2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Fast Sequence Rejection for Multi-Goal Planning with Dubins Vehicle

Jan Faigl, Petr Váňa, and Jan Drchal

Abstract— **Multi-goal curvature-constrained planning such as the Dubins Traveling Salesman Problem (DTSP) combines NP-hard combinatorial routing with continuous optimization to determine the optimal vehicle heading angle for each target location. The problem can be addressed as combinatorial routing using a finite set of heading samples at target locations. In such a case, optimal heading samples can be determined for a sequence of targets in polynomial time, and the DTSP can be solved as searching for a sequence with the minimal cost. However, the examination of sequences can be computationally demanding for high numbers of heading samples and target locations. A fast rejection schema is proposed to quickly examine unfavorable sequences using lower bound estimation of Dubins tour cost based on windowing technique that evaluates short subtours of the sequences. Furthermore, the computation using small problem instances can benefit from reusing stored results and thus speed up the search. The reported results indicate that the computational burden is decreased about two orders of magnitude, and the proposed approach supports finding high-quality solutions of routing problems with Dubins vehicle.**

I. Introduction

In this paper, we investigate multi-goal planning to determine a cost-efficient solution on how to visit a set of target locations. The addressed problem can be considered as the robotic variant of the combinatorial optimization problem such as the Traveling Salesman Problem (TSP), where the connections between the locations have to satisfy motion constraints of the utilized robot [1]. Multi-goal planning is a part of the sequencing task [2] and also aerial surveillance planning [3]. In particular, we address the problem with curvature-constrained trajectories modeled as Dubins vehicle [4] that is known as the Dubins TSP (DTSP) [5].

The DTSP stands to determine a shortest closed path connecting a given set of n target locations such that the path is feasible for a vehicle moving with a constant forward velocity and limited minimal turning radius ρ. Although a closed-form solution is available for the optimal point-to-point Dubins path between two locations with prescribed leaving and arrival vehicle headings [4], in the DTSP, the vehicle headings are not known in advance. The DTSP is computationally challenging because of the combined combinatorial optimization of the sequencing part and continuous optimization of n variables to determine the optimal vehicle heading at each target location.

Existing DTSP solvers can be categorized into transformation approaches, decoupled approaches, and evolutionary-based direct methods. The transformation approach is based on a discretization of possible heading angles and combinatorial optimization. In contrast, decoupled approaches are focused on finding the optimal headings for a given sequence determined independently on the headings. Both optimization parts are addressed simultaneously in the direct methods that are, generally, computationally demanding. Finding optimal headings is mostly essential for dense instances with mutually close target locations, and high-quality multi-goal Dubins paths (further called Dubins tours) can be ensured by tight lower bound [6]. Hence, the transformation and decoupled approaches can be bridged by searching for the optimal sequence using an examination of candidate sequences based on high-quality Dubins tours. Solutions with the optimality gap 0.1 % are reported in [7], but computation is prohibitively demanding to examine a considerable number of sequences; and the solution does not scale with n.

Therefore, we propose to utilize a sliding window technique to estimate the cost of Dubins tour using a lower bound estimation based on the tour's subtours. Moreover, we propose to incrementally prolong the subtours to make an early rejection of a candidate sequence using a shorter, and thus faster to compute, window. Although the worst-case asymptotic complexity is not improved, the empirical results indicate a significant speedup in practice. Furthermore, the windowing technique enables to organize the computation using small subtours that can be efficiently reused during the sequence optimization with many examinations of possible node exchanges within the sequence. Subtours are part of lengthy sequences and are queried multiple times. Therefore, storing results for small problem instances and re-usage of the results significantly reduce the computational burden.

The paper is organized as follows. Related DTSP approaches are reviewed in the following section. The DTSP and its discretized variant are formally defined and introduced in Section III. The utilized combinatorial metaheuristic employed in the sequence determination is overviewed in Section IV. The proposed sequence rejection based on the sliding window technique is presented in Section V. Empirical results on proposed fast sequence rejection in the solution of the DTSP are reported in Section VI. The concluding remarks are dedicated to Section VII.

