Algorithms for Solving the Vehicle Routing Problem with Drones

Daniel Schermer, Mahdi Moeini(✉), and Oliver Wendt

BISOR, University of Kaiserslautern, Postfach 3049,
Erwin-Schrödinger-Str., 67653 Kaiserslautern, Germany
{daniel.schermer,mahdi.moeini,wendt}@wiwi.uni-kl.de

Abstract. The *Vehicle Routing Problem (VRP)* and its variants are well-studied problems in Operations Research. They are related to many real-world applications. Recently, several companies like Amazon, UPS, and Deutsche Post AG showed interest in the integration of autonomous drones in delivery of parcels. This motivates researchers to extend the classical VRP to the *Vehicle Routing Problem with Drones (VRPD)*, where a drone works in tandem with a vehicle to reduce delivery times. In this paper, we focus on solving the VRPD. In particular, we introduce two heuristic algorithms for solving this problem and, through numerical experiments on large-scale instances, we evaluate the performance of the heuristics and show the potential benefit that can be expected when using drones.

Keywords: Last-mile Logistics · Vehicle Routing Problem
Drone Delivery · Large-scale Instances · Heuristic

1 Introduction

The *Unmanned Aerial Vehicles (UAVs)*, i.e., drones, have started to play an increasing role in delivery applications from the point of view of both, researchers and large companies, such as Amazon Inc., Deutsche Post AG, UPS, and Google (see e.g., [7] and references therein).

The UAVs are not considered as an alternative to the conventional delivery vehicles such as truck, but rather as complementary delivery tools. Table 1 highlights the complementary features of trucks and drones for delivery applications. Since drones are not restricted to the road network or congestion, they can generally move, between two locations, faster than trucks. Furthermore, drones and their payload are far more lightweight than trucks', which causes drones to consume much less energy for the movement between two points. However, a drone's carrying capacity is typically limited to one or few parcels. Furthermore, since drones rely on comparatively small batteries for powering their flight, their range is somewhat limited compared to a truck using a fossil fuel.

The *Vehicle Routing Problem (VRP)* is one of the most well-studied problems in operations research. As a generalized case of the well known *Traveling*

N. T. Nguyen et al. (Eds.): ACIIDS 2018, LNAI 10751, pp. 352–361, 2018.
https://doi.org/10.1007/978-3-319-75417-8_33

Table 1. Qualitative differences between trucks and drones.

	Speed	Weight	Capacity	Range	Energy consumption
Truck	Low	Heavy	Many	Long	High
Drone	High	Light	One	Short	Low

Salesman Problem (TSP), for a given fleet of vehicles and a set of customers, the VRP seeks for the optimal set of routes in order to deliver goods to the customers and satisfy their demand. In [9], Toth and Vigo provide an extensive overview of the problem and its variants.

In the academic literature, the interest in integrating drones in delivery applications has surged, following Murray and Chu, who introduced two new NP-hard problems related to deliveries with a vehicle working in tandem with a drone [6]. In their *flying sidekick traveling salesman problem (FSTSP)*, a drone travels along with a vehicle, both of which start from a common *distribution center (DC)* and must return to the same DC at the end of the tour. At any customer, the drone might be launched from the vehicle, starting a sortie. In this case, it will begin delivery to a customer and will then rendezvous with the vehicle at a later customer. In the *parallel drone scheduling TSP (PDSTSP)*, it is assumed that the DC is in close proximity to most of the customers. In this case, it can be beneficial to let the drone make its deliveries independently from the vehicle. In both problems, the objective consists in minimizing the time required to serve all customers and the drone and vehicles must return to the DC. Murray and Chu proposed Mixed Integer Linear Programs (MILP) as well as two simple heuristic methods for solving the FSTSP and PDSTSP, respectively. They note, that only small size instances can be solved up to optimality due to the NP-hard nature of the proposed problems. Later, Ha et al. [2] employed two heuristics to solve the FSTSP: *route first, cluster second* and *cluster first, route second*. The latter heuristic performed better according to their numerical results.

Mathew et al. [4] introduced the *Heterogeneous Delivery Problem (HDP)* which shares most features of the previously introduced FSTSP. They propose a solution by reducing the HDP to the *Generalized Traveling Salesman Problem (GTSP)*. They suggest splitting the HDP into two traceable sub-problems: first a TSP is used to generate the optimal tour, then convex optimization is applied to compute the specific deployment points for the drone and vehicle.

