Multi-Vehicle Close Enough Orienteering Problem with Bézier Curves for Multi-Rotor Aerial Vehicles

Jan Faigl, Petr Váňa, and Robert Pěnička

***Abstract*— This paper introduces the Close Enough Orienteering Problem (CEOP) for planning missions with multi-rotor aerial vehicles considering their maximal velocity and acceleration limits. The addressed problem stands to select the most rewarding target locations and sequence to visit them in the given limited travel budget. The reward is collected within a non-zero range from a particular target location that allows saving the travel cost, and thus collect more rewards. Hence, we are searching for the fastest trajectories to collect the most valuable rewards such that the motion constraints are not violated, and the travel budget is satisfied. We leverage on existing trajectory parametrization based on Bézier curves recently deployed in surveillance planning using unsupervised learning, and we propose to employ the learning in a solution of the introduced multi-vehicle CEOP. Feasibility of the proposed approach is supported by empirical evaluation and experimental deployment using multi-rotor vehicles.**

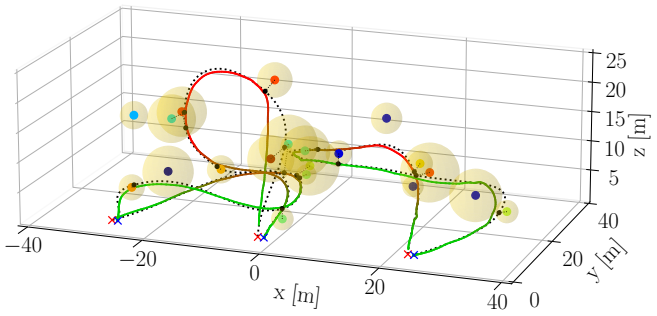


Fig. 1. Example of the found solution to the introduced multi-vehicle Close Enough Orienteering Problem (CEOP) with Bézier curves for three vehicles and its experimental verification. The planned trajectories are dotted, and the real trajectories are the solid curves colored according to the altitude.

I. Introduction

Multi-rotor micro aerial vehicles (MAVs) are increasingly applied in a variety of deployments, but the surveillance and data collection missions, where the vehicles are requested to collect data from the desired locations, can still be considered as a dominant application task for drones [1]. For a (semi)autonomous mission execution, a cost-efficient trajectory to visit the given set of locations can be determined by routing optimization formulated as the Traveling Salesman Problem (TSP) [2]. However, the operational time of MAVs is limited, e.g., by the capacity of their batteries, and thus all the locations may not be visited within such limited budget. Therefore, the mission planning can be formulated as the Orienteering Problem (OP) [3] to determine valuable locations that can be visited within the travel budget T_{max}.

The travel cost can be saved by exploiting remote sensing capabilities with non-zero sensing range or wireless data transfer. Then, the problems become variants of the TSP and OP with Neighborhoods, and for a continuous disk-shaped neighborhood around each target location, the problem is called the Close Enough TSP [4] and Close Enough OP [5], respectively. These problems can be addressed by sampling the neighborhoods to a discrete set of possible waypoints and a solution of the created Generalized TSP, e.g., by [6], [7], [8], and in the case of the OP as the Set OP [9], [10]. However, there is still one more challenge to be addressed when planning for MAVs that is related to requested curvature-constrained trajectories to satisfy their motion constraints.

A possible solution can be based on the Dubins vehicle model [11] for which the Dubins OP has been introduced in [12]. However, Dubins vehicle does not fit the motion constraints of MAVs that are not limited by a minimal turning radius, but mostly by the maximal velocity and acceleration limits. Therefore, we need a more suitable parametrization of the requested multi-goal trajectory that allows determining the velocity profile satisfying the motion limits such that the trajectory realization by the MAVs fits T_{max}. In this paper, we propose to build on our previous work on unsupervised learning algorithm called Growing Self-Organizing Array (GSOA) [13] that has been deployed to the Close Enough TSP with the suitable trajectory parametrization as a sequence of Bézier curves in [5] and extended for multi-vehicle setups in [14]. Due to our relatively extensive previous work, we consider the particular contributions presented in this paper as follows.

- Generalization of the Close Enough OP to motion constraints of the multi-rotor vehicles with trajectory parametrization as a sequence of Bézier curves.
- Novel heuristic solution of the introduced multi-vehicle Close Enough OP with Bézier curves.
- Report on the deployment of the proposed solution in the field experiment with three real MAVs, see Fig. 1.

The paper is organized as follows. The addressed problem is formally introduced in Section II and the proposed solution is presented in Section III. Evaluation results and experimental field deployment are reported in Section IV. Concluding remarks are in Section V.

Authors are with the Czech Technical University, Faculty of Electrical Engineering, Technicka 2, 166 27, Prague, Czech Republic {faiglj|vanapet1|penicrob}@fel.cvut.cz

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 19-20238S. The authors acknowledge the support of the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”.