II. Related Work

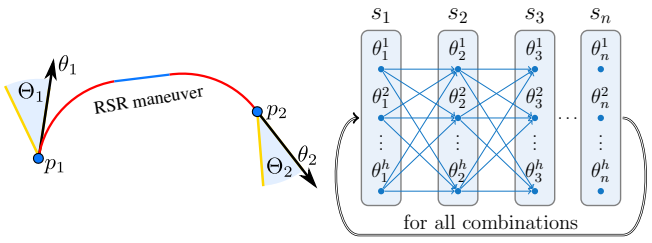
Existing approaches to the DTSP [8] can be classified into transformation methods [9], decoupled approaches [5], and direct evolutionary techniques that address both the sequencing part and headings optimization simultaneously, e.g., [10], [11]. A discrete set of heading values for each target location is sampled in the transformation methods.

The authors are with the Faculty of Electrical Engineering, Czech Technical University, 166 27 Prague, Czech Republic. E-mails: {faiglj,vanapet1,drchajan}@fel.cvut.cz

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 19-20238S.

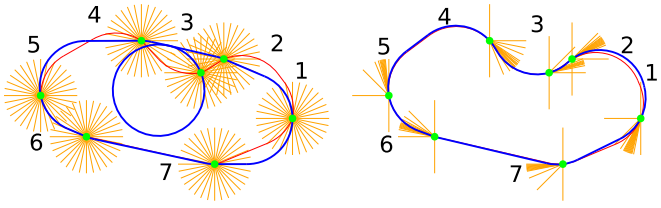
Hence, the DTSP becomes an instance of the Generalized Asymmetric TSP (GATSP) [12] that can be transformed into the regular TSP [13] using Noon-Bean transformation [14].

In decoupled approaches, the sequencing part is solved as a solution of the Euclidean TSP (ETSP), followed by optimization of the headings. For a fixed sequence, the DTSP becomes the *Dubins Touring Problem* (DTP) [7]. Simple heuristic Alternating Algorithm (AA) has been presented in [5] that has been surpassed by receding horizon approach [15], k-step look-ahead algorithm [16], and by hill-climbing heuristic called Local Iterative Optimization (LIO) [17]. The DTP can also be addressed as a discrete problem with a fixed number of heading samples similar to transformation methods [18]. Then, the optimal headings (among sampled angles) can be determined in polynomial time by a graph search, see Fig. 1b.



(a) Dubins Interval Problem (DIP) (b) Search graph (discretized DTP)

Fig. 1. Dubins Interval Problem (DIP) [6] and search graph for finding the optimal heading angles of the discretized instance of the DTP.



(a) Uniform heading sampling (b) Informed heading sampling

Fig. 2. Solution of the DTSP as the DTP with the sequence of visits found by the Euclidean TSP. The target locations are small green disks, a feasible solution is the blue curve, and heading samples are yellow. The uniform sampling uses 224 heading samples in the total (32 samples per each location) with the Dubins tour length $\mathcal{L} = 19.8$ while the informed sampling (IRIS method [7]) uses only 128 samples and provides a tour with $\mathcal{L} = 14.4$ [7]. The lower bound solution (determined using DIP) is the red curve. It is not a feasible Dubins trajectory because the arrival heading angle might not be the same as the leaving heading angle at a particular location.

Although dense heading samples improve the solution quality [19], the first systematical study of the tight lower bound of the DTSP has been proposed by Manyam et al. [6] using the introduced Dubins Interval Problem (DIP), where the leaving and arrival heading angles are constrained to intervals, see Fig. 1a. The authors provide a closed-form solution of DIP and a tight lower bound of the DTSP for a particular sequence of visits [20]. DIP is employed in the Iteratively-Refined Informed Sampling (IRIS) [7], where heading intervals are split based on the lower bound solution to improve convergence to a high-quality solution, see Fig. 2.

Direct methods simultaneously optimize the sequence and headings, and they can provide high-quality solutions at the cost of high computational requirements [3]. Having an efficient DTP solver, a direct method can be considered a generator of possible sequences that can be examined as DTP instances. Although the DTP is an integral part of solving the DTSP, it is also a part of the routing with profits and limited travel budget called the *Dubins Orienteering Problem* (DOP) [21], where a subset of locations that can be visited has to be determined together with Dubins tour to visit the subset. Therefore, an efficient solution of the DTP is an enabler for multi-goal planning with Dubins vehicle.

In [22], a combinatorial metaheuristic, the *Variable Neighborhood Search* (VNS) is employed for the combinatorial part only, and headings are determined as a solution of the DTP that shows to be less demanding than a direct VNS-based solution of the discretized problem as a whole [23]. Thus, a separate solution of the DTP might bridge transformation and decoupled approaches. Moreover, it can also improve the performance of combinatorial heuristics and evolutionary methods, which might provide better solutions, but are slower, than decoupled approaches [24].