Mourelo et al. [5] worked on a variant of the FSTSP, where they allow multiple drones per vehicle. They introduced a heuristic based on *K-means* for clustering purposes and a *Genetic Algorithm* for routing. They conducted their computational experiments taking into account various drone speeds and they noticed that the drones should be at least twice as fast as the vehicles in order to allow for *significant* improvements.

Wang et al. [12] introduced the *Vehicle Routing Problem with Drones (VRPD)*, where a fleet of trucks, each truck equipped with a given number of drones, delivers packages to customers. According to the classification of Toth and Vigo [9], the VRPD can be classified as a variant of the *Distance-Constrained Capacitated Vehicle Routing Problem (DCVRP)* with a set of heterogeneous vehicles. The objective consists in minimizing the completion time, i.e., the time required to serve all customers and return the fleet to the depot. Wang et al. introduced several lower bounds on the amount of time that can be saved by employing drones. The lower bounds depend on the relative speed of the drones compared to the vehicles and the number of drones per vehicle.

Ulmer and Barrett proposed the *same-day delivery routing problems with heterogeneous fleets (SDDPHF)*, where drones and vehicles serve customers without a need for synchronization [10]. The customers are not known a priori; hence, dynamic routing of the heterogeneous fleet based on the incoming requests is required. Therefore, the previously introduced PDSTSP, where all customers are known a priori, can be viewed as a relaxed version of the SDDPHF. Using computational experiments, Ulmer et al. can show that the combination of vehicles and drones can reduce the delivery costs significantly.

Jiang et al. [3] applied drone logistics to the *Vehicle Routing Problem with Time Windows (VRPTW)* and derived an MILP model. Their problem consists of assigning a swarm of drones to serve customers within predefined time frames. They used an adapted version of *Particle Swarm Optimization* for drone routing. They conclude that their implemented heuristic is well suited for solving their version of the VRPTW.

In this article, we focus on the VRPD introduced by Wang et al. [12]. Our contributions are twofold: first, we introduce two heuristics for solving VRPD. Second, in order to assess the performance of the implemented heuristics, we carried out numerical experiments on *large-scale* TSP instances that we adjusted for using in the context of the VRPD. On the one hand, our numerical results highlight usefulness of incorporating drones in last-mile logistics. On the other hand, based on our experiments, we can derive implications for the way future heuristics should be designed when solving the large-scale VRPD instances.

This paper is organized as follows. Section 2 introduces the notation and describes the VRPD. In Sect. 3, we present our heuristics for solving the VRPD. Section 4 is dedicated to the computational experiments and the numerical results. Finally, we draw some concluding remarks and derive some future research questions in Sect. 5.

2 The Vehicle Routing Problem with Drones

Suppose that a set of n customers and a fleet of m homogeneous trucks, each carrying k homogeneous drones, are given. We might assume that the capacity of each truck is C. The *Vehicle Routing Problem with Drones (VRPD)* looks for minimizing the maximum completion time, i.e., the time required to serve all customers using the trucks and the drones such that, by the end of mission all trucks (carrying drones as well) must be at the depot [7,12]. By t^T_{n+1}, we denote the objective value of the VRPD solution indicating the time that the set of trucks and drones needs for serving all customers and then, returning to the depot. This time corresponds to the mission time of the latest truck or drone that arrives to the depot, as soon as all customers have been served by the set of trucks and drones (other trucks and/or drones arrive ahead of this time). In t^T_{n+1}, T stands for the *total* time and $n+1$ indicates that the solution includes the set of n customers and the depot.

Furthermore, we make the following assumptions about the drone's behavior in a risk-free environment [6,7,12]:

- A drone can carry exactly one parcel when airborne.
- A drone has a limited battery life of $\mathcal{E}$ time units. After returning to the truck, the battery life of the drone is recharged instantaneously with no service delay.
- The trucks and drones follow the same distance metric.
- The service time to launch and reunite with a drone is assumed to be negligible.
- The service time required to serve a customer is assumed to be negligible.
- Without loss of generality, the speed of each truck is set to 1 and the speed of each drone is assumed to be α times the speed of the truck.
- A drone must always return to the truck from which it was launched.
- Drones may only be dispatched and picked up from vertices, i.e., at the depot or any other customer location. Furthermore, a drone may not be picked up from the same vertex, where it has been launched from.
- Since the drones are assumed to be in constant flight and can not conserve battery while in flight; consequently, if a truck arrives earlier to a pick-up node, then the truck has to wait for its corresponding drone.
- We assume that when a drone is launched, then its delivery will be successful.

