Human Performance in Solving Multi-UAV Over-Constrained Dynamic Vehicle Routing Problems

Gupta, Ankit; Borst, Clark; Mulder, Max

DOI
10.1016/j.ifacol.2022.10.234

Publication date
2022

Document Version
Final published version

Published in
IFAC-PapersOnline

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-29 (2022) 73–78

Human Performance in Solving Multi-UAV Over-Constrained Dynamic Vehicle Routing Problems

Ankit Gupta, Clark Borst [1] and Max Mulder

Control and Simulation, Faculty of Aerospace Engineering, TU Delft, 2629 HS, Delft, The Netherlands

Abstract: For many logistics applications, such as drone delivery missions, finding an optimized network of routes yields a Vehicle Routing Problem (VRP). Such optimizations are mostly conducted offline prior to actual operations for reasons of computational complexity. In case disturbances arise during operations, for example a sudden loss of a vehicle, the VRP needs to be re-optimized in real-time and this raises concerns regarding obtaining a solution within time. In a previous study, it was demonstrated that humans, when supported through a human-machine interface, can quickly deal with these routing problems through satisficing, providing workable solutions. This paper extends our previous research by exposing human operators to an over-constrained VRP with different mission priorities and vehicle capabilities. Experiment results ($n = 16$) indicate that the mission type had the largest impact on how participants used the interface and what constraints were relaxed. In particular, during a search-and-rescue context the mission emphasis was put on delivering (medical) payload (close) to as many customers as possible, even if this would involve sacrificing vehicles and relaxing the depot constraint. Ethical aspects of the VRP are taken into account which algorithms do not by themselves, underlining the importance of involving humans in automation. Human operators complement algorithms with their context awareness, yielding more safe, resilient and responsible systems.

Keywords: Human operator support, Decision making and cognitive processes

1. INTRODUCTION

Vehicle routing problems (VRPs) play a central role in streamlining and optimizing many logistic challenges, for example drone delivery services and responding to emergencies (Dorling et al. (2017); Wang and Sheu (2019)). In its most succinct form, a VRP can be defined as *"the problem of designing optimal routes from a depot to geographically scattered customers, subject to side constraints"* (Laporte (1992)) and is illustrated in Figure 1.

The most basic VRP is the so-called Distance-Constrained Capacitated VRP (DCVRP). Here, the side constraints are the range endurance for each vehicle (governed by e.g., fuel capacity), vehicle payload and a depot constraint which limits the number of vehicles that can take-off and land simultaneously at the depot. Hence, every vehicle is expected to depart from the depot, deliver payload to its assigned customers and ultimately return to the depot.

Most VRP algorithms assume a well-defined problem and an unlimited number of vehicles. The objective is then to gather an optimal solution that would either account for the least number of vehicles or minimize the total travel cost while satisfying all constraints (Hiermann et al. (2016)). In reality and during operations, however, the number of vehicles is limited and part(s) of the problem may be unknown a priori and only revealed dynamically during the execution phase of the planned routes. Operational disturbances, such as changing demands from the customer or a vehicle failure, may require the routes to be re-optimized in real-time. For reasons of limiting the computational complexity and extensive modeling requirements involved in solving dynamic VRPs using algorithms, an alternative approach is to make use of human ingenuity in solving VRPs by creative and adaptive problem solving.

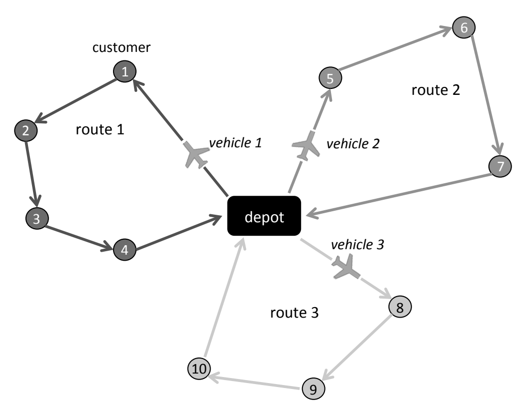


Fig. 1. Graphical illustration of the VRP [taken from Klein Koerkamp et al. (2019)].