The informed sampling-based solution of the DTP [7] is more efficient than uniform sampling; however, it is still computationally demanding. Therefore, we propose to address the sequencing part of the DTSP by a sliding window approach to quickly examine the sequence quality and thus support a quick rejection of candidate sequences with many examinations of DTP instances. The windowing technique has been utilized in [25] to generalize surrogate approximator learned for small tours to arbitrarily long sequences. In this paper, we consider windowing as a sole technique to support a quick rejection of examined sequences and also as a way how to organize computation to exploit reuse solutions of a part of the sequences (subtours) during searching for sequences in the DTSP. Although the proposed rejection procedure can be employed with any solver of the sequencing part, motivated by [23], we employed the VNS-based solver that is overviewed in Section IV.

III. Problem Statement

The proposed sequence rejection is intended to support a solution of curvature-constrained multi-goal planning problems with Dubins vehicle. It is considered in the solution of the DTSP that stands to determine a cost-efficient tour to visit a given set of n target locations $S = \{s_1, \ldots, s_n\}$, $s_i \in \mathbb{R}^2$ by Dubins vehicle [4]. Dubins vehicle is a model of curvature-constrained trajectory. The vehicle moves only forward with constant velocity v and can turn with the limited minimal turning radius ρ. The vehicle state can be described as the position $(x, y) \in \mathbb{R}^2$ and heading angle $\theta \in \mathbb{S}^1$; thus, the state $q = (x, y, \theta)$ is from the Special Euclidean group $q \in SE(2)$. The vehicle motion according to bounded control input u can be expressed as

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{u}{\rho} \end{bmatrix}, \quad |u| \leq 1. \tag{1}$$

W.l.o.g., $v = 1$ and $\rho = 1$ are considered in this paper.

A solution of the DTSP consists of the sequence of visits Σ to the targets S and corresponding heading angles $\Theta = \{\theta_i, \ldots, \theta_n\}$. The sequencing part of the DTSP is to determine the optimal permutation $\Sigma = (\sigma_1, \ldots, \sigma_n)$, $1 \leq \sigma_i \leq n$ and $\sigma_i \neq \sigma_j$ for $i \neq j$ while the continuous part is to optimize Θ. The DTSP can then be formulated as the optimization Problem 3.1 to minimize the length of the closed multi-goal Dubins path (Dubins tour) visiting S.

Problem 3.1 (DTSP):

$$\begin{aligned} &\text{minimize}_{\,\Sigma,\Theta} \\ &\quad C(\Sigma,\Theta) = \sum_{i=1}^{n} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) \\ &\quad \text{s.t.} \\ &\quad \Sigma = (\sigma_1, \ldots, \sigma_n),\ \sigma_i \in \{1, \ldots, n\}, \sigma_i \neq \sigma_j \text{ for } i \neq j, \\ &\quad \Theta = \{\theta_1, \ldots, \theta_n\}, \\ &\quad q_i = (s_i, \theta_i),\ q_i \in SE(2),\ s_i \in S, \end{aligned} \tag{2}$$

where $\mathcal{L}(q_i, q_j)$ is the length of Dubins path connecting q_i with q_j determined optimally using closed-form solution [4]. We define $\sigma_j \triangleq \sigma_{j-n}$ for $j > n$ to reflect the closed-loop Dubins tour of the DTSP.

Having a sequence of visits Σ to S, the problem of finding the shortest Dubins tour is a continuous optimization problem with n variables to determine the best heading angles Θ. The problem is called the DTP [7] to distinguish it from the DTSP, where the sequence of visits is generally not known. The DTP can be addressed by sampling possible heading angles of each target location $s_i \in S$ into k discrete samples $\Theta_i = \{\theta_i^1, \ldots, \theta_i^k\}$. Thus, it becomes the discretized DTP with the discretization $\mathbb{H}_k = \{\Theta_1, \ldots, \Theta_n\}$. The cost of Dubins tour defined by the sequence Σ can be then determined as

$$C_{\mathbb{H}_k}(\Sigma) = \min_{\theta_j \in \Theta_j,\, j \in \{1,\ldots,n\}} \sum_{i=1}^{n} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) \tag{3}$$

that can be solved optimally in $O(nk^3)$ as the shortest path in the auxiliary search graph shown in Fig. 1b.

