Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 38–47

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

On a cooperative truck-and-drone delivery system

Gloria Cerasela Crişan[a,]*, Elena Nechita[a]

[a]*Vasile Alecsandri University of Bacău, Cal. Mărăşeşti 157, Bacău 600115, Romania*

Abstract

During the last decade drones, or unmanned aerial vehicles, have been intensively studied from various perspectives. Important advances in drone technology and numerous experiments concerning drone infusion in various services and businesses have generated intensive research on modeling delivery systems that include drones. Combinatorial Optimization Problems such as Traveling Salesman Problem and Vehicle Routing Problem have been extended by considering drones and optimize various cost functions. This paper introduces a new greedy heuristic for minimizing the total transportation time of a truck-and-drone delivery system, proposes a new cost function to include the flying time of the drone and performs a comparison with previous similar approaches. Experiments with different parameter settings on two large area Traveling Salesman Problem instances show significant total time savings for the proposed approach and supports future research when considering real-world scenarios.

Keywords: Truck-and-drone systems; Traveling Salesman Problem; Heuristics.

1. Introduction

Producers and retailers are nowadays developing emergent supply chain configurations that integrate different modalities, extending them directly to the consumers' front door [2]. The drone-based delivery models respond to the new consumer behaviors and purchase patterns. To design such systems is a recent research preoccupation. Several works on Traveling Salesman Problem (TSP) are presented in [1, 13, 16, 25, 30]; other works are based on Vehicle Routing Problem (VRP) [6, 11, 17, 29].

* Corresponding author. Tel.: +4-023-454-2411; fax: +4-023-4545753.
E-mail address: ceraselacrisan@ub.ro

1877-0509 © 2019 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2019.09.158

In this paper we introduce a new heuristic approach for designing a truck-and-drone delivery system. It starts from a solution to a TSP instance and constructs, in a greedy way, short drone sub-routes by excluding from the truck path the nodes that offer the biggest savings from the truck cost point of view. As the final truck route preserves the order of the nodes in the initial solution and the drone flies between neighbor nodes, this approach is computationally very efficient.

In order to test its results, two large-area TSP instances are used. We are aware that these instances have no practical meaning; we used them here for intensive tests, to capture our method's behavior when it iteratively and massively sets drone subroutes. The results are compared with the heuristic approach proposed in [25], showing more sensitivity and higher computing efficiency.

The remainder of the article is organized as follows. Section 2 reviews drones' usage and applications. Section 3 presents several models available in the scientific literature, designed to include drones in last-mile delivery. Section 4 summarizes the TSP and FSTSP models and introduces the instances on which the computational tests are developed. Section 5 describes the new proposed heuristic. Section 6 presents and interprets the results of the experiments. Conclusions and future research directions are drawn in the final section.

2. Literature review on drones' usage

An Unmanned Aerial Vehicle (UAV) (or Unmanned Aircraft System UAS, or a Remotely Piloted Aircraft RPA), commonly known as drone, is an aircraft with no onboard human pilot. As described in [20], the UVAs are recorded since 1918. Their first deployments were in the military zone [18] but today, due to affordable costs, broad categories of users are implementing applications in civilian areas [12].

Several projects in Transportation Engineering are presented in [20]: road geometry modification, traffic incidents reconstruction, and documentation of traffic control placements. Integrated aerial solutions for special projects (in precision agriculture for example) include customized kits and packages with sensors, accessories, tablets [23]. Drones may also be very useful in forensics and law enforcement applications [38]. Drones in logistics, first aid or mining are considered a young technology, with great potential. Numerous experiments have been lately deployed, to test such possibilities [22, 31, 34].

Many applications start from a TSP relaxed solution, as there are several very good exact TSP solvers (Concorde [5] for example) or dedicated heuristic implementations [17, 28, 33, 39]. This work iteratively integrates the drone in a given TSP tour, in the most promising positions. Current drone technologies are presented in [7]. A comprehensive survey on UAVs applications is [27].

3. Previous models for last-mile delivery with drones

Within a supply chain, the last mile is considered to be expensive, inefficient and polluting, due to very specific delivery needs. In [14], the authors present the *last-mile problems* related to the different types of reception of goods and synthesize the main factors that influence the efficiency and the cost structure of the supply chain.

In [25], the authors introduce a new variant of the traditional TSP, namely the *Flying Sidekick Traveling Salesman Problem* (FSTSP) in a mixed integer linear programming formulation. In FSTSP, each customer must be served exactly once by a delivery truck or by a UAV which operates in coordination with the truck. Once launched, the drone must visit a customer and return, within its flight endurance limit, to the truck or to the depot. The objective is to minimize the total service time (all customers are visited; both vehicles are at the depot).

