Noname manuscript No.
(will be inserted by the editor)

Data Collection Planning with Non-zero Sensing Distance for a Budget and Curvature Constrained Unmanned Aerial Vehicle

Robert Pěnička · Jan Faigl · Martin Saska · Petr Váňa

Received: date / Accepted: date

Abstract Data collection missions are one of the many effective use cases of Unmanned Aerial Vehicles (UAVs), where the UAV is required to visit a predefined set of target locations to retrieve data. However, the flight time of a real UAV is time constrained, and therefore only a limited number of target locations can typically be visited within the mission. In this paper, we address the data collection planning problem called the Dubins Orienteering Problem with Neighborhoods (DOPN), which sets out to determine the sequence of visits to the most rewarding subset of target locations, each with an associated reward, within a given travel budget. The objective of the DOPN is thus to maximize the sum of the rewards collected from the visited target locations using a budget constrained path between predefined starting and ending locations. The variant of the Orienteering Problem (OP) addressed here uses curvature-constrained Dubins vehicle model for planning the data collection missions for UAV. Moreover, in the DOPN, it is also assumed that the data, and thus the reward, may be collected from a close neighborhood sensing distance around the target locations, e.g., taking a snapshot by an onboard camera with a wide field of view, or using a sensor with a long range. We propose a novel approach based on the Variable Neighborhood Search (VNS) metaheuristic for the DOPN, in which combinatorial optimization of the sequence for visiting the target locations is simultaneously addressed with continuous optimization for finding Dubins vehicle waypoints inside the neighborhoods of the visited targets. The proposed VNS-based DOPN algorithm is evaluated in numerous benchmark instances, and the results show that it significantly outperforms the existing methods in both solution quality and computational time. The practical deployability of the proposed approach is experimentally verified in a data collection scenario with a real hexarotor UAV.

Keywords Unmanned Aerial Vehicles · Non-holonomic Motion Planning · Data Collection Planning · Orienteering Problem

This work was supported by the Czech Science Foundation under research project No. 16-24206S, No. 17-16900Y, and No. 19-20238S. The authors acknowledge the support of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”. Access to computing at National Grid Infrastructure MetaCentrum, provided under the CESNET LM2015042 programme, is greatly appreciated. The support of CTU in Prague grant No. SGS17/187/13 is also gratefully acknowledged.

Robert Pěnička · Jan Faigl · Martin Saska · Petr Váňa
Czech Technical University, Faculty of Electrical Engineering, Technicka 2, 166 27, Prague, Czechia
E-mail: {penicrob | faiglj | saskam1 | vanapet1}@fel.cvut.cz

1 INTRODUCTION

Unmanned Aerial Vehicles (UAV) are effective systems for long-range data collection (Ergezer and Leblebicioğlu 2014) or for information gathering scenarios (Nguyen et al. 2016), where a UAV has to gather data from specified locations in the environment. Such a scenario consists of a UAV equipped with an onboard sensor that is required to reach particular target locations and measure or collect the desired data. For example, in a Wireless Sensor Network (WSN), the sensors are placed in the environment, and the UAV can be used for retrieving the measured data from the sensor units by wireless communication with a limited range (Jawhar et al. 2014; Wang et al. 2015). Hence, the objective of data collection planning can be to minimize the required time to retrieve the requested data (i.e., to minimize the length of the data collection path) or/and to maximize the information collected by a single path.

Data collection planning can be formulated as a variant of the Traveling Salesman Problem (TSP) (Oberlin et al. 2010), where a path visiting all the given locations with minimal length is to be found. However, the required visits to all locations may not be possible with the budget limitation of a real vehicle (limited flight time).

Nowadays, the typical flight time of a small UAV is limited to tens of minutes, and the time is further decreased if the UAV is equipped with an additional payload, e.g., onboard sensors. Therefore, the Orienteering Problem (OP) (Tsiligirides 1984) formulation seems to be more suitable for data collection planning with a limited travel budget. Rather than minimizing the path length as in the TSP, the OP set out to find a path maximizing the sum of the rewards collected from a selected subset of target locations that can be reached using the given travel budget.

In this work, we consider that the data collecting vehicle with a budget constraint has to follow a curvature-constrained path, and thus we model the UAV as Dubins vehicle (Dubins 1957). Dubins vehicle can be used for modeling car-like robots (Tokekar et al. 2014), fixed-wing aerial vehicles (Lugo-Cárdenas et al. 2014) or Vertical Take-Off and Landing (VTOL) multirotor UAVs traversing the planned path at a constant speed (Pěnička et al. 2017a).

For Dubins vehicle, the TSP becomes the Dubins Traveling Salesman Problem (DTSP) (Savla et al. 2005), where it is required to find not only the optimal sequence for the visits to all target locations, but also optimal heading angles of the vehicle at the locations, as they greatly influence the final path length. Since each heading angle can be arbitrarily selected from 0 to 2π, the problem becomes computationally demanding due to the required non-linear continuous optimization of the additional dimension of the heading angles.

For a limited travel budget and Dubins vehicle, the OP becomes the Dubins Orienteering Problem (DOP), which was introduced and solved by a Variable Neighborhood Search (VNS) based approach in (Pěnička et al. 2017a). In the DOP, it is required to search over all possible heading angles at the target locations to find the most rewarding curvature-constrained path within the limited budget. Note that both the OP and the DOP are NP-hard similarly to the TSP and the DTSP (Le Ny et al. 2007).

In data collection planning, the solution quality, i.e., the path length in the (D)TSP or the sum of the rewards collected in the (D)OP, can be increased by introducing a non-zero sensing distance in which the

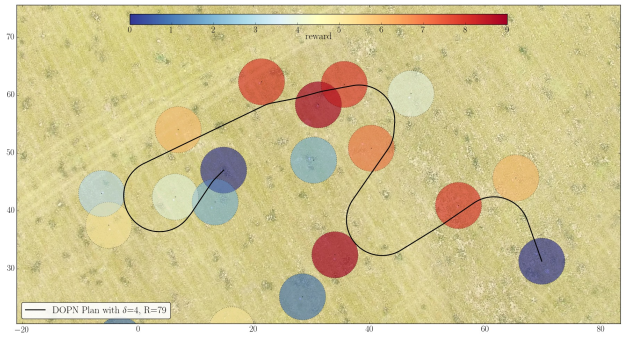


Fig. 1: A snapshot of the workspace for experimental verification of the proposed Dubins Orienteering Problem with Neighborhoods taken by a UAV flying 100 m above the ground. The solution of the DOPN used in the real experiment with a hexarotor UAV is calculated using the proposed Variable Neighborhood Search method with target neighborhood radius $\delta = 4\,\mathrm{m}$ and budget constraint $T_{max} = 150\,\mathrm{m}$.

data can be collected from the particular target locations. An extension of the DTSP for the non-zero sensing distance is called the Dubins Traveling Salesman Problem with Neighborhoods (DTSPN) (Obermeyer 2009; Isaacs et al. 2011; Váňa and Faigl 2015). In this paper, we consider a similar extension of the DOP to the Dubins Orienteering Problem with Neighborhoods (DOPN), initially introduced in (Pěnička et al. 2017b). Although exploiting the neighborhood in most cases increases the quality of the solutions (regarding the collected rewards), solving the DOPN is more challenging due to the additional determination of the most suitable waypoint locations to retrieve the rewards within the neighborhood of target locations. The DOPN thus includes both a combinatorial part and a continuous optimization part. Determining the subset of target locations and determining the sequence for visiting them are the combinatorial parts of the DOPN. The continuous optimization part involves determining the waypoint locations within the neighborhood of the target locations and the determining the waypoint heading angles of Dubins vehicle at the selected waypoint locations. An illustration of the DOPN solution from the experimental verification of the proposed method with a hexarotor UAV is shown in Fig. 1.

The novel method for the DOPN is based on the Variable Neighborhood Search (VNS) metaheuristic (Mladenović and Hansen 1997). It consists of both combinatorial and continuous optimization operators to solve the DOPN. Initially, low-density equidistant sampling of both the waypoint heading angles and the waypoint locations within the neighborhoods is considered, in order to create waypoint graph for the combinatorial optimization to maximize the sum of the collected rewards. The particular waypoint samples of the solutions for a given sequence of visited target locations are selected such that the path length is minimized. An initial greedy solution is then found by adding target locations that maximize the reward per tour prolongation, while the maximally allowed budget is still fulfilled. The VNS method afterward uses a set of neighborhood operators to randomly change and locally improve the best found solution. The proposed VNS consists of the combinatorial optimization operators extended from the VNS-based solution to the original OP in (Sevkli and Sevilgen 2006) and also utilized for solving the DOP in (Pěnička et al. 2017a). However, the herein VNS-based DOPN solver contains novel continuous optimization operators to minimize the path length over the selected sequence of target locations by optimizing both the heading angles and the waypoint locations within the neighborhoods of the target locations. The proposed operators shorten the solution found on the low-density sampled waypoint graph, and update the locally optimized values to the graph for combinatorial optimization. The path length is optimized to allow addition of previously unvisited target locations while satisfying the budget constraint.

A preliminary version of this work appears in (Pěnička et al. 2017b), where the DOPN is addressed by a purely sampling-based approach. This paper is considered to make the following contributions. The method introduced here significantly improves the solution quality and decreases the overall required computational time, which allows onboard online planning and anytime behavior. The proposed method combines combinatorial optimization and continuous optimization in a single VNS-based framework, which outperforms the previous purely combinatorial sampling-based solution (Pěnička et al. 2017b) and also the competitive Self-Organizing Map (SOM) based solution (Faigl and Pěnička 2017). The initial low-density waypoint sampling allows us to obtain high quality initial solutions ($\approx 90\%$ of the best-known rewards) within a few seconds, and due to the continuous optimization of the waypoints, the solution quality is improved above the so far best-known solutions created by dense waypoint sampling with required initialization in tens of minutes. The performance and quality improvements are mainly caused by the proposed tight coupling between combinatorial optimizations and continuous optimization in a single algorithm, which is also considered as one of the main contributions of our work. Furthermore, the designed VNS-based algorithm minimizes the path length in addition to the main OP objective of maximizing the sum of the collected rewards, which can be useful when all target locations can be feasibly collected within the defined budget. Last but not least, the experimental verification in the data collection scenario demonstrates the practical usefulness of the addressed problem and the proposed method.

The remainder of this paper is organized as follows. An overview of related work is presented in the next section. A formal definition of the DOPN is introduced in Section 3, and the novel VNS-based approach is proposed in Section 4. Section 5 shows the computational results and the experimental verification in a real data collection scenario. The conclusion and future work are outlined in Section 6.

2 Related Work

The Dubins Orienteering Problem with Neighborhoods belongs to a wider class of orienteering problems (Gunawan et al. 2016), where the objective is to find a limited length path between a starting location and an ending location which maximizes the sum of the rewards collected from a subset of the specified target locations. Therefore, this section presents an overview of existing approaches for the Orienteering Problem and relevant variants for UAVs. The DOPN is also related to the Traveling Salesman Problem (TSP) and its variants involving Dubins vehicle and neighborhoods; therefore a brief overview of relevant solutions of the TSP is provided in this section.

The Euclidean version of the OP, further denoted as the EOP, was introduced by Tsiligirides (Tsiligirides 1984) in 1984, together with the deterministic D-algorithm and stochastic S-algorithm approaches for the EOP. The S-algorithm is based on the Monte-Carlo method, which creates multiple feasible paths and selects the best solution according to the reward. The D-algorithm is based on the method for the vehicle routing problem (Wren and Holliday 1972). Furthermore, Tsiligirides created three OP benchmark instances (Vansteenwegen 2018), further denoted as Set 1, Set 2 and Set 3, with up to 33 target locations.

