

Reactive Dubins Traveling Salesman Problem for Replanning of Information Gathering by UAVs

Robert Pěnička¹, Martin Saska¹, Christophe Reymann² and Simon Lacroix²

Abstract—We introduce a novel online replanning method for robotic information gathering by Unmanned Aerial Vehicles (UAVs) called Reactive Dubins Traveling Salesman Problem (RDTSP). The considered task is the following: a set of target locations are to be visited by the robot. From an initial information gathering plan, obtained as an offline solution of either the Dubins Traveling Salesman Problem (DTSP) or the Coverage Path Planning (CPP), the proposed RDTSP ensures robust information gathering in each given target location by replanning over possible missed target locations. Furthermore, a simple decision making is a part of the proposed RDTSP to determine which target locations are marked as missed and also to control the appropriate time instant at which the repair plan is inserted into the initial path. The proposed method for replanning is based on the Variable Neighborhood Search metaheuristic which ensures visiting of all possibly missed target locations by minimizing the length of the repair plan and by utilizing the preplanned offline solution of the particular information gathering task. The novel method is evaluated in a realistic outdoor robotic information gathering experiment with UAV for both the Dubins Traveling Salesman Problem and the Coverage Path Planning scenarios.

I. INTRODUCTION

This paper presents a new robust approach for solving the robotic information gathering by Unmanned Aerial Vehicles. In this task, a set of locations is to be visited by the robot to collect sensory data. The proposed method called Reactive Dubins Traveling Salesman Problem (RDTSP) is an online approach that uses the UAV's onboard computational resources for replanning and decision making to ensure that all given target locations are visited, regardless of possible disturbances during the information gathering task, that can lead to missing measurements from some of the target locations.

The first and typical application of the proposed method arises in the Coverage Path Planning (CPP) [1] scenarios where a predefined region is given, and the task is to find appropriate waypoints (sensor measurement locations) such that the whole area is scanned by the onboard sensor. For the aerial robotics, a simple zigzag path (further also called 'sweeping' path) is usually sufficient as there is no need for avoiding obstacles during the coverage mission.

The second considered scenario is the information gathering specified as the Dubins Traveling Salesman Problem (DTSP) [2], where its multi-robot case is shown in

¹Authors are with Faculty of Electrical Engineering, Czech Technical University, Technická 2, Prague, Czech Republic {robert.penicka|martin.saska}@fel.cvut.cz

²Authors are with LAAS-CNRS, INSA, Université de Toulouse, 7 Avenue du Colonel Roche, Toulouse, France {christophe.reymann|simon.lacroix}@laas.fr

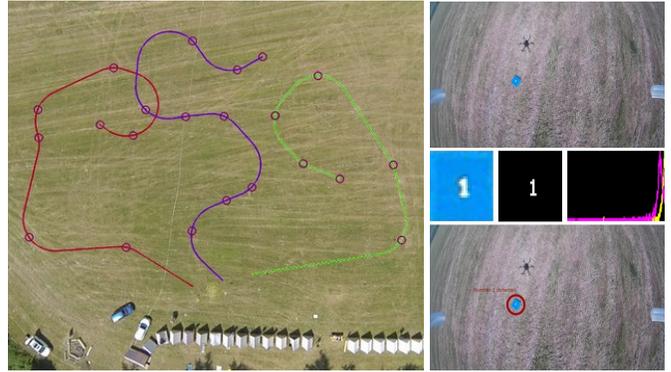


Fig. 1: Example of information gathering as a solution of Dubins Traveling Salesman Problem for multiple UAVs (MDTSP) on left, and the data collection as a camera detection of colored objects with number recognition (on right)

Figure 1. The DTSP is a variant of the well known Traveling Salesman Problem (TSP) [3] for Dubins vehicle, which we use as a simple model of considered UAV for constructing smooth constant speed paths. Contrary the CPP, the DTSP uses a predefined set of target locations, and the task is to find a minimal length path over all target locations. Notice that after finding the appropriate waypoints in the area, the CPP is a special case of the TSP or in our case a special case of the DTSP.