A reference framework dealing with combined truck and drone delivery system is given in [1]. In *Traveling Salesman Problem with Drone* (TSP-D), there are N customer locations to be served from a single depot, by a truck or by a drone, typically considered to be faster than the truck. Some customers (the *truck nodes*) are not suitable for drone delivery. The objective of the TSP-D is to find the shortest tour, in terms of time, to serve all customer locations under the following assumptions: a) the drone has unit-capacity and returns to the truck after each delivery; b) the drone can land on and depart from the truck while the truck is parked at a customer location or at the depot; c) recharging time of the drone, the pickup and delivery times of packages are neglected.

A computational study is done on various randomly generated TSP-D instances with different characteristics, depending on: number of drone nodes and truck nodes, waiting times and travel distances of trucks and drones, different drone speeds and/or different distribution of delivery locations. Other similar models are in [8, 35].

4. The Flying Sidekick Traveling Salesman Problem (FSTSP)

The goals of this section are to briefly present the Traveling Salesman Problem and to introduce the premises for two *Flying Sidekick Traveling Salesman Problem* (FSTSP) large instances.

4.1. Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) is one of the most known and studied Combinatorial Optimization Problems. An extensive view on TSP approaches and related problems is in [3].

Definition. In a complete graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is the set of n vertices, let $C = (c_{ij})_{1\le i,\, j\le n}$ being the cost matrix associated with E. The goal of TSP is to find a minimum cost Hamiltonian circuit (i.e. a minimum cost closed path that contains any vertex once and only once).

If the matrix C is symmetric then TSP is symmetric. If the costs form a Euclidean norm, then TSP is Euclidean. The most known TSP benchmarks are [36, 37].

TSP is a NP-hard problem from the computational complexity point of view; therefore it is a very low probability that a polynomial time exact algorithm will ever be found for all cases [19]. As large TSP instances are difficult to solve, besides the exact solving methods, the researchers also focus on non-exact methods: the approximate and heuristic approaches.

The best exact solver for symmetric TSP currently is Concorde [5]. Approximate methods always deliver solutions with known quality. One such algorithm, based on the minimum spanning tree and designed for the Euclidean TSP is presented in [9]. Heuristic methods usually have good empirical behavior. Lin-Kernighan method is an old and very efficient heuristic [21]. Meta-heuristic approaches which exhibit successful behavior on TSP are: Tabu Search [15], Genetic Algorithms [24] and Ant Colony Optimization [10].

A collection of national TSP instances with nodes specifying real points-of-interest and the Euclidean 2D distance is maintained at [26].

The Romanian TSP instance has 2950 nodes representing the main localities from Romania, each specified by a pair of decimal values for the geographic coordinates (latitude, longitude) [32]. The distance between two nodes is computed using the great circle connecting them, rounded to the next integer and expressed in kilometers (the GEO norm). The exact solver Concorde provided the optimum length: 21,683 km. The Bulgarian TSP instance has 1954 nodes [4]; its optimum length is 10,839 km.

4.2. An insight into the Flying Sidekick Traveling Salesman Problem (FSTSP)

In *Flying Sidekick Traveling Salesman Problem* (FSTSP) a truck and a drone cooperate in order to visit exactly once each client. The truck and the drone depart from a single depot together or independently, perform the clients' service and return to the same depot. Some clients are visited by the drone, others are visited by the truck, but when traveling in tandem, the drone is transported by the truck. The objective of the FSTSP is to minimize the time required by the truck-and-drone system to service all customers and to return to the depot [25].

To visit a customer, the drone performs a three-node *sortie* (i, j, k) within its flight endurance limit: it starts in node i (at the depot or at a customer location), visits the client in the node j and meets the truck in node k.

Let $C = \{1, 2, ..., c\}$ be the set of all customers and $C' \subseteq C$ the subset of customers that may be serviced by the drone. The depot (a single physical location) is set as node 0 at the departure of the truck and drone, and is set as node $c+1$ at their return. Therefore, the sets to operate with are: $N = \{0, 1, ..., c+1\}$ - the set of all nodes, $N_0 = \{0, 1, ..., c\}$ - the set of all nodes the two vehicles may depart from, and $N_+ = \{1, 2, ..., c+1\}$ - the set of all nodes visited by a vehicle along a tour.

Let τ_{ij} be the time required for the truck to travel from node i to node j, $i \in N_0$ and $j \in N_+$ and τ'_{ij} the analogous flying time for the drone. Given the logical restrictions, τ_{ii} and τ'_{ii} are not defined and $\tau_{0,c+1} \equiv 0$.

The following parameters (measured in units of time) are considered: S_L - the time required to prepare the drone for launch; S_R - the time required for the drone recovery, after the truck and drone have met, and e – the flight endurance of the drone. The traveling speeds are constant for both truck and drone. Let α be the speed-up when using the drone instead of the truck, defined as the ratio between the drone speed and the truck speed.