II. Problem Statement

The Close Enough Orienteering Problem (CEOP) for multi-rotor vehicles stands to determine the most valuable trajectory to collect rewards from a given set of n objects of interest. The objects are assumed to be placed at the locations $O = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n\}$, $\boldsymbol{o}_i \in \mathbb{R}^3$, each with the associated reward $r_i \in \mathbb{R}_0^+$ that can be retrieved by the vehicle if it passes the location $\boldsymbol{o}_i$ within the distance $\delta_i \geq 0$ such that the trajectory satisfies the motion constraints of the vehicle. We search for a subset of k objects $O_k \subseteq O$ with the corresponding waypoints $P_k = \{\boldsymbol{p}_{\sigma_1}, \ldots, \boldsymbol{p}_{\sigma_k}\}$, $\boldsymbol{p}_{\sigma_i} \in \mathbb{R}^3$ such that for each $\boldsymbol{o}_{\sigma_i} \in O_k$ there is a waypoint $\boldsymbol{p}_{\sigma_i}$ within δ_{σ_i} distance from $\boldsymbol{o}_{\sigma_i}$, i.e., $\left\|(\boldsymbol{o}_{\sigma_i}, \boldsymbol{p}_{\sigma_i})\right\| \leq \delta_{\sigma_i}$. Following the notation introduced in [5], the requested trajectory consists of a sequence of Bézier curves $\mathbf{X}_{\sigma_i}$, $\mathcal{X} = (\mathbf{X}_{\sigma_1}, \ldots, \mathbf{X}_{\sigma_{k-1}})$, with the travel time estimation (TTE), denoted $\mathcal{T}(\mathcal{X})$, satisfying the vehicle maximal velocity and acceleration limits, i.e.,

$$\mathcal{T}(\mathcal{X}) = \sum_{i=1}^{k-1} \mathcal{T}(\mathbf{X}_{\sigma_i}) \leq \mathrm{T}_{\max}. \tag{1}$$

The trajectory $\mathcal{X}$ is a smooth connection of $k-1$ Bézier curves connecting the sequence of waypoints P_k such that each $\mathbf{X}_{\sigma_i}$ starts at $\boldsymbol{p}_{\sigma_i}$, terminates at $\boldsymbol{p}_{\sigma_{i+1}}$, and $\mathcal{T}(\mathcal{X})$ satisfies motion constraints of the vehicle.

For a team of m vehicles, we are searching for m individual trajectories $\mathcal{X}^v$ for $v \in \{1, \ldots, m\}$, such that each trajectory $\mathcal{X}^v$ starts at $\boldsymbol{p}_s^v$, terminates at $\boldsymbol{p}_t^v$, and collects rewards of $O^v \subseteq O$; and each reward is collected by one vehicle at maximum, i.e., $O^i \cap O^j = \emptyset$ for any $i \neq j$, $1 \leq i, j \leq m$, which can be defined as Problem 2.1.

Problem 2.1 (Multi-Vehicle CEOP):

$$\begin{aligned} &\text{maximize}_{\mathcal{X}^v \text{ for } v \in \{1,\ldots,m\}} \quad R = \sum_{v=1}^{m} \sum_{i=1}^{k^v} r(\sigma_i^v) \\ &\text{s.t.} \\ &\qquad \mathcal{T}(\mathcal{X}^v) \leq \mathrm{T}_{\max} \end{aligned}, \tag{2}$$

where the budget $\mathrm{T}_{\max}$ represents the maximal flight time.

We further follow the notation of the regular OP, where it is requested the vehicle starts and terminates at the specified locations $\boldsymbol{o}_1$ and $\boldsymbol{o}_n$, respectively, and thus $\sigma_1 = 1$ and $\sigma_k = n$, and $\delta_1 = \delta_n = 0$, and also $r_1 = r_n = 0$. However, we consider individual start and terminal locations for each vehicle v as $\boldsymbol{p}_s^v$ and $\boldsymbol{p}_t^v$, respectively, because of m vehicles.

A detail description of the considered multi-goal trajectory parametrization using Bézier curves is presented in [5], and therefore, only a brief overview of how the TTE is determined is presented here. In particular, we rely on the trajectory following using the Model Predictive Controller (MPC) [15] that allows us to consider a simplified model of the velocity profile with decoupled axes and separated motion generation [16] with an independent limitation of the vertical and horizontal movements. For the velocity profile that satisfies vertical and horizontal limits of the accelerations and velocities, the utilized MPC guarantees following the planned trajectory. Hence, we need to determine the velocity profile, and thus the TTE, for the maximal vertical speed and corresponding maximal magnitude of the acceleration in vertical and horizontal axes that are denoted v_{vert}, a_{vert}, v_{horiz}, and a_{horiz} in the rest of this paper.

Since the vertical and horizontal axes are separated, the altitude differences (i.e., the changes of the curve along the z-axis) are used for computing the vertical velocity profile. In the horizontal axis, the tangent a_{tan} and radial acceleration a_{rad} have to satisfy the horizontal acceleration limit a_{horiz}

$$a_{tan}^2 + a_{rad}^2 \leq a_{horiz}^2. \tag{3}$$

The radial acceleration is defined by the horizontal curvature κ_{horiz} as $a_{rad} = v^2 \kappa_{horiz}$. The horizontal curvature also defines the maximal horizontal velocity along the path

$$v_{pos} = \min\left(v_{horiz}, \sqrt{a_{horiz}/\kappa_{horiz}}\right) \tag{4}$$

and the maximal tangent acceleration

$$a_{tan}^2 = a_{horiz}^2 - v_{pos}^4 \kappa_{horiz}^2. \tag{5}$$

See [5] for further details and procedure of the TTE determination for a sequence of Bézier curves.

