM network.

ISBN: 97819384961-9-6 

### 3.1 SOM Algorithm for the TSP

In order to determine the initial solution to the TSP, a SOM algorithm was used. The SOM algorithm works by creating a continuous chain of nodes called neurons. The recommendation made by previous literature was to generate a network larger than the population size. This is done in order to better capture the features of the optimal route as well as to reduce the model sensitivity to the initialized weights that are determined randomly (Bai, et al., 2006). The network size used for this study was eight times the population size, resulting in a SOM network consisting of 1,552 neurons.

The SOM network algorithm used in this study is shown in Equation 1. The equation calculates the weight updates for a neuron *n* at time *t+1* based on the initial weights of the neuron, the decaying learning rate $\alpha_t$, and a Gaussian neighborhood function $g$ based on the winning neuron $w_e$ as well as the neighborhood radius $r_t$. The learning rate is also multiplied by the Euclidean distance between the customer node element and the neuron *n* at time *t*.

$$n_{t+1} = n_t + \alpha_t \cdot g(w_e, r_t) \cdot \Delta(e, n_t) \quad (1)$$

Table 2. SOM model parameters

| SOM Parameter | Value |
|---|---|
| Learning Rate α | 0.9 |
| Learning Rate Decay | 0.003% |
| Network Size | 1552 |
| Neighborhood Radius | 155 |
| Radius Decay Rate | 0.03% |
| Stopping Parameters | α < 0.001 |
| | radius ≤1 |

### 3.2 SOM Algorithm for the TSP

After determining a feasible route for the vehicle, the next phase in the solution was determining the ideal candidates for drone insertion. Drone insertion refers to the process of dropping one of the nodes from the truck route and instead, adding it to the drone route. The objective of the algorithm was to determine which nodes if selected for drone delivery would most significantly reduce the total distance travelled by the truck. This process of inserting drone nodes was iterated until there were no more potential candidates for drone delivery in order to minimize the distance travelled by the truck.

The first phase in the drone insertion algorithm was listing all possible *sorties*. A sortie is a combination of three nodes (*i*,*j*,*k*). In a sortie, a truck equipped with a drone is said to visit node *i*. At this node *i*, the truck driver loads the drone with a package to be delivered to node *j*. After making its delivery at node *j*, the drone meets the truck at node *k* and the route is continued. In this process, the distance travelled by the truck is reduced because the truck does not have to travel from node *i* to *j* and then *k* and instead, the truck takes a shorter path between nodes *i* and *k*. The objective of the greedy insertion algorithm is to maximize this distance not traveled by the truck. This is done by listing all sorties along with the potential travel distance to be saved by the truck if the sortie is chosen. Next, the algorithm sorts the sorties based on maximum distance savings achieved by each sortie. The sortie with the highest potential distance reduction is accepted, which consequently makes the two sorties above and below the selected sortie infeasible. This is because if the sortie (*i*,*j*,*k*) is accepted, the drone is expected to make a delivery at node *j* and thus, the drone must be with the truck at nodes *i* and *k* since the drone is only allowed one delivery per sortie. This means that sorties (*h*,*i*,*j*) and (*j*,*k*,*l*) are eliminated from the list of potential sorties. This process is iterated until there are no more possible sorties left in order to maximize utilization of the drone and minimize the distance travelled by the truck. Although this algorithm is not guaranteed to result in an optimal solution, its simplicity provides a sub-optimal solution in a very short time frame.

ISBN: 97819384961-9-6

## 4. Experiment and Analysis

Figure 1 shows the iterative results of the TSP solution generated using the SOM. The red dots in the figures represent the 194 customer nodes and the blue loop consists of the 1,552 neurons. As the number of iterations progresses, the loop is shown to stretch and reach out to all of the customer nodes. The far-right panel in Figure 1 shows that the SOM learning process stopped after 24,487 iterations. The total computing time required generate this solution was approximately 25.8 seconds. It should also be mentioned that although the number of iterations was initialized to 100,000, the learning process stopped less than 25% of the way there. This is because the neighborhood decay function had reduced the neighborhood size from its initial length of 155 nodes to just one node, meaning that only the winning neuron would learn if iterated any further. This triggered the early stopping mechanism in the algorithm. The total length of the generated tour was 10,040, which is 7% higher than the optimal tour length of 9,352, as shown in Table 3.