A triplet (i, j, k) is a valid *sortie* if $j \in C'$, $j \neq i$, $i \in N_0$, (the node j and the launch point differ) and also if $k \in N_+, k \neq j, k \neq i, \tau'_{ij} + \tau'_{jk} \leq e$ (the node where the drone reunites with the truck differs from i and from j and the drone's travel time from i to j and further to k does not exceed its endurance limit).

In [25], the saving of serving the node j by the drone is computed for each $j \in C'$. The method computes the greatest saving by testing all the possible sorties of the current truck subroute, and also by testing all possible insertions of other truck nodes in other position in the current truck subroute. Starting from the depot, the drone is set to fly to j if the greatest saving is positive, and the method is iterated for the remaining truck route.

The heuristic proposed in Section 5, hereinafter referred as *NGH* (*N*ew *G*reedy *H*euristic) has a lower complexity than the one proposed in [25], hereinafter referred as *MC* (*M*urray and *C*hu).

5. The new heuristic approach for FSTSP

This section describes the new heuristic method for FSTSP and a small example on how it works.

5.1. The new heuristic approach NGH

The proposed algorithm starts from a complete truck circuit and ends with a tour where the most promising nodes are iteratively extracted from the truck path and are assigned to the drone. A node is *promising* if it may be visited by the drone and the time saved by the truck, when skipping that node is significant. The list of the saved time durations for the nodes in *C'* can be computed in O(n) time and is sorted in O($n$log(n)) time. For each node in the sorted list, the drone insertion procedure is applied. As this procedure is done in constant time, this final step of the proposed method has O(n) computational complexity.

The *droneInsertion* procedure works on the truck path, takes a node i and assigns it to the drone if:

- the nodes $i-1$ and $i+1$ are not already assigned to the drone;
- the time needed by the drone to fly from $i-1$ to i and from i to $i+1$ is not greater than the time needed by the truck to travel from $i-1$ to $i+1$ (therefore, the drone waits for the truck in node $i+1$ or both vehicles reach this node at the same moment; we call this a *small sortie*);
- $i \in C'$ and drone endurance allows the flight on edges $(i-1, i)$ and $(i, i+1)$.

Otherwise, the procedure exits with no new drone assignment.

The major advantage of this new heuristics approach is its speed. The algorithm is presented in the following.

Algorithm 1. Pseudocode for the new heuristic *NGH*
//Requires: a complete tour, starting from the *depot*
for all (i in C')
  compute *sav*[i] = $\tau_{i-1,i} + \tau_{i,i+1} - \tau_{i-1,i+1} - S_L - S_R$
end for
sort the array *sav* in decreasing order
keep in *sav* only the strictly positive values
for all (i in *sav*)
  call *droneInsertion*(i)
end for
compute the cost of the new tour
end

Algorithm 2. Pseudocode for *droneInsertion*
//Requires: a node i
droneTime = $\tau'_{i-1,i} + \tau'_{i,i+1}$
  if (*droneTime* $\leq \tau_{i-1,i+1}$ AND $i \in C'$ AND both $i-1$ and $i+1$ are not served by drone AND $e \geq$ *droneTime* $+ S_L + S_R$)
    then make the new sortie $(i-1, i, i+1)$
  end if
end

5.2. A small example

Suppose that we have 7 nodes to be served from the node 0 (the *depot*), $S_L = S_R = 0$ and the drone speed is twice as the truck speed. One solution using truck only is presented in Figure 1 (nodes are in circles, truck service times are on the edges); its travel time is 68. The idea is to extract only *small sorties*, where the drone flies between adjacent (in the initial route) nodes. The nodes are inspected in the order given by the magnitude of the time saved by replacing the truck with the drone. A sortie is set only if the drone is able to fulfill its duty in the same or less time than the truck. The demonstration in **Annex 1** shows that no better sortie is possible in this case.

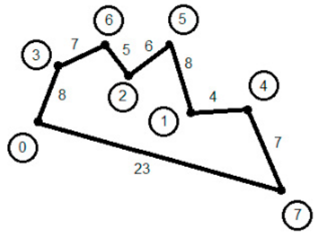


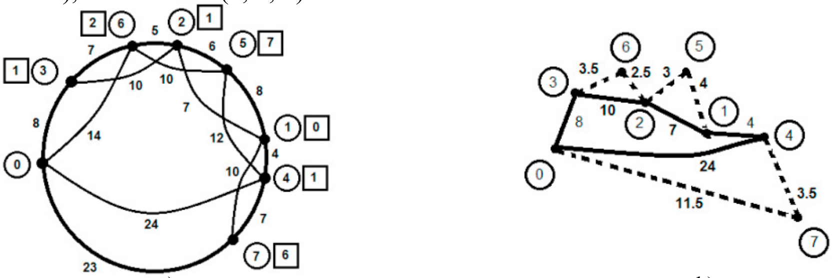
Figure 1. Small TSP instance, initial solution

The same solution is represented on a circle in Figure 2 a); it also displays the saved truck-time for all nodes, in squares. These values, stored in the array *sav*, are presented in Table 1. For example, if the node 3 is serviced by the drone, then the truck goes to node 6, the time to reach node 6 is 14, and so the saved time computed for node 3 is 1.