3 Algorithms for Solving the VRPD

In this section, we present two heuristic algorithms that we introduce for solving the VRPD. The heuristics are composed of several components; in particular, they have two main stages: an initialization step and an improving (optimization) phase.

Indeed, as a natural approach, we need an initial solution in order to start the optimization phase. For this purpose, we choose a *route-first cluster-second (RFCS)* heuristic. The RFCS heuristics has different components. First, we create a single tour using the *nearest neighbor heuristic (NHH)* [8]. Afterwards, having created a single tour that contains all vertices, the tour is split equally in m segments, where m is the number of available trucks.

The improvement phase of the heuristics is inspired from the VRP literature. In fact, improvement heuristics for the VRP can be categorized into *single-route improvement* and *multi-route improvement* heuristics [9]. Single-Route improvements focus on improving a single tour at a time. Multi-Route improvements, on the other hand, try to improve the objective function by considering multiple distinct tours simultaneously. In [11], van Breedam has classified the multi-route improvement operations into *String Cross (SC)*, *String Relocation (SR)*, *String Exchange (SE)*, and *String Mix* (SM), where *string* stands for a tour. While SM is a combination of SE and SR, SC is a generalization of the k-opt operator [1].

Since the VRPD imposes additional constraints, that do not allow for a direct application of the introduced single- and multi-route improvement operators, we introduce a variation of the 2-opt and SM operators with the aim of handling the additional constraints imposed by the VRPD. More precisely, the launch and rendezvous vertices must be updated when inverting a sub-sequence of the tour in order to maintain synchronization of the truck and the drone. Additionally, endurance constraints must not be violated when applying the 2-opt or any other operator.

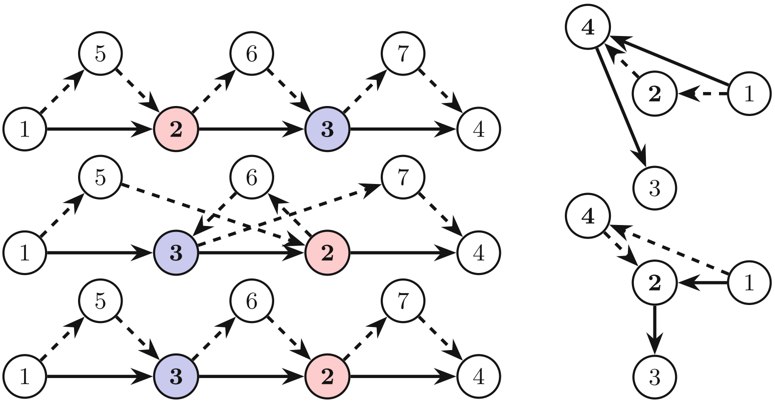

Fig. 1. A visual example of the implemented single-route improvement heuristics: 2-opt (left) and delivery exchange (right). Truck routes are indicated by solid lines and drone routes indicated by dashed lines. In the case of 2-opt, inverting a sub-sequence (top to middle) requires adjusting delivery and rendezvous vertices (bottom) to maintain feasible routing. In all cases, any change must not violate the endurance constraints of the drone.

Furthermore, in order to explore the solution space of the problem, we define two operators that focus mainly on drones: *Drone Insertion (DI)* operator and *Delivery Exchange (DE)* operator. The DI operator is based on the concept of *first-improvement*. More precisely, starting from the first vertex in a tour, whenever the first feasible sortie is possible and reduces the time required to complete the tour, the drone is inserted. The DE operator works as follows: Assume that both, the drone and the truck, serve exactly one customer after a launch and before meeting at a rendezvous vertex, then the DE operator will attempt to change the method of delivery. If the exchange is feasible and the time required to complete the tour decreases, then the change is kept; otherwise, the move is reverted.

Figure 1 shows a visual example of the single-route improvement operators: 2-opt and DE. Figure 2 visualizes the *String Exchange (SE)* and the *String Relocation (SR)* operators. The SE operator exchanges two random customers between two distinct tours. The SR moves a vertex from one tour to another one. The affected customers can be served by drone or truck before the move and will always be assigned to a truck after the move.