Since the first deterministic and stochastic algorithms for the OP, a large number of solutions for the EOP and other variants of the OP have been proposed (Vansteenwegen et al. 2011; Gunawan et al. 2016) with results that outperform the first solutions. The OP can be solved optimally using the Branch and Bound algorithm (Ramesh et al. 1992) or by the Branch and Cut (Fischetti et al. 1998) algorithm; however, the optimal solution of the EOP requires significant computational resources, and the solutions are provided in several minutes or hours for instances with tens of target locations.

For the Dubins Orienteering Problem or its variant with neighborhoods, additional waypoint sampling is required for each target location. Hence it is optimally solvable only for a given, rather low, sampling density with reasonable computational resources. Therefore, numerous heuristic solutions for the EOP, such as the approaches in (Ramesh and Brown 1991; Chao et al. 1996a; Schilde et al. 2009; Sevkli and Sevilgen 2006), have been proposed, with results that can achieve a solution close to the optimal one within a fraction of the computational time required for the optimal solution. The Fast and Effective heuristic for the EOP by Chao et al. (Chao et al. 1996a) considers only target locations reachable within the prescribed budget (i.e., target locations inside the respective ellipse around the prescribed starting and ending locations). This reduces the number of target locations in solutions with low budgets. The heuristic by Chao et al. uses a set of operators consisting of two-point exchange and one-point movement together with the 2-Opt operation to find high-quality EOP solutions. Furthermore, two symmetrical benchmark sets were created in (Chao et al. 1996a), the diamond shaped Set 64 and the square shaped Set 66 with up to 66 target locations.

The OP has also been proposed for path and data collection planning for UAVs. A variant of the OP, called the Correlated Orienteering Problem (COP) (Yu et al. 2016), introduced for persistent monitoring and data collection tasks with UAVs, proposed a variant of the OP where the rewards of target locations are correlated on the basis of their mutual distances. The COP is motivated by the correlation in sensory measurements of neighboring target locations, and its solution can be found optimally using mixed integer quadratic programming for a small number of target locations. A version of the COP involving Dubins vehicle has been proposed recently by Tsiogkas and Lane (2018).

Thakur et al. (2013) proposed a variant of the Team Orienteering Problem (TOP) (the multi-vehicle variant of the OP proposed by Chao et al. (1996b)) for Dubins vehicle in environments with obstacles. However, the definition of the problem proposed in (Thakur et al. 2013) consists of a given set of waypoints for Dubins vehicle and does not consider an arbitrary heading angle at the target locations or the non-zero sensing distance, as in the DOPN. An optimal multilevel graph search technique is proposed for optimizing the TOP on a given set of Dubins vehicle waypoints for up to 15 target locations. The multi-robot variant of the OP is also proposed in (Jorgensen et al. 2018) for so-called Team Surviving Orienteers (TSO), where the budget is replaced by the constraining probabilities that each robot survives to its destination.

The proposed DOPN method is based on the Variable Neighborhood Search (VNS) (Mladenović and Hansen 1997) metaheuristic by Hansen and Mladenović for combinatorial optimization applicable to numerous problems (Hansen and Mladenović 2001). The VNS employs predefined neighborhood operators used for iterative improvement of the initial solution inside the *shaking* and *local search* procedures. The first VNS-based approach to the EOP (Sevkli and Sevilgen 2006) uses neighborhood structures that motivate the combinatorial optimization part of the proposed solution of the DOPN. The VNS-based method for the EOP randomly changes the current best solution by either path move operator or path exchange operator in the *shaking* procedure to get from the possible local maximum. Then, the method tries to improve the randomly changed path by multiple one point moves or exchanges in the *local search* procedure in order to find a more rewarded path than the incumbent solution.

In our previous work (Pěnička et al. 2017a), the DOP was introduced together with the VNS-based method to solve it. The method uses similar neighborhood structures as the VNS method for the EOP (Sevkli and Sevilgen 2006). However, to tackle the continuous optimization problem of finding a suitable path for curvature-constrained Dubins vehicle, equidistant sampling of the heading angle at the target locations was proposed. The VNS-based method then searches for the most rewarding path, together with the appropriate sequence of sampled heading angles to fit the path length within the budget constraint. The DOPN and its heuristic VNS-based solution was introduced in (Pěnička et al. 2017b) with a straightforward extension of the pure sampling-based approach by additional sampling of visit positions in the circular neighborhood of each target location. In this paper, the solution of the DOPN is further improved by a combination of combinatorial optimization of the DOPN with continuous optimization of the waypoint samples in a single VNS-based algorithm. Furthermore, the deployment of the proposed method is shown in an experimental verification with a hexarotor UAV.

The first approach addressing the generalization of the OP to the Euclidean variant of the Orienteering Problem with Neighborhoods (OPN) was proposed in (Best et al. 2016), and was further improved in (Faigl et al. 2016). The multi-robot variant of the OPN for active perception has been studied in (Best et al. 2018). The approach is based on unsupervised learning of the Self-Organizing Map (SOM) for the Prize-Collecting Traveling Salesman Problem with Neighborhoods (PC-TSPN) (Faigl and Hollinger 2014), i.e., a variant of the TSP that combines maximization of the rewards (prizes) and minimization of the path length. The approach has been further extended to variants with multiple vehicles in the OP (Faigl 2017) and also multi-vehicle PC-TSPN (Faigl and Hollinger 2018). The SOM has also been applied to the DTSP and DTSPN in (Faigl and Váňa 2017). Recently, the SOM-based approach has been adopted for solving the Close Enough Dubins Orienteering Problem (CEDOP) (Faigl and Pěnička 2017), which is the DOPN with name emphasized usage of disk-shaped neighborhoods. The VNS-based solution of the DOPN proposed here significantly outperforms the SOM-based approach for CEDOP, both in the maximal achievable solution quality and also regarding computational time.

The proposed DOPN is also related to existing approaches to the DTSP (Cohen et al. 2017) and the DTSPN (Váňa and Faigl 2015). The most relevant approaches are sampling-based variants of the DTSP, where the heading angles at the target locations are sampled, and the problem is transformed to the Asymmetric TSP (ATSP) (Noon and Bean 1993), which can be solved optimally for the specified sampling. A similar approach can be used for the DTSPN (Obermeyer et al. 2010), where both the heading angles and the positions within the neighborhood are sampled. The problem is then transformed into the Generalized TSP (GTSP) and further to the ATSP, which can be solved, e.g., by the LKH solver (Helsgaun 2000). The solutions of sampling based methods, however, can be further improved by employing the Dubins Touring Problem (DTP) (Faigl et al. 2017), which sets out to find the optimal heading angles of Dubins vehicle for a given sequence of target locations in order to minimize the path length in the DTSP. For the DTSPN, the DTP can be further extended to the Dubins Touring Regions Problem (DTRP), recently addressed as the Generalized Dubins Interval Problem (Váňa and Faigl 2018), where both the heading angles of Dubins vehicle and the visit position inside the neighborhoods of target locations are optimized for a given sequence of target locations. The proposed VNS-based solution of the DOPN uses the adopted version of the Local Iterative Optimization (LIO) procedure (Váňa and Faigl 2015) (originally designed for the DTRP) in continuous optimization VNS operators. It iteratively optimizes individual heading angles and neighborhood positions at each target location to minimize the required path length. The related DTSPN and its DTRP subproblem, however, does not contain subset selection with maximization of the collected rewards, and the budget constraint, as in the DOPN, which is formally introduced in the next section.

3 Problem Statement

In this section, we formally define the DOPN. The problem studied here consists of two main optimization parts. The first part is the combinatorial optimization part of the OP, which sets out to maximize the sum of the collected rewards by selecting a subset of the target locations such that the path length visiting them is within the specified travel budget. The second part is the continuous optimization of the DTRP which, for a given sequence of target locations themselves, sets out to find appropriate waypoint heading angles of Dubins vehicle and also the waypoint locations themselves in the neighborhoods of the selected target locations. Both parts have to be addressed at the same time, as the OP subset selection influences the continuous DTRP optimization, which on the other hand influences the path length constrained by the combinatorial OP. The addressed DOPN is therefore incrementally formulated from the OP and the DTRP in the following subsections.

3.1 Orienteering Problem (OP)

The OP assumes a given set of target locations to be visited $S = \{s_1, \cdots, s_n\}$, where each target location $s_i = (t_i, r_i)$ consists of its position in the plane $t_i \in \mathbb{R}^2$ and the associated reward r_i. The reward of all target locations is expected to be strictly positive $r_i \in \mathbb{R}_{>0}$, with the exception of the predefined starting location s_1 and ending location s_n with zero rewards $r_1 = r_n = 0$. Furthermore, the problem is constrained by the given maximal allowed travel budget T_{max}, i.e., the path length of the vehicle is limited by this value.

The objective of the OP is to maximize the sum of the collected rewards $R = \sum_{r_i \in S_k} r_i$ by selecting a subset of k target locations $S_k \subseteq S$. However, the length of the tour to visit all the locations of subset S_k is constrained by T_{max}, and therefore, the path length has to be taken into account during the selection of S_k. The path can be described as a sequence of target location indexes Σ_k, in which the path visits the selected target locations $\Sigma_k = (\sigma_1, \cdots, \sigma_k)$, with $1 \le \sigma_i \le n$, $\sigma_i \neq \sigma_j$ for $i \neq j$, $s_{\sigma_h} \in S_k$ where $h \in (1, \ldots, k)$ and $\sigma_1 = 1$, $\sigma_k = n$. Using the predefined starting and ending locations in the permutation ($\sigma_1 = 1$, $\sigma_k = n$), the solution of the OP is determined by searching over all possible values of k, S_k, and Σ_k. In the ordinary OP (Gunawan et al. 2016), the Euclidean distance $\mathcal{L}_e(s_{\sigma_i}, s_{\sigma_j})$ is used as the travel cost between two target locations s_{σ_i} and s_{σ_j}. Having these preliminaries, the OP can be formulated as the optimization problem:

Problem 1 (Orienteering Problem (OP))

$$\begin{aligned}&\text{maximize}_{k,S_k,\Sigma_k}\; R = \sum_{i=1}^{k} r_{\sigma_i}\\&\text{subject to}\\&\sum_{i=2}^{k} \mathcal{L}_e(t_{\sigma_{i-1}}, t_{\sigma_i}) \le T_{max}\\&\sigma_1 = 1, \sigma_k = n\ .\end{aligned} \tag{1}$$

3.2 Dubins Touring Regions Problem (DTRP)

In the DTRP, Dubins vehicle model is utilized to plan a curvature-constrained data collection path. Dubins (1957) showed that the shortest path between two configurations of Dubins vehicle can be found by a closed-form expression, and the path is one of six possible maneuvers of CSC or CCC type, where 'C' stands for turning right or left and 'S' means going straight. A configuration of Dubins vehicle $q = (p, \theta)^T = (x, y, \theta)^T$ can be described by its position $p = (x, y)$ in the plane, i.e., $p \in \mathbb{R}^2$, and the vehicle heading angle θ, $\theta \in \mathbb{S}^1$. The kinematic model of Dubins vehicle shown in (2) uses a constant forward velocity v_c and a control input u, which steers the vehicle. The minimal turning radius ρ of Dubins vehicle is assumed to be constant.

$$\dot{q} = \begin{bmatrix} \dot{p}^T \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v_c \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{u}{\rho} \end{bmatrix}, u \in [-1, 1] \tag{2}$$

In multi-goal path planning with curvature-constrained Dubins vehicle formulated as the DTSP or the DOP and their variants with neighborhoods, the important issue of the continuity of heading angles has to be solved. The analytical solution of optimal Dubins maneuvers (Dubins 1957) provides the shortest path between two target locations with known heading angles. However, the heading angles have to be appropriately found to connect multiple Dubins maneuvers into a path of minimal length over multiple target locations with a priori unknown heading angles. For a given sequence of waypoint locations, the problem of determining the optimal heading values is called the Dubins Touring Problem (DTP) (Faigl et al. 2017). For the purposes of the OP, we can consider a variant of the DTP in which the target locations in S_k are visited in the sequence defined by $\Sigma_k = (\sigma_1, \cdots, \sigma_k)$ with specified starting and ending locations $\sigma_1 = 1$ and $\sigma_k = n$, respectively. The problem is then to find a vector of the waypoint heading angles $\Theta_k = (\theta_{\sigma_1}, \cdots, \theta_{\sigma_k})$ that connects Dubins maneuvers at the target locations. The solution of the DTP minimizes the sum of the length of Dubins maneuvers, where $\mathcal{L}_d(q_{\sigma_i}, q_{\sigma_j})$ denotes the length of the shortest Dubins maneuver (Dubins 1957) between configurations q_{σ_i} and q_{σ_j}.