Table 1. Order of the computation of the values in squares in Figure 2 a).

nodes	0	3	6	2	5	1	4	7	0
sav	-	1	2	1	7	0	1	6	-

The node order for testing if a small sortie can be done is: 5, 7, 6, 3, 2, 4. If node 5 is served by the drone, the drone time is 6/2+8/2 = 7, which is equal to the truck time to reach node 1 from node 2; the sortie (2, 5, 1) is set for the drone and the nodes 2 and 1 are forbidden to drone service. The next try is node 7. The drone time is 7/2+23/2 = 15, which is less than the truck travelling time from node 4 to depot, so a new sortie (4, 7, 0) is set and node 4 is flagged as forbidden. The next tested node is 6; the time for the drone is 7/2+5/2 = 6, it is less than 10 (the truck time to arrive from 3 to 1), a new sortie (3, 6, 2) is set and nodes 3 and 2 are forbidden for future drone allocation.



a) b)

Figure 2. a) Example with nodes labeled with saved time; b) Truck route (solid line) and drone sorties (dashed lines).

All the remaining nodes are now unavailable for tries, so the NGH algorithm exits with the path in Figure 2 b), which has the travel time 53. The drone service is represented by dotted lines, with flying times labeled on them.

6. Experiments and analysis

Although they may start from any complete tour, the experiments use optimum tours as initial truck solutions. Moreover, we consider that all the nodes may be serviced by the drone ($C' = C$) and the drone endurance is infinite ($e = \infty$). The complete set of raw data is available upon request from the authors.

We have designed the tests under the following parameters setting:

- α (the ratio between the drone speed and the truck speed) took 21 values from 1.5 to 2.5, with step 0.05;
- $S_L + S_R$ (time needed for drone launching and recovery) took 5 values between 0 and 0.8 hours, with step 0.2.

The starting truck tours for both instances were the optimum tours, obtained with *Concorde*. For both instances, for every pair (α , $S_L + S_R$) we recorded: the total time TT; the drone flying time TD; the number of sorties NS.

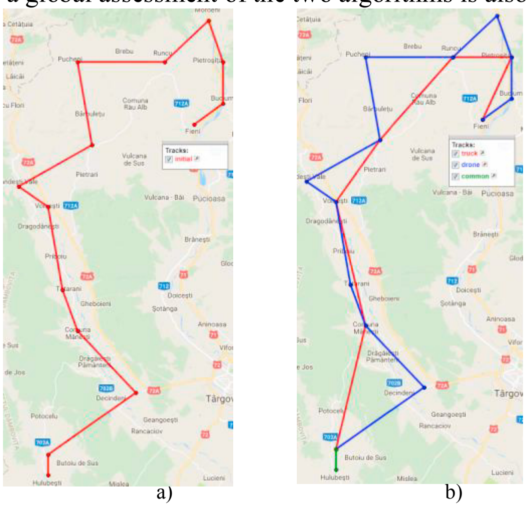
The applications implementing *MC* and *NGH* were developed in C++ under Windows OS. The tests ran on a Pentium dual-core processor at 2.5 GHz with 2 GB RAM. The average values of the running times for *MC* tests were: 90s for the Romanian instance and 50s for the Bulgarian instance, while the corresponding values for the proposed heuristic *NGH* were 6s and 2.8s. There were 420 executions of the applications. The differences between this work and two previous experiments on integrating drone delivery into the TSP are presented in Table 2.

Table 2. Characteristics of TSP-gp [1], MC [25] and NGH.

Characteristic	TSP-gp heuristic	FSTSP heuristic (MC)	NGH
Known optimum initial solution?	Yes; Concorde	No	No
Multiple visits allowed?	Yes	No	No
Distance	Euclidean 2D	Manhattan (truck); Euclidean 2D (drone)	GEO
Number of instances	30	72	2
Instances available?	Yes	No	Yes
Number of nodes	10	10	2950, 1954
Computational complexity	O(nlog(n))	O(n^4)	O(nlog(n))

A fragment from the initial solution of the Romanian instance is in Figure 3 a). Figure 3 b) presents the result of the proposed heuristic *NGH*: a common path for the truck carrying the drone, pictured with green line (on the bottom of the image), the truck path is in red line and six small sorties of the drone are in blue.

