Investigating Alternative Clustering Algorithms for a Cluster-First, Route-Second Truck and Drone Scheduling Algorithm

Sarah Dillon, Rudolf Ernst, and Jacomine Grobler(✉)

Department of Industrial Engineering, Stellenbosch University, Stellenbosch, South Africa
jacomine.grobler@gmail.com

Abstract. This paper investigates alternative clustering algorithms as part of a cluster-first, route-second method for solving a vehicle routing problem with drones. The problem consists of multiple purpose-built trucks, each having a single drone that can be launched to make deliveries, which set out to deliver products to customers. The drones are able to intercept and land on a moving truck after making a delivery, which speeds up the synchronisation and delivery time. Various different clustering algorithms are evaluated as part of a self-adaptive neighbourhood search differential evolution algorithm and applied to fifteen publicly available datasets, containing three types of customer distributions: double-centred, single-centred, and uniform. The customer nodes are segmented into clusters and each cluster is serviced by one truck and drone pair. Each cluster is solved as a travelling salesman problem with drone with interceptions using the nearest neighbour heuristic for initial solutions self-adaptive neighbourhood search differential evolution algorithm. A k-means clustering algorithm, the Gaussian mixture model, and agglomerative nesting are evaluated. From the study, it is conclusive that the Gaussian mixture model is the best clustering algorithm for double-centred datasets. The k-means algorithm is the best clustering algorithm for single-centred and uniform datasets. The contributions of this paper include the only cluster first-route second approach to solving the vehicle routing problem with drone with interceptions. This paper contributes to the computational sciences by using advanced computing capabilities for decision support to improve the performance of an intelligent transport system.

Keywords: Clustering · Drone delivery · Vehicle routing problem with drones

1 Introduction

Drone technology has been on the rise in recent years, and it continues to improve, which broadens the application of drones for personal and commercial use. Due to their unique capabilities, drones can play an essential role in a

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13957, pp. 147–162, 2023.
https://doi.org/10.1007/978-3-031-36808-0_10

variety of industries, including agriculture, security, distribution, and commercial delivery [1]. Last-mile delivery refers to the transportation of goods from the warehouse or depot to the final destination [2]. Despite the shorter delivery time expected by customers, the e-commerce industry has remained chiefly dependent on trucks for last-mile delivery operations [1]. Generally, products are housed in warehouses and transported to local depots. From the depots, the products are delivered to the customer's door. Trucks complete the movement of goods from the warehouses and the movement of goods from the depots to the customers, also known as the "last mile." However, due to increased customer expectations and the growth of the e-commerce industry, the traditional method of truck-only deliveries might not necessarily be the most optimal. In addition, an increase in traffic congestion and fuel prices affects the price that the customer pays for the delivery, which needs to be factored in along with the desired delivery speed [3].

One method to improve a last-mile delivery system is to include drones in the delivery process. Drones have the advantage of travelling faster than trucks and do not have to follow a fixed route, such as a road [4]. However, both the capacity and flight distance of drones are limited. If drones were launched directly from the depot, deliveries could not be made to customers outside of the drone's service range. The limitations mentioned above can be overcome by using a combination of trucks and drones. A *Drone-truck combined operations* (DTCO) system refers to a drone and truck working together to perform tasks [4].

Clustering refers to the grouping of elements that are of the same type or appear close together [5]. An optimally clustered group of orders will result in an effective delivery schedule. A truck and drone would service each cluster individually; therefore, each cluster is assigned a truck with a drone onboard. The truck would drive through the cluster and service specific customers, while the drone would assist with the other customers. Therefore, the overall delivery route distance that the truck travels is impacted by the effectiveness of the clustering algorithm applied. The DTCO system can significantly improve the last-mile delivery process by harnessing the advantages of both vehicles involved. The new system will optimise the delivery process by decreasing the delivery price and time, as efficient routes cannot be obtained without effective clustering.

In this paper, a *VRP with drones with interceptions* (VRPDi) is solved using a cluster-first, route-second approach. The results indicate that the k-means algorithm is best for clustering double-centred and larger uniform datasets, while the Gaussian mixture model is preferred for single-centred and smaller uniformly distributed datasets.

The paper is significant, since the clustering algorithm utilised in a cluster-first, route-second algorithm, has a marked impact on routing performance. Investigating the impact of alternative clustering algorithms has, however, not yet received priority in drone scheduling research.