III. Proposed Solution of the Multi-Vehicle CEOP with Bézier Curves

The introduced multi-vehicle CEOP with Bézier curves is addressed by the Growing Self-Organizing Array (GSOA) [13], which is a heuristic procedure based on unsupervised learning that combines a solution of the sequencing part with continuous optimization to determine the waypoints. The GSOA is a growing array of M nodes $\mathcal{N} = \{\nu_1, \ldots, \nu_M\}$ that encodes a sequence of visits to the selected target locations. The learning is an iterative procedure in which the array adapts to the target locations and also grows for new target locations added to the solution. The deployment of the GSOA [13] to a single vehicle solution of the CEOP with Bézier curves is a modification of the solution of the CEOP for Dubins vehicle [17], where instead of Dubins maneuvers, Bézier curves are utilized similarly as in [5] using the velocity profile and TTE.

In the herein addressed multi-vehicle CEOP, an individual array of nodes $\mathcal{N}^v$ is created for each vehicle $1 \leq v \leq m$. The arrays (vehicles) then compete to collect rewards from the objects $\boldsymbol{o} \in O$. Thus, all the arrays are conditionally adapted to the particular $\boldsymbol{o}$ (if so) and the array adaptation with the highest increase of the total sum of the rewards (2) and shortest prolongation of $\mathcal{T}(\mathcal{X}^v)$ is kept and the local changes to all other arrays are reverted to the state before the examination of $\boldsymbol{o}$. The adaptation of $\mathcal{N}^v$ to the particular $\boldsymbol{o}$ is a determination of the new node ν^* as the closest point $\boldsymbol{p}_o$ of the trajectory represented by $\mathcal{N}^v$ to the location $\boldsymbol{o}$ and movement of ν^* and its neighbors towards $\boldsymbol{o}$.

A trajectory of each vehicle is encoded by the array $\mathcal{N}^v$ where each node $\nu \in \mathcal{N}^v$ is associated to (i) a position in the input space denoted $\boldsymbol{\nu} \in \mathbb{R}^3$; (ii) the particular object $\nu.\boldsymbol{o} \in O$ to be visited; and (iii) the waypoint $\nu.\boldsymbol{p} \in \mathbb{R}^3$ inside the δ-neighborhood of $\boldsymbol{o}$, i.e., $\|(\nu.\boldsymbol{o}, \nu.\boldsymbol{p})\| \leq \delta(\nu.\boldsymbol{o})$,

which allows collecting the reward of the object within δ-radius from $\boldsymbol{o}$. Besides, the trajectory is a sequence of Bézier curves, and thus two consecutive nodes ν_i and ν_{i+1} of $\mathcal{N}^v$ define each particular Bézier curve where the waypoints $\nu_i.\boldsymbol{p}$ and $\nu_{i+1}.\boldsymbol{p}$ are the curve endpoints. Hence, two additional control points defining the departure and terminal tangents are associated to ν_i and ν_{i+1} that are used for determining the TTE similarly to the solution of the TSP with Bézier curves presented in [5]. On the other hand, in the solution of the OP, each trajectory has to satisfy $\mathrm{T_{max}}$, and therefore, the adaptation of $\mathcal{N}^v$ to $\boldsymbol{o}$ is conditioned to increase the sum of the collected rewards R while satisfying $\mathrm{T_{max}}$.

Even though a local optimization of Bézier curves is performed every time a new node is added to $\mathcal{N}^v$, it might not be possible to collect the reward from $\boldsymbol{o}$ unless nodes associated to less rewarding objects or nodes with relatively far waypoints are removed from $\mathcal{N}^v$, see [17] for further details. Hence, such nodes are conditionally removed from the array to shorten the trajectory in the benefit of collecting the reward of additional $\boldsymbol{o}$. However, the new node ν^* conditionally added to $\mathcal{N}^v$ is preserved only if the trajectory satisfies $\mathrm{T_{max}}$ and the solution is improved by collecting more rewards; otherwise all local changes to $\mathcal{N}^v$ are discarded.

The unsupervised learning is performed in c_{max} epochs or until all rewards are collected. During each epoch, all objects O are examined in a random order, and thus the learning procedure can be considered as stochastic search. The arrays are initialized by the requested start and terminal locations and for each of m vehicles, the array of the v-th vehicle is set to $\mathcal{N}^v = \{\nu_1^v, \nu_{end}^v\}$ with $\boldsymbol{\nu}_1^v = \boldsymbol{p}_s^v$ and $\boldsymbol{\nu}_{end}^v = \boldsymbol{p}_t^v$. Because the start and terminal locations are prescribed, the nodes ν_1^v and ν_{end}^v are never removed or changed, and their waypoints are identical to the node locations $\boldsymbol{\nu}_1^v$ and $\boldsymbol{\nu}_{end}^v$.