The following subsections present considerations on the behavior of *MC* and *NGH*, with respect to TT, TD and NS. At the end of the section, a global assessment of the two algorithms is also presented.



a) b)

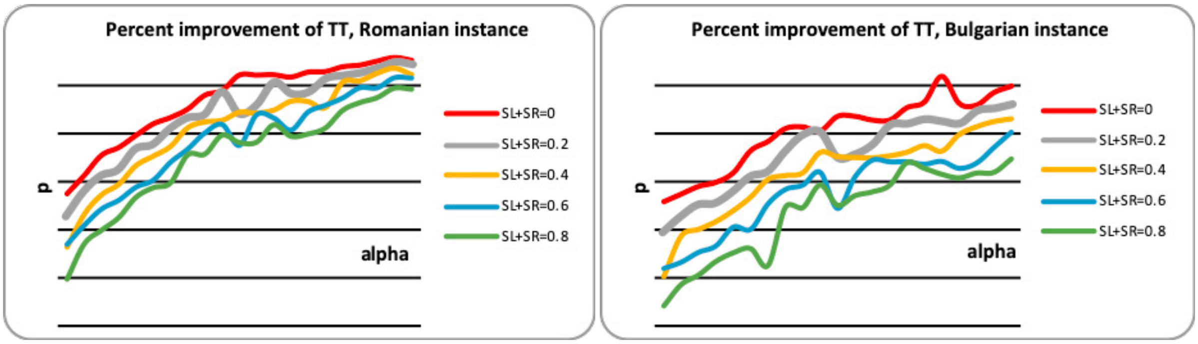
Figure 3. a) Part of the initial truck tour for the Romanian instance; b) the same nodes with 6 consecutive small sorties.

6.1. Remarks concerning the total delivery time

In order to compare the results generated by the two algorithms, we used the *percent improvement*, expressed as

$$p = \frac{TT_{MC} - TT_{NGH}}{TT_{MC}} * 100 \tag{1}$$

The positive values of p identify the cases when *NGH* provides better results than *MC*. The percent improvement of TT is displayed for the Romanian instance in Figure 4 a) and for the Bulgarian instance in Figure 4 b).



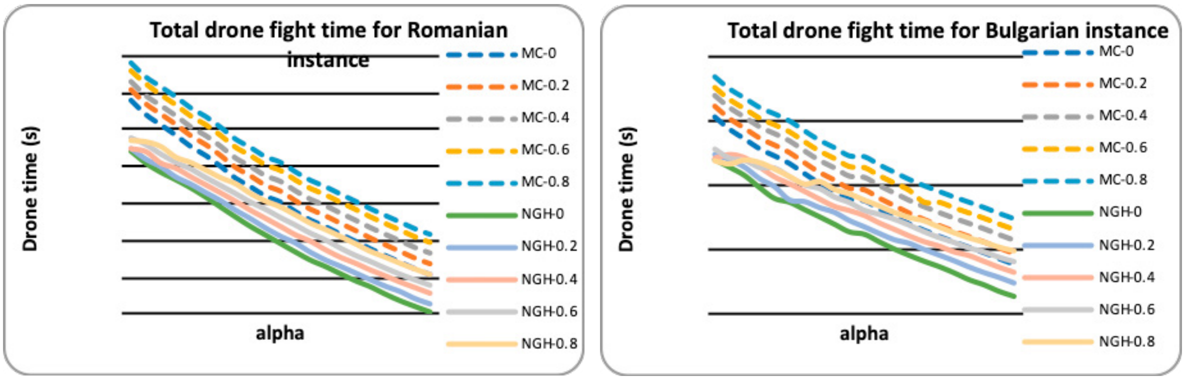
a) b)

Figure 4. Percent improvement of truck-and-drone system time for a) the Romanian instance; b) the Bulgarian instance.

As these two representations show, the number of the points scattered above the horizontal axis is bigger for the Romanian instance, which has more nodes than the Bulgarian instance. In both cases, for a given α, the percent improvement decreases as $S_L + S_R$ increases. This behavior shows a balance between MC and NGH, when considering different launching and recovery time values.

6.2. Remarks concerning drone operations

Under the setting $S_L + S_R = x$, the results are denoted $MC - x$ and $NGH - x$ in the legends of Figure 5. The charts show that the values of TD returned by MC are bigger than those returned by NGH, for all the pairs $(\alpha, S_L + S_R)$. The nature of the variation of TD will be investigated in future research. As the charts display, this variation slightly differs with the size of the instance. For the Romanian instance, TD appears to exhibit more regularity than in the case of the smaller-size Bulgarian instance. For the algorithm MC, NS does not depend on $S_L + S_R$. On the other side, the proposed heuristic is sensitive to $S_L + S_R$ and always generates fewer subroutes.



a) b)

Figure 5. Drone times for (a) Romanian instance; (b) Bulgarian instance, for MC and NGH algorithms.