Using UAVs for obtaining sensory data in such predefined locations in the environment is always influenced by uncertainties in robot motion, due to wind disturbances and imprecision of the dynamic model, among other factors. Another source of uncertainty is introduced into the system by imperfect sensors used for the particular application. All these aspects together may cause that some places of interest are not perceived, or the obtained information is not sufficient. Often it is possible to detect online (during the mission) that a required location was not visited properly by identification of a deviation from the pre-planned trajectory, or that the obtained sensory measuring is not valid. The current approach is to collect all the locations with identified missing sensory information after the mission and to plan an additional flight over these places to complement the data. The proposed RDTSP, however, identifies the missing target locations online and applies the repair plans during the mission.

The typical solution of the information gathering mission either for CPP or DTSP is done offline prior to the mission, and during the execution, a simple trajectory following is used. However, defining the path for sweeping scenarios is rather simple by using the zigzag path, the solution of NP-hard DTSP is computationally demanding. The optimal solution of the DTSP (for a given sampling of heading angles of Dubins vehicle) can be obtained using a transformation of the DTSP into an Asymmetric TSP (ATSP) [4] and then solved by e.g. Concorde TSP solver [5]. Therefore the further introduced RDTSP method uses the offline precalculated path for the information gathering and replans over the missed target locations such that the new repair part of the plan (generally a constrained solution of DTSP over missed locations) is inserted into the original plan.

The proposed Reactive Dubins Traveling Salesman Problem is an online algorithm that uses the predefined path from either CPP or DTSP and ensures the visit of all target locations by replanning over the missed target locations. Since the optimal solution of the DTSP is computational demanding, the proposed RDTSP method utilizes the pre-computed plan and inserts the repair plan over the missed target locations into the existing plan between adjacent target locations. The existing approaches for the information gathering and surveillance by UAVs [6], [7], [8], [9] focus on the planning of the whole mission offline (as a solution of the DTSP) and do not consider possible replanning over missed target locations. The proposed VNS-base DTSP planner keeps planning during the mission and minimizes the overall path length not only by considering a different sequence of visiting the missed targets, but also by finding the location of applying the repair plan. Furthermore, a decision making approach is used to determine when the replanning is applied to revisit the missed locations and also to determine which target locations are considered as missed.

The remainder of this paper is organized as follows. The problem statement of the proposed RDTSP is specified in the next section. The overall system architecture is introduced in section III. Section IV describes the proposed method for RDTSP. First experimental results are presented in V and section VI concludes the paper.

II. PROBLEM STATEMENT

The proposed Reactive Dubins Traveling Salesman Problem is based on the idea of utilizing the offline computed initial plan for information gathering in the online iterative improvement of the repairing plan that covers the missed target locations. The RDTSP method uses an initial path P_{init} which is a feasible Dubins path for information gathering over the required target locations. We expect the target locations $S = \{s_1, \dots, s_n\}$ to be ordered according to the occurrence inside P_{init} .

For the purpose of RDTSP for the targeted Unmanned Aerial Vehicle, we use the kinematic model of Dubins vehicle [10]. The state of the vehicle $q = (p, \theta)^T = (x, y, \theta)^T$ is described by the position $p = (x, y) \in \mathbb{R}^2$ and the

heading angle $\theta \in \mathbb{S}^1$. We expect a constant velocity v of the information gathering vehicle that is controlled by input u to either go straight or turn.

$$\dot{q} = \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{u}{\rho} \end{bmatrix}, u \in [-1, 1]. \quad (1)$$

The Dubins vehicle (1) is specific for its minimal turning radius ρ which can be, for the considered UAV, derived from the equation of circular motion $\rho = v_d^2/a_{max}$, where v_d is desired constant speed and a_{max} is the maximal acceleration allowed by the UAV. The distance $\mathcal{L}_d(q_{\sigma_i}, q_{\sigma_j})$ between two states q_{σ_i} and q_{σ_j} of the Dubins vehicle is then the shortest Dubins maneuver out of the six possible maneuvers [10].

For the RDTSP we expect an online deployment where the replanning of the whole initial path after missing some locations is not possible due to the fact that the DTSP is NP-hard, thus very computationally demanding for a large number of target locations. Instead, we propose the VNS-based method that uses the initial DTSP plan, and try to find a Dubins path over the missed locations such that the path can be inserted into the original plan between two adjacent locations.

After starting the execution of the initial plan, the current position inside the plan is described by an index number $c \in (1, n)$ that belongs to the location s_c which is the last location that has been attempted to visit. The set S_m contains k missed target locations. A solution of the RDTSP can be described as a permutation $\Sigma = (\sigma_1, \dots, \sigma_{k+2})$ to visit the missed locations in S_m , where $\sigma_1 > c$ is location on the initial plan somewhere in the future, $\sigma_{k+2} = \sigma_1 + 1$ is the location where the repair plan connects back to the initial plan, and $s_{\sigma_i} \in S_m$ for $i \in (2, k+1)$ being the missed target locations.