For a single examination of $\boldsymbol{o} \in O$, all the arrays are tried to adapt to $\boldsymbol{o}$ but only the array with the highest reward improvement is selected, and the shortest prolongation is preferred in a case of the same reward; the other arrays are reverted to the state before examination of $\boldsymbol{o}$. After examination of all O (after one learning epoch), all nodes from the previous epoch are removed from the arrays by the $\mathrm{regenerate}()$ procedure, because only the newly added nodes with their associated waypoints represent the requested trajectories. Besides, the Local Iterative Optimization (LIO) [18] is utilized to shorten the Bézier curves for the all arrays. Then, trajectories represented by the arrays are extracted, and since the learning is stochastic search, the best solution found so far is maintained.

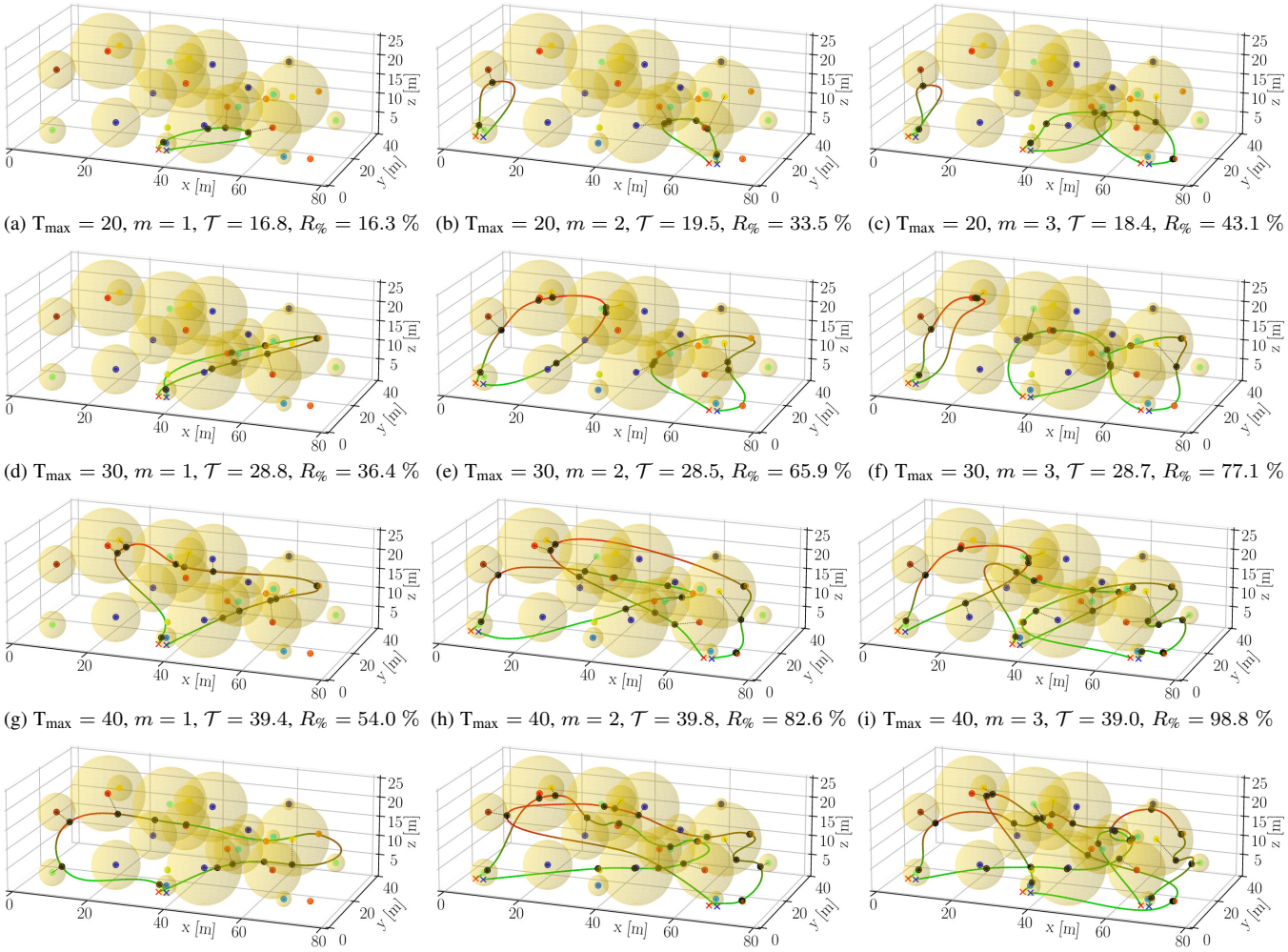
The array $\mathcal{N}^v$ grows by adding new node ν^* that is determined as the closest point $\boldsymbol{p}_o$ of the current trajectory represented by $\mathcal{N}^v$ to the particular location $\boldsymbol{o}$ of the examined object. The trajectory is a sequence of Bézier curves connecting the waypoints associated to the nodes; hence, $\boldsymbol{p}_o$ is on the Bézier curve defined by two consecutive nodes ν_i and ν_{i+1}; and therefore, new node ν^* is added to $\mathcal{N}^v$ between the nodes ν_i and ν_{i+1} with the location set to $\boldsymbol{\nu}^* = \boldsymbol{p}_o$. Moreover, the point $\boldsymbol{p}_o$ is utilized to determine the waypoint $\nu^*.\boldsymbol{p}$ to collect the reward as the farthest point from $\boldsymbol{o}$ on the segment $(\boldsymbol{p}, \boldsymbol{o})$ that is inside the δ-neighborhood of $\boldsymbol{o}$. The new node ν^* is then adapted towards $\nu^*.\boldsymbol{p}$ together with its neighboring nodes $\nu \in \mathcal{N}^v$ that are in the distance (in the number of nodes) $d \leq 0.2M$ from ν^*, where M is the current number of nodes in $\mathcal{N}^v$. The adaptation of the node ν is an update of the node location $\boldsymbol{\nu} \leftarrow \boldsymbol{\nu} + \mu f(\sigma, d)(\nu^*.\boldsymbol{p} - \boldsymbol{\nu})$ using the neighboring function $f(\sigma, d) = \exp(-d/\sigma^2)$, where μ is the learning rate and σ is the learning gain. Note, the learning gain is initially set to a sufficiently high value that is decreased after each learning epoch using the gain decreasing rate denoted α [13]. The learning procedure is summarized in Algorithm 1.