The DTRP additionally requires to find the waypoint locations within a disk-shaped neighborhood of each target location. The non-zero sensing distance in the DTRP is denoted as the neighborhood radius δ defining a δ-radius disk centered at the respective target location. The same neighborhood radius is used for all the target locations in the given sequence Σ_k, with the exception of the starting s_1 and ending s_k locations, which are assumed to have a zero neighborhood radius due to the vehicle taking off and landing at these locations. The DTRP extends the DTP to a variant where an additional vector of the waypoint locations $P_k = (p_{\sigma_1}, \cdots, p_{\sigma_k})$ has to be found. Each $p_{\sigma_i} \in \mathbb{R}^2$ defines the location within the δ neighborhood of the target location $s_{\sigma_i} = (t_{\sigma_i}, r_{\sigma_i}) \in S_k$ such that $\|p_{\sigma_i}, t_{\sigma_i}\| \le \delta$ for $i \in (2, k-1)$ and $\|p_{\sigma_i}, t_{\sigma_i}\| = 0$ for $i = 1, k$. The DTRP sets out to minimize the length $\mathcal{L}(\Theta_k, P_k)$ of the Dubins tour over the given sequence of targets Σ_k by optimizing both the vector of the waypoint heading angles Θ_k and the vector of the waypoint locations P_k that are inside the neighborhoods of the particular selected targets. This type of continuous optimization problem is complex, as any change of, e.g., heading angle θ_{σ_i} at a single target location, influences not only the optimal location p_{σ_i} of the same waypoint, but also other adjacent waypoint heading angles and locations. The same applies to changes in the waypoint locations P_k. The DTRP can be summarized as the following optimization problem.

Problem 2 (Dubins Touring Regions Problem (DTRP))

$$\begin{aligned}&\text{minimize}_{\Theta_k,P_k}\; \mathcal{L}(\Theta_k, P_k) = \sum_{i=2}^{n} \mathcal{L}_d(q_{\sigma_{i-1}}, q_{\sigma_i})\\&\text{subject to}\\&\quad q_{\sigma_i} = (p_{\sigma_i}, \theta_{\sigma_i}),\, p_{\sigma_i} \in P_k,\, \theta_{\sigma_i} \in \Theta_k,\, i \in (1, k)\ ,\\&\quad \|p_{\sigma_i}, t_{\sigma_i}\| \le \delta\ ,\ i \in (2, k-1)\ ,\\&\quad \|p_{\sigma_1}, t_{\sigma_1}\| = 0\ ,\ \|p_{\sigma_k}, t_{\sigma_k}\| = 0\ ,\\&\quad \sigma_1 = 1\ ,\ \sigma_k = n\ .\end{aligned} \tag{3}$$

3.3 Dubins Orienteering Problem with Neighborhoods

The DOPN combines combinatorial OP reward maximization with continuous path length minimization of the DTRP. However, both optimization problems combined in the DOPN have to be addressed simultaneously, due to their mutual influence. The DOPN can be therefore expressed in a single optimization formulation:

Problem 3 (Dubins Orienteering Problem with Neighborhoods (DOPN))

$$\text{maximize}_{k,S_k,P_k,\Sigma_k,\Theta_k}\ R = \sum_{i=1}^{k} r_{\sigma_i}$$

subject to

$$\begin{aligned}
&\sum_{i=2}^{k} \mathcal{L}_d(q_{\sigma_{i-1}}, q_{\sigma_i}) \le T_{max}\ , \\
&q_{\sigma_i} = (p_{\sigma_i}, \theta_{\sigma_i}),\ p_{\sigma_i} \in P_k,\ \theta_{\sigma_i} \in \Theta_k,\ i \in (1,k)\ , \\
&\|p_{\sigma_i}, t_{\sigma_i}\| \le \delta\ ,\ i \in (2, k-1)\ , \\
&\|p_{\sigma_1}, t_{\sigma_1}\| = 0\ ,\ \|p_{\sigma_k}, t_{\sigma_k}\| = 0\ , \\
&\sigma_1 = 1\ ,\ \sigma_k = n\ .
\end{aligned} \tag{4}$$

4 Proposed Approach for the DOPN

The proposed approach for the DOPN is a novel variant of the Variable Neighborhood Search (VNS) metaheuristic, which combines combinatorial optimization and continuous optimization. The preliminary VNS-based method for the DOPN proposed in (Pěnička et al. 2017b) is purely sampling-based combinatorial optimization, which requires significantly longer computational times and achieves lower quality solutions. In this paper, we propose the VNS-based method, which additionally contains continuous optimization to improve the solution quality and to reduce the computational burden. The method proposed here uses low-density initial sampling for solving the DTRP subproblem (sampling the waypoint heading angles and the waypoint locations); however, it also employs continuous optimization of the waypoints. This kind of waypoint optimization can shorten the actual tour to cover the same subset of target locations, and thus it potentially allows visits to additional as yet unvisited target locations, without violating the travel budget constraint. Optimized waypoints that shorten the actual path are therefore added to the initial sampled waypoints to be used further for OP optimization.

The proposed method for the DOPN is based on the VNS metaheuristic (Mladenović and Hansen 1997), which has been introduced for combinatorial optimization in various problems (Hansen and Mladenović 2001) and its principles are also applicable for continuous optimization (Mladenović et al. 2008). VNS uses *shake* and *local search* procedures to iteratively improve the best achieved incumbent solution. Both procedures use the l_{max} predefined operators in the context of the VNS described as neighborhood structures $N_l, l = 1, \ldots, l_{max}$, where, in each VNS iteration, the neighborhood N_l is gradually increased when no better solution is found. The *shake* procedure uses the incumbent solution and randomly changes it using one of its operators to get from possible local optima. The randomly changed incumbent solution is then used by the *local search* procedure in an attempt to increase the quality of the solution above the incumbent solution.

The VNS-based algorithm for the DOPN uses equidistant initial sampling of the waypoints in the δ-radius neighborhood disk centered at the respective target locations $s_{\sigma_i} \in S$. Each waypoint consists of the waypoint location p_{σ_i} on the circumference of the neighborhood circle and also the heading angle of Dubins vehicle θ_{σ_i} at the waypoint location. The initial sampling uses o equidistantly placed waypoint locations along the circumference of the δ-radius circle. Each such waypoint location is described throughout the VNS-based algorithm by its directional angle $\langle 0, 2\pi)$ from the respective target location. This allows the waypoint location to be described by only one parameter and, like the description by two parameters (x, y), does not restrict solutions of the DOPN. Zero neighborhood radius is used for both the starting locations and the ending location specified by the DOPN, as the exact start and end position of the vehicle is considered, and therefore the $o = 1$ location sample is used. The heading angle is similarly sampled into m values from the interval $\langle 0, 2\pi)$ for each of the o waypoint location samples. The sampling approach requires $(o \cdot m)$ samples per target location, which is sufficient for the initial solution of the DOPN (Pěnička et al. 2017b). However, a high sampling rate, which is needed for finding high-quality solutions, is very computationally demanding, and most of the waypoint samples are never used in the improvements to the solution. Therefore, we propose to use low-density sampling of the waypoints from the initialization, together with the online addition of the optimized waypoints, which shorten the current paths, to the set of initial waypoint samples. DOPN paths are then created on the optimized waypoint samples where the appropriate waypoints, i.e., vectors Θ_k and P_k, are selected for the target sequence Σ_k using the shortest path in the graph of samples between the starting and ending target locations.

The neighborhood operators used for combinatorial optimization inside the VNS algorithm for the DOPN, namely **Path Move**, **Path Exchange**, **One Point Move**, and **One Point Exchange**, were introduced for the Euclidean OP in (Sevkli and Sevilgen 2006). The modified version of the same operators was also used in the initial solution of the purely sampling-based DOPN in (Pěnička et al. 2017b). The novel VNS *shake* procedure for the DOPN consists of the following $l = 1, \dots, 3$ neighborhood operators: **Path Move** and **Path Exchange**, which are further described in detail in Section 4.1, and **Waypoint Shake**, which is described in Section 4.3. The particular l for the individual operators of the *shake* procedure are:

- **Waypoint Shake** ($l = 1$);
- **Path Move** ($l = 2$);
- **Path Exchange** ($l = 3$).

The *local search* procedure consists of three operators, **One Point Move** and **One Point Exchange**, which are discussed in Section 4.2, and **Waypoint Improvement**, which is described in Section 4.3. The particular l for the individual operators of the *local search* procedure are:

- **Waypoint Improvement** ($l = 1$);
- **One Point Move** ($l = 2$);
- **One Point Exchange** ($l = 3$).

The proposed continuous optimization of the waypoints is performed by a combination of the Waypoint Shake operator in the *shake* procedure and the Waypoint Improvement operator in the *local search* procedure. The operators randomly change the waypoint of the current solution of the DOPN, and then improve the waypoints by iterative usage of local improvements. The continuous optimization operators ($l = 1$) are prioritized in order to shorten any newly found solution (see Algorithm 1) and thus to allow the combinatorial operators ($l = 2, 3$) to add previously unvisited target locations within the same budget. Local optimization of the waypoint is also performed during the *local search* One Point Move and One Point Exchange operators, when a new unvisited target location is added to the path. The improvement ratio α_{imp} defines the minimal collected reward $R_{imp} = \alpha_{imp} R_{init}$ when the newly-added target location is optimized for its waypoint samples. Value R_{init} denotes the sum of the rewards collected by the initial greedy solution of the DOPN. This immediate shrinking of the path allows more unvisited target locations to be added within the same travel budget, and at the same time, improvement ratio α_{imp} ensures that only waypoints of promising paths are improved. Ratio α_{imp} thus represents a tradeoff between exploration and exploitation. While low α_{imp} attempts waypoint improvement for all new target location additions made by the local search, a high value of α_{imp} (up to the point where $\alpha_{imp} R_{init}$ is equal to the current maximal reward) tends to exploit (improve) only the best found solution. Having high α_{imp} can thus lead to an even better solution being missed by not continuously optimizing the waypoints of promising solutions. On the other hand, optimizing the waypoints of low-quality solutions is more computationally demanding, mainly due to the large number of additional waypoint samples that are never used in further solutions. The influence of ratio α_{imp} is shown in Section 5.1.

The internal representation of the DOPN solution in the designed VNS-based method consists of the vector $v = (s_{\sigma_2}, \dots, s_{\sigma_{k-1}}, s_{\sigma_{k+1}}, \dots, s_{\sigma_n})$, where the first $k - 2$ elements are the selected target locations of set S_k, together with the starting $s_{\sigma_1} = s_1$ and ending $s_{\sigma_k} = s_n$ target locations ordered according to Σ_k. The rest of the vector elements are the unvisited target locations $(s_{\sigma_{k+1}}, \dots, s_{\sigma_n})$. Any solution of the DOPN is describable only by v on existing waypoint samples, as the appropriate waypoints Θ_k and P_k at the target locations are selected from the samples waypoint graph in such a way that the path over Σ_k is minimal. During combinatorial optimization by the operators Path Move, Path Exchange, One Point Move, and One Point Exchange, the whole solution vector v is used, such that the same operators can change the order of the visited target locations, and new unvisited targets can also be introduced to the solution path.