In a previous study, Klein Koerkamp et al. (2019) demonstrated that humans are indeed capable to quickly deal with a dynamic DCVRP in real-time, when supported by a human-machine interface which visualizes the route patterns, mission constraints and depot arrival capacity

[1] E-mail: c.borst@tudelft.nl

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2022.10.234

over time and also allows for 'what-if?' probing and the manipulation of vehicle routes. In that study, the VRP involved one or two vehicle failures, but the problem could always be solved by using the remaining vehicles. In reality, a VRP could become over-constrained, meaning that vehicles would not be able to deliver to all customers while satisfying all constraints (Ding et al. (2005)). Over-constrained problems typically require certain constraints to be relaxed in order to find a (sub-optimal) solution. However, deciding on what constraint(s) to relax is often difficult as that may depend on the mission context. Such decisions could be motivated by factors that are not easily captured in algorithms, for example ethical considerations (Tsamados et al. (2022)). Hence, over-constrained problems may increase the need for more human involvement.

This study investigates how humans work with the previously-developed interface and decide on which constraints to relax when faced with an over-constrained DCVRP in two delivery missions having different contexts. To further increase the complexity of the DCVRP compared to the previous experiment, the scenarios in this study also included two vehicle types (having different capabilities in terms of battery capacity and travel speed) that may influence decisions and strategies.

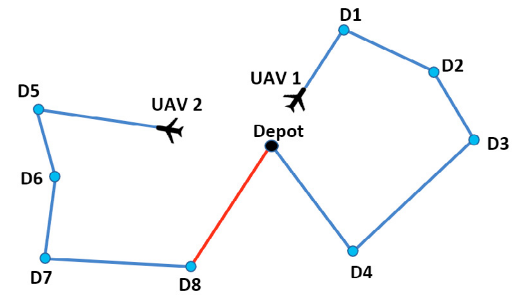
2. HUMAN-MACHINE INTERFACE

2.1 Extensions to Previous work

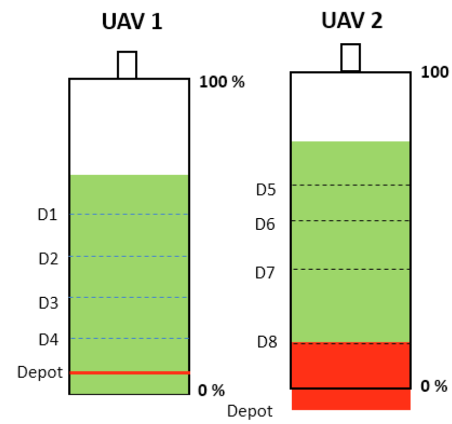
In the previous study by Klein Koerkamp et al. (2019) an interface prototype was developed in Java (using OpenGL graphics) that visualized the following information: 1) a map of geographical customer locations and their demands, 2) vehicle payload capacity, 3) vehicle routes, 4) and endurance/range ellipses (based on battery capacity), 5) temporal arrival schedule of the vehicles and 6) depot capacity limit. For the purpose of this study, two additional parameters needed to be visualized: 7) UAV battery capacity along route segments and 8) UAV icon size to distinguish between vehicle capabilities.

When dealing with dynamic and over-constrained problems, there might be a need to explicitly display the battery capacity of each UAV along each route segment. Such an indicator would allow the human operator to perceive the required battery level to complete the entire mission and mission elements (= route segments), the predicted battery level when changing the route and lastly, the current battery level. For this purpose, an interactive battery indicator, inspired by the work of Fuchs et al. (2014), has been added to the interface prototype, see Figure 2b. The battery indicator of a UAV is only shown when the human operator selects that UAV.

The green color represents that there is sufficient battery to cover all the waypoints. Anything below the red line represents the surplus in battery capacity after reaching the depot, as it can be seen for UAV 1 in Figure 2b. A battery capacity shortage is colored red, meaning that certain waypoints would not be reached. When considering UAV 2 in Figure 2b, it has insufficient capacity to return to the depot and will thus (crash)land anywhere between D8 and the depot. The same information is also depicted on the map view (Figure 2a), allowing operators to link battery



(a) Map view. showing the area covered by the two UAVs. The leg where UAV 2 will run out of battery is visualized in red.