Algorithm 1: Multi-vehicle CEOP with Bézier curves

Input: $O = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n\}$; start and terminal locations $(\boldsymbol{p}_s^1, \boldsymbol{p}_t^1), \ldots, (\boldsymbol{p}_s^m, \boldsymbol{p}_t^m)$; travel budget $\mathrm{T_{max}}$
Output: $\mathcal{X}^1, \ldots, \mathcal{X}^m$ – trajectories visiting subset of O

$\sigma \leftarrow 12.41n + 0.6$, $\alpha = 0.1$, $\mu \leftarrow 0.5$; $c \leftarrow 1$;
for $v \in \{1, \ldots, m\}$ initialize $\mathcal{N}^v \leftarrow (\boldsymbol{p}_s^v, \boldsymbol{p}_t^v)$;
$S_{best} \leftarrow \{\mathcal{X}^1 \ldots, \mathcal{X}^m\}$ for $\mathcal{X}^v \leftarrow \mathrm{Extract}(\mathcal{N}^v)$
while $c \leq c_{max}$ and $\mathrm{Rewards}(S_{best}) < \sum_{i=1}^{n} r_i$ **do**
  foreach $v \in \{1, \ldots, m\}$ **do** $R^v \leftarrow 0$ **end**
  foreach object $\boldsymbol{o}$ in a random permutation of O **do**
    foreach $v \in \{1, \ldots, m\}$ with $(R^v, \mathcal{N}^v)$ **do**
      $(R^v_{prev}, \mathcal{N}^v_{prev}) \leftarrow (R^v, \mathcal{N}^v)$ // Save array
      $(\boldsymbol{p}, \nu_i, \nu_{i+1}) \leftarrow \mathrm{closestsPoint}(\mathcal{N}^v, \boldsymbol{o}, \delta(\boldsymbol{o}))$
      $R^v, \mathcal{N}^v \leftarrow \mathrm{insert\&adapt}(\mathcal{N}^v, \nu_i, \nu_{i+1}, \boldsymbol{p})$
    $v_b \leftarrow \arg\max_{1 \leq v \leq m \text{ and } \mathcal{X}^v \leq \mathrm{T_{max}}} \left(R^v - R^v_{prev}\right)$
    foreach vehicle $v \in \{1, \ldots, m\} \setminus \{v_b\}$ **do**
      $(R^v, \mathcal{N}^v) \leftarrow (R^v_{prev}, \mathcal{N}^v_{prev})$ // Revert
  ▷ *Update arrays and the best solution found so far*
  $\mathcal{N}^v, \ldots, \mathcal{N}^m \leftarrow \mathrm{regenerate}(\mathcal{N}^1, \ldots, \mathcal{N}^m)$
  $c \leftarrow c + 1$ and $\sigma \leftarrow (1 - \alpha)\sigma$ // Decrease gain
  if $\mathrm{Rewards}(\mathcal{N}^1, \ldots, \mathcal{N}^m) \geq \mathrm{Rewards}(S_{best})$ **then**
    $S_{best} \leftarrow \{\mathcal{X}^1 \ldots, \mathcal{X}^m\}$ for $\mathcal{X}^v \leftarrow \mathrm{Extract}(\mathcal{N}^v)$
return S_{best}

A solution of the CEOP is available after each learning epoch, but few epochs are needed to improve the solution. In particular, $c_{max} = 20$ is used for the results reported in this paper. Regarding the structure of the learning procedure depicted in Algorithm 1, its computational complexity depends on the number of nodes that depends on the number of objects n and vehicles m. However, the nodes are distributed in particular arrays, and therefore, the total number nodes is always below $2n + 2m$ because nodes from the previous epoch are removed in the $\mathrm{regenerate}()$ procedure. Hence, the number of nodes is increased by the number of vehicles only in the additional $2m$ nodes corresponding to the prescribed start and terminal locations of the vehicles. Thus, the computational complexity of the adaptation to n objects can be bounded by $O(n(n+m))$, which also holds for the whole single learning epoch because the velocity profile is determined in $O(n)$ [5] and LIO can also be bounded by $O(n)$ for some small fixed number of iterations [18].