The proposed VNS-based method for the DOPN is summarized in Algorithm 1. The method starts with the *getReachableLocations* procedure, which filters out all target locations unreachable within the budget to reduce the number of target locations considered to be visited by the travel budget T_{max}. The reachable set of target locations S_r then contains $s_i \in S_r$ such that $\mathcal{L}_e(s_1, s_i) + \mathcal{L}_e(s_i, s_1) - 2\delta \leq T_{max}$. Note that the Euclidean distance with subtracted neighborhood radius is used as the lower bound on the required distance to the target location. This can add some unreachable target locations for Dubins vehicle; however, it does not require to determine the waypoint location and the heading angle that visits the neighborhood of the target location.

Using the set of reachable target locations S_r, the initial solution is created by a method denoted as *createInitialPath*. This method greedily adds target locations into the initial path between the starting location and the ending location with respect to the additional reward per length increase of the data collection path. The initial solution then consists of all such added target locations that fit within the budget constraint T_{max}. Afterward, the VNS uses the neighborhood operators in the *shake* and *local search* procedures, which are described in detail in the following subsections, to improve the incumbent solution P, either by increasing the sum of the collected rewards or by shrinking the length of the equally rewarded solution. The termination condition for the proposed method can be the number of per-

Algorithm 1: VNS method for the DOPN

Input : S – Set of the target locations
Input : T_{max} – Maximal allowed travel budget
Input : o – Initial number of position waypoints for each target
Input : m – Initial number of heading values for each waypoints
Input : α_{imp} – Local waypoint improvement ratio
Input : l_{max} – Maximal neighborhood number
Output: P – Found data collecting path defined by k, S_k, Σ_k, Θ_k, and P_k

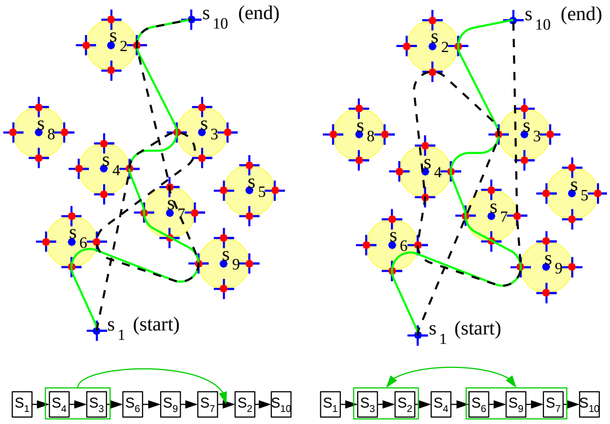
$S_r \leftarrow$ getReachableLocations(S, T_{max})
$P \leftarrow$ createInitialPath(S_r,T_{max}) // greedy
while *Stopping condition is not met* **do**
  $l \leftarrow 1$
  while $l \leq l_{max}$ **do**
    $P' \leftarrow$ shake(P, l)
    $P'' \leftarrow$ localSearch(P', l, α_{imp})
    if $\mathcal{L}_d(P'') \leq T_{max}$ ***and*** $[R(P'') > R(P)$ ***or*** $[R(P'') == R(P)$ ***and*** $\mathcal{L}_d(P'') < \mathcal{L}_d(P)]]$ **then**
      $P \leftarrow P''$
      $l \leftarrow 1$
    else
      $l \leftarrow l + 1$

formed iterations, or the number of iterations without any improvement, or the elapsed computational time, or a targeted sum of collected rewards. For brevity, the solution DOPN path (defined by k, S_k, Σ_k, Θ_k, and P_k) is denoted P, and the sum of the rewards collected by the vehicle traveling along path P is denoted $R(P)$ and its length is denoted as $\mathcal{L}_d(P)$.

4.1 Combinatorial *shake* Operators

The combinatorial part of the *shake* procedure consists of two operators, **Path Move** and **Path Exchange**. Both operators are intended to randomly change the currently best achieved incumbent solution to escape from possible local optima. Changes are made to the underlying sequence Σ_k and subset selection S_k by random reordering of the solution vector v, which internally represents the DOPN solution. Corresponding waypoints of the target locations for the reordered solution are selected from the existing graph of waypoint samples to minimize the overall length of the solution. A DOPN solution is selected from the first $k-2$ target locations in vector v that fit within the budget constraint between the starting location and the ending location.

Operator **Path Move** ($l = 2$), illustrated in Fig. 2a, randomly selects a part of the existing solution and moves it into a different randomly selected place within the solution vector. The operator is implemented by selecting three random indexes inside the solution vector, e.g., $i_1 \in \langle 2, n-1\rangle$, $i_2 \in \langle i_1+1, n-1\rangle$,



(a) **Path Move** (b) **Path Exchange**

Fig. 2: **Path Move** and **Path Exchange** operators with a random change of the initial incumbent solution of the DOPN (dashed black) into a shorter solution (green) by changing the sequence of target locations and selecting the optimal waypoint samples. The combinatorial *shake* operators are shown with waypoint sampling, which consists of $o = 4$ number of neighborhood position samples and $m = 4$ heading samples of Dubins vehicle in each position sample.

$i_3 < i_1$ or $i_3 > i_2$, and $i_{1\ldots3} \neq k$. The DOPN solution vector $v = (s_{\sigma_2}, \ldots, s_{\sigma_n})$ of the initial incumbent solution is then changed, e.g., for the case of $i_3 > i_2$, into $v = (s_{\sigma_2}, \ldots, s_{\sigma_{i_1-1}}, s_{\sigma_{i_2+1}}, \ldots, s_{\sigma_{i_3}}, s_{\sigma_{i_1}}, \ldots, s_{\sigma_{i_2}}, s_{\sigma_{i_3+1}}, \ldots, s_{\sigma_n})$. Note that the operator can change not only the used part of the solution, the part until index $k-1$ that fits within T_{max} between the starting and ending locations, but it can also change the order of the unused target locations. The same property applies to all the other operators for combinatorial optimization.

The **Path Exchange** operator ($l = 3$) randomly selects two non-overlapping parts of the existing solution and switches their position inside the solution. Such a random exchange can be realized by selecting four feasible random indexes $i_1 \in \langle 2, n-1\rangle$, $i_2 \in \langle i_1+1, n-1\rangle$, $i_3 \in \langle i_2+1, n-1\rangle$, and $i_4 \in \langle i_3+1, n-1\rangle$ with $i_{1\ldots4} \neq k$. The initial solution $v = (s_{\sigma_2}, \ldots, s_{\sigma_n})$ is then modified by the operator into $v = (s_{\sigma_2}, \ldots, s_{\sigma_{i_1-1}}, s_{\sigma_{i_3}}, \ldots, s_{\sigma_{i_4}}, s_{\sigma_{i_2+1}}, \ldots, s_{\sigma_{i_3-1}}, s_{\sigma_{i_1}}, \ldots, s_{\sigma_{i_2}}, s_{\sigma_{i_4+1}}, \ldots, s_{\sigma_n})$. An example of the operator is shown in Fig. 2b.

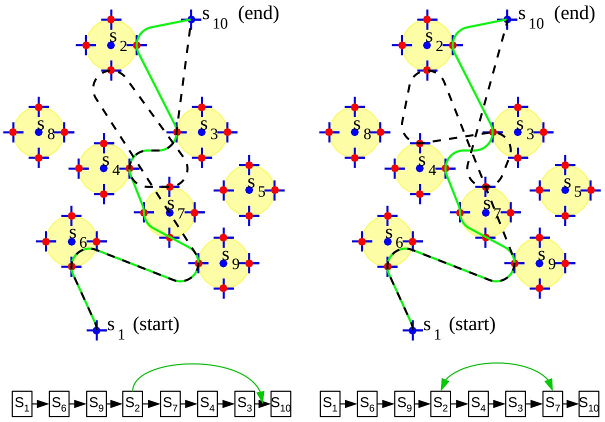
4.2 Combinatorial *local search* Operators

The *local search* operators for combinatorial optimization of the DOPN are **One Point Move** and **One Point Exchange**. The proposed method for the DOPN is based on the Randomized Variable Neighborhood

Search (RVNS), a variant of the VNS where the *local search* procedure is randomized. In the ordinary VNS, the *local search* uses a systematic local search in the solution space; however, in the proposed RVNS-based variant, the solution space is searched by numerous random operations. In both combinatorial local search operators, each random operator is tested for a number of times equal to the square of the number of reachable target locations. Every such random operation that increases the quality of the solution, i.e., increases the sum of the collected reward or decreases the path length for the same reward, is applied, and the operators continue with testing other random changes. Using *local search* randomized optimization, the solution created in the *shake* procedure is deeply searched for local optima in pursuit of a solution that improves the current best incumbent solution. Both the One Point Move operator and the One Point Exchange operator use the graph of the existing waypoint samples, and for any testing sequence Σ_k, represented by solution vector v, they select the waypoint samples that minimize the total path length.

For solutions with promising rewards (i.e., with the sum of the rewards equal to or higher than $R_{imp} = \alpha_{imp} R_{init}$, where R_{init} is the reward of the initial greedy solution created by *createInitialPath*), the operators also perform local optimization of the waypoint samples. When a new target location s_{σ_i} is added into the existing solution with a reward equal to or higher than R_{imp}, the currently selected waypoint heading sample θ_{σ_k} and the waypoint location sample inside the target neighborhood p_{σ_i} are optimized by a hill climbing method, similar to the method used in Waypoint Improvement, introduced in Section 4.3 for decreasing the length of the data collection path. Additionally, the waypoint samples of the adjacent target locations (in the current solution) are optimized, and if the path length after adding the new target location meets the budget constraint the solution is modified, and the optimized samples are inserted into the global graph of the waypoint samples. In this manner, the *local search* operators can fit more target locations within the same budget T_{max}, even with low initial sampling density determined by o and m.

The **One Point Move** operator ($l = 2$) shown in Fig. 3a randomly selects one target location within solution vector v and moves it into a different randomly chosen position inside v. By selecting two random indexes i_1 and i_2, $i_1 \neq i_2$, $i_1 \neq k$, $i_2 \neq k$, without loss of generality $i_1 < i_2$, inside $v = (s_{\sigma_2}, \ldots, s_{\sigma_n})$, the solution of a single move operation is $v = (s_{\sigma_2}, \ldots, s_{\sigma_{i_1-1}}, s_{\sigma_{i_1+1}}, \ldots, s_{\sigma_{i_2-1}}, s_{\sigma_{i_1}}, s_{\sigma_{i_2}}, \ldots, s_{\sigma_n})$. If such a move operation improves the quality of the solution,



(a) **One Point Move** (b) **One Point Exchange**

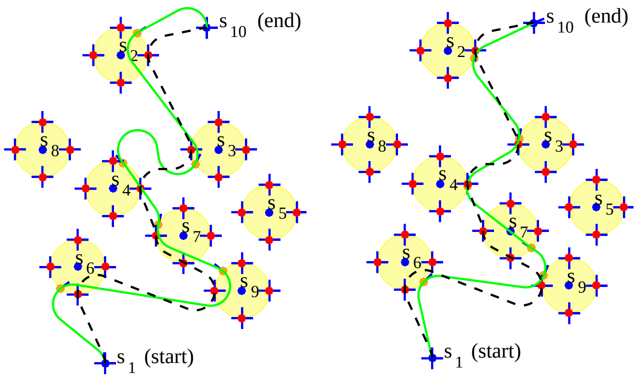
Fig. 3: *Local search* operators **One Point Move** and **One Point Exchange** with waypoint sampling $o = 4$ and $m = 4$. The combinatorial optimization operators randomly move one target location within the solution in the case of One Point Move or randomly exchange two target locations in the solution sequence by the One Point Exchange operator.

the change is applied, and further random One Point Move operations are tested.