The paper organisation is as follows: A literature review is presented in Sect. 2. The problem is described in Sect. 3. The empirical evaluation of the alternative clustering algorithms is documented in Sect. 4. A conclusion of the paper is presented in Sect. 5.

2 Literature Review

The clustering algorithms evaluated in this study are introduced in Sect. 2.2, along with information about where they have been applied to DTCO problems. Literature on DTCO problems is covered in Sect. 2.1. Next in Sect. 2.3, the nearest neighbour heuristic for initial solutions self-adaptive neighbourhood search differential evolution (NNHis SaNSDE) algorithm is introduced, which is used to schedule the truck and drone deliveries of each cluster.

2.1 DTCO Research

DTCO problems have been addressed in multiple survey papers. Otto et al. (2018) [6], Khoufi et al. (2019) [7], Chung et al. (2020) [4], Poikonen and Campbell (2020) [8]; Moshref-Javadi and Winkenbach (2021) [9], and Liang and Luo (2022) [10] all published survey papers where they addressed the most prominent and important research with regard to DTCO problems.

Research on where drones and trucks can intercept each other between customer nodes is sparse, as it is a new addition to current DTCO problem variants. Notable examples include Li et al. [11], Salama and Srinivas [12], and Thomas et al. [13], where drones can be dispatched and retrieved on the arc between two customers.

2.2 Clustering Literature

The k-means clustering algorithm is a centroid-based method for clustering a dataset with the objective of minimising the variance of the data points within each cluster [14]. The k-means clustering algorithm is commonly applied in a cluster-first, route-second approach when solving drone and truck scheduling problems. Ferrandez (2016) [15], investigated an optimisation algorithm that aimed to minimise the total delivery time of the DTCO system. The truck travelled through the centres of clusters that were obtained using the k-means clustering algorithm. Chang and Lee (2018) [16], proposed a new approach for routing the DTCO system, which used shift weights to move the cluster centres closer to the depot. The clusters used in this approach were determined using k-means.

Distribution-based clustering methods group data points based on the likelihood of the points belonging to the same probability distribution within the dataset [17]. Each cluster has a centre, and as the distance from that centre increases, the probability of a data point being a member of that cluster decreases [14]. The probability distribution applied in this paper is the *Gaussian* distribution (GMM) [18]. To the best of the author's knowledge, the Gaussian mixture model has not been used to perform the clustering of a cluster-first, route-second approach for solving DTCO problems.

Hierarchical-based clustering algorithms separate the dataset into clusters based on a measure of similarity. The objective is not to find the best cluster, but to merge the closest pair of data points or split the farthest pair [19].

agglomerative nesting (AGNES) is applied in this paper. Hierarchical clustering has frequently been applied to solve TSPs. Changyou Wu and Xisong Fu (2020) [20] applied an *agglomerative greedy brainstorm optimization* (AG-BSO) algorithm to a TSP. The algorithm applied a greedy algorithm to the TSP and replaced the traditional k-means clustering algorithm with the AG-BSO algorithm. Marco Lama and John Mittenthal (2013) [21], proposed a hierarchical clustering-based heuristic for a multi-depot location-routing problem. Abdulah Fajar (2011) [22] applied hierarchical clustering to a Euclidean TSP. To the best of the author's knowledge, hierarchical clustering has not been used to perform the cluster-first, route-second approach for solving DTCO problems.

2.3 NNHis SaNSDE Algorithm

The NNHis SaNSDE algorithm is a population-based metaheuristic applied in this paper to determine the routing and allocation of nodes to drones and trucks after clustering the data. The NNHis SaNSDE was introduced by Ernst et al. (2023) [23], where it was applied with great success to a TSP with drones with interceptions. The NNHis SaNSDE built on the SaNSDE-based algorithm introduced by Yang et al. [24], combined with problem-specific nearest neighbour heuristics that introduced good solutions into the initial population to guide the search process.

By introducing several good initial solutions into the starting population, it was possible for the NNHis SaNSDE to improve these solutions and find even better routes with shorter delivery times in the neighbourhood of the solutions. The six initial solutions introduced into the first population of the NNHis SaNSDE are explained below:

- The first solution is generated using a *nearest neighbour heuristic* (NNH). The depot is considered the first node. The node with the smallest distance from the depot is chosen as the second node to visit. Thereafter, the node closest to the second node is allocated as the next stop. This process is continued until all nodes have been allocated and the final node is again assigned as the depot. All nodes are to be visited by the truck, and the drone is stationary.
- The second solution is generated using two parallel NNHs. Two different routes are generated, both starting from the depot. After allocating all nodes, the second route is reversed and connected onto the end of the first half of the route to ensure that the depot is visited first and last. Again, all nodes are visited by a truck.
- The third solution is generated using two parallel NNHs, as in solution 2. However, in the third solution, nodes are continuously added to either route depending on which endpoint is nearest to any unallocated node. The second route is again reversed after all nodes are allocated and connected to the first route to construct one full route where the depot is visited at the start and end of the route with truck-only nodes.