6.3. A new cost function to globally assess the truck-and-drone system

We further propose a *unified cost* (UC) to include both TT and TD, as a simple, linear combination:

$$\text{UC} = \text{TT} + \textit{weight} * \text{TD} \tag{2}$$

where the *weight* was computed such that, for α =1.50, UC(*MC*)=UC(*NGH*). For the recorded data, this equation gave *weight*=0.264808. The averaged UC values are displayed in Figure 6. The trendline for the scattered values of UC for *NGH* is logarithmic, with R^2=0.9913. *NGH* produces better UC values, the gap increasing with α .

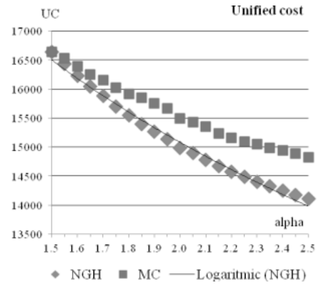


Figure 6. The new proposed cost UC for the two algorithms.

7. Conclusions and future work

In this paper we have proposed a new heuristic for solving the FSTSP and verified it experimentally. It provided promising results when compared to previous successful proposals. The test instances allowed intensive analysis of the behavior of the proposed method, based on the ratio between the truck and the drone speeds and the time needed by drone launch and recovery. We also propose a *unified cost* which includes the time needed by the drone service.

Future work is intended to validate the introduced *NGH* heuristic on a significant number of real world instances and on test instances to be generated with various characteristics, such as those proposed in [1]. To start from multiple randomly generated truck paths is also foreseen. Another research direction concerns the study of the proposed *unified cost* and of the nature of its behavior. Moreover, to consider new constraints (such as the drone endurance) in testing the *NGH* behavior stays in our attention.

Acknowledgements

This work was supported by the project CNFIS-FDI-2019-0453: Support actions for excellence in research, innovation and technological transfer at "Vasile Alecsandri" University of Bacău (ACTIS-Bacău), financed by the National Council for Higher Education, Romania.

Appendix A.

Theorem 1. If $\alpha \geq 1$ and a small sortie is set for a given node, then no better sortie exists for that node.

Proof. Suppose that the *droneInsertion* procedure is applied to node i, and a new sortie ($i-1, i, i+1$) is set. We will show that no better and longer (with more nodes for the truck) sortie exists.

In Figure 7 a) the fragment of the truck path ($i-2$, $i-1$, i, $i+1$, $i+2$) is represented. We compare the time needed for the truck-and-drone system to travel from $i-2$ to $i+1$ for the small sortie ($i-1$, i, $i+1$) (Figure 7 b)) and if the drone starts earlier, from $i-2$ (Figure 7 c)). As the *droneInsertion* is applied, it results that

$$\tau'_{i-1,i} + \tau'_{i,i+1} \leq \tau_{i-1,i+1} \tag{3}$$

From (3), the duration for moving from $i-2$ to $i+1$ in Figure 7 b) is:

$$A = \tau_{i-2,i-1} + \tau_{i-1,i+1} + S_L + S_R \tag{4}$$

and the duration for moving from $i-2$ to $i+1$ in Figure 7 c) is:

$$B = \max(\tau_{i-2,i-1} + \tau_{i-1,i+1}, \tau'_{i-2,i} + \tau'_{i,i+1}) + S_L + S_R \tag{5}$$

We are interested in comparing A to B. Working on (5) we have:

$$B = \tau_{i-2,i-1} + \max(\tau_{i-1,i+1}, \tau'_{i-2,i} + \tau'_{i,i+1} - \tau_{i-2,i-1}) + S_L + S_R \tag{6}$$

as it is obviously true that $\max(x+y,z) = x + \max(y, z-x)\ \forall x,\ y,\ z \in R$.

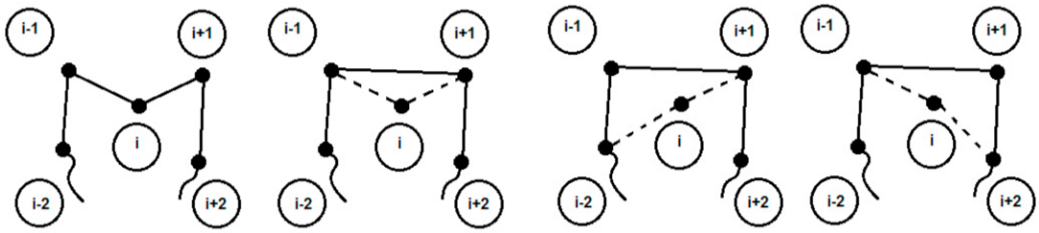
If we prove that:

$$\tau'_{i-2,i} + \tau'_{i,i+1} - \tau_{i-2,i-1} \leq \tau'_{i-1,i} + \tau'_{i,i+1} \tag{7}$$

then from (3) and (6) and (7) we get:

$$B = \tau_{i-2,i-1} + \tau_{i-1,i+1} + S_L + S_R = A, \tag{8}$$

which shows that an earlier choice for the initial node of the sortie produces no improvement. So, under the assumptions we made, it is better to restrict the search for drone nodes assignment to small sorties only, in order to let more nodes available for further subroute constructions. Of course, the time for drone launching and recovery could influence the number of small sorties. If this time becomes large, then the drone becomes inefficient.



a) b) c) d)

Figure 7. a) Initial path, truck only; b) small sortie, the drone flies to i; c) the drone starts from i -2; d) the drone arrives at i +2.

Let us prove the inequality (7). It is equivalent to:

$$\tau'_{i-2,i} - \tau'_{i-1,i} \leq \tau_{i-2,i-1} \tag{9}$$

As we have from the definition of α that $\tau_{i-2,i-1} = \alpha\tau'_{i-2,i-1}$, we therefore have to show that

$$\tau'_{i-2,i} - \tau'_{i-1,i} \leq \alpha\tau'_{i-2,i-1} \tag{10}$$

Multiplying (10) by the constant speed of the drone, we have to prove that

$$d_{i-2,i} - d_{i-1,i} \leq \alpha d_{i-2,i-1} \tag{11}$$

where d is the distance matrix.

As the triangle inequality holds for every distance, we have $d_{i-2,i} \leq d_{i-2,i-1} + d_{i-1,i}$ which is equivalent to

$$d_{i-2,i} - d_{i-1,i} \leq d_{i-2,i-1} \tag{12}$$

Because $\alpha \geq 1$, from (12) we also have that

$$d_{i-2,i} - d_{i-1,i} \leq d_{i-2,i-1} \leq \alpha d_{i-2,i-1} \tag{13}$$

which concludes our proof.

We can show the same when adding one more node to the right, as in Figure 7 d), due to the symmetry of the image and the computations.

References

[1] Agatz, Niels, Paul Bouman, and Marie Schmidt. (2016) "Optimization Approaches for the Traveling Salesman Problem with Drone", *ERIM Report Series Research in Management*, ERS-2015-011-LIS, Erasmus University Rotterdam.

[2] Alfayad, Fadye. (2017) "Marketing consumer product goods in a shifting supply chain environment: same-day delivery and future drone delivery" *Journal of Accounting and Marketing* **6 (1)**: 1-8.

[3] Applegate, David, Robert Bixby, Vasek Chvátal, and William Cook. (2011) *The Traveling Salesman Problem: A Computational Study* Princeton Series in Applied Mathematics, Princeton University Press, Princeton.

[4] Bulgarian TSP instance, https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/QCHUMD, accessed February 2019.

[5] Concorde solver, http://www.math.uwaterloo.ca/tsp/concorde/index.html, accessed February 2019.

[6] Chang, Yong Sik, and Hyun Jung Lee. (2018) "Optimal delivery routing with wider drone-delivery areas along a shorter truck-route" *Expert Systems with Applications*, **104**: 307-317.

[7] Chen, Siyuan, Debra F. Laefer, and Eleni Mangina. (2016) "State of technology review of civilian UAVS" *Recent Patents on Engineering.* **10 (3)**: 160-174.

[8] Choi, Youngmin, and Paul M. Schonfeld. (2017) Optimization of multi-package drone deliveries considering battery capacity (No. 17-05769), in *96th Annual Meeting of the Transportation Research Board*, Washington DC.

[9] Christofides, Nicos. (1976) “Worst-case analysis of a new heuristic for the travelling salesman problem”, Report 388, *Graduate School of Industrial Administration*, Carnegie Mellon University, Pittsburgh, USA.

[10] Dorigo, Marco, and Thomas Stützle. (2004) *Ant Colony Optimization*. MIT Press.

[11] Dorling, Kevin, Jordan Heinrichs, Geoffrey G. Messier, and Sebastian Magierowski. (2017) “Vehicle Routing Problems for Drone Delivery” *IEEE Transactions on Systems Man and Cybernetics: Systems*, **47 (1):** 70-85.

[12] Floreano, Dario, and Robert J. Wood. (2015) “Science, technology and the future of small autonomous drones”, *Nature*, **521**: 460-466.

[13] de Freitas, Júlia Cária, Puca Huachi Vaz Penna. (2018) “A Randomized Variable Neighborhood Descent Heuristic to Solve the Flying Sidekick Traveling Salesman Problem” *Electronic Notes in Discrete Mathematics*, **66**: 95-102, 2018.