The **One Point Exchange** operator ($l = 3$) illustrated in Fig. 3b is similar to the first $l = 2$ operator; however, instead of moving one target location within the solution v, the operator exchanges two randomly selected target locations. The operator can be realized by selecting two random indexes i_1 and i_2, $i_1 \neq i_2$, $i_1 \neq k$, $i_2 \neq k$, within the existing solution $v = (s_{\sigma_2}, \ldots, s_{\sigma_n})$ and by exchanging the target location with the selected indexes $v = (s_{\sigma_2}, \ldots, s_{\sigma_{i_1-1}}, s_{\sigma_{i_2}}, s_{\sigma_{i_1+1}}, \ldots, s_{\sigma_{i_2-1}}, s_{\sigma_{i_1}}, s_{\sigma_{i_2+1}}, \ldots, s_{\sigma_n})$. Like the One Point Move operator, the One Point Exchange operator tests numerous such operations and applies those that improve the quality of the solution.

4.3 Continuous Optimization Operators for the DOPN

This section presents two novel operators used for continuous optimization of the underlying DTRP to minimize the required path length for visiting a selected sequence of target locations. Minimization of the path length is motivated by the idea of fitting additional target locations that slightly violate the budget constraint T_{max}; however, the path length can fulfill T_{max} after optimizing the waypoint samples. Two proposed operators: **Waypoint Shake** inside the *shake* procedure, and **Waypoint Improvement** in the *local search*, are used within the proposed VNS algorithm as the first Neighborhood operators $l = 1$.



(a) **Waypoint Shake** (b) **Waypoint Improvement**

Fig. 4: **Waypoint Shake** and **Waypoint Improvement** with $o = 4$ samples of waypoint locations and $m = 4$ samples of heading angles at each waypoint location. Waypoint Shake randomly changes the current waypoint samples used in the incumbent solution. Waypoint Improvement optimizes the waypoint samples to minimize the length of the solution.

The **Waypoint Shake** operator randomly changes the waypoints currently used by the incumbent solution within the interval $\langle 0, 2\pi)$ for the heading angle and within the δ-radius circle for the waypoint location inside the target neighborhoods. Note that the waypoint location on the δ-radius circle can also be described by the angle within the interval $\langle 0, 2\pi)$, and it can therefore be changed and optimized in a similar way as the heading angle. The operation corresponds to a random change of vectors Θ_k and P_k that describe a solution of the DTRP, and similarly to the other *shake* operators it is intended to get the solution from possible local optima. Waypoint Shake is illustrated in Fig. 4a.

The **Waypoint Improvement** operator is a procedure which utilizes continuous local optimization of both the waypoint locations and the corresponding heading angles to improve the solution produced by Waypoint Shake. This continuous optimization enables solutions to be found closer to the optimum because the configurations are no longer selected from a discrete set of initial samples. This problem is formalized as the DTRP, as introduced in Section 3.2.

Although the sequence of visits to the targets is given, the DTRP remains challenging due to $2k + 2$ continuous variables, where k is the number of currently selected targets to be visited. One variable defines the location of the waypoint on the boundary of the respective target neighborhood, and the other variable defines the waypoint heading angle. The DTRP is addressed by dividing the problem into smaller optimization sub-problems, where each variable is treated separately. The modification of a single variable influences not more than two adjacent Dubins maneuvers, which makes the optimization very fast. However, the variables are mutually affected, and the local optimizations of a single variable are therefore repeated several times. This method has been adopted from the LIO procedure (Váňa and Faigl 2015), originally designed for the DTSPN.

5 Results

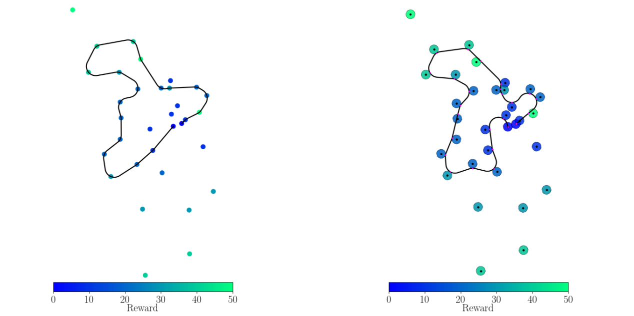
The proposed VNS-based solution to the DOPN has been evaluated on benchmark datasets for the regular OP from the literature, and has also been verified experimentally in a data collection scenario with a real UAV. The computational results on the OP datasets show that the proposed solution of the DOPN increases the so far best achieved collected rewards (Pěnička et al. 2017b) in numerous benchmark instances. The proposed approach also outperforms the only other existing solution of the DOPN, which is based on the Self-Organizing Map (SOM) (Faigl and Pěnička 2017). Moreover, in comparison to the preliminary purely sampling-based approach (Pěnička et al. 2017b), the computational time is significantly decreased by the continuous optimization and solutions with $\approx 90\%$ of the maximally collected rewards are found within several seconds. The experimental verification with the hexarotor UAV shows the benefit of using the neighborhood distance δ in a data collection scenario with a wide field of view camera.

5.1 Computational Results

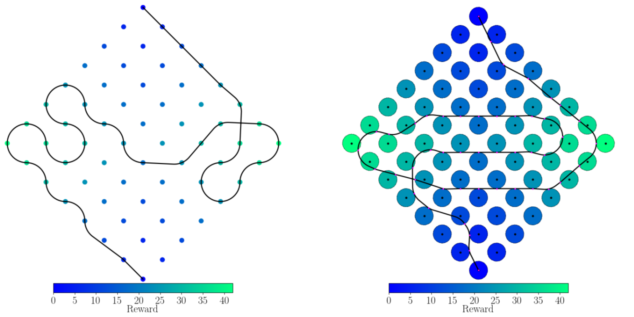
The VNS-based DOPN method has been evaluated on two existing benchmark[1] groups for the ordinary OP (Vansteenwegen 2018). The first group consists of **Set 3**, created by Tsiligirides (1984) with up to 34 randomly placed target locations and with various instances for different budget constraints T_{max}. The second group consists of two larger sets, **Set 64**, with a diamond-shaped structured placement, and **Set 66**, with a square-shaped structured placement, with up to 66 target locations proposed by Chao et al. (1996a). Example solutions of the DOPN for all benchmark sets used here are presented in Fig. 5, showing the benefits of using a non-zero neighborhood radius for maximizing the collected reward.

The computational times reported in this section have been achieved using a single core of the Intel i7

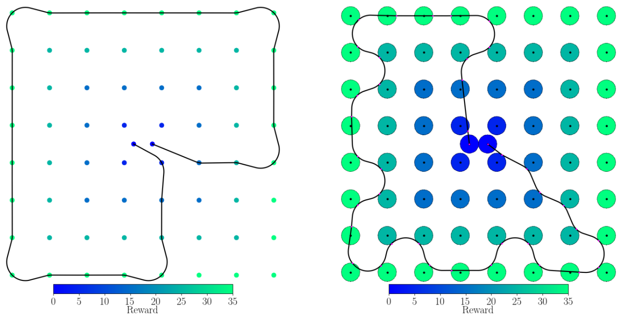
[1] Available online `https://www.mech.kuleuven.be/en/cib/op/#OP`



(a) Set 3, $\delta = 0$, $R = 510$ (b) Set 3, $\delta = 0.5$, $R = 570$



(c) Set 64, $\delta = 0$, $R = 894$ (d) Set 64, $\delta = 0.5$, $R = 1116$



(e) Set 66, $\delta = 0$, $R = 890$ (f) Set 66, $\delta = 0.5$, $R = 1055$

Fig. 5: Solutions of the DOPN for Set 3, Set 64, and Set 66, using $T_{max} = 50$, 55, and 60, respectively. All solutions are shown for the turning radius $\rho = 1.0$ and the neighborhood distance $\delta = 0$ (i.e., a solution of the DOP) on the left and $\delta = 0.5$ on the right. The indicated sum of the collected rewards R is shown to increase in all three instances, in comparison with the rewards of $\delta = 0$, with the VNS-base solution of the DOPN for $\delta = 0.5$.

3.4GHz CPU and C++ implementation of the proposed algorithm. The VNS is a stochastic method, and thus each benchmark instance has been solved 10 times to obtain meaningful statistical results. The initial waypoint sampling of the possible vehicle headings and the waypoint locations are $o = 8$ and $m = 8$. For instances with the zero neighborhood radius $\delta = 0$, only $o = 1$ has been used, and for the zero turning radius of Dubins vehicle $\rho = 0$, the algorithm automatically uses $m = 1$ samples for the heading angle at the target locations. The local waypoint improvement ratio α_{imp} used for defining the minimal collected reward in which the *local search* combinatorial operators start the local optimization of the waypoints during the target location addition has been set to $\alpha_{imp} = 0.95$. The termination criterion used in the computation of the presented results is a combination of the maximum of 10 000 iterations together with the maximal number of 5 000 iterations without any improvement.

Table 1: Maximal Collected Rewards for Set 3

T_{max}	δ=0			δ=0.5			δ=1		
	ρ=0	ρ=0.5	ρ=1	ρ=0	ρ=0.5	ρ=1	ρ=0	ρ=0.5	ρ=1
15	170	160	160	180	180	180	210	210	***200**
20	200	190	180	250	240	230	300	290	***290**
25	260	260	250	320	320	310	370	370	360
30	320	320	**320**	380	370	370	450	450	**450**
35	390	380	380	450	450	440	**510**	500	***490**
40	430	430	*420	500	500	480	570	570	***550**
45	470	460	***460**	550	550	***540**	600	600	***590**
50	*520	520	***510**	580	570	*570	630	630	***620**
55	550	550	530	620	620	600	670	670	***660**
60	580	580	560	650	650	630	710	710	***700**
65	610	600	590	680	670	***660**	750	740	***730**
70	640	630	***620**	720	710	***700**	790	780	***760**
75	670	660	***650**	750	740	**730**	800	800	***790**
80	**710**	690	**680**	790	780	***760**	800	800	800
85	740	730	***710**	800	800	*790	800	800	800
90	770	760	740	800	800	800	800	800	800
95	790	780	770	800	800	800	800	800	800
100	800	800	790	800	800	800	800	800	800
105	800	800	800	800	800	800	800	800	800
110	800	800	800	800	800	800	800	800	800

Table 2: Maximal Collected Rewards for Set 64

T_{max}	δ=0			δ=0.5			δ=1		
	ρ=0	ρ=0.5	ρ=1	ρ=0	ρ=0.5	ρ=1	ρ=0	ρ=0.5	ρ=1
15	96	96	96	204	204	**204**	***312**	312	***306**
20	294	294	252	432	426	***384**	576	570	*552
25	390	384	**342**	564	558	***510**	744	738	***714**
30	474	468	420	714	696	***630**	***954**	948	***936**
35	576	570	**516**	**894**	852	***774**	***1170**	1146	***1128**
40	714	696	624	1068	1026	***900**	***1296**	1272	***1242**
45	816	792	**708**	1164	1134	***990**	1344	1344	**1326**
50	900	882	*798	1248	1212	***1026**	1344	1344	*1344
55	984	972	894	1320	1296	***1116**	1344	1344	1344
60	1062	1044	**954**	1344	1344	**1188**	1344	1344	1344
65	1116	1098	**1020**	1344	1344	***1236**	1344	1344	1344
70	1188	1170	**1092**	1344	1344	***1290**	1344	1344	1344
75	1236	1218	**1134**	1344	1344	***1308**	1344	1344	1344
80	1284	1266	***1176**	1344	1344	***1344**	1344	1344	1344