- The fourth solution is generated using an *alternating nearest neighbour heuristic* (ANNH). The first node, which is the depot, is allocated to the truck. Thereafter, the node with the smallest distance to the first node is allocated to the drone. The nearest node to the drone node is then chosen and allocated to the truck. This procedure is continued until all the nodes have been allocated, and the last node is then again allocated as the depot.
- The fifth solution is generated using two parallel ANNHs, combining solutions two and four, by alternating between allocating truck and drone nodes. This process is continued until all nodes have been added. The second half of the route is again reversed and added onto the end of the first half of the route to ensure that the depot is visited first and last.
- The sixth solution is generated similarly to the third solution, except that nodes are now alternated between the truck and the drone (thus combining solutions three and four).

The SaNSDE NNHis makes use of decision variables, which are stochastic continuous values between a an upper bound, u_b, and lower bound, l_b, which are converted to determine the allocation and routing of solutions. The SaNSDE alters these decision variable values as it moves through the search space. Each solution contains 2 * n decision variable values, where n is the number of customer nodes to be visited. The first n decision variables, called priority values, are sorted in descending order, which then gives the permutation or order that nodes are visited. The second n decision variables, called allocation variables, are used for allocating nodes. Any decision variable value bigger than or equal to zero is allocated to the truck, while values smaller than zero are allocated to the drone, thereby discretising the search space.

One of the restrictions of the VRPDi model is that only one drone delivery is allowed per drone flight, whereafter the drone has to return to the truck. The continuous values, which are used to determine the routing and allocation of nodes, can often lead to solutions that have multiple consecutive drone nodes. Consecutive drone nodes are not allowed, and thus repair mechanisms are used to turn these infeasible solutions into valid ones. The repair mechanism changes the second consecutive drone node into a truck node by changing the decision variable value to $u_b/2$ (from the negative number it was before). The repair process ensures that the previously infeasible nodes will now be visited by a truck node, but also that the SaNSDE's search process can still change the decision variable values to find different solutions.

3 Problem Description

The VRPDi is based on a set of trucks and drones that work together to complete the last-mile delivery of a predetermined number of packages. A set of trucks and drones depart from the depot simultaneously. Each of the drones can be launched from its corresponding stationary truck at any of the nodes. Once the drone has completed a delivery, it either intercepts the truck while it is in motion or flies

straight to the next truck node if an interception is not possible due to the truck arriving at the next node before the drone is able to intercept it. This model seeks to optimise the drone's flying time since the drone attempts to not have to fly to the next customer node. Rather, the drone can fly to a predetermined optimal point where it intercepts the truck, resulting in less waiting time. After an interception, the drone remains on the truck until the truck reaches the next customer node. Once the truck has reached the next customer, the drone can again be launched to complete another delivery. The model explanation follows as provided in [25].

- n is the number of customers to be serviced.
- T_{ijk} denotes the total time for the scenario where a drone delivers to node j while the truck delivers to nodes i and k. T_{ijk} is calculated based on the drone and truck intercepting each other after the drone delivery. This interception point is calculated using the standard mathematical equations for two objects interceptions explained in [25].
- T_{ik} denotes the total time for the scenario where the truck delivers to nodes i and k when there is not a drone delivery in between.
- w is the drone offload time

With the following decision variables:

- u_i is used to eliminate sub-tours.

$$x_{ijk} = \begin{cases} 1 & \text{if the truck services customer } i \text{ before } k \text{ while the drone services } j \text{ in between} \\ 0 & \text{otherwise} \end{cases}$$

$$b_j = \begin{cases} 1 & \text{if the customer } j \text{ is serviced by the drone} \\ 0 & \text{if customer } j \text{ is serviced by the truck} \end{cases}$$

$$z_{ik} = \begin{cases} 1 & \text{if the truck travels between customer } i \text{ and } k \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n} x_{ijk}T_{ijk} + \sum_{i=1}^{n}\sum_{k=1}^{n} z_{ik}T_{ik}(1 - \sum_{j=1}^{n} x_{ijk}) - w \tag{1}$$