(a) $\mathrm{T_{max}} = 20$, $m = 1$, $\mathcal{T} = 16.8$, $R_{\%} = 16.3$ % (b) $\mathrm{T_{max}} = 20$, $m = 2$, $\mathcal{T} = 19.5$, $R_{\%} = 33.5$ % (c) $\mathrm{T_{max}} = 20$, $m = 3$, $\mathcal{T} = 18.4$, $R_{\%} = 43.1$ %

(d) $\mathrm{T_{max}} = 30$, $m = 1$, $\mathcal{T} = 28.8$, $R_{\%} = 36.4$ % (e) $\mathrm{T_{max}} = 30$, $m = 2$, $\mathcal{T} = 28.5$, $R_{\%} = 65.9$ % (f) $\mathrm{T_{max}} = 30$, $m = 3$, $\mathcal{T} = 28.7$, $R_{\%} = 77.1$ %

(g) $\mathrm{T_{max}} = 40$, $m = 1$, $\mathcal{T} = 39.4$, $R_{\%} = 54.0$ % (h) $\mathrm{T_{max}} = 40$, $m = 2$, $\mathcal{T} = 39.8$, $R_{\%} = 82.6$ % (i) $\mathrm{T_{max}} = 40$, $m = 3$, $\mathcal{T} = 39.0$, $R_{\%} = 98.8$ %

(j) $\mathrm{T_{max}} = 50$, $m = 1$, $\mathcal{T} = 49.9$, $R_{\%} = 64.5$ % (k) $\mathrm{T_{max}} = 50$, $m = 2$, $\mathcal{T} = 49.6$, $R_{\%} = 95.0$ % (l) $\mathrm{T_{max}} = 50$, $m = 3$, $\mathcal{T} = 47.0$, $R_{\%} = 100.0$ %

Fig. 2. Selected best found solutions for the particular $\mathrm{T_{max}}$ and number of vehicles m, where $\mathcal{T} = \max_{1 \leq v \leq m} \mathcal{T}(\mathcal{X}^v)$ and $R_{\%}$ is the sum of the collected rewards in the percentage points of the maximal reward $R_{max} = 250.58$.

IV. Results

Due to lack of any other solver capable of solving the CEOP with Bézier curves, the proposed solver has been empirically evaluated in 3D instances of the CEOP with 22 target locations with up to six additional locations (5 m above the ground) where the MAVs start and terminate their reward collecting tours[1]. The found solutions have also been deployed in a real experiment to verify the feasibility of the planned trajectories, and the results are reported in Section IV-A. The particular considered velocity and acceleration limits are $v_{vert} = 1\,\mathrm{ms}^{-1}$, $a_{vert} = 1\,\mathrm{ms}^{-2}$, $v_{horiz} = 5\,\mathrm{ms}^{-1}$, and $a_{horiz} = 2\,\mathrm{ms}^{-2}$.

For the empirical evaluation, we consider $\mathrm{T_{max}} \in \{20, 30, 40, 50\}$ and $m \in \{1, 2, 3\}$ which gives 12 instances with $n = 22$ target locations that are at different altitudes. Each location has an individual neighborhood radius δ_i and reward value r_i randomly drawn from the interval $\delta_i \in [0, 10]$ and $r_i \in [0, 20]$, respectively. The total maximal sum of the rewards in the scenario is 250.58. The proposed solver has been implemented in C++, compiled by the Clang version 6.0.1 with the -O3 and -march=native flags, and run on a single core of the Intel i7-6700K CPU running at 4 GHz. The computational requirements depend on $\mathrm{T_{max}}$ because small values allow only a short array with relatively fast evaluation.

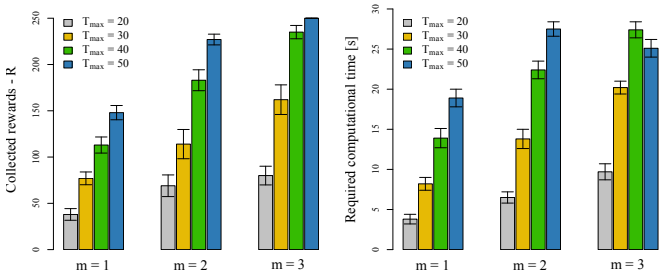
TABLE I

Sum of the collected rewards for the found solutions

$\mathrm{T_{max}}$	Best found R			Average found R_{avg}		
	$m = 1$	$m = 2$	$m = 3$	$m = 1$	$m = 2$	$m = 3$
20	40.9	84.1	107.9	38.3	68.8	79.7
30	91.2	165.0	193.1	76.7	114.1	161.6
40	135.3	207.0	247.7	113.2	183.1	234.8
50	161.7	238.2	250.6	147.7	227.5	250.5

Each instance has been solved 20 times for each $\mathrm{T_{max}}$ and m, and the sum of the collected rewards R for the best-found solutions together with the average sum of the collected rewards R_{avg} are reported in Table I and Fig. 3. The best found solutions are visualized in Fig. 2. The

[1]Problem instances of the addressed Close Enough OP and found solutions are available at `https://purl.org/comrob/sw`.