Furthermore the considered Dubins vehicles requires determining respective heading angles at the target locations from Σ . The heading angles can be described as a vector $\Theta = (\theta_{\sigma_1}, \dots, \theta_{\sigma_{k+2}})$, where θ_{σ_1} and $\theta_{\sigma_{k+2}}$ are set to the values from the initial plan P_{init} and the heading angles at the missed target locations need to be found together with the permutation of the missed locations Σ .

In other words we want to find the starting location inside the future part of the plan, from which the repair plan would be started, together with the sequence of visits to the missed locations such that the produced path has minimal length. The addressed RDTSP can be described as an optimization problem to minimize the length L_{path} of the Dubins tour over the missed target locations:

$$\begin{aligned} \underset{\Sigma, \Theta}{\text{minimize}} \quad & \mathcal{L}_{path} = \sum_{i=2}^{k+2} \mathcal{L}_d(q_{\sigma_{i-1}}, q_{\sigma_i}) \\ \text{subject to} \quad & \sigma_1 = s \in (c+1, n), \\ & \sigma_{k+2} = \sigma_1 + 1. \end{aligned} \quad (2)$$

III. SYSTEM ARCHITECTURE

The software architecture of the proposed system is built on top of the system developed at Czech Technical University and University of Pennsylvania for our participation at the MBZIRC competition in Abu Dhabi (see [11] for a description of the initial version of the CTU system primarily designed for formation flying [12], [13], [14] and swarm applications [15], [16], [17]; experiments with the current version of the MBZIRC system can be found at <http://mrs.felk.cvut.cz/projects/mbzirc>). The core of the system, based on Robot Operating System (ROS), is a Model Predictive Control (MPC) algorithm capable of following any given trajectory feasible for UAVs (the MPC controller was built from our solution designed for flying in GPS-denied environment, which is capable of UAV stabilization and trajectory tracking using only embedded micro-controller [18], see <http://mrs.felk.cvut.cz/projects/cesnet> for deployment of the system in task of scanning of large historical buildings). The system allows giving a trajectory specified by a sequence of waypoints sampled at a predefined frequency, which is post-processed and smoothed using known UAV motion constraints. For precise trajectory following, a state estimation mechanism is integrated fusing data from a variety of sensors. GPS data, IMU output, an output from image processing from two cameras, a range finder for altitude measurement, and in some applications also Differential GPS can be used in the feedback of the controller.

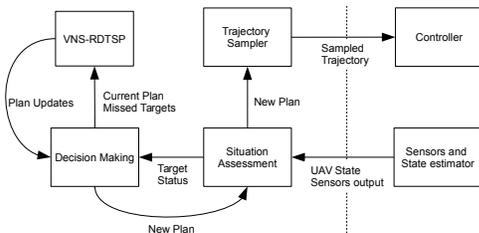


Fig. 2: Software architecture. Arrows represent messages, blocks ROS components. The dashed line symbolizes communication with lower level components.

Figure 2 illustrates the components of the system relevant to the scope of this paper. The mission consists of visiting a predefined list of targets (positions where the sensory data have to be captured), which are collected into a sweeping trajectory or for which a predefined DTSP solution can be computed in advance and fed as an input to the system. The *situation assessment* node processes the sensory outputs as well as the current estimated UAV state to track the status of the current target in the plan. When flying over a target, this node will assess if the measurements are good enough and upon failure of the sensor reading or error in position exceeding an allowed threshold, declares the target as *missed*.

This status is passed to the *decision making* node, which gathers the list of missed targets. Positions of the missed targets are used by the *VNS-DTSP* node (we use VNS - Variable Neighborhood Search based DTSP) solving the defined RDTSP problem to repair the plan by optimizing a

DTSP tour over these targets. Each time the *VNS-DTSP* node finds a better solution, in the form of a local plan update including at least one previously missed node, a decision is made whether to use it or wait for a potentially better one. When the updated plan is deemed good enough for its execution, it is fed back to the *situation assessment* node which waits for the UAV to be in a state that allows seamless switching to the new plan, which is then sampled, post-processed and sent to the controller.