Although the complexity of solving the discretized DTP is polynomial, it is still prohibitively demanding for dense sampling required for high-quality tours connecting mutually close locations [7]. The computational burden of solving the DTP is even more critical in cases where a vast number of candidate sequences need to be evaluated, such as for evolutionary IRIS-EA [25] and VNS-based methods [23]. We address the computational requirements by assessment of the sequence of visits to the target locations and quickly decide if a particular sequence would yield an improved Dubins tour than, e.g., the current best tour. The problem addressed in this paper is to quickly decide that a particular sequence Σ_i would be shorter Dubins tour than a sequence Σ_j.

We can further assume k fixed heading samples per each target location $s \in S$, and thus the fixed discretization $\mathbb{H}_k$. Dubins tour is defined by the sequence Σ_i to S with the corresponding headings Θ that can be determined as a solution of the DTP. Hence, we simplify the notation by denoting the cost of Dubins tour as $\mathcal{L}(\Sigma)$. Thus, the problem is to quickly determine the relation between $\mathcal{L}(\Sigma_j)$ and $\mathcal{L}(\Sigma_i)$, i.e., whether $C_{\mathbb{H}_k}(\Sigma_j) < C_{\mathbb{H}_k}(\Sigma_i)$ is true or not.

IV. VNS-BASED SOLUTION TO THE DTSP

The VNS is a combinatorial metaheuristic [26] that consists of two procedures called *shake* and *local search*. Both procedures try to improve the current best incumbent solution Σ by searching predefined Neighborhood structures $N(l_1, \ldots, l_{max})$, where l_i is the maximal distance between two solutions in the neighborhood. The *shake* procedure randomly moves Σ to different solution Σ' within the neighborhood, and it is intended to leave possible local optima. The *local search* procedure systematically searches the neighborhood of Σ' to find the best solution within the neighborhood. A systematical search can be too demanding, and therefore, the Randomized VNS [27] is used to perform a random change of Σ' for a predefined number of iterations. The VNS-based solution of the DTSP is overviewed in Alg. 1, which primarily works on sequences Σ, but each candidate sequence is examined as a solution of the DTP using the sampled headings $\mathbb{H}_k$ to determine the cost of the corresponding Dubins tours $\mathcal{L}(\Sigma)$.

Algorithm 1: VNS-based Solver for the DTSP

Input: S – Set of the target locations to be visited
Output: Σ – Found sequence of visits to locations S

$\Sigma \leftarrow$ Initial sequence found by cheapest insertion
while *terminal condition is not met* **do**
  $\Sigma' \leftarrow \text{shake}(\Sigma, l)$
  $\Sigma'' \leftarrow \text{localSearch}(\Sigma', l)$
  if $\mathcal{L}(\Sigma'') \leq \mathcal{L}(\Sigma)$ **then**
    $\Sigma \leftarrow \Sigma''$

The VNS-based solver itself can be tuned, and various Neighborhood structures can be utilized; however, we consider such tuning to be out of this paper's scope. Therefore, we consider a straightforward implementation with a single Neighborhood structure. Thus, the shake procedure uses *path move* and *path exchange* operators that exchange or move a part of the sequence Σ to create a new sequence Σ'. The *path move* operator selects a part of Σ and moves it to a randomly selected position. The *path exchange* operator exchanges two random non-overlapping parts of the sequence. Similarly, two operators are considered for *local search*, which search the neighborhood of Σ' for a local optimum in *one point move* and *open point exchange* that randomly moves one target in Σ' and exchange two randomly selected targets in Σ', respectively. The particular number of generated sequences (examinations of possible improvements) in a single *local search* call is n^2, as suggested for TSP-like routing problems.

Regarding the search process of the VNS-based algorithm, it might be observed that it is very intensive in the examination of possible sequences, including repeated queries to the same sequences. Hence, the proposed rejection procedure based on the sliding window can provide huge computational benefits in solving instances of the DTSP with tens of targets.

V. Assessment of Sequences for Comparison of Dubins Tours based on Windowing Technique

Two principled ways on how to decrease the computational burden of computing $\mathcal{L}(\Sigma)$ can be considered for a solution of the discretized DTP. The first is reducing the heading sample count, which is, however, not desirable because dense sampling might be needed for a high-quality solution. The second way is to relax the requirement on the closed Dubins tour and compute the so-called open DTP, which reduces $O(nk^3)$ to $O(nk^2)$ at the cost of the approximated tour cost, which would be a lower bound of the closed Dubins tour.