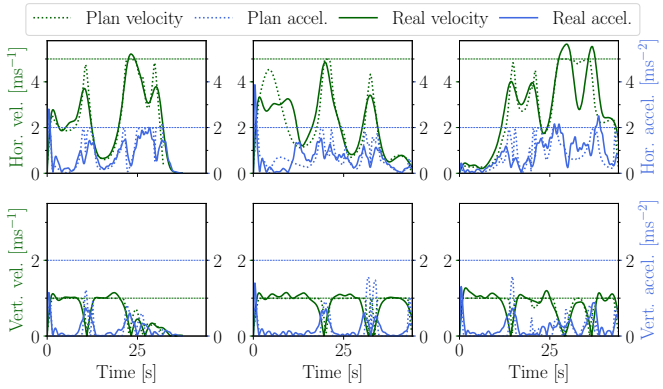


(a) Collected rewards (b) Computational time

Fig. 3. Influence of the increased T_{max} and number of vehicles m to the sum of the collected rewards and required computational time for the same scenario with $n = 22$ and variable sensing range per each target location.

results indicate that increasing T_{max} increases the sum of the collected rewards R similarly as adding more vehicles. The real computational requirements increase with T_{max}, but also with m. It is mainly because a longer T_{max} allows visiting more locations, and thus more nodes are in the arrays and the trajectories consist of more Bézier curves for which the used local optimization is more demanding. More vehicles have a similar effect as more trajectory segments are evaluated. A noticeable decrease of the computational requirements can be seen for $T_{max} = 50$ and $m = 3$ in which the total travel budget is sufficiently high to collect all the rewards.

In Fig. 2, it can be seen that for small travel budgets, trajectories at the altitude similar to start and terminal locations are preferred, which is indicated by the green parts of the trajectories. For longer T_{max}, there is typically one peek to reach locations at a higher altitude that is visualized by the red color.



(a) $\mathcal{T}(\mathcal{X}^1) = 32.2$ s (b) $\mathcal{T}(\mathcal{X}^2) = 43.2$ s (c) $\mathcal{T}(\mathcal{X}^3) = 44.0$ s

Fig. 4. Planned and real velocity and acceleration profiles for three vehicles (each in one column) divided to their horizontal (top) and vertical (bottom) components for the experiment with $T_{max} = 45$ s and non-mutually crossing trajectories depicted in Fig. 1.

The optimal solution of the addressed problem is not available, and therefore, the solution quality can be studied by the saturation of the velocity and acceleration limits for which we can assume that a high-quality solution should quickly reach and keep the maximal velocity. The planned and real velocity and acceleration profiles are shown in Fig. 4 that have been recorded from the experimental deployment.

It is worth noting that the start and terminal locations of the vehicles are selected to support a division of the workspace to particular vehicles, but they are also pragmatically motivated to have sufficient space around the vehicles for the autonomous take-off and landing. Besides, the locations also support mutual collision-free trajectories which are not explicitly guaranteed by the proposed planning approach. The influence of the local mutual collision avoidance in the employed MPC [19] is reported in the following section.

A. Real Experimental Deployment

The practical experimental verification of the proposed solution to the introduced variant of the multi-vehicle CEOP with Bézier curves has been performed with three identical MAVs, see Fig. 5. The most important part of the field deployment is the MPC-based trajectory following controller [15] which is capable to precisely navigate the vehicle along the planned trajectory (that satisfies the vehicle maximal velocity and acceleration limits) using the RTK GPS. Besides, local collision avoidance is implemented in the MPC to steer the vehicle to a collision-free trajectory in a case of nearby vehicles [19]. Therefore, we consider two solutions of the same problem instance: one without mutually crossing paths and one with crossing paths, and thus with a possible collision.

Fig. 5. A snapshot from the real experimental verification of the proposed solution to the multi-vehicle CEOP with Bézier curves.

For the mutually non-crossing trajectories, only negligible differences between the planned and real executed trajectories can be seen in Fig. 1. However, for a solution with the mutually crossing trajectories, the first and third vehicles have to significantly adjust their trajectories while the second vehicle follows almost completely the planned trajectory, see Fig. 6. A combination of the collision avoidance with the optimal MPC-based trajectory following controller causes that some of the planned waypoints are missed and not all planned rewards are collected. The issue can be eventually addressed by hovering at a location instead of active collision avoidance, but it will not save the energy which can be very similar for hovering as for motion, and thus the sum of the collected rewards would be lower anyway. Hence, we need guaranteed mutually non-crossing trajectories as a solution to the multi-vehicle CEOP. Alternatively, a new instance of the problem reflecting the current situation can be created, and the mission plan for each particular vehicle can be updated, for which we need a new solution quickly. Both of these ideas are considered for our future work.