IV. PROPOSED APPROACH FOR THE RDTSP

Because the Dubins TSP problem is NP-hard, the exact optimization scheme requires, in general, a long computation time and is not suitable to find good solutions in online settings. In fact, no algorithm is capable of finding optimal solutions to the DTSP (without sampling the heading angles) for very large number of points at this moment, which is the case of our predefined overall mission plan. Moreover, in most of the information gathering applications, a simple sweeping or DTSP trajectory over the required places of interest is preferred, and there is no need to change the overall plan completely. Therefore we seek only to locally repair the given plan by re-visiting the missed sensory locations during the mission.

As new missed targets may come in at any moment, and due to the anytime nature of the VNS algorithm, we need to be able to decide when to stop the optimization and switch to the current best plan. We control the *VNS-DTSP* scheme from the *decision making* node by providing:

- the current plan, serving as a blueprint for plan updates
- a list of missed targets S_m , to be inserted in the current plan
- current position c inside the current plan

The decision to stop optimizing the current set of missed targets and to execute the newly adapted repair plan is then taken based on three criteria: the proximity to the node at which the current plan update starts (the plan is updated just before reaching the repair plan start), the time elapsed since the current best solution was found, and based on number of missed locations together with chance of missing currently approaching location.

A. Reactive Dubins Traveling Salesman Problem

The proposed solution of the Reactive Dubins Traveling Salesman Problem (RDTSP) is based on the Variable Neighborhood Search (VNS) metaheuristic for combinatorial optimization [19]. The used VNS method iteratively switches between *shake* and *local search* procedures, where the *shake* randomly changes the actual best solution of the problem and *local search* then tries to find on such randomly created solution a new and better solution than the currently best one. VNS employs a predefined neighborhood structures $N_{1, \dots, l_{max}}$ further described as an operation on the problem solution which is the planned repair path. A Randomized variant of the VNS (RVNS) is used, which tries iteratively a number of same random operations during the *local search* procedure.

For finding the appropriate heading angles at the missed target locations, we propose a sampling based approach where the heading angles $\theta_i \in (0, 2\pi)$ are equidistantly sampled into m values. The heading angles are then determined by a graph search over all heading samples for location sequence Σ and the ones with the minimal path length are used.

The proposed VNS-based solution of RDTSP uses following neighborhood structures. The randomized *shake* procedure uses:

- **Path Move** ($l=1$) randomly selects part of the repair plan between s_{σ_2} and $s_{\sigma_{k+1}}$, and moves it to a different position inside the plan.
- **Path Exchange** ($l=2$) works similar to the path move, but selects two random non overlapping parts of the plan between s_{σ_2} , $s_{\sigma_{k+1}}$ and exchange their position.
- **Start Exchange** ($l=3$) is a special neighborhood structure that randomly changes the location s_{σ_1} where the repairing path starts which also changes the end of repairing plan to $s_{\sigma_{k+2}} = s_{\sigma_1} + 1$. To limit the computational demand and also to focus on prior start of the repairing path, the start exchange is limited by the maximal distance d_{max} which is a number of locations ahead of the current one s_c to which the s_{σ_1} is randomly changed.

During one iteration of the *local search* procedure, the Randomized VNS tries one of the following neighborhood structures for a number of time that is equal to k^2 .

- **Point Move** ($l=1,3$) randomly moves one location inside the solution into different position inside the plan.
- **Point Exchange** ($l=2$) selects randomly two target locations and switch their positions.

The VNS-based algorithm for the proposed RDTSP is summarized in Alg. 1. The planning algorithm is started every time S_m contains at least one missed location and the local optimization keeps trying to improve the repair path until it is applied by the *decision making* node which clears the set of missed target locations S_m .

The algorithm is written for the online execution during the mission, where both the set of missed locations S_m and the current index of location c are changing. During the deployment, the current index c increases which requires to change the considered starting locations s_{σ_1} and ending locations $s_{\sigma_{k+2}}$ of the repair path to be $s_{\sigma_1} > c$ and $s_{\sigma_{k+2}} = s_{\sigma_1} + 1$ which is done by *adjustStart* on line 3. Whenever the c increases, the algorithm changes automatically the starting position to the one inside d_{max} that produces the shortest repair path. Furthermore, the introduction of new missed locations into S_m requires addition of the location to the existing repair plan (*addToPath* on line 5) which is done greedily. By this manner, the proposed VSN-base algorithm for RDTSP benefits from reusing the existing repair plan during the deployment and tries to optimize it to minimize the additional path length required to revisit the missed target locations.