(b) UAV 1 has enough energy to visit all the waypoints, however, UAV 2 does not have not have enough energy to reach the depot after delivering to customer D8.

Fig. 2. Side by side view of the map view along with the battery indicator.

capacity components to the different route segments and endurance range.

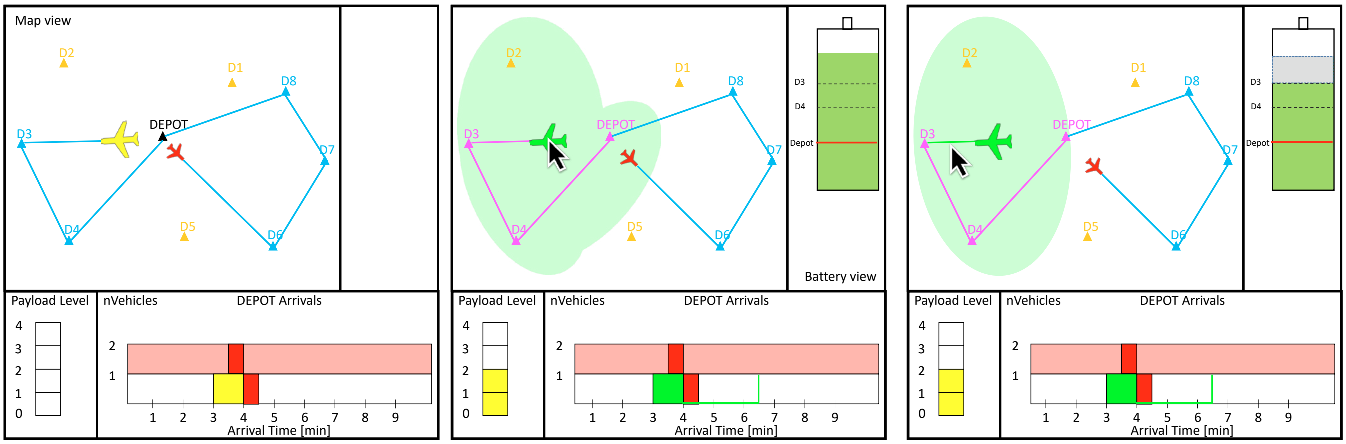
The battery indicator also allows for interaction; by hovering the mouse cursor over the different capacity segments, the corresponding route segments on the map view will be highlighted. Additionally, when the UAV route is manipulated on the map view, the predicted capacity will be displayed, allowing the operator to probe solutions before committing them.

Finally, different UAV icon sizes were used to indicate UAVs of different capabilities. For example, smaller UAVs that commonly travel with lower speeds and smaller batteries are displayed with small icons.

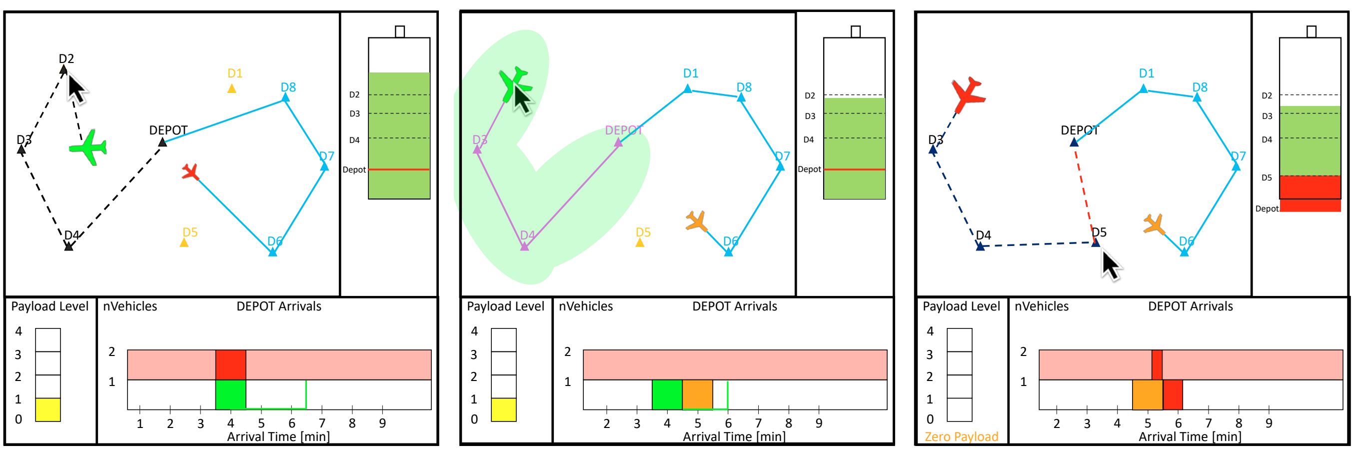
2.2 Layout, Structure and Functionality of the interface