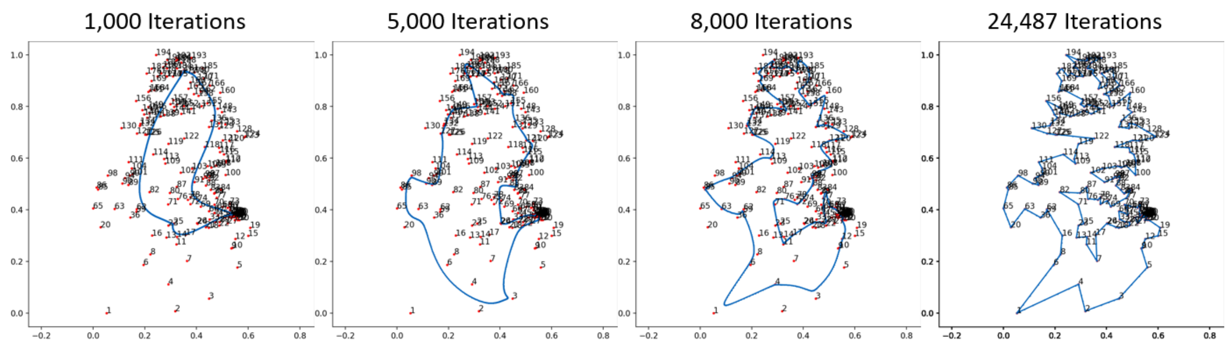


Figure 1. SOM Learning Progression

Table 3. Summary of Results

| Methodology | Value | Percent of Optimal |
|---|---|---|
| Optimal Value (Concorde) | 9352 | 100% |
| SOM Solution | 10040 | 107% |
| TSP-D Solution | 5672.8 | 61% |

Figure 2 shows the path traveled by the truck and drone pair. The path in purple is the reduced path to be traveled by the truck while the dotted green line tours represent each drone sortie. One thing that can be observed in the figure is that there are some missed opportunities for potential sorties that could have significantly reduced the travel distance for the truck, which is to be expected since the greedy algorithm rarely produces an optimal solution. Nonetheless, the solution generated using the greedy algorithm was calculated to be 39% lower than optimal TSP solution with no drones. An added benefit of using the greedy algorithm was that the computation time was extremely low at less than three seconds to generate and plot the route. Thus, the combined time required to compute the entire final solution was 28.8 seconds.