Subject to:

$$z_{ii} = 0 \qquad \forall i \in \{1, ..., n\} \tag{2}$$

$$x_{iii}, x_{iij}, x_{iji}, x_{ijj} = 0 \qquad \forall i, j \in \{1, ..., n\} \tag{3}$$

$$z_{ik} = (\sum_{j=1}^{n} x_{ijk} = 1) => (z_{ik} = 1) \qquad \forall i, k \in \{1, ..., n\} \tag{4}$$

$$b_j = \sum_{i=1}^{n}\sum_{k=1}^{n} x_{ijk} \qquad \forall j \in \{1, ..., n\} \tag{5}$$

$$\sum_{i=1}^{n} z_{iq} - \sum_{k=1}^{n} z_{qk} = 0 \qquad \forall q \in \{1, ..., n\} \tag{6}$$

$$(b_i = 0) => (\sum_{k=1}^{n} z_{ik} = 1) \qquad \forall i \in \{1, ..., n\} \tag{7}$$

$$(b_k = 0) => (\sum_{i=1}^{n} z_{ik} = 1) \qquad \forall k \in \{1, ..., n\} \tag{8}$$

$$\sum_{j=1}^{n} x_{ijk} \leq 1 \qquad \forall i, k \in \{1, ..., n\} \tag{9}$$

$$u_i + z_{ij} \leq u_j + (n-1)(1 - z_{ij}) \qquad \forall i, j, j \neq 1 \in \{1, ..., n\} \tag{10}$$

$$u_1 = 0 \tag{11}$$

$$x_{ijk}, b_i, z_{ik} \in \{0, 1\} \qquad \forall\, i, j, k \in \{1, ..., n\} \tag{12}$$

- The objective function (1) minimises the total time that the drone and truck travel, which includes the travel time of all truck deliveries which do not include a drone release and intercept.
- Constraints (2) and (3) simplify the model by ensuring that the truck and drone cannot travel from the current node back to a previously visited node.
- Constraints (4) links z_{ik} and x_{ijk}.
- Constraints (5) links b_j and x_{ijk}.
- Constraints (6) ensure that if a delivery is conducted at a customer location, the truck or drone leaves that same customer.
- Constraints (7) and (8) ensure that each customer is arrived at from exactly one other customer (by truck), and that from each customer there is a departure to exactly one other customer (by truck). If node j is serviced by a drone, then no trucks enter or exit node j.
- Constraints (9) ensure that only one drone delivery can occur between two truck deliveries.
- Constraints (10) and (11) eliminate any sub-tours.
- Constraints (12) restricts the decision variables to binary values.

4 Empirical Evaluation

The VRPDi is solved by first clustering the customer nodes and then routing each cluster using a *Nearest Neighbour Heuristic for initial solutions Self-adaptive Differential Evolution with Neighbourhood Search* (NNHis SaNSDE) algorithm. A single truck and drone are assigned to each cluster. The clustering is performed on multiple different datasets from Bouman [26], with varying distributions. Single-centred datasets contain a central location where most nodes are scattered around it; double-centred datasets contain two centres with nodes scattered around both; and uniform datasets contain uniformly scattered nodes and no centre.

Each clustering algorithm is applied once per dataset, whereafter the NNHis SaNSDE is applied to each cluster for 30 optimisation runs. The resulting delivery times are then taken as the average over the 30 optimisation runs, which are reported in this section. The boundary values of u_b and l_b are chosen as –100 and 100.

Tables 1, 2, 3 provide a summary of the most important results. Tables 1, 2, and 3 present the average delivery time of the cluster with the longest delivery time for each dataset-clustering algorithm pair. The longest average delivery time from the clusters represents the overall average delivery time for the truck and drone systems to service the dataset, since the longest average delivery time is the limiting time.

4.1 Double-Centred Datasets Results

Examples of the final clusters for the datasets of size 250 are plotted in Fig. 1, 2, Fig. 3. The cluster size distribution for the three clustering algorithms was relatively even due to the clear separation of the double-centred datasets. While this trend is not present in the smaller datasets, the larger the dataset, the more apparent the trend becomes.