Figure 3 shows the interface layout, structure and operating modes by using an example scenario. This scenario includes two vehicles with four payload levels each to deliver to five customers from the depot. In Figure 3a, the interface has four separate views: the map view, the payload details and the depot arrival schedule. The battery view and UAV range envelope (map view) will appear when clicking on an individual vehicle (see Figure 3b).

The range envelope, based on the UAV's battery capacity, represents the available re-routing space for the selected UAV. In this case, the selected UAV can be re-routed



(a) Mission overview. (b) Selecting a UAV. (c) Selecting a route segment.



(d) Including a customer. (e) Exploring more customer to add. (f) Solution after including the an out-of-range customer.

Fig. 3. Step-by-step overview of the interface workings for a simple scenario.

to include customer D2 as it lies within the envelope. To include D2 in the flight plan, the operator selects the nearest flight leg (Figure 3c) and clicks on the D2 point to include it. The battery indicator is then updated accordingly and when the operator is satisfied, she presses ENTER on the keyboard to commit the revised route.

In Figure 3d it can be seen that the depot capacity constraint is still violated, as the two UAVs are expected to arrive at the depot at the same time. Path-stretching actions can be undertaken to the other UAV to solve this depot capacity issue. Figure 3e illustrates that the selected UAV does not have sufficient battery capacity to fly to customer D5 *and* back to the depot. By previewing that action (Figure 3f), it can be seen that the UAV can deliver payload to D5, but will crashland somewhere between D5 and the depot. It is at the operator's discretion to either commit this action, resulting in a vehicle loss, or leave the route as it was and not deliver the payload to D5.

3. EXPERIMENT

The experiment aimed to examine how participants used the interface elements and what constraints were relaxed in an over-constrained DCVRP with different mission types.

3.1 Participants and instructions

Sixteen participants volunteered, all graduate students (10) or staff members (6) from Delft University of Technology (TU Delft), with an average age of 25.38 (SD = 7.19). The group had fourteen males and two females. Eight of the participants considered themselves as regular gamers.

After participants were briefed and trained on how to use the interface, they were instructed to solve several VRP scenarios to the best of their ability.

3.2 Independent variables

Three within-participant variables were manipulated:

(1) *Payload Capacity*: The problem size of the over-constrained DCVRP dependent on the payload capacity of a single UAV. There were four payload levels used: 4, 5, 6 and 7 payloads for each vehicle. Each of the vehicles was provided with the same number of payloads in every scenario.
(2) *Perturbation Severity*: To produce an over-constrained problem, each scenario was initiated by either a single low-battery level vehicle, or two low-battery level vehicles. The low-level vehicle was defined as the vehicle which will initially deliver to two customers while

having the least amount of battery capacity. In other words, this vehicle would have the highest payload margin with the lowest battery capacity. In that case, the operator would lose at least one vehicle if they choose to serve all customers in a scenario.

(3) *Mission Objective*: Two distinct delivery missions were chosen, one within a search-and-rescue context (delivering medical supplies) and one with supplying customers with commercial items (here, packages of coffee beans). Before the start of each scenario, the mission type for that scenario was verbally communicated to participants.