In addition, we can further think about the sequence generation in a combinatorial search, where an improvement of the sequence can be based on the local exchange of short subsequence. In contrast, other parts of the sequence would remain the same. Besides, a comparison can be mostly expected between some best sequence found so far and a candidate sequence being its slight modification. Such a behavior has been empirically observed for combinatorial metaheuristics and evolutionary algorithms, where many parts of the sequences remain identical. Hence, we aim to find a computational schema allowing us to reuse already examined subsequences and thus reduce computational requirements.

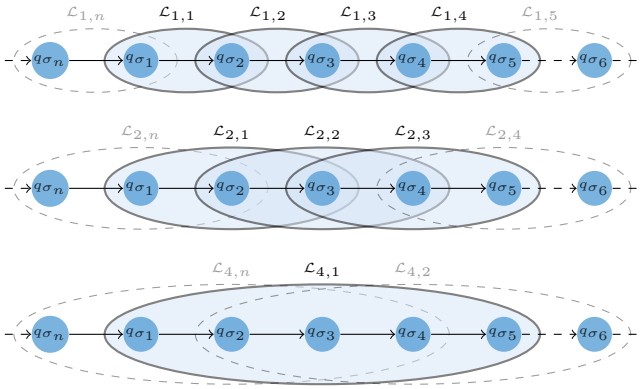


Fig. 3. Example of the sliding window for estimating the cost of Dubins tour using subtours with the window of the size $w \in \{1, 2, 4\}$.

Motivated by repeatable computations of subtours among several Dubins tours during the examination of possible sequences of visits to S, we propose to evaluate a cost of Dubins tour using a sliding window of the limited size w, see Fig. 3. The cost of Dubins tour for a sequence Σ is estimated using costs of subtours having w links (connecting $w+1$ targets) for each target location in the sequence. The individual cost from the i-th location and window size w can be expressed as

$$\mathcal{L}_{(w,i)}(\Sigma, \mathbb{H}_k) = \min_{\mathbb{H}_k} \sum_{j=i}^{i+w-1} \mathcal{L}(q_{\sigma_j}, q_{\sigma_{j+1}}), \tag{4}$$

where we consider $\sigma_j \triangleq \sigma_{j-n}$ for $j > n$ to partially reflect the closed-loop Dubins tour of the DTSP and also to have the same window size for each location. The cost of Dubins path connecting locations $q_{\sigma_i}, \dots q_{\sigma_{i+w-1}}$ is computed using the sampling schema $\mathbb{H}_k$ such that the individual heading angles are selected from the samples $\theta_i \in \Theta_i$. Since the subsequence is $(w+1)$ targets long, the cost of opened Dubins subtour defined by the window w can be computed in $O(wk^2)$.

The cost of the whole sequence Σ is then estimated as an aggregated cost of the individual subtours

$$\mathcal{L}_w(\Sigma, \mathbb{H}_k) = \frac{1}{w} \sum_{i=1}^{n} \mathcal{L}_{(w,i)}(\Sigma, \mathbb{H}_k). \tag{5}$$

For the given discretized schema $\mathbb{H}_k$, the aggregated cost is a lower bound of the discretized DTP that can be solved optimally in $O(nk^3)$ using graph search (see Fig. 1b). It is the lower bound because of the relaxed closed path, and the individual subtours for the limited sized window w can take advantage of freely selected arbitrary heading angles for the subtour endpoints. The minimal cost of the subtour is always lower or equal to the corresponding part of the final closed tour among all targets and thus $\mathcal{L}_w(\Sigma, \mathbb{H}_k) \leq C_{\mathbb{H}_k}(\Sigma)$.

The overlapping windows allow us to consider a single-window size regardless of the current number of targets in the sequence. It also allows us to store the result for a particular subtour and reuse it for a repeated query on the same subtour. In particular, we can implement result storage as a hash map, where the key is defined by the labels of the target locations in the sequence, and the cost is the length of the Dubins subtour.

Furthermore, we can exploit the symmetry of the subtour because of free endpoints, and thus the key always starts with the label of lower value. Therefore, we revert the subtour sequence labels if it starts with the target location of a higher label value than it ends. Based on empirical observations, the overlapping windows have a negligible impact on real computational requirements because of the great benefits of storing and reusing results for small subtours.