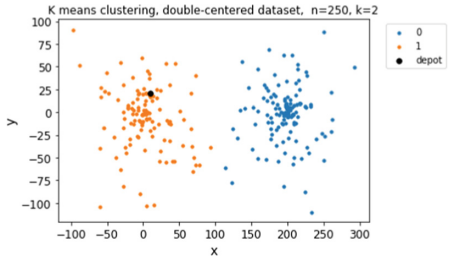

Fig. 1. k-means doublecenter-3-n250 clusters

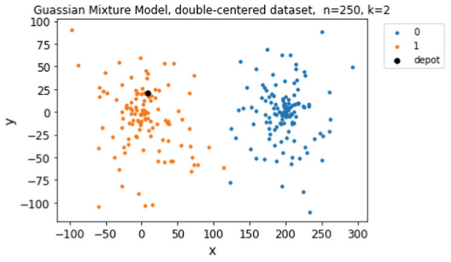

Fig. 2. GMM doublecenter-3-n250 clusters

The average delivery times for the double-centred datasets are presented in Table 1. The Gaussian mixture model resulted in the shortest average delivery time for each of the double-centred dataset sizes. The k-means clustering and AGNES algorithms resulted in longer average delivery times. The k-means clustering algorithm and AGNES obtained similar results for most datasets, but the AGNES algorithm performed the worst for the doublecenter-5-n500 by a significant margin of almost 30 time units.

4.2 Single-Centred Datasets Results

Examples of the final clusters for the datasets of size 250 are plotted in Fig. 4, 5, Fig. 6, showing the distributions of the different clustering algorithms. The

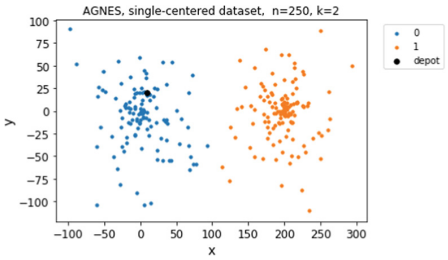

Fig. 3. AGNES doublecenter-3-n250 clusters

Table 1. Double-centred dataset results

Name	Clustering algorithm	k	Average delivery time
doublecenter-alpha3-61-n20	k-means	2	57.32
	Gaussian Mixture Model	2	**57.29**
	AGNES	2	57.32
doublecenter-91-n100	k-means	2	96.91
	Gaussian Mixture Model	2	**94.52**
	AGNES	2	99.36
doublecenter-3-n250	k-means	2	173.84
	Gaussian Mixture Model	2	**166.12**
	AGNES	2	173.69
doublecenter-3-n375	k-means	2	243.40 7
	Gaussian Mixture Model	2	**214.52**
	AGNES	2	243.40
doublecenter-5-n500	k-means	2	240.99
	Gaussian Mixture Model	2	**240.86**
	AGNES	2	270.65

clusters are spread fairly evenly in terms of the number of customers per cluster, with a clear central cluster and three spread-out clusters around it.

The average delivery times for the single-centred datasets are presented in Table 2. For the single-centred datasets, the k-means clustering algorithm resulted in the shortest average delivery time for the datasets of sizes 100, 250, 375, and 500. For the size 20 dataset, the Gaussian mixture model resulted in the shortest average delivery time, and the AGNES algorithm resulted in the shortest total truck distance. However, despite the short truck distance, the AGNES algorithm also resulted in the longest average delivery time.

Figs. 7, 8, 9 present the average delivery time per cluster for the singlecenter-6-n500 dataset. The graphs show that the average delivery time is highly consistent across the clusters for the k-means clustering algorithm. The variance for the k-means clustering algorithm's average delivery times is 124.04. The Gaussian mixture model and AGNES algorithm have greater variances of 482.37 and

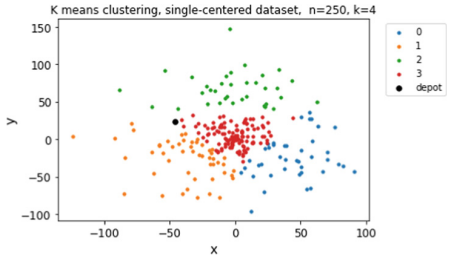

Fig. 4. k-means singlecenter-1-n250 clusters

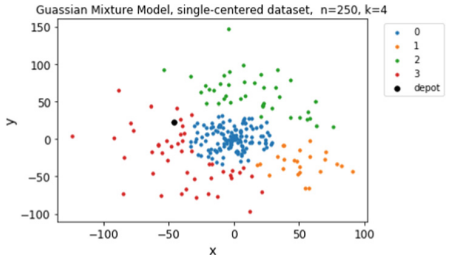

Fig. 5. GMM singlecenter-1-n250 clusters

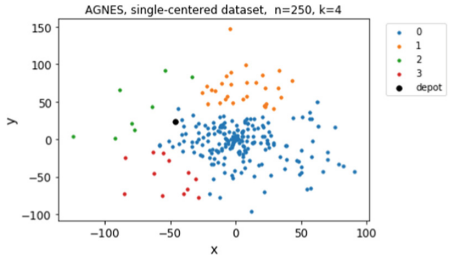

Fig. 6. AGNES singlecenter-1-n250 clusters

1149.40, respectively. Therefore, the k-means clustering algorithm results in the most uniform average delivery time distribution when the NNHis SaNSDE is applied to each cluster.