3.3 Scenarios

In all scenarios (lasting five minutes each), a batch of UAVs was deployed every thirty seconds (equal to the depot service time). The number of UAVs deployed was equivalent to the depot capacity. Only lateral waypoint modifications could be made. Additionally, there were two types of vehicles used in the scenario. One of them had a higher maximum flight time of 900 s and airspeed of 20 m/s, and the other vehicle had a maximum flight time of 750 s and an airspeed of 13 m/s. In case of an even number of vehicles, the two types of vehicles were equally divided, whereas, in the case of an odd number of vehicles, there was one more of the lower performing vehicle.

To create the scenarios for the experiment, there was an offline VRP optimization algorithm developed, which allowed the inputs for different properties of vehicles used and varying payload level. The scenario was first optimized for a static case (before the addition of customers during mission operation). The level of customers, in this case, was determined by the payload margin for each vehicle and the number of low-battery level vehicles. Disregarding the low-level vehicle, the rest of the fleet was given a payload margin of one during mission operation. Once the number of customers was determined, their locations were randomly generated. A minimum distance criterion was applied to avoid clusters of locations in one particular area. The optimizing algorithm then routed this static scenario considering the DCVRP. For this algorithm, the Google Optimization (Google-OR) Tools were adapted to feature the variable fleet capacity and its properties.

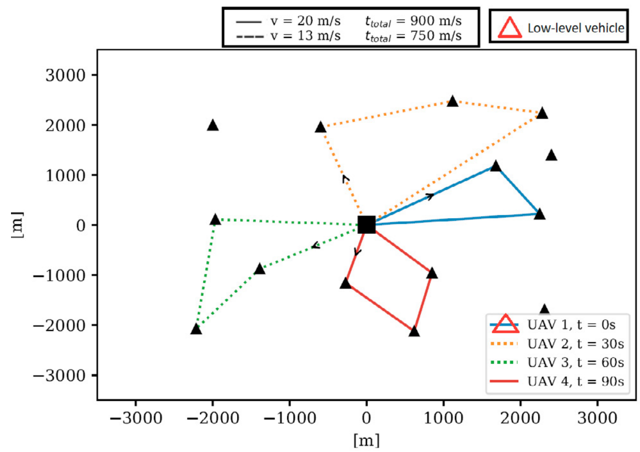
Once the static scenario was completed, the dynamic elements were added and the remaining customers were then placed randomly around the area. Two example scenarios are shown in Figure 4. To validate whether the scenarios were indeed over-constrained, it was run again through the Google optimization algorithm to check if the resulting scenarios indeed produced no solutions.

3.4 Control variables

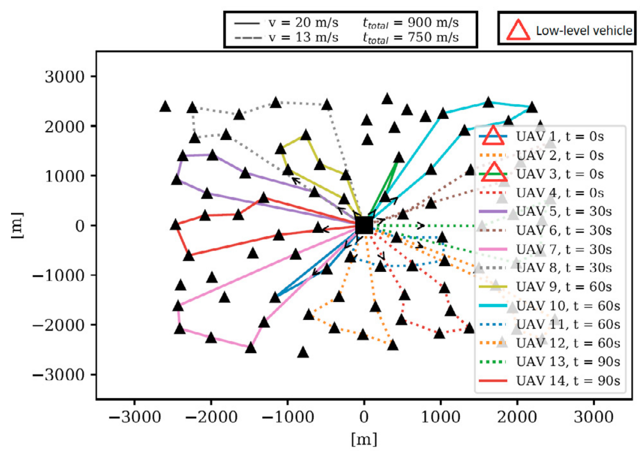
The control variables (summarized in Table 1) were the depot service time, sector size, depot capacity, scenario duration, UAV fleet, and payload margin for each vehicle.

3.5 Dependent Measures

It was of particular interest to learn what constraints participants preferred to relax (measured by the percentage of meeting and violating the constraints), how they



(a) Payload capacity 4 and one low-level vehicle.



(b) Payload capacity 7 and two low-level vehicles.

Fig. 4. Two of the scenarios which were given to the participants.

Table 1. Control variables in the experiment.