One of the most important factors that shows the performance of the OP solver, or the DOPN solver in our case, is the maximally achievable sum of the collected rewards. The maximal rewards for various budget constraints T_{max} and for the neighborhood distance $\delta \in \{0, 0.5, 1.0\}$ and turning radii $\rho \in \{0, 0.5, 1.0\}$ are shown in the tables. In particular, the results for Set 3, Set 64, and Set 66 are presented in Table 1, Table 2, and Table 3, respectively. Due to the computational requirements for computing all the instances for various δ and ρ, the results for Tables 1–3 have been obtained with a grid of Xeon CPUs running at 2.2 GHz to 3.4 GHz. The solutions with an improved maximal collected reward with respect to the previously best found

Table 3: Maximal Collected Rewards for Set 66

T_{max}	δ=0			δ=0.5			δ=1		
	ρ=0	ρ=0.5	ρ=1	ρ=0	ρ=0.5	ρ=1	ρ=0	ρ=0.5	ρ=1
5	10	10	0	20	15	0	35	25	0
10	40	40	40	70	70	55	105	100	90
15	120	100	***100**	160	160	***160**	***225**	220	***205**
20	*205	200	195	265	265	***260**	***385**	370	***360**
25	280	280	275	**400**	380	***375**	540	540	540
30	400	380	370	***500**	495	***475**	685	685	*655
35	*465	465	455	605	595	***590**	870	845	835
40	**575**	570	***545**	**735**	710	***680**	**985**	965	***960**
45	650	650	*645	**840**	815	***775**	1135	1100	***1130**
50	730	710	**710**	920	920	***865**	1275	1240	*1235
55	**825**	825	820	***1050**	1015	***950**	1390	1370	***1380**
60	915	895	**890**	**1165**	1120	***1055**	**1555**	1500	***1485**
65	980	950	955	**1265**	1205	***1155**	1620	1605	***1590**
70	**1070**	1050	1070	**1360**	1315	***1260**	**1680**	1650	***1620**
75	1140	1090	**1120**	**1450**	1410	***1325**	1680	1680	***1680**
80	1215	1185	**1175**	1535	1490	***1375**	1680	1680	*1680
85	1270	1255	***1240**	1605	1570	***1410**	1680	1680	1680
90	1340	1310	1295	1635	1620	***1500**	1680	1680	1680
95	1395	1390	1365	1680	1665	***1565**	1680	1680	1680
100	**1465**	1445	1420	1680	1680	***1605**	1680	1680	1680
105	1520	1505	***1480**	1680	1680	***1670**	1680	1680	1680
110	1550	1550	**1535**	1680	1680	***1680**	1680	1680	1680
115	1595	1580	1565	1680	1680	***1680**	1680	1680	1680
120	1625	1625	**1610**	1680	1680	1680	1680	1680	1680
125	1670	1655	1640	1680	1680	1680	1680	1680	1680
130	1680	1680	1670	1680	1680	1680	1680	1680	1680

solutions provided by the purely sampling VNS-based solution of the DOPN (Pěnička et al. 2017a) without continuous optimization are displayed in bold font. In all cases, the maximal collected rewards are the same as, or better than, the previously best found solutions.

The instances with significantly better rewards using significance level $\alpha = 0.05$ are denoted by '*' based on a t-test comparison between the proposed method and the purely sampling VNS-based solution. As is shown in Tables 1–3, the maximal collected reward is improved mainly for non-zero neighborhood distances δ and turning radii ρ. This is caused by the fact that the improvements are mainly due to the continuous optimization, which is only effective when at least $\delta > 0$ or $\rho > 0$. Furthermore, the longer δ and ρ are, the greater their influence on the path length (continuously optimized in the proposed approach), and thus on the maximally achievable reward. Increasing the turning radius enlarges the limited path length and thus decreases the collected reward. Larger neighborhood radius, on the other hand, can increase the collected reward due to the distance savings. Both effects can be observed in Tables 1–3 by comparing $\rho = 0$ and $\rho = 1$ for the turning radius, and $\delta = 0$ and $\delta = 1$ for the neighborhood radius. The results for $\rho = 0.5$ do not contain any improvement, as this particular turning radius has no previously known best found solutions (Pěnička et al. 2017a). Furthermore, the maximal reward cannot be improved for a large number of instances where the created path visits all possible target locations. This is mainly noticeable for $\delta > 0$ and higher budgets T_{max}, where from a certain budget the maximally collected reward does not increase. Significantly better results (based on the t-test comparison) are in most cases in the same instances where the maximal collected reward is improved. However, for several maximal reward improvements, the newly found maximal reward tends to be an outlier, and the reward is not systematically higher. On the other hand, several instances have significantly better rewards without improving the maximal score. This is caused by the closeness of the average reward of the proposed method to the known maximum, together with a small standard deviation in comparison with the purely sampling VNS-based solution. We refer to an enlarged variants[2] of Tables 1–3, which contain the average rewards and the standard deviations of all instances.

The continuous optimization of the waypoint samples is one of the main improvements of the proposed DOPN algorithm in comparison to the previous algorithm proposed in (Pěnička et al. 2017b). Waypoint optimization is used in two main parts of the proposed VNS-based algorithm. The first part is used in the combinatorial *local search* operators One Point Move and One Point Exchange while testing an additional target insertion into a solution with the minimal reward of $R_{imp} = \alpha_{imp} R_{init}$. The waypoint samples of the inserted and adjacent target locations are locally optimized to shrink the length of the solution below T_{max}. The second waypoint optimization is through the Waypoint Shake and Waypoint Improvement operators, which randomly change the waypoint samples and optimize the randomly changed waypoints to minimize the length of the incumbent solution. Fig. 6 shows a comparison of the solution quality as the average sum and as the maximal sum of the collected rewards over the computational time. The solution obtained for 'High sampling DOPN' uses $o = 12$ and $m = 12$ samples and the 'Low waypoint sampling' solution uses $o = 4$ and $m = 4$ samples. Both use only combinatorial optimization with sampled waypoints, as proposed in (Pěnička et al. 2017b). The reward improvement for the solutions denoted as 'Local optimization *local search*' uses local waypoint improvement during the combinatorial *local search* operators, together with low initial waypoint sampling. The solution denoted as 'With Waypoint Improvement' shows the reward improvement with waypoint optimization employed both in local waypoint improvement during combinatorial *local search* and by using the Waypoint Shake and Waypoint Improvement operators. Low waypoint sampling $o = 4$ and $m = 4$ is also used for the solution with both waypoint optimizations.

[2] `https://archive.org/download/vns-dopn/results.pdf`

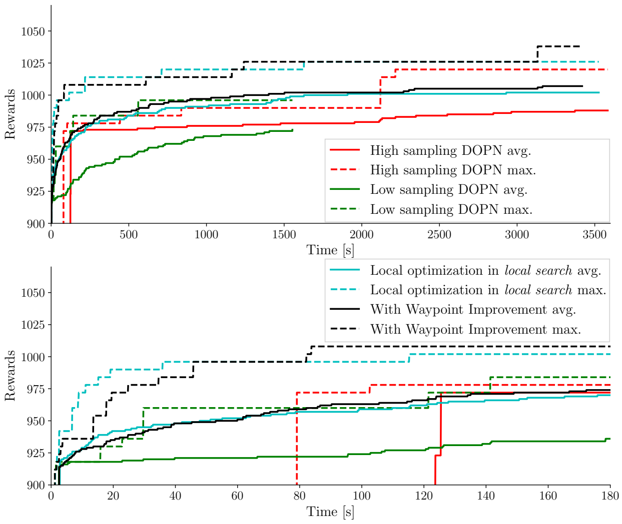


Fig. 6: Comparison of the average sum and the maximal sum of the collected rewards over time for high $o = m = 12$ and low $o = m = 4$ dense initial sampling of the DOPN without and with waypoint optimization. The solution denoted as 'Local optimization in *local search*' uses waypoint optimization only in the combinatorial *local search* operators and the 'With Waypoint Improvement' solution uses additionally the Waypoint Shake and Waypoint Improvement operators. The algorithm performance is shown for Set 64 with $T_{max} = 50$, $\rho = 1.0$ and $\delta = 0.5$. The upper plot shows the reward progress over one hour of maximal computational time, while the lower plot shows a detail of the initial 180 seconds of computation.

The comparison shows that waypoint continuous optimization increases the maximal achieved sum of the collected rewards within the maximal one hour of computational time, which was used in these tests as an additional stopping criterion. The maximum sum of the achieved reward is (similarly to the results in Table 1–3) considered as one of the most important aspects of the OP solver that shows the limiting extreme-most performance of the proposed method. The initial solution for low-density sampling together with waypoint optimization (denoted as 'With Waypoint Improvement') provides solutions with more than 90% of the best solutions within a few seconds only. The initial solution for a high sampling solution, however, takes approximately two minutes to compute, and in that time its maximally achieved reward is outperformed by the initially low sampling solution with both waypoint optimizations. The high sampling approach has a bigger average reward than the proposed method shortly after initialization (between time 125 s and 149 s) due to the quality of the initial solution, which influences the average. However, the high sampling approach is limited by static samples, which also limit the maximally achievable reward during the computation and also result in a lower average reward than that of the proposed method after 167 s of calculation.

The local optimization in *local search* is highly influenced by the selection of the improvement ratio α_{imp}, which determines how rewarded (compared to the reward of initial solution) a solution has to be in order to perform the local waypoint optimization of newly added target locations. Fig. 7 shows a comparison of the selected values of α_{imp} with the average and maximal sum of the collected rewards over the computational time.

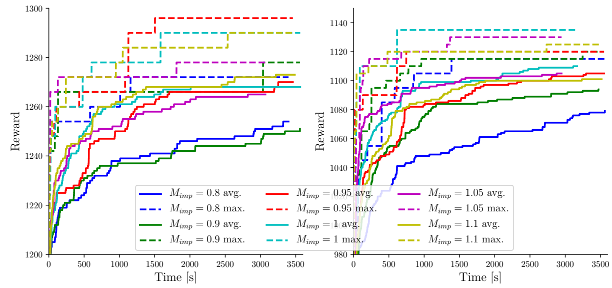


Fig. 7: Comparison of the average and maximal sum of the collected rewards for improvement ratio $\alpha_{imp} \in \{0.8, 0.9, 0.95, 1.0, 1.05, 1.1\}$ over the computational time for Set 64, on the left, with $T_{max} = 55$, $\rho = 0.5$, $\delta = 0.5$ and for Set 66, on the right, with $T_{max} = 60$, $\rho = 0.5$, $\delta = 0.5$.

The comparison shows that higher $\alpha_{imp} \in \{1.05, 1.1\}$ tends to optimize only the current incumbent solution which is demonstrated by the faster initial growth of the average rewards. In the same time, however, the higher α_{imp} prohibits continuously optimizing enough promising solutions in order to find even better than the currently best found solution, which is reflected in the fact that it cannot find the best known solutions. On the other hand, the lower $\alpha_{imp} \in \{0.8, 0.9\}$ slows down the computation of local search operators (by performing waypoint optimization on more non-promising solutions), which results in both lower average rewards and lower maximal rewards. Maximal rewards are achieved by either $\alpha_{imp} \in \{0.95, 1.0\}$ based on the dataset instance being solved.

To the best of our knowledge, the only other existing algorithm for solving the DOPN is the SOM-based approach for the so-called Close Enough Orienteering Problem, which is in fact the DOPN emphasizing the usage of circular neighborhoods. A comparison of the proposed VNS-based and SOM-based algorithms for various neighborhood radii is presented in Fig. 8 for 20 runs of VNS per instance, and for 80 runs in the case of the SOM-based algorithm. The results show that for all tested neighborhood distances, the proposed method produces solutions with similar or significantly better results than the SOM-based algorithm (Faigl and Pěnička 2017).