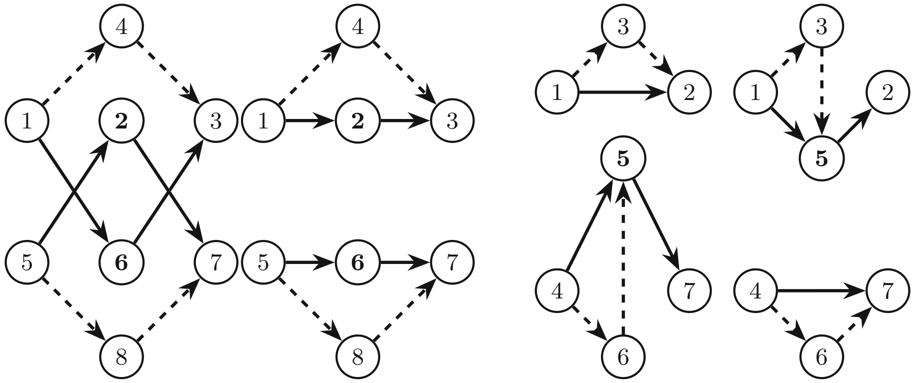

Fig. 2. A visual example of the implemented multi-route improvement operators: string exchange (left) and string relocation (right). In the case of string exchange, the vertices labeled 2 and 6 are exchanged between two tours. In the case of string relocation, the vertex labeled 5 is moved from one tour to another tour.

Algorithm 1. Two-Phase Heuristic (TPH)

```
procedure OPTIMIZE($m, \alpha, \beta$, Instance)
    solution ← routeFirstClusterSecond(Instance,m);
    while !stoppingCriterion do
        if phaseOne then
            solution.twoOpt();
            solution.stringMix();
        else
            solution.insertDrones();
            solution.deliveryExchange();
        end if
    end while
    return solution;
end procedure
```

Algorithm 2. Single-Phase Heuristic (SPH)

```
procedure OPTIMIZE($m, \alpha, \beta$, Instance)
    solution ← routeFirstClusterSecond(Instance,m);
    solution.insertDrones();
    while !stoppingCriterion do
        solution.twoOpt();
        solution.stringMix();
        solution.insertDrones();
        solution.deliveryExchange();
    end while
    return solution;
end procedure
```

Assume that t_a^T and t_b^T are the objective values of two tours a and b before applying the operators and $t_{a'}^T$ and $t_{b'}^T$ are the objective values after applying the operators. We keep the change if $max(t_{a'}^T, t_{b'}^T) < max(t_a^T, t_b^T)$ holds true.

Using the described operators, i.e., 2-opt, SM (i.e., SE & SR), DI, and DE, we introduce two heuristics for solving the VRPD. Both heuristics are started by the NNH in order to construct initial tours. The first heuristic, called *Two-Phase Heuristic (TPH)*, initially ignores the drones and focuses on constructing good VRP tours by means of 2-opt and SM. As soon as a given amount of time has passed, the drones are inserted into the existing tours using DI and the drone placement optimized using DE, transforming the initial VRP solution into a VRPD solution. The second heuristic, named *Single-Phase Heuristic (SPH)*, inserts drones right from the very beginning (i.e., just after NNH) and tries to create good VRPD solutions by using a combination of 2-opt, SM, DI, and DE. Since some sorties might be removed after using 2-opt or SM, due to violation of the endurance constraint, it is necessary to continuously utilize DI. Algorithms 1 and 2 illustrate the pseudo-code of the introduced heuristics (where, α and β are instance and test parameters. See also Sect. 4).

4 Computational Experiments

This section is dedicated to the presentation of our computational experiments and their numerical results. In particular, we have carried out a total of 750 experiments. For this purpose, we rely on instances from the TSPLIB[1] in order to generate VRPD instances. For this purpose, we start by introducing the following metric for setting comparable endurance constraints among all instances. Since each instance can be described by a graph $G(V, E)$, we first consider the adjacency matrix $\mathcal{A}(G)$. Then, through introducing a new parameter β, we set the endurance to $\mathcal{E} = \beta \cdot max(\mathcal{A})$ and use the following parameter values and TSP instances in our experiments:

- Six TSP instances: $rd400$, $att532$, $u574$, $gr666$, $rat783$, $dsj1000$.
- Three different values for $\alpha \in \{2, 3, 4\}$.
- Five different values for $\beta \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$.
- Five runs per problem instance and each pair of parameters of α and β. The case $\beta = 0.0$ corresponds to the classical VRP (i.e., there is no drone). Hence, for the case of $\beta = 0.0$, we only test for one value of α, as no drones will be used, regardless of the value of α.
- We test with $m = 3$ trucks and $k = 1$ drone per truck.
- There is no limit on the capacity C of the trucks but the capacity of each drone is limited to 1.