Algorithm 1: VNS based method for the RDTSP

Input : P_{init} – initial plan
Input : d_{max} – start lookahead distance
Input : l_{max} – maximal number of neighborhoods
Input : m – Number of heading angles samples
Changing Input: S_m – Set of missed locations
Changing Input: c – actually passed location index in P_{init}
Output : P – Actual RDTSP repair plan

```

1  $P \leftarrow \text{createInitialPath}(S_m)$  ; // greedily
2 while Repair plan is not applied do
3   if  $c$  changes then
4      $\text{adjustStart}(P, c)$  ; // ensures  $s_{\sigma_1} > c$ 
5   if new missed target  $s_i \in S_m$  then
6      $\text{addToPath}(P, s_i)$  ; // greedily
7    $l \leftarrow 1$ 
8   while  $l \leq l_{max}$  do
9      $P' \leftarrow \text{shake}(P, l)$ 
10     $P'' \leftarrow \text{localSearch}(P', l)$ 
11    if  $\mathcal{L}_{path}(P'') < \mathcal{L}_{path}(P)$  then
12       $P \leftarrow P''$ 
13       $l \leftarrow 1$ 
14    else
15       $l \leftarrow l + 1$ 

```

V. EXPERIMENTAL RESULTS

The proposed RDTSP method has been tested in a realistic outdoor experiment with the UAV depicted in Fig. 3. The Dubins vehicle model is used for the hexarotor UAV to produce smooth paths over the target locations using constant speed trajectories that are preferred due to the precision of sensory measurements in the information gathering scenario.



Fig. 3: The hexarotor UAV used during the experimental verification of the proposed RDTSP in scenario with initial plan obtained as the DTSP (for more details see <http://mrs.felk.cvut.cz/ecmr17rdtsp>)

Two different scenarios have been used to verify the performance of the online replanning of the RDTSP. The first scenario is based on precomputed coverage path plan ('sweeping' plan) where the given area is covered by the simple zigzag plan (see section V-A). The second scenario uses an offline open-loop DTSP plan over specified target locations. Section V-B describes the second scenario.

For both scenarios¹, the same configuration of the proposed RDTSP has been used. The maximal distance in which the VNS-based method considers the disconnection from the original plan and applying the repair path has been set to $d_{max} = 20$. The nominal forward speed of UAV during the experiment has been set to constant speed $v_c = 2.5 \frac{m}{s}$. The RDTSP further uses the number of sampled heading angles $m = 16$ and turning radius of the Dubins vehicle $\rho = 4m$. The turning radius has been selected with respect to the maximal acceleration $a_{max} = 3 \frac{m}{s^2}$ of the used hexarotor UAV.

The *decision making* node in the conducted RDTSP experiments has been used for determining whether the specified target locations were visited and also for determining when the calculated repair plan is applied. The comparison of the target locations position and UAV odometry was used to assess the missing of a target location with decision tolerance distance of 1m. Furthermore, to test the real replanning ability of the proposed method, the decision making used an artificially introduced probability $p_{miss} = 0.2$ of missing the target location (which simulates an error in the sensory data measurements). Furthermore, the decision of applying the repair path (also done by the *decision making* node) has been set to a simple variant where the new plan is applied every time the UAV reaches a starting position of the repair path with two or more missing target locations.

A. RDTSP in coverage scenario

The conducted experiment of RDTSP in the Coverage Path Planning scenario uses a simple zigzag plan over the specified area with equidistantly sampled target locations (see Figure 4).

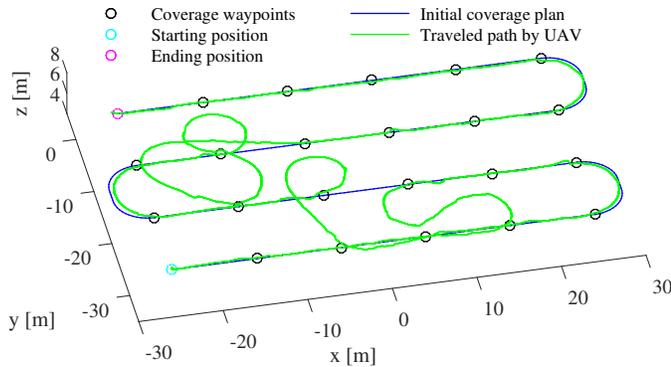


Fig. 4: Initial coverage zigzag path plan between starting and ending positions with equidistantly sampled target locations. The positions of the UAV ('Traveled path by UAV') shows the odometry measurements of the UAV that includes the two applied repair plans to revisit the target locations missed during the mission.