The computational complexity to evaluate (5) can be bounded by $O(nwk^2)$ that immediately indicates that the possible theoretical speedup can be achieved for $w \ll k$. Thus, a quick rejection of the sequence Σ'' in comparison to sequence Σ' can be based on an incremental increase of the window size w_j and a comparison of the estimated cost of Σ'' using w_j with Σ' and its lower bound estimation computed for the window size w_i.

Moreover, during the solution of the sequencing part of the DTSP, we can further assume there is a current best sequence, Σ^* found so far. Hence, it does not make sense to use a long window size once the lower bound estimation of Σ' (using short window w_i) is above the cost of Dubins tour Σ^* computed by the full solution of the closed DTP, $C_{\mathbb{H}_k}(\Sigma^*)$. Thus, for $\mathcal{L}_{w_i}(\Sigma', \mathbb{H}_k) > C_{\mathbb{H}_k}(\Sigma^*)$, we can consider Σ'' be more promising than Σ' if its cost estimation $\mathcal{L}_{w_j}(\Sigma'', \mathbb{H}_k)$ does not exceed $\mathcal{L}_{w_i}(\Sigma', \mathbb{H}_k)$ up to the window size w_i, i.e., $w_j \leq w_i$.

On the other hand, if w_i reaches the maximum window size w_m and Σ'' is still cheaper for w_m than Σ' using lower bound estimates, we cannot be sure which sequence is supposed to be better. Therefore, we can ensure selection

between Σ'' and Σ' by comparing the optimal solutions $C_{\mathbb{H}_k}(\Sigma'')$ and $C_{\mathbb{H}_k}(\Sigma')$, respectively, which can be however, computationally demanding. Hence, we can approximate the comparison of Dubins tours, and consider Σ'' is better than Σ', because once the candidate sequence is selected as Σ^*, its cost is determined by the full DTP as $C_{\mathbb{H}_k}(\Sigma^*)$. Thus, in randomized searching such as in the VNS, the sequence Σ' might be eventually determined multiple times. Based on the early evaluation, we found the approximation improves computational performance without significantly impacting the solution quality. Therefore, the approximate comparison is considered for the reported results, but full examination is listed in the proposed procedure for completeness.

The proposed rejection procedure with a sequence of window sizes $W = \{w_1, \ldots, w_m\}$, such that $w_1 < \ldots < w_m$, is summarized in Algorithm 2. The determination of w_i at Lines 2–5 can be saved between the function calls by explicit usage of the variable w_i. The part of the algorithm is listed to make the description self-contained. Similarly, the test at Line 7 can be omitted within the context of the employed VNS-based solution of the DTSP. It is because it examines randomly generated sequences, and promising sequences would be selected as Σ^* for which the cost of the optimal DTP solution is always determined.

Regarding the computational complexity, depending on the value of w_m, the proposed comparison of sequences using tour cost estimation can be, in the worst case, even more demanding than the optimal solution of the DTP using discretization $\mathbb{H}_k$. The main expected computational benefits are not in a single comparison of two sequences but in quick rejections of many candidate sequences examined in combinatorial heuristics or evolutionary algorithms. It is expected that a vast majority of comparison would be resolved using $w = 1$ with the theoretical complexity $O(nk^2)$.

Moreover, the query results for short subtours can be stored (e.g., in a hash map data structure) and reuse that can practically further reduce the computational burden up to $O(n)$ with the average access to the hash map with $O(1)$. Such storing and reusing would not be effective for a sequence as a whole, as it is unlikely the case to examine the same sequence multiple times. Small subsequences enable to reuse queries as most of the sequence will remain the same while generating new candidate sequences using a local exchange of the targets in the sequence. The practical impact of the proposed windowing-based rejection to the real computational time and solution quality has been empirically evaluated in the solution of the DTSP, and the achieved results are reported in the following section.

Algorithm 2: Examination of the candidate sequence

Input: Σ'' – The candidate sequence under examination, Σ' – The sequence to which Σ'' is compared to, Σ^* – The current best sequence found so far.

Parameters: m – the number of window sizes $W = \{w_1, \ldots, w_m\}$ with $w_i = 2^{i-1}$ for $i \in \{1, \ldots m\}$; Sampling schema $\mathbb{H}_k$ used in the sequence cost evaluation.