We implemented the algorithms in Java SE 8. We limit the run time of each algorithm to 5 min and record the best value achieved after passing the time-limit. The algorithms were run on an Intel 5200U CPU limited to the base clock

[1] https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

speed of 2.2 GHz with a maximum of 8 GB of RAM available. In the case of TPH, we ran the tour phase for 4 min and used the remaining time to insert drones and applied the delivery exchange heuristic afterwards. For the sake of comparison, we computed reference values, generated using TPH with a tour phase of 5 min and $\beta = 0$ (no drones are inserted). For all instances, we set the first vertex as the depot. Figure 3 shows sample output for an instance with 51 vertices and limited drone endurance. Table 2 shows the numerical results of the experiments. For each instance, the objective value of the VRP solution ($\beta = 0$) is taken as the reference (base) value and the remaining values in the same row are the average values (of five runs) scaled relative to the VRP solution.

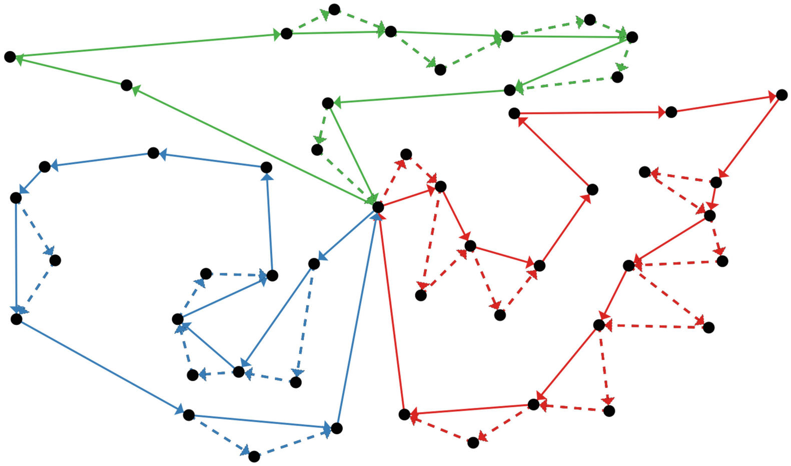

Fig. 3. Sample output for a problem with 51 vertices and limited drone endurance. The truck's and the drone's paths are shown using solid lines and dashed lines, respectively.

According to the numerical results, we can make the following observations:

- The TPH can typically achieve good results among a wide range of α and β values. Additionally, the TPH always produces better results than the base value ($\beta = 0$).
- The SPH only achieves good results for large values of α and β. Additionally, the SPH often produces worse results than the base value ($\beta = 0$).
- Our experiments indicate, that the TPH creates better solutions than the SPH in most cases - in particular on large-scale instances (i.e., u574, gr666, dsj1000). Nevertheless, in some instances (e.g., some cases of rd400, att532, rat783), the SPH can provide better or competitive results in comparison to the TPH. A possible explanation might be due to the presence of the additional constraints imposed by the VRPD. In fact, these constraints limit the search space and make it more difficult to find valid and high-quality VRPD solutions by means of a single-stage heuristic.

Table 2. Summary of the numerical results of the TPH and the SPH algorithms across different instances.