Variable	Value
UAV 1 - Max flight time [s]	900
UAV 1 - Airspeed [m/s]	20
UAV 2 - Max flight time [s]	750
UAV 2 - Airspeed [m/s]	13
Service Time [s]	30
Scenario Duration [s]	300
Payload Margin for high-battery vehicles [-]	1
Sector Size [m^2]	5000 x 5000
Depot Capacity	30% of nVehicles

approached solving the scenarios (observations during the experiment and post-experiment interviews) and the subjective workload (measured by Rating Scale Mental Effort (RSME) scores (Zijlstra, 1993) and mouse clicks).

4. RESULTS

4.1 Constraint adherence

Figure 5 shows the feasibility of each of the constraints for every condition and mission type. The constraints were considered infeasible if a solution was: 1) unable to serve as many customers as possible by using the limited payload of the vehicle, 2) overrunning the provided flight time limit of the UAVs (i.e., exceed battery capacity limits) and/or 3) by exceeding the depot capacity limit.

There appears to be no effect of payload level and the number of the low-level vehicles. From Figure 5a it can be seen that the main priority has been to serve all customers, especially in the search-and-rescue context. This priority came at the expense of violating the battery constraint and the depot constraint. In the context of delivering coffee beans, the emphasis was put on making sure the UAV returns to the depot rather than serving all customers.

4.2 Control strategy

How participants prioritized different aspects is also reflected by the remaining payload per experiment condition as shown in Figure 6. There is no effect on the number of low-level vehicles on the percentage of payload that is remaining at the end of the scenario. However, it can be seen that there is an effect caused by the payload level and the mission type. As the payload capacity increases, the percentage of payload remaining decreases. A repeated-measures ANOVA adopting a threshold of $\alpha = 0.05$ confirmed this $(F(3, 45) = 100.899, p < 0.01)$. When observing the different mission types, there is also a higher payload margin remaining in case of delivering coffee beans compared to the search-and-rescue mission $(F(1, 15) = 68.655, p < 0.01)$.

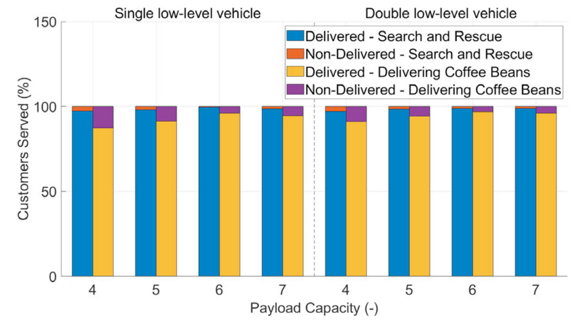
Observations during the experiment showed that some participants adopted an interesting strategy. While sacrificing the vehicles in a search-and-rescue context, some participants were able to deliver the payload to as many customers as possible and made sure the UAV would only fail when flying its last segment towards the depot. The underlying rationale was that the failed vehicle would drop down somewhere close to the depot, making it more convenient to pick it up later.

Additionally, the strategy adopted by most participants was to use the vehicles with the highest battery capacity and higher speed to go to customers further away from the depot, whereas the vehicles with the lower battery capacity and speed were used to provide the customers closer to the depot. Moreover, participants tended to sacrifice the smaller UAV to provide to as many customers as possible.

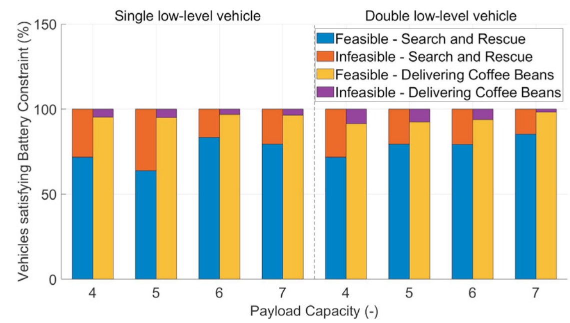
4.3 Workload

Figure 7 shows the clustered boxplots of the RSME subjective workload scores for each condition. The mission type had a significant effect on the perceived workload $(F(1, 15) = 9.784, p < 0.05)$ as well as number of low-battery level vehicles $(F(1, 15) = 59.331, p < 0.01)$. In a search-and-rescue context participants apparently felt a higher pressure to serve all customers and make a decision on what constraint(s) to relax.