[14] Gevaers, Roel, Eddie van de Voorde, and Thierry Vanelslander. (2009) “Characteristics of innovations in last-mile logistics - using best practices, case studies and making the link with green and sustainable logistics”. In Proceedings of the European Transport Conference (ETC 2009), 1-21, London.

[15] Glover, Fred, and Manuel Laguna. (1997) *Tabu Search*, Kluwer Academic Publications.

[16] Ha, Quang Minh, Yves Deville, Quang Dung Pham, and Minh Hoàng Hà. (2018) “On the min-cost Traveling Salesman Problem with Drone” *Transportation Research Part C: Emerging Technologies*, **86**: 597-621.

[17] Ham, Andy. (2018) “Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming” *Transportation Research Part C: Emerging Technologies*, **91**: 1-14.

[18] Harder, R.W., Raymond Hill, and J. T. Moore. (2004) “A Java™ universal vehicle router for routing unmanned aerial vehicles” *International Transactions in Operational Research*, 11 (3): 259-275.

[19] Karp, Richard. (1972) “Reducibility among Combinatorial Problems” in R.E. Miller, J.W. Thatcher (Eds.), *Complexity of Computer Computations*, The IBM Research Symposia, Plenum. Press, New York, 85–103.

[20] Kwasniak, Andrew, and Anita Kerezman. (2017) “Drones in Transportation Engineering: A discussion of current drone rules, equipment, and applications” *Institute of Transportation Engineers Journal*, **87 (2)**: 40-45.

[21] Lin, Shen, and Brian Kernighan. (1973) “An effective heuristic algorithm for the Traveling–Salesman Problem” *Operations Research*. **21**: 498–516.

[22] Livingston, Molly. (2016) “Flirtey and 7-Eleven complete first month of routine commercial drone deliveries. Deliver 77 packages to customer homes in United States” *PRNewsWire*, 20 Dec.

[23] Microdrones, https://www.microdrones.com/, accessed February 2019.

[24] Mitchell, Melanie (1996) *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

[25] Murray, Chase, and Amanda Chu. (2015) “The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery” *Transportation Research Part C*, **54**: 86-109.

[26] National Traveling Salesman Problems, http://www.math.uwaterloo.ca/tsp/world/countries.html, accessed February 2019.

[27] Otto, Alena, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch (2018) “Optimization approaches for civil applications for unmanned aerial vehicles (UAVs) or aerial drones: A survey” *Networks*, **72(4)**: 411-458.

[28] Poikonen, Stefan, and Bruce Golden. "The Mothership and Drone Routing Problem." to appear in *INFORMS Journal on Computing*.

[29] Poikonen, Stefan, Xingyin Wang, and Bruce Golden. (2017) “The Vehicle Routing Problem with Drones: Extended models and connections” *Networks*, **70 (1)**: 34-43.

[30] Ponza, Andrea. (2016) *Optimization of Drone-Assisted Parcel Delivery*, Tesi di Laurea Magistrale, Università degli Studi di Padova.

[31] Raptopoulos, Andreas. (2013) *No Roads? There's a Drone for that*. TED Talks.

[32] Romanian TSP instance, http://cadredidactice.ub.ro/ceraselacrisan/cercetare/, accessed February 2019.

[33] Schermer, Daniel, Mahdi Moeini, and Oliver Wendt (2018) Algorithms for Solving the Vehicle Routing Problem with Drones. in: Nguyen N., Hoang D., Hong TP., Pham H., Trawiński B. (eds): *Intelligent Information and Database Systems*. ACIIDS 2018. Lecture Notes in Computer Science, 10751. Springer, Cham

[34] Swiss Post. *Swiss WorldCargo and Matternet start drone tests*. Online communication, 2015.

[35] Tavana, Madjid, Kaveh Khalili-Damghani, Francisco J. Santos-Arteaga, and Mohammad H. Zandi. (2017) “Drone shipping versus truck delivery in a cross-docking system with multiple fleets and products” *Expert Systems with Applications*. **72**: 93-107.

[36] TSP test data, http://www.math.uwaterloo.ca/tsp/data/index.html, accessed February 2019.

[37] TSPLIB collection, https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, accessed February 2019.

[38] Vergouw, Bas, Huub Nagel, Geert Bondt, and Bart Custers. (2016) Drone technology: types, payloads, applications, frequency spectrum issues and future developments. in: Custers B.H.M. (red): *The Future of Drone Use. Opportunities and Threats from Ethical and Legal Perspective*. Springer, **27**: 21-45.

[39] Wang, Xingyin, Stefan Poikonen, and Bruce Golden. (2017) "The Vehicle Routing Problem with Drones: Several Worst-Case Results" *Optimization Letters*. **11 (4)**: 679-697.