Output: `true` if Σ'' is rejected; otherwise `false`.

```
1  Function Reject(Σ'', Σ', Σ*):
2      i ← 1
3      for i to m by 1 do
4          if L_{w_i}(Σ', H_k) > C_{H_k}(Σ*) then
5              break    // Do not use a large w_i
                        for not promising sequence
6      for j ← 1 to i by 1 do
7          if EXACT_COMPARISON and j == m then
8              return C_{H_k}(Σ'') > C_{H_k}(Σ')  // Decide
                      based on true costs
9          if L_{w_j}(Σ'', H_k) > L_{w_i}(Σ', H_k) then
10             return true  // Σ'' is unlikely to
                        be cheaper than Σ'
11     return false     // Σ'' might be shorter
                    than Σ'
```

VI. Empirical Evaluation

The proposed sequence rejection is motivated to quickly assess Dubins tours determined as a permutation of the target locations S. The rejection is intended to be employed in the improvement search for a new best sequence in combinatorial meta-heuristics such as VNS-based or evolutionary algorithms. Since the rejection will unlikely to provide the desired real-time speedup as a single procedure, we deploy the rejection within the VNS-based solver to the DTSP described in Section IV. Besides, we further demonstrate the benefit of the windowing technique with surrogate approximator [25], learned for four-point Dubins tours ($w = 3$). For all the presented results using the proposed sequence rejection, `EXACT_COMPARISON` is set to `false`, and thus only approximated sequence costs are used in Alg 2.

The computational benefits of the rejection have been studied for instances of the DTSP with relatively dense target locations that require dense sampling of the headings and also an explicit solution of the sequencing part because the ETSP would provide a weak solution in decoupled approaches [7]. The examined instances are defined by the number of target locations n and density d for which the target locations are drawn from the uniform distribution to fill a squared area with the side length $\sqrt{n/d}$. The used evaluation instances are therefore created for $n = 50$ and $d = 0.1$. The number of heading samples for each target location is $k = 16$, which is uniformly sampled. Based on the early evaluation, six sizes ($m = 6$) of the sliding window $W = \{1, 2, 4, 8, 16, 32\}$ are considered in the rejection procedure to provide a suitable trade-off between the solution quality and computational requirements.

The proposed rejection procedure (Alg. 2) has been employed in the VNS-based solver for the DTSP (Alg. 1) both prototyped in Julia 1.3.1 [28], and the computational times have been obtained using a single core of the Intel Xeon Gold 6146 running at 3.2 GHz. The solution quality

TABLE I

COMPUTATIONAL REQUIREMENTS OF EXAMINING $N_s = 100\,000$ SEQUENCES DURING VNS-BASED SOLUTION OF THE DTSP INSTANCE WITH $n = 50$, $k = 16$, $d = 0.1$

DTP	**VNS$_{\text{base}}$**		
	# calls	T [s]	T$_a$ [μs]
Full	100k	659	6 590
Average	-	**659**	**6 590**

DTP	**VNS$_{\text{win}'}$**			**VNS$_{\text{win}}$**		
	# calls	T [s]	T$_a$ [μs]	# calls	T [s]	T$_a$ [μs]
$w = 1$	100k	65.2	651	100k	0.3	3
$w = 2$	13.1k	15.5	1 180	13.4k	0.5	36
$w = 4$	4.39k	9.9	2 250	4.33k	0.5	105
$w = 8$	998	4.5	4 510	1.12k	0.5	411
$w = 16$	352	3.1	8 930	410	0.4	844
$w = 32$	167	2.8	17 000	234	0.4	1 700
Average	-	**101**	**1 010**	-	**2.6**	**26**

is reported as the normalized cost of the particular DTSP instance found as the DTP with the resolution of the heading samples 1024 for the sequence found as the optimal solution of the corresponding ETSP provided by Concorde [29]. The instances and reference values are adopted from [25].

A. Sequence Rejection in the Solution of the DTSP

Computational benefits of the proposed rejection have been studied for the VNS-based DTSP algorithm considered in three variants denoted as VNS$_{\text{base}}$, VNS$_{\text{win}'}$, VNS$_{\text{win}}$ algorithms. VNS$_{\text{base}}$ uses the full solution of the DTP [7], the windowing technique is employed in VNS$_{\text{win}'}$ without the storing and reusing subtours results, while VNS$_{\text{win}}$ represents the enhanced implementation with reusing query results via the hash map. The number of calls and the corresponding computational times per particular window size is depicted in Table I for $N_s = 10^5$ queries on different sequences of the VNS-based solution of the DTSP, where T and T$_a$ denote the total and average computational time per query, respectively. The evolution of the solution quality with the number of examined sequences and computational time is reported in Fig. 4, where quicker comparison enables more examinations of sequences and thus a better solution.