4.3 Uniform Dataset Results

Examples of the final clusters for the datasets of size 250 are plotted in Figs. 10 and 12. The clusters obtained by k-means and GMM are very similar, while the clusters obtained by AGNES have a much larger variance in size and an average number of nodes per cluster. The average delivery times for the uniform datasets are presented in Table 3.

The uniform datasets do not appear to have one clustering algorithm that tends to deliver a shorter average delivery time or total truck distance. The k-means clustering algorithm has the best performance on the larger datasets, while the AGNES algorithm greatly outperformed the other algorithms on the uniform-62-n20 dataset.

Figs. 13, 14, 15 show the cluster average delivery time distribution of the uniform-8-n500 dataset. The most uniform average delivery time distribution is observed for the k-means clustering algorithm. The variance in average delivery time over all clusters for the k-means clustering algorithm is 29.25. The variances for the Gaussian mixture model and the AGNES algorithm's average delivery

Table 2. Single-centred dataset results

Name	Clustering algorithm	k	Average delivery time
singlecenter-61-n20	k-means	4	24.79
	Gaussian mixture model	4	**24.61**
	AGNES	4	29.23
singlecenter-100-n100	k-means	4	**36.84**
	Gaussian mixture model	4	47.22
	AGNES	4	47.64
singlecenter-1-n250	k-means	4	**59.46**
	Gaussian mixture model	4	64.35
	AGNES	4	131.24
singlecenter-1-n375	k-means	4	**62.17**
	Gaussian mixture model	4	70.01
	AGNES	4	143.52
singlecenter-6-n500	k-means	5	**77.49**
	Gaussian mixture model	5	101.35
	AGNES	5	109.18

times are 151.34 and 174.57, respectively. The trend is observed across all the dataset sizes for the k-means clustering algorithm.

Next, the results of the VRPDi study are validated in the form of Mann-Whitney U statistical tests. The statistical tests compare the delivery time results of the different clustering algorithms for a statistical difference with a significance level of 0.05. The results are measured as wins-draws-losses and presented in Table 4. A win (or loss) indicates the number of times each algorithm statistically significantly differs in average delivery time when compared.

It is clear from Table 4 that no clustering algorithm, after a NNHis SaNSDE is applied to solve each cluster, significantly outperforms the other clustering