Instance	α	Two-Phase Heuristic (TPH)					Single-Phase Heuristic (SPH)			
		β					β			
		0.00	0.25	0.50	0.75	1.00	0.25	0.50	0.75	1.00
rd400	2	100%	84.5%	85.1%	81.6%	84.6%	108.0%	99.7%	99.3%	99.8%
	3	100%	81.2%	82.1%	84.4%	81.4%	99.7%	99.3%	99.8%	84.5%
	4	100%	82.6%	83.2%	83.1%	83.5%	99.3%	99.8%	84.5%	85.1%
att532	2	100%	87.5%	91.9%	85.9%	87.7%	92.6%	92.6%	94.0%	92.6%
	3	100%	87.6%	87.7%	87.8%	85.6%	92.6%	94.0%	92.6%	87.5%
	4	100%	86.4%	88.8%	89.4%	87.4%	94.0%	92.6%	87.5%	91.9%
u574	2	100%	93.5%	94.6%	93.9%	94.2%	104.4%	106.1%	107.9%	102.4%
	3	100%	84.8%	82.6%	86.1%	84.5%	106.1%	107.9%	102.4%	93.5%
	4	100%	83.4%	84.6%	82.5%	84.0%	107.9%	102.4%	93.5%	94.6%
gr666	2	100%	87.8%	88.1%	83.5%	87.3%	124.0%	122.0%	116.1%	114.5%
	3	100%	85.8%	83.2%	83.1%	84.2%	122.0%	116.1%	114.5%	87.8%
	4	100%	83.4%	83.2%	84.9%	82.9%	116.1%	114.5%	87.8%	88.1%
rat783	2	100%	86.9%	88.4%	88.7%	89.1%	103.2%	106.7%	97.8%	106.0%
	3	100%	89.2%	87.9%	85.5%	89.9%	106.7%	97.8%	106.0%	86.9%
	4	100%	89.5%	85.4%	93.4%	96.1%	97.8%	106.0%	86.9%	88.4%
dsj1000	2	100%	88.0%	86.7%	87.0%	84.7%	106.4%	104.6%	103.4%	98.5%
	3	100%	83.5%	81.4%	86.1%	85.6%	104.6%	103.4%	98.5%	88.0%
	4	100%	84.6%	82.9%	85.3%	81.6%	103.4%	98.5%	88.0%	86.7%

- Although the TPH has a higher-average solution quality, we notice that the TPH fails to provide better solutions in most of the instances that correspond to larger values of α and β. However, the SPH shows a clear progression towards higher objective values with an increase in α and β.

5 Conclusion

In this paper, we introduced two new heuristics, *Two-Phase Heuristic (TPH)* and *Single-Phase Heuristic (SPH)*, for solving the VRPD. To the best of our knowledge, it is the first time that numerical results for large-scale VRPD instances are presented. While the TPH, as a two-stage approach, is based on first creating good VRP tours and then creating a VRPD solutions through insertion of drones, the SPH focuses on creating good VRPD solution right from the scratch. Our observations, based on preliminary numerical results, confirm that the TPH provides better results than the SPH in most cases, and there are a few cases where the SPH produces better or competitive results. Hence, it seems reasonable to claim that good VRPD solutions can be constructed by using a two-stage heuristic, starting from good VRP solutions.

Future research might focus on heuristics that allow for a good exploration of the search space by means of more effective methods for handling additional constraints of the VRPD. Metaheuristics, such as Simulated Annealing, seem to be a reasonable choice for this task. Finally, design of algorithms for solving VRPD in presence of more realistic conditions, e.g., limited truck capacity or multiple drones per truck, can also be considered as interesting future research plans. The research in these directions are in progress and the results will be published in future.

Acknowledgements. The authors acknowledge the chair of Business Information Systems and Operations Research (BISOR) at the TU-Kaiserslautern (Germany) for the financial support, through the research program "CoVaCo".

References

1. Gendreau, M., Potvin, J.-Y.: Handbook of Metaheuristics. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-4419-1665-5
2. Ha, Q.M., Deville, Y., Pham, Q.D., Ha, M.H.: Heuristic methods for the traveling salesman problem with drone, pp. 1–13. Technical report, CTEAM/INGI/EPL (2015)
3. Jiang, X., Zhou, Q., Ye, Y.: Method of task assignment for UAV based on particle swarm optimization in logistics. In: Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence - ISMSI 2017, pp. 113–117. ACM Press (2017)
4. Mathew, N., Smith, S.L., Waslander, S.L.: Planning paths for package delivery in heterogeneous multirobot teams. IEEE Trans. Autom. Sci. Eng. **12**(4), 1298–1308 (2015)
5. Mourelo, F.S., 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. **9**(2), 374–388 (2016)
6. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp. Res. Part C: Emerg. Technol. **54**, 86–109 (2015)
7. Poikonen, S., Wang, X., Golden, B.: The vehicle routing problem with drones: extended models and connections. Networks **70**(1), 34–43 (2017)
8. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)
9. Toth, P., Vigo, D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia (2002)
10. Ulmer, M., Barrett, W.T.: Same-day delivery with a heterogeneous fleet of drones and vehicles. Technical report, pp. 1–30 (2017)
11. van Breedam, A.: An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints contents. Ph.D. dissertation, University of Antwerp (1994)
12. Wang, X., Poikonen, S., Golden, B.: The vehicle routing problem with drones: several worst-case results. Optim. Lett. **11**(4), 679–697 (2016)