¹We refer to <http://mrs.felk.cvut.cz/ecmr17rdtsp> for more information and video from the outdoor experiment that further visualizes the real time performance of RDTSP method.

During the experiment, a total number of six target locations were missed, either due to the real displacement of the UAV position from the original plan or due to the artificially introduced random missing of the target. The initial three targets, as shown in Fig. 5, were revisited by the first repair path and the other three by the second replanning, always by connecting the repair plan between two adjacent target locations on the original coverage plan.

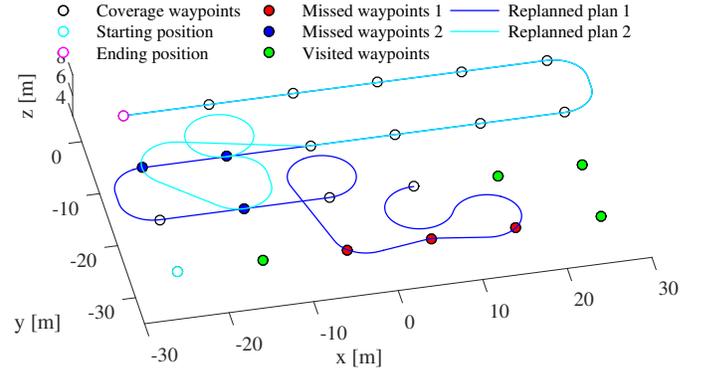


Fig. 5: Two repair plans applied by the RDTSP method during the execution of the coverage scenario.

B. RDTSP in DTSP scenario

In the RDTSP scenario with the initial path as the Dubins Traveling Salesman Problem, a total number of 20 target locations were specified. The initial plan, an open-loop version of the DTSP, was found by conversion of the DTSP into the ATSP and then optimally solved using the Concorde solver. The results of the experiment are depicted in Figure 6 and Figure 7, where the two missed target locations at the beginning of the mission are revisited by one repair path that is significantly shorter compared to the case of visiting the missed locations after the mission.

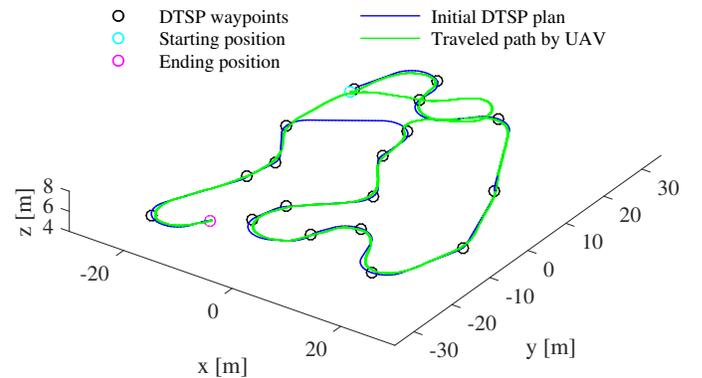


Fig. 6: The initial Dubins Traveling Salesman Problem plan over specified target locations together with the odometry measurements of the UAV including the additional repair path over missed target locations.

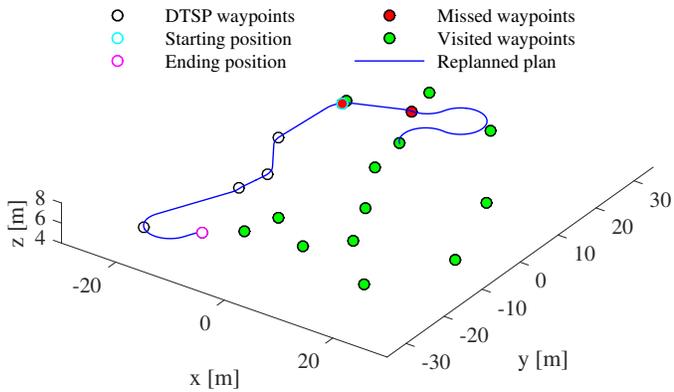


Fig. 7: The repair path applied after missing two target locations in the beginning of the DTSP scenario

The experimental verification shows that the proposed RDTSP is a robust method for the information gathering tasks, both for the CPP and DTSP scenarios, where all the missed target locations were revisited by the repair paths. The introduced online method minimizes the overall length by connecting the repair plan into the initially computed offline plan. Especially Figure 6, with the DTSP scenario, shows the benefits of the online replanning with lookahead distance, where the final path length is minimized by selecting the appropriate target location of connection the repair plan.