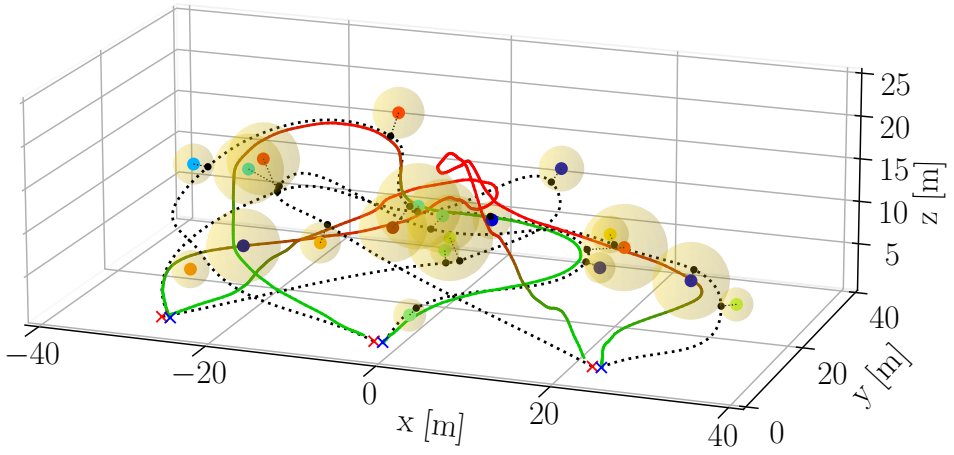


Fig. 6. Example of the real execution of the mutually crossing trajectories for $m = 3$ vehicles. The used MPC-based trajectory following controller avoids possible collisions; however, it is therefore not possible to follow the whole planned trajectory (dotted), and thus vehicles do not visit all the planned locations. The not visited locations are marked by small black disks.

V. Conclusion

We introduce a novel variant of the multi-vehicle curvature-constrained multi-goal trajectory planning that is denoted multi-vehicle Close Enough Orienteering Problem (CEOP) with Bézier curves. The problem is addressed by a novel heuristic approach that is based on the unsupervised learning technique called the GSOA. The proposed solution leverages on our previous work on the developed GSOA for routing problems and also on a solution of the problems with curvature-constrained vehicles, i.e., the CEOP with Dubins vehicle and CETSP with Bézier curves. Therefore, the main contributions of the paper are in reporting on the empirical evaluation and experimental verification of the proposed approach in several scenarios with up to three vehicles. The particular problem instances with the found solutions are made available to support further research on the problem.

Our particular future research directions are motivated by the raised issue of the local collision avoidance which may reduce the sum of the collected rewards as it has been observed in the reported experimental deployment. Therefore, we plan to extend the proposed solution to guarantee mutually non-crossing trajectories explicitly, but we also plan to consider a more general problem formulation to determine solutions that would be robust to local disturbances and which can provide initial solution for the fast plan update because of influences that are not known before the planning but only during the real mission execution.

References

[1] M. Dunbabin and L. Marques, "Robots for Environmental Monitoring: Significant Advancements and Applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, Mar. 2012.

[2] D. G. Macharet and M. F. M. Campos, "A survey on routing problems and robotic systems," *Robotica*, p. 1–23, 2018.

[3] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering Problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, Dec. 2016.

[4] D. J. Gulczynski, J. W. Heath, and C. C. Price, *The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics*. Boston, MA: Springer US, 2006, pp. 271–283.

[5] J. Faigl and P. Váňa, "Surveillance planning with Bézier curves," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 750–757, 2018.

[6] G. Laporte, A. Asef-Vaziri, and C. Sriskandarajah, "Some applications of the generalized travelling salesman problem," *The Journal of the Operational Research Society*, vol. 47, no. 12, pp. 1461–1467, 1996.

[7] K. Helsgaun, "Solving the equality generalized traveling salesman problem using the lin–kernighan–helsgaun algorithm," *Mathematical Programming Computation*, vol. 7, no. 3, pp. 269–287, Sep 2015.

[8] S. L. Smith and F. Imeson, "GLNS: An effective large neighborhood search heuristic for the generalized traveling salesman problem," *Computers & Operations Research*, vol. 87, no. C, pp. 1–19, Nov. 2017.

[9] C. Archetti, F. Carrabs, and R. Cerulli, "The set orienteering problem," *European Journal of Operational Research*, vol. 267, no. 1, pp. 264–272, 2018.

[10] R. Penička, J. Faigl, and M. Saska, "Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants," *European Journal of Operational Research*, 2019.

[11] 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*, pp. 497–516, 1957.

[12] R. Pěnička, J. Faigl, P. Váňa, and M. Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210–1217, April 2017.

[13] J. Faigl, "GSOA: growing self-organizing array - unsupervised learning for the close-enough traveling salesman problem and other routing problems," *Neurocomputing*, vol. 312, pp. 120–134, 2018.

[14] J. Faigl, P. Váňa, R. Pěnička, and M. Saska, "Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles," *Journal of Field Robotics*, vol. 36, no. 1, pp. 270–301, 2019.

[15] T. Báča, D. Heřt, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6753–6760.

[16] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, Dec 2015.

[17] J. Faigl and R. Pěnička, "On close enough orienteering problem with dubins vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5646–5652.

[18] P. Váňa and J. Faigl, "On the dubins traveling salesman problem with neighborhoods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4029–4034.

[19] V. Spurný, T. Báča, M. Saska, R. Pěnička, T. Krajník, J. Thomas, D. Thakur, G. Loianno, and V. Kumar, "Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles," *Journal of Field Robotics*, vol. 36, no. 1, pp. 125–148, 2019.