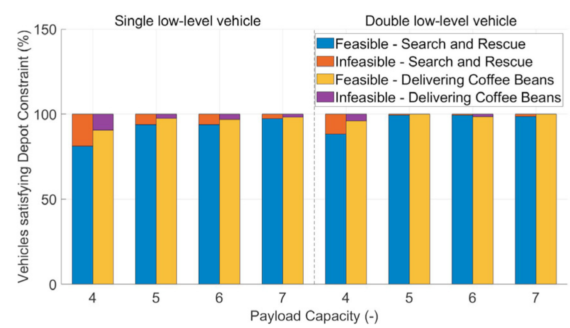
Regarding physical workload (mouse click events), Figure 8 shows that the click events have opposite trends compared to the subjective workload, especially for 'smaller' problems (in terms of low-battery level vehicles and payload capacities). In a search-and-rescue context, subjective workload is higher than in a generic delivery mission, but the number of mouse clicks is significantly lower



(a) Percentage of customers served.



(b) Percentage of vehicles satisfying the battery constraint.



(c) Percentage of vehicles satisfying the depot constraint.

Fig. 5. Clustered Bar chart for the feasibility of each of the constraints for every condition and mission objectives.

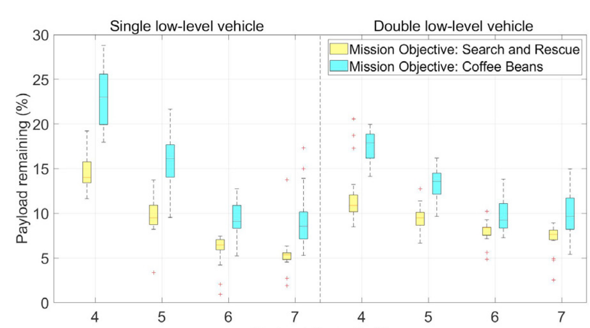


Fig. 6. Clustered Box plot for the total payload remaining after each condition and mission objectives.

$(F(1, 15) = 4,929, p < 0.05)$. This can be explained by the fact that when delivering coffee beans, participants more often tweaked flight routes to optimize solutions, whereas in the search-and-rescue mission the preference was to satisfice. This behaviour can clearly be observed by the number of path-stretch events shown in Figure 9.

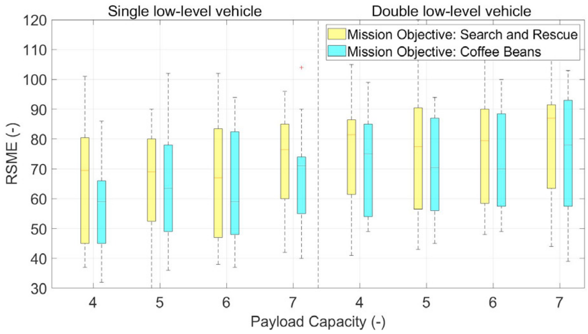


Fig. 7. Clustered box plots of the Rating Scale Mental Effort (RSME) scores.

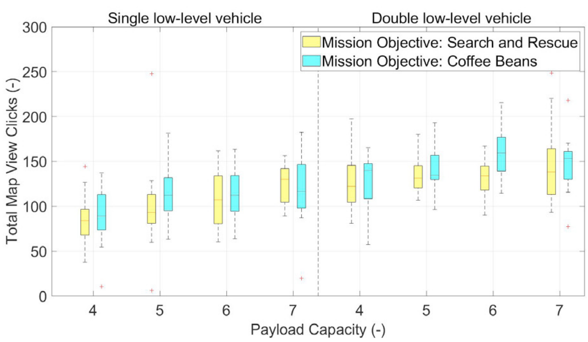


Fig. 8. Clustered box plots for the total map view clicks.