VI. CONCLUSIONS

In this paper, we introduced a novel method for information gathering by UAVs called Reactive Dubins Traveling Salesman Problem. The method uses a precomputed information gathering path from either the Coverage Path Planning for scanning of a given area or from the Dubins Traveling Salesman Problem where the length of the path over all given target locations is minimized. The proposed VNS-based online algorithm for the RDTSP uses the pre-defined path during the mission execution and ensures the robustness of information gathering task by replanning the initial plan for visiting possible missed target locations due to disturbances in sensory measurements or localization. The proposed method was verified in realistic outdoor information gathering experiments with hexarotor UAV and shows the robustness of the proposed online replanning approach in both CPP and DTSP scenarios. For the future work, we intend to investigate different strategies and improvements for the decision making, where the application of the repair plan is based on the real sensor measurements.

ACKNOWLEDGMENT

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 17-16900Y. The support of the Grant Agency of the Czech Technical University in Prague under grant No.

SGS17/187/OHK3/3T/13 to Robert Pěnička is also gratefully acknowledged. The support of the Czech Ministry of Education, Youth and Sports and of the French Ministries of Foreign Affairs for the Czech-France MOBILITY project 7AMB16FR017 is greatly appreciated.

REFERENCES

- [1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258 – 1276, 2013.
- [2] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and traveling salesman problems for dubins' vehicle," in *ACC*, 2005, pp. 786–791 vol. 2.
- [3] C. Rego, D. Gamboa, F. Glover, and C. Osterman, "Traveling salesman problem heuristics: Leading methods, implementations and latest advances," *European Journal of Operational Research*, vol. 211, no. 3, pp. 427 – 441, 2011.
- [4] C. E. Noon and J. C. Bean, "An efficient transformation of the generalized traveling salesman problem," *INFOR: Information Systems and Operational Research*, vol. 31, no. 1, pp. 39–44, 1993.
- [5] "Concorde tsp solver," cited on 2017-05-14. [Online]. Available: <http://www.math.uwaterloo.ca/tsp/concorde.html>
- [6] K. J. Obermeyer, "Path planning for a UAV performing reconnaissance of static ground targets in terrain," in *AIAA Conf. on Guidance, Navigation and Control*, Chicago, IL, USA, Aug. 2009.
- [7] J. Faigl and P. Váňa, *Self-Organizing Map for the Curvature-Constrained Traveling Salesman Problem*. Cham: Springer International Publishing, 2016, pp. 497–505.
- [8] X. Yu and J. Y. Hung, "A genetic algorithm for the dubins traveling salesman problem," in *2012 IEEE International Symposium on Industrial Electronics*, May 2012, pp. 1256–1261.
- [9] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, Sept 2009, pp. 82–85.
- [10] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [11] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots.*, vol. 41, no. 4, pp. 919–944, 2017.
- [12] M. Saska, V. Vonasek, T. Krajnik, and L. Preucil, "Coordination and Navigation of Heterogeneous MAV&UGV Formations Localized by a "hawk-eye"-like Approach Under a Model Predictive Control Scheme," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, 2014.
- [13] M. Saska, V. Spurny, and V. Vonasek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, vol. 75, no. Part B, pp. 379–397, 2016.
- [14] M. Saska, T. Krajnik, V. Vonasek, Z. Kasl, V. Spurny, and L. Preucil, "Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups," *Journal of Intelligent and Robotic Systems*, vol. 73, no. 1–4, pp. 603–622, 2014.
- [15] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, "Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles," *Journal of Intelligent & Robotic Systems.*, vol. 84, no. 1, pp. 469–492, 2016.
- [16] M. Saska, J. Vakula, and L. Preucil, "Swarms of Micro Aerial Vehicles Stabilized Under a Visual Relative Localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [17] M. Saska, "MAV-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.
- [18] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *MMAR*, 2016.
- [19] P. Hansen and N. Mladenovi, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.