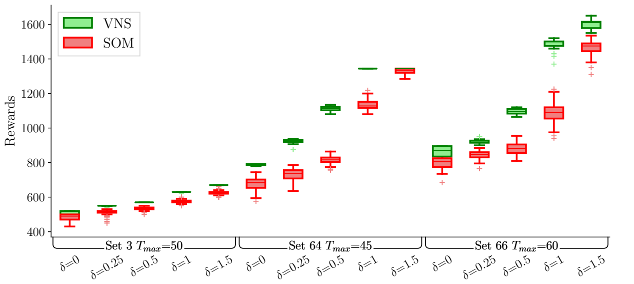


Fig. 8: Comparison of the proposed VNS-based solution and the SOM-based solution (Faigl and Pěnička 2017) of DOPN for selected neighborhood distances. The collected rewards are shown for Dubins vehicle with turning radius $\rho = 0.5$ for all Set 3, Set 64, and Set 66.

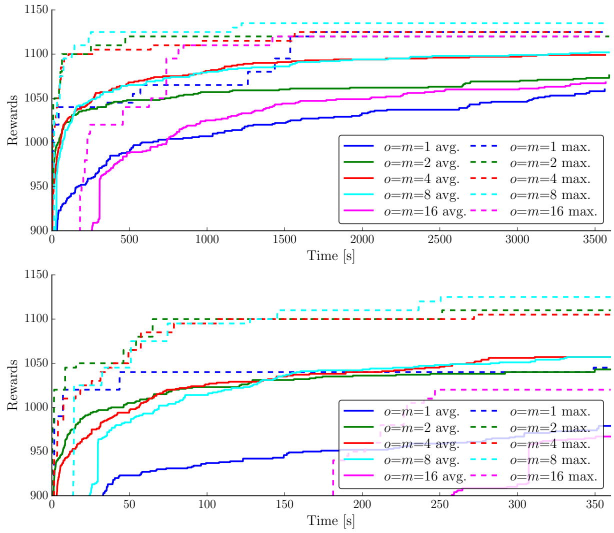


Fig. 9: Comparison of different initial waypoint sampling densities for Set 66 with $T_{max} = 60$, $\rho = 0.5$, and $\delta = 0.5$. In the upper plot, the average and maximal collected rewards are shown over one hour of the maximal computational time. The lower plot shows a detail of the average and maximal rewards within the initial 360 seconds of computation.

The performance of the proposed algorithm is significantly influenced by the initial waypoint sampling density defined by the number of Dubins vehicle heading samples m at each of the o waypoint location samples. Waypoint sampling mainly influences the computational time and the maximally achievable sum of the collected rewards. Using high sampling density such as $o = 12$ and $m = 12$ requires much more computational time in the local search One Point Move and One Point Exchange operators for selecting the samples with the shortest path for a given sequence of target locations. However, with high sampling density, the quality of the solution as the sum of the collected rewards is higher than for low-density sampling. For the newly proposed VNS-based solution of the DOPN with optimization of the waypoint samples, it is possible to achieve the same solution quality with low initial sampling through optimization of the samples. The initial sampling therefore mainly influences the evolution of the solution quality, i.e., the maximal sum and the average sum of the collected rewards, over the computational time, as shown in Fig. 9.

The comparison of the initial waypoint sampling shows that both for very low sampling $o = m = 1$ and for very high sampling $o = m = 16$, the average and maximal collected rewards are below other medium sampling densities. The highest maximal and average rewards are collected with waypoint sampling $o = m = 8$, with similar results for $o = m = 4$. However, the computational time required for creating the initial solution with $o = m = 8$ is approximately 17 s, while for a lower sampling density $o = m = 4$, it is within 1.5 s.

Fig. 9 also shows the computational requirements of the proposed VNS-based solution for the DOPN, and can be used for a comparison of the computational time for achieving a certain sum of collected rewards. The SOM-based solution (Faigl and Pěnička 2017) requires, for Set 66, $T_{max} = 60$, $\delta = 0.5$ and $\rho = 0.5$, an average computational time of 33.8 s with the maximal achieved rewards $R = 955$ and average $\overline{R} = 880$. For the same configuration of the problem, Fig. 9 shows that in 33.8 s the average sum of the collected rewards is 970 for $o = m = 8$, 985 for $o = m = 4$, and 997 for $o = m = 2$. The maximal collected rewards at the same time are 1045 for $o = m = 8$, 1040 for $o = m = 4$, and 1050 for $o = m = 2$. The proposed VNS-based solution outperforms the state-of-the-art algorithm not only in the maximally achievable sum of collected rewards but also regarding the computational time required for achieving a certain sum of collected rewards. This is very important for a real deployment of the method, as it is demonstrated in the following section.

5.2 Experimental Verification

The proposed method was experimentally verified in a visual data collection scenario with a real hexarotor UAV in an outdoor environment. Although the Vertical Take-Off and Landing (VTOL) UAV does not necessarily have to be modeled as Dubins vehicle, the model is convenient for the VTOL UAV when traversing a curvature-constrained path at a constant speed. Constant speed flights of the VTOL can be beneficial for visual data collection missions, where additional visual information during flights between target location neighborhoods can be further used, and constant speed improves the quality of the images that are taken. From the selected constant speed v_c and the maximal acceleration a_{max} of the UAV, the minimal turning radius of Dubins vehicle can be computed using the equation of circular motion with constant speed $\rho = v_c^2/a_{max}$. Note that for the VTOL (not so much for the fixed-wing UAV), the solution quality can be improved by allowing acceleration to the maximal speed during straight line segments of Dubins maneuvers. This, however, is only a technical consideration, which does not require any change to the proposed algorithm. Considering these accelerated Dubins maneuvers would only require the time of flight cost instead of length cost, and would change the length-budget constraint to a time-budget constraint. We therefore consider for experimental verification the DOPN as defined in Problem 3 with standard Dubins maneuvers for VTOL, which also provides a better comparison of the results. We refer to Pěnička et al. (2017a) for a comparison of an experimental deployment of the DOP and the ordinary OP with a straight line trajectory and sharp turns. The advantage of using Dubins vehicle model for fast and reliable visits to multiple locations to be scanned by the onboard camera was also demonstrated by our team in the third challenge of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) competition, which also motivated the research presented here. The effectiveness of fast flying using Dubins vehicle model (e.g., demonstrated in the MBZIRC challenge in the solution of the DTSP) resulted in the best performance among all 143 competing teams.

The used hexarotor UAV was designed for the MBZIRC competition[3] and was built on the DJI hexacopter F550 frame with E310 DJI motors and with the PixHawk Autopilot low-level flight controller (Meier et al. 2012). The low-level localization of the PixHawk Autopilot is realized as a combination of the standard GPS with a compass and with the accelerometers and gyroscopes at the lowest level. To increase the localization precision, the system uses PRECIS-BX305 (Tersus-GNSS 2018) RTK GPS with centimeter accuracy and also the *TeraRanger One* laser rangefinder for measuring the distance from the ground. The onboard Intel NUC-i7 mini PC provides enough resources for calculating the plan for the addressed data collection scenario formulated as the DOPN. In addition, the onboard computer realizes the Model Predictive Control (MPC) trajectory controller (Báča et al. 2016) for trajectory tracking, UAV localization estimation, and sensor fusion. For a visual information gathering task, a high-resolution wide field of view Mobius ActionCam camera was used. The hardware components are summarized in Fig. 10.

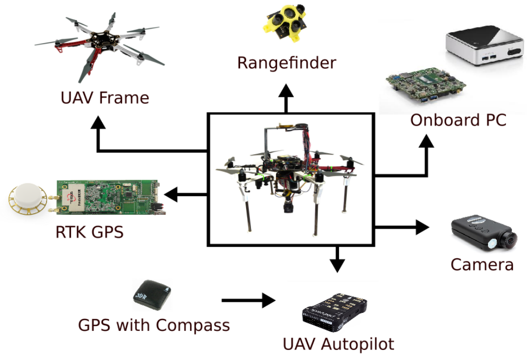


Fig. 10: Hardware components of the hexarotor UAV for experimental verification of the data collection scenario formulated as the Dubins Orienteering Problem with Neighborhoods.

The results from the experimental verification of the proposed DOPN method in a realistic data collection scenario using the onboard camera are shown in Fig. 11. During the experiment, constant vehicle speed $v_c = 4\,\mathrm{ms}^{-1}$ was used together with maximal acceleration $a_{max} = 2.6\,\mathrm{ms}^{-2}$, which resulted in $\rho = 6.15\,\mathrm{m}$ turning radius of Dubins vehicle. The scenario for the experimental verification consists of 19 target locations, including the starting and ending locations, with the budget constraint set to $T_{max} = 150\,\mathrm{m}$. The planned trajectories for the various neighborhood radii $\delta = \{0, 3, 6\}$ m together with the real flown trajectories of the UAV are depicted in Fig. 11.

The real trajectories deviate slightly from the planned trajectory due to the strong wind conditions and the tightly set maximal acceleration of the vehicle. However, the presented camera images show the target markers placed throughout the experimental area, which were taken at the respective waypoints of the targets, and thus the mission was successfully fulfilled.

The sum of the collected rewards for the increasing neighborhood radii shows the main benefits of using the VNS-based solver for the DOPN with non-zero turning radii, where the collected reward increases with each incrementation of the neighborhood distance. The onboard camera images in Fig. 11 show that the high neighborhood radius of 6 m, with a high collected reward, is usable for a visual data collection scenario of this kind. The complete set of radii used during the experimental verification, together with the corresponding sum of the collected rewards and the path lengths, is shown in Table 4.

[3] See `http://mrs.felk.cvut.cz/mbzirc` for examples of the experimental deployment of the system.

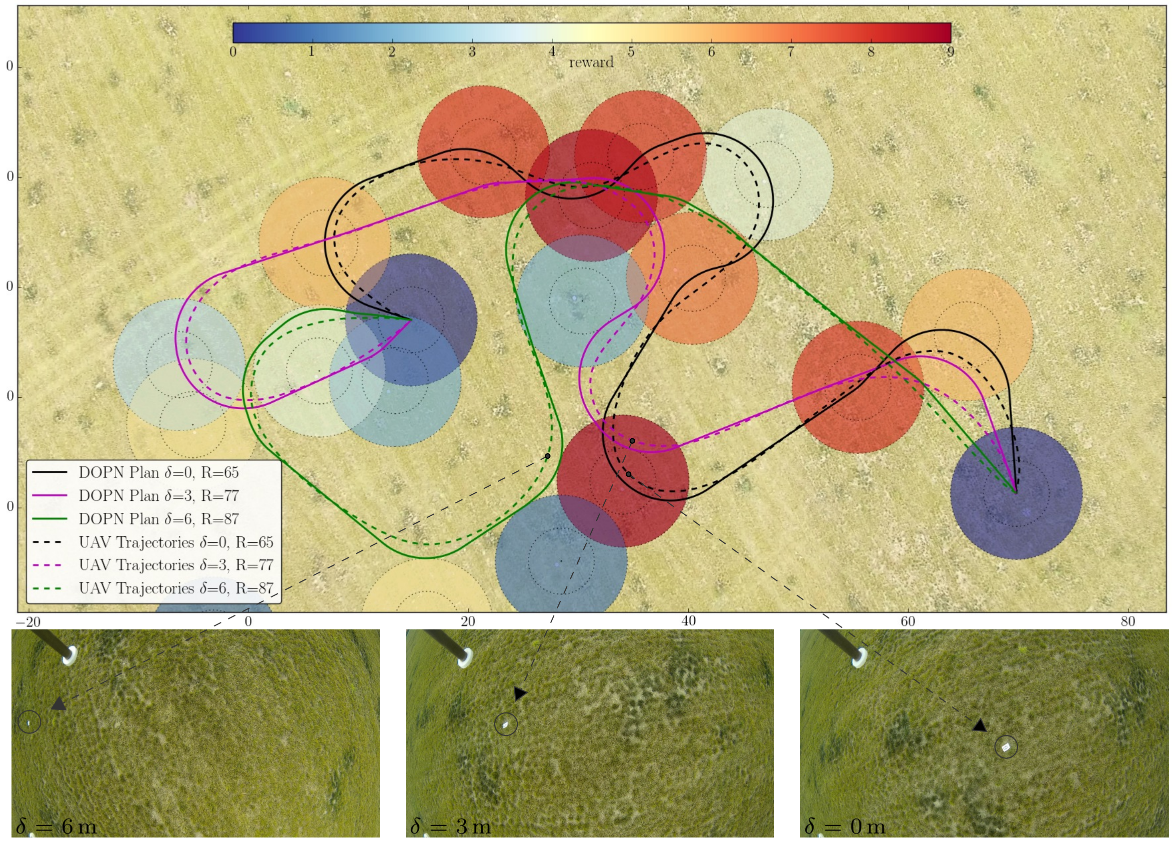


Fig. 11: Snapshots from the experimental verification of the proposed VNS-based algorithm for the DOPN in a data collection scenario with various neighborhood distances δ. We refer to `https://youtu.be/zPXZahW33-w` for supporting video material from the experimental verification of the method.