Fig. 7. k-means sc-6-n500 delivery time distribution



Fig. 8. Gaussian sc-6-n500 delivery time distribution

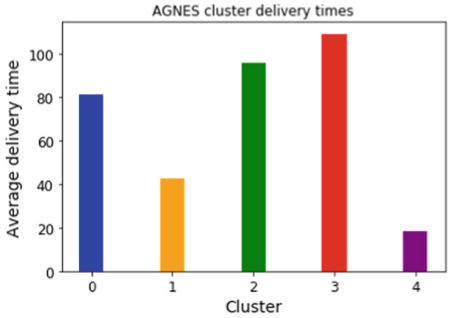

Fig. 9. AGNES sc-6-n500 delivery time distribution

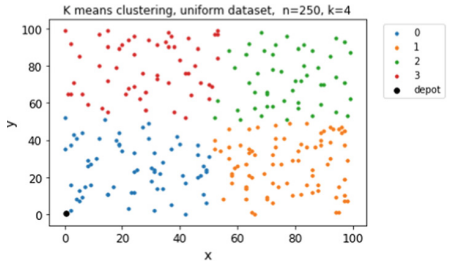

Fig. 10. k-means uniform-8-n250 clusters

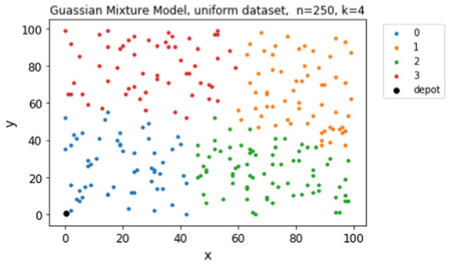

Fig. 11. GMM uniform-8-n250 clusters

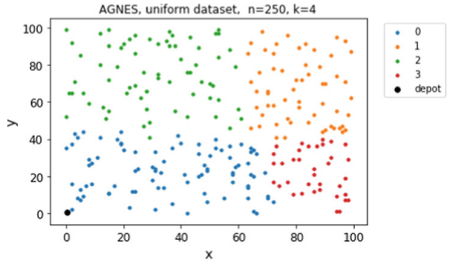

Fig. 12. AGNES uniform-8-n250 clusters

algorithms when the three distributions are combined. Therefore, in order to choose an algorithm that will have a higher statistical chance of performing well, it is necessary to know the distribution of the dataset.

Table 3. Uniform dataset results

Name	Clustering algorithm	k	Average delivery time
uniform-62-n20	k-means	4	21.74
	Gaussian Mixture Model	4	19.50
	AGNES	4	**15.80**
uniform-91-n100	k-means	4	49.91
	Gaussian Mixture Model	4	**48.08**
	AGNES	4	48.66
uniform-8-n250	k-means	4	49.91
	Gaussian Mixture Model	4	**48.09**
	AGNES	4	48.67
uniform-1-n375	k-means	4	**57.43**
	Gaussian Mixture Model	4	60.65
	AGNES	4	66.33
uniform-8-n500	k-means	4	**66.62**
	Gaussian Mixture Model	4	70.51
	AGNES	4	81.03

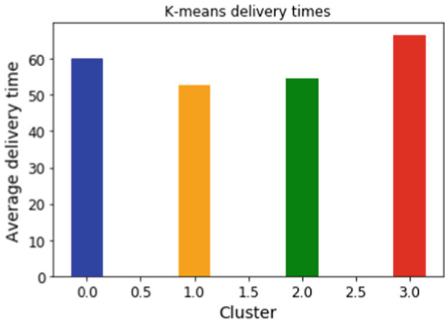

Fig. 13. k-means u-8-n500 delivery time distribution

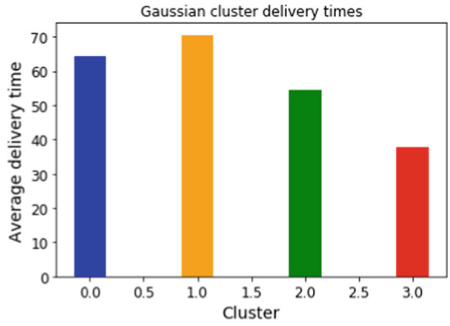

Fig. 14. Gaussian u-8-n500 delivery time distribution

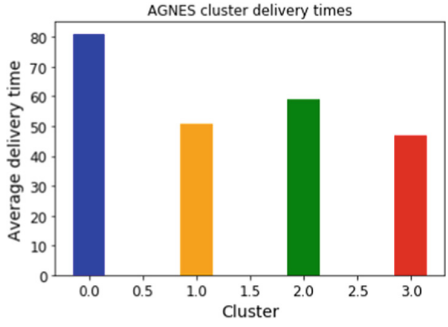

Fig. 15. AGNES u-8-n500 delivery time distribution

Table 4. Combined hypothesis test results

	k-means	Gaussian mixture model	AGNES
k-means	X	7-2-6	6-2-7
Gaussian mixture model	6-2-7	X	6-3-6
AGNES	7-2-6	6-3-6	X

5 Conclusion

This paper considered a VRP with drones with interceptions [25], where a cluster-first, route-second approach was used to solve the last-mile delivery problem. K-means, a Gaussian mixture model, and the AGNES algorithm were applied to cluster customers from pre-determined datasets. Each dataset was then solved in the form of a TSP, where a truck with a drone that can be launched from it to make deliveries was applied to each cluster. Each cluster was routed with the NNHis SaNSDE algorithm [23]. The results indicated that the distribution of the dataset is an important factor when deciding which clustering algorithm to apply. The Gaussian mixture model performed the best for double-centred datasets. The k-means algorithm significantly outperformed the other clustering algorithms for single-centred datasets. The k-means algorithm performed the best for larger, uniformly distributed datasets but was outperformed by the Gaussian mixture model and AGNES for smaller, uniform datasets.