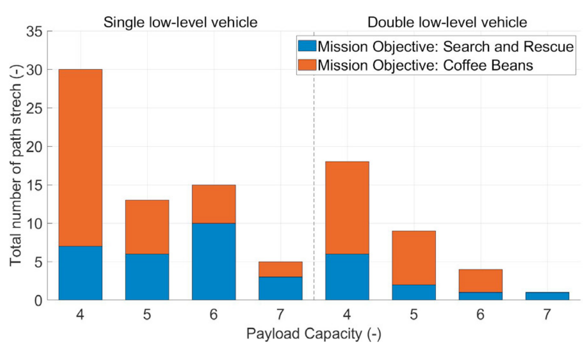


Fig. 9. Bar chart of the total path stretch events.

5. DISCUSSION AND RECOMMENDATIONS

The work described in this article aimed to let humans solve an over-constrained Vehicle Routing Problem (VRP) of various complexities by means of a human-machine interface. Results from a human-in-the experiment ($n = 16$) showed that participants were able to use the interface in formulating workable solutions, despite the increased scenario complexity compared to the previous experiment conducted by Klein Koerkamp et al. (2019).

Most interestingly, participants' control strategy and the decision on what constraints to relax changed with the mission type. When faced with a delivery mission in a search-and-rescue context, the emphasis was put on delivering to as many customers (far away from the depot) as possible, even when this would involve sacrificing vehicles. In a more generic mission of delivering coffee beans, the participants' priorities changed and the main objective became to let the vehicle safely return to the depot, even if this would lead to unserved customers.

The main lessons one can learn from this result are three-fold. First, we can learn from human solutions how to (re-)shape computer algorithms such that they generate solutions that are most appropriate for the problem context (in terms of respecting human values). Second, human involvement in automated processes can be beneficial, especially in cases where the automation cannot find a solution and creative problem solving is required. A prerequisite for successful human involvement, however, is the availability of an effective human-machine interface that allows operators to perceive the problem's structure and all degrees of freedom to formulate solutions and act accordingly. Third, being ethical creatures, humans can adopt strategies unforeseen by and unsupported by optimization algorithms. Without being instructed to do so, our participants changed their strategy based on the moral consequences of failing either the customer, or sacrificing the vehicle. Faced with the ethical dilemma posed by the experimental scenarios, most of our participants adopted a 'humans come first, technology comes second' approach.

It is recommended to study how operators can use the interface in complementing an algorithm in real-time. That is, when faced with an over-constrained problem due to unanticipated circumstances during operations, humans could use the interface to adjust the mission objective *and* its constraints, then an algorithm could optimize the solution within the new objective. True human-automation teamwork could then be achieved.

REFERENCES

Ding, H., Lim, A., Rodrigues, B., and Zhu, Y. (2005). The over-constrained airport gate assignment problem. *Computers & Operations Research*, 32(7), 1867–1880.

Dorling, K., Heinrichs, J., Messier, G.G., and Magierowski, S. (2017). Vehicle routing problems for drone delivery. *IEEE Trans. on SMC: Systems*, 47(1), 70–85.

Fuchs, C., Borst, C., De Croon, G.C.H.E., Van Paassen, M.M., and Mulder, M. (2014). An ecological approach to the supervisory control of uav swarms. *International Journal of Micro Air Vehicles*, 6(4), 211–229.

Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R.F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *Eur. J. of Operational Research*, 252(3), 995–1018.

Klein Koerkamp, N.W., Borst, C., Mulder, M., and Van Paassen, M.M. (2019). Supporting Humans in Solving Multi-UAV Dynamic Vehicle Routing Problems. In *14th IFAC Symp. on Human-Machine Systems*, 359–364.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3), 345–358.

Tsamados, A., Aggarwal, N., Cowls, J., Morley, J., Roberts, H., Taddeo, M., and Floridi, L. (2022). The ethics of algorithms: Key problems and solutions. *AI & SOCIETY*, 37, 215–230.

Wang, Z. and Sheu, J.B. (2019). Vehicle routing problem with drones. *Transp. Res. Part B: Meth.*, 122, 350–364.

Zijlstra, F.R.H. (1993). *Efficiency in Work Behaviour – a design approach for modern tools*. Ph.D. dissertation, TU Delft.