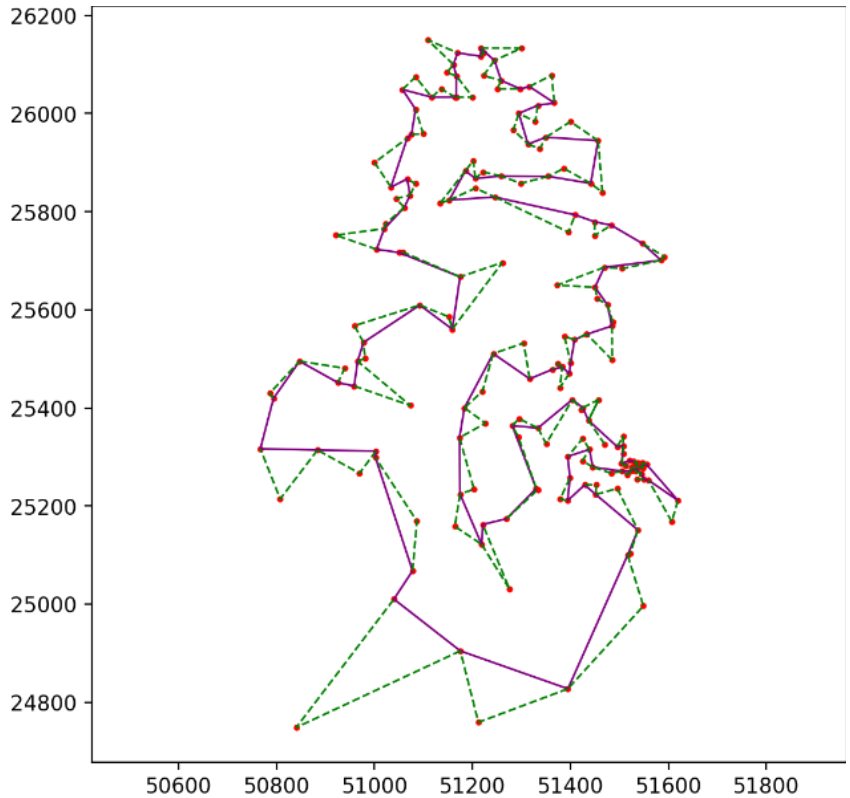


Figure 2. Optimized path for truck/drone pair

## 5. Conclusions and Future Work

In conclusion, the SOM algorithm was able to generate a 7% sub-optimal solution to a 194 node TSP in under half a minute. Additionally, the simple yet effective greedy insertion heuristic significantly reduced the total distance travelled by the truck by choosing locally optimal node candidates for drone delivery. The final solution to the large TSP-D was calculated in 28.8 seconds and yielded a result 39% lower than the optimal TSP solution, which suggests that pairing a drone with a delivery truck can help significantly reduce transportation distance and time if planned efficiently. One potential idea for future research is to run multiple experiments with the SOM algorithm while also changing the values of the neighborhood size, learning rate, and decay rates to determine the ideal network parameters for better TSP solutions. Additionally, other interesting avenues of research include adding realistic constraints such as packages that exceed the drone payload or flight radius and comparing the present results to results using two or more drones per vehicle.

## 6. References

Bai, Y., Zhang, W., & Jin, Z. (2006). An new self-organizing maps strategy for solving the traveling salesman problem. *Chaos, Solitons & Fractals*, *28*(4), 1082-1089.

Brocki, Ł., & Koržinek, D. (2007). Kohonen self-organizing map for the traveling salesperson problem. In *Recent Advances in Mechatronics* (pp. 116-119). Springer, Berlin, Heidelberg.

Burns, S. (2017). *Drone Meets Delivery Truck.* UPS, Retrieved 22 Feb. 2017 from www.ups.com/us/es/services/knowledge-center/article.page?kid=cd18bdc2.

Crişan, G. C., & Nechita, E. (2019). On a cooperative truck-and-drone delivery system. *Procedia Computer Science, 159*, 38-47.

de Freitas, J. C., & Penna, P. H. V. (2018). A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. *Electronic Notes in Discrete Mathematics, 66*, 95-102.

Garcia, S. (2019). *F.A.A. Allows U.P.S. to Deliver Medical Packages Using Drones.* The New York Times, The New York Times, Retrieved 2 Oct. 2019 from www.nytimes.com/2019/10/02/us/UPS-drone-deliveries.html

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. Transportation Research Part C: *Emerging Technologies, 86*, 597-621.

Karak, A., & Abdelghany, K. (2019). The hybrid vehicle-drone routing problem for pick-up and delivery services. Transportation Research Part C: *Emerging Technologies, 102*, 427-449.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering, 129*, 14-30.

Liu, Y. (2019). An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research, 111*, 1-20.

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: *Emerging Technologies, 54*, 86-109.

Olivares, V., Cordova, F., Sepúlveda, J. M., & Derpich, I. (2015). Modeling internal logistics by using drones on the stage of assembly of products. *Procedia Computer Science, 55*, 1240-1249.

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers & Operations Research, 113*, 104802.

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transportation Research Part C: *Emerging Technologies, 102,* 289-315.