In this paper, only three clustering algorithms were investigated and tested. Further research can investigate clustering algorithms that were not considered. To allow for direct comparison between the clustering algorithms, the elbow method was used to solve for the k-value for all the clustering algorithms. However, there are other methods that can be used to solve for the optimal k-value. It is possible that a different method, such as the silhouette metric, could provide a better solution. Future work can thus consider applying different methods for determining the k-value. Different algorithms can also be applied instead of the NNHis SaNSDE algorithm when routing each of the clusters. Finally, the realism of the model can be improved to, for example, take into account weather conditions and traffic congestion.

Acknowledgements. This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number: 129340).

References

1. Salama, M., Srinivas, S.: Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. Transp. Res. Part C: Emerg. Technol. **114**, 620–642 (2020)
2. DoveTail. 5 strategies for effective last-mile delivery logistics. https://www.dovetail.co.za/5-strategies-for-effective-last-mile-delivery-logistics/

3. Francois Emil Knoetze. Solving a last mile truck and drone delivery schedule to optimality (2021)
4. Chung, S.H., Sah, B., Lee, J.: Optimization for drone and drone-truck combined operations: a review of the state of the art and future directions. Comput. Oper. Res. **123**, 105004 (2020)
5. Simpson, J.A., Weiner, E.S.C.: Oxford English Dictionary. Claredon Press, Oxford (1989)
6. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. Networks **72**(4), 411–458 (2018)
7. Khoufi, I., Laouiti, A., Adjih, C.: A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. Drones **3**(3), 66 (2019)
8. Poikonen, S., Campbell, J.F.: Future directions in drone routing research. Networks **77**(1), 116–126 (2021)
9. Moshref-Javadi, M., Winkenbach, M.: Applications and research avenues for drone-based models in logistics: a classification and review. Expert Syst. Appl. **177**, 114854 (2021)
10. Liang, Y.J., Luo, Z.X.: A survey of truck-drone routing problem: literature review and research prospects. J. Oper. Res. Soc. China **10**, 1–35 (2022)
11. Li, H., Chen, J., Wang, F., Zhao, Y.: Truck and drone routing problem with synchronization on arcs. Naval Res. Logist. (NRL) **69**(6), 884–901 (2022)
12. Salama, M.R., Srinivas, S.: Collaborative truck multi-drone routing and scheduling problem: package delivery with flexible launch and recovery sites. Transp. Res. Part E: Logist. Transp. Rev. **164**, 102788 (2022)
13. Thomas, T., Srinivas, S., Rajendran, C.: Collaborative truck multi-drone delivery system considering drone scheduling and EN-route operations. Available at SSRN 4080821
14. McGregor, M.: 8 clustering algorithms in machine learning that all data scientists should know. https://www.freecodecamp.org/news/8-clustering-algorithms-in-machine-learning-that-all-data-scientists-should-know/
15. Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R., Rich, R.: Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J. Ind. Eng. Manag. (JIEM) **9**(2), 374–388 (2016)
16. Chang, Y.S., Lee, H.J.: Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst. Appl. **104**, 307–317 (2018)
17. Prasad, S.: Different types of clustering methods and applications. https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms
18. Delua, J.: Unsupervised learning: what's the difference? https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning
19. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: a new data clustering algorithm and its applications. Data Min. Knowl. Disc. **1**(2), 141–182 (1997)
20. Changyou, W., Xisong, F.: An agglomerative greedy brain storm optimization algorithm for solving the tsp. IEEE Access **8**, 201606–201621 (2020)
21. Lam, M., Mittenthal, J.: Capacitated hierarchical clustering heuristic for multi depot location-routing problems. Int. J. Logist. Res. Appl. **16**(5), 433–444 (2013)
22. Fajar, A., Herman, N.S., Abu, N.A., Shahib, S.: Hierarchical approach in clustering to Euclidean traveling salesman problem. In: Shen, G., Huang, X. (eds.) ECWAC 2011. CCIS, vol. 143, pp. 192–198. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20367-1_30

23. Ernst, R., Grobler, J., Moremi, T., Knoetze, F., Knoetze, F.: Differential evolution algorithms for the travelling salesman problem with drones with interceptions. Submitted Comput. Ind. Eng. (2023)
24. Yang, Z., Tang, K., Yao, X.: Self-adaptive differential evolution with neighborhood search. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 1110–1116. IEEE (2008)
25. John, M.T., et al.: An Ant Colony Optimisation Approach to Scheduling Truck and Drone Delivery Systems. PhD thesis, Stellenbosch University (2022)
26. Bouman, A.N.P., Schmidt, M.: Instances for the tsp with drone, April 2015. https://dx.doi.org/10.5281/zenodo.22245