Table 4: Collected reward and path length from the real experiment for various neighborhood radius δ

δ [m]	0	1	2	3	4	5	6	7
Reward R	65	67	71	77	79	85	87	88
Length [m]	148.0	143.3	146.9	148.0	148.6	147.9	139.0	143.7

6 CONCLUSIONS

In this paper, we introduce a novel approach for curvature-constrained data collection planning with UAVs that is formulated as the Dubins Orienteering Problem with Neighborhoods (DOPN). The DOPN sets out to find a path for Dubins vehicle that maximizes the sum of the collected rewards by visiting a subset of the given target locations with prescribed starting and ending locations and a constrained travel budget. The DOPN uses a predefined circular neighborhood at each target location, motivated by remote data collection from the target locations to save the required travel cost, and thus to increase the sum of the collected rewards within the same budget constraint.

The proposed Variable Neighborhood Search-based method uses a set of neighborhood operators that perform a combinatorial optimization to maximize the sum of the collected rewards by selecting a subset of target locations to be visited, and also by determining the sequence of the visits. The proposed method employs initial low-density waypoint sampling consisting of sampling both Dubins vehicle headings and waypoint locations within the neighborhood of each target location, to quickly determine an initial data collection path by a greedy maximization of reward per tour prolongation. The continuous optimization employed in the proposed VNS neighborhood operators is used for the optimizing the initial waypoint samples to minimize the length of the Dubins path visiting the neighborhoods of the selected target locations. The proposed waypoint optimization increases the sum of the collected rewards by adding unvisited target locations within the prescribed budget constraint.

The computational results show that the proposed VNS-based algorithm is a viable method for solving the DOPN. The continuous optimization employed in the novel approach significantly improves the required computational time, and also improves the best-known solutions in several benchmark instances. The method also outperforms the only other existing SOM-based DOPN approach in both solution quality and computational time. Finally, the experimental verification of the proposed method with a real hexarotor UAV demonstrates the deployment of the proposed solution of the DOPN in a data collection scenario with an onboard camera. The solutions found for the experimental deployment also show the benefits of using a non-zero neighborhood distance on the sum of the collected rewards.

For our future work, we intend to extend the proposed approach to a variant of multi-UAV data collection scenarios and to employ more complex maneuvers, such as cubic splines, which are suitable for the non-constant forward velocity of the VTOL UAV, e.g., as in Faigl and Váňa (2018) for solving TSP-like scenarios.

References

Báča T, Loianno G, Saska M (2016) Embedded model predictive control of unmanned micro aerial vehicles. In: International Conference on Methods and Models in Automation and Robotics (MMAR), pp 992–997

Best G, Faigl J, Fitch R (2016) Multi-robot path planning for budgeted active perception with self-organising maps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3164–3171

Best G, Faigl J, Fitch R (2018) Online planning for multi-robot active perception with self-organising maps. Autonomous Robots 42(4):715–738

Chao IM, Golden BL, Wasil EA (1996a) A fast and effective heuristic for the orienteering problem. European Journal of Operational Research 88(3):475–489

Chao IM, Golden BL, Wasil EA (1996b) The team orienteering problem. European Journal of Operational Research 88(3):464–474

Cohen I, Epstein C, Isaiah P, Kuzi S, Shima T (2017) Discretization-based and look-ahead algorithms for the dubins traveling salesperson problem. IEEE Transactions on Automation Science and Engineering 14(1):383–390

Dubins LE (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of Mathematics 79(3):497–516

Ergezer H, Leblebicioğlu K (2014) 3d path planning for multiple uavs for maximum information collection. Journal of Intelligent & Robotic Systems 73(1):737–762

Faigl J (2017) On self-organizing maps for orienteering problems. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp 2611–2620

Faigl J, Hollinger GA (2014) Unifying multi-goal path planning for autonomous data collection. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 2937–2942

Faigl J, Hollinger GA (2018) Autonomous data collection using a self-organizing map. IEEE Transactions on Neural Networks and Learning Systems 29(5):1703–1715

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

Faigl J, Váňa P (2017) Unsupervised learning for surveillance planning with team of aerial vehicles. In: International Joint Conference on Neural Networks (IJCNN), pp 4340–4347

Faigl J, Váňa P (2018) Surveillance planning with bézier curves. IEEE Robotics and Automation Letters 3(2):750–757

Faigl J, Pěnička R, Best G (2016) Self-organizing map-based solution for the orienteering problem with neighborhoods. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp 1315–1321

Faigl J, Váňa P, Saska M, Báča T, Spurný V (2017) On solution of the dubins touring problem. In: European Conference on Mobile Robotics (ECMR), pp 151–156

Fischetti M, González JJS, Toth P (1998) Solving the orienteering problem through branch-and-cut. INFORMS Journal on Computing 10(2):133–148

Gunawan A, Lau HC, Vansteenwegen P (2016) Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research 255(2):315–332

Hansen P, Mladenović N (2001) Variable neighborhood search: Principles and applications. European Journal of Operational Research 130(3):449–467

Helsgaun K (2000) An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. European Journal of Operational Research 126(1):106–130

Isaacs JT, Klein DJ, Hespanha JP (2011) Algorithms for the traveling salesman problem with neighborhoods involving a dubins vehicle. In: American Control Conference (ACC), pp 1704–1709

Jawhar I, Mohamed N, Al-Jaroodi J, Zhang S (2014) A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks. Journal of Intelligent & Robotic Systems 74(1):437–453

Jorgensen S, Chen RH, Milam MB, Pavone M (2018) The team surviving orienteers problem: routing teams of robots in uncertain environments with survival constraints. Autonomous Robots 42(4):927–952

Le Ny J, Frazzoli E, Feron E (2007) The curvature-constrained traveling salesman problem for high point densities. In: IEEE Conference on Decision and Control, pp 5985–5990

Lugo-Cárdenas I, Flores G, Salazar S, Lozano R (2014) Dubins path generation for a fixed wing uav. In: International Conference on Unmanned Aircraft Systems (ICUAS), pp 339–346

Meier L, Tanskanen P, Heng L, Lee GH, Fraundorfer F, Pollefeys M (2012) Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. Autonomous Robots 33(1):21–39

Mladenović N, Hansen P (1997) Variable neighborhood search. Computers & Operations Research 24(11):1097–1100

Mladenović N, Dražić M, Kovačevic-Vujčić V, Čangalović M (2008) General variable neighborhood search for the continuous optimization. European Journal of Operational Research 191(3):753 – 770

Nguyen JL, Lawrance NRJ, Fitch R, Sukkarieh S (2016) Real-time path planning for long-term information gathering with an aerial glider. Autonomous Robots 40(6):1017–1039

Noon CE, Bean JC (1993) An efficient transformation of the generalized traveling salesman problem. INFOR: Information Systems and Operational Research 31(1):39–44

Oberlin P, Rathinam S, Darbha S (2010) Today's traveling salesman problem. IEEE Robotics Automation Magazine 17(4):70–77

Obermeyer KJ (2009) Path planning for a uav performing reconnaissance of static ground targets in terrain. In: AIAA Guidance, Navigation, and Control Conference

Obermeyer KJ, Oberlin P, Darbha S (2010) Sampling-based roadmap methods for a visual reconnaissance uav. In: AIAA Guidance, Navigation, and Control Conference

Pěnička R, Faigl J, Váňa P, Saska M (2017a) Dubins orienteering problem. IEEE Robotics and Automation Letters 2(2):1210–1217

Pěnička R, Faigl J, Váňa P, Saska M (2017b) Dubins orienteering problem with neighborhoods. In: International Conference on Unmanned Aircraft Systems (ICUAS), pp 1555–1562

Ramesh R, Brown KM (1991) An efficient four-phase heuristic for the generalized orienteering problem. Computers & Operations Research 18(2):151–165

Ramesh R, Yoon YS, Karwan MH (1992) An optimal algorithm for the orienteering tour problem. ORSA Journal on Computing 4(2):155–165

Savla K, Frazzoli E, Bullo F (2005) On the point-to-point and traveling salesperson problems for dubins' vehicle. In: American Control Conference (ACC), vol 2, pp 786–791

Schilde M, Doerner KF, Hartl RF, Kiechle G (2009) Metaheuristics for the bi-objective orienteering problem. Swarm Intelligence 3(3):179–201

Sevkli Z, Sevilgen FE (2006) Variable neighborhood search for the orienteering problem. In: International Symposium on Computer and Information Sciences (ISCIS), pp 134–143

Tersus-GNSS (2018) PRECIS-BX305 GNSS RTK Board. URL `https://www.tersus-gnss.com`, (cited on 2018-07-21)

Thakur D, Likhachev M, Keller J, Kumar V, Dobrokhodov V, Jones K, Wurz J, Kaminer I (2013) Planning for opportunistic surveillance with multiple robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 5750–5757

Tokekar P, Karnad N, Isler V (2014) Energy-optimal trajectory planning for car-like robots. Autonomous Robots 37(3):279–300

Tsiligirides T (1984) Heuristic methods applied to orienteering. The Journal of the Operational Research Society 35(9):797–809

Tsiogkas N, Lane DM (2018) Dcop: Dubins correlated orienteering problem optimizing sensing missions of a nonholonomic vehicle under budget constraints. IEEE Robotics and Automation Letters 3(4):2926–2933

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

Váňa P, Faigl J (2018) Optimal solution of the generalized dubins interval problem. In: Robotics: Science and Systems (RSS)

Vansteenwegen P (2018) The Orienteering Problem: Test Instances – Department of Mechanical Engineering, University of Leuven. URL `http://www.mech.kuleuven.be/en/cib/op/#OP`, cited on 2018-07-21

Vansteenwegen P, Souffriau W, Oudheusden DV (2011) The orienteering problem: A survey. European Journal of Operational Research 209(1):1–10

Wang C, Ma F, Yan J, De D, Das SK (2015) Efficient aerial data collection with uav in large-scale wireless sensor networks. International Journal of Distributed Sensor Networks 11(11)

Wren A, Holliday A (1972) Computer scheduling of vehicles from one or more depots to a number of delivery points. Operational Research Quarterly (1970-1977) 23(3):333–344

Yu J, Schwager M, Rus D (2016) Correlated orienteering problem and its application to persistent monitor-

ing tasks. IEEE Transactions on Robotics PP(99):1–
13