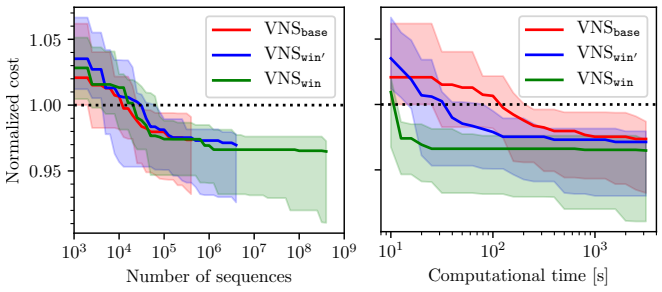


Fig. 4. Solution quality evolution with the number of examined sequences (left) and real computational times (right) reported as median values among ten instances of the discretized DTSP with $n = 50$, $k = 16$, $d = 0.1$ accompanied with 80 % non-parametric confidence intervals.

The results indicate that the solution quality is similar for all algorithm variants, and thus the proposed heuristic estimation of the Dubins tour cost is sufficient. VNS$_{\text{base}}$ is computationally demanding (see Table I), and the proposed rejection procedure decreases the computational burden, albeit T_a for $w = 16$ and $w = 32$ is more demanding than the optimal solution of the full DTP in VNS$_{\text{base}}$. It is, however, the expected behavior as the theoretical speedup holds only for $w \ll k$. On the other hand, the number of queries for $w = 16$ and $w = 32$ is significantly lower than that for small window sizes, and the overall computational requirements of VNS$_{\text{win}'}$ are about 6.5 times lower than for VNS$_{\text{base}}$.

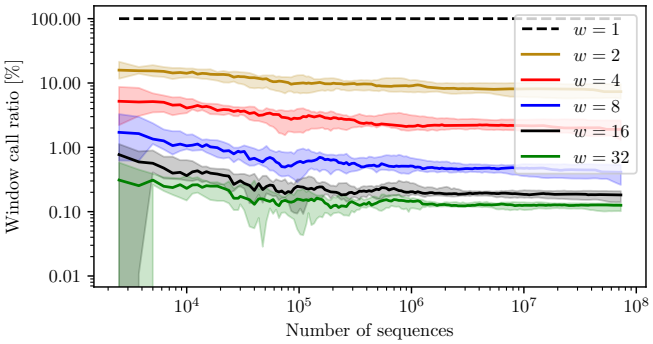


Fig. 5. Evolution of the number of sequence examinations using the window size w relative to the total number of queries on sequence cost. Notice, both axes are in the logarithmic scale.

The evolution of how many times the estimation of the sequence cost is called for the individual window size is depicted in Fig. 5 as the relative ratio to the number of queried sequences. Each sequence is examined with $w = 1$ but less than 20 % of sequences are examined using $w = 2$. The number of examinations quickly decreases with each additional enlargement of the window size, and only about 0.1 % of queried sequences are examined for $w = 32$.

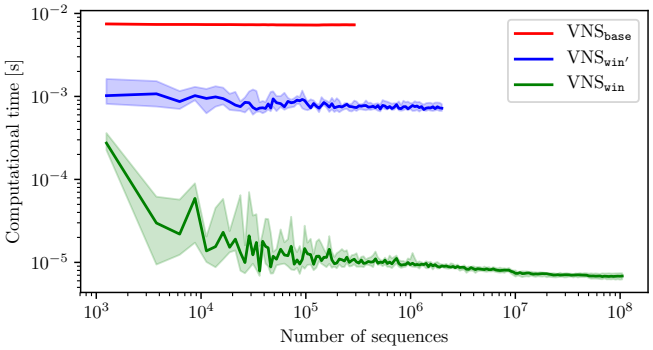


Fig. 6. Evolution of the mean required computational time to examine a single sequence query during VNS-based solution of ten instances of the DTSP instance with $n = 10$, $k = 16$, and $d = 0.1$. The VNS-based solver is terminated after 3600 s, and therefore, much more sequences are examined by the proposed VNS$_{\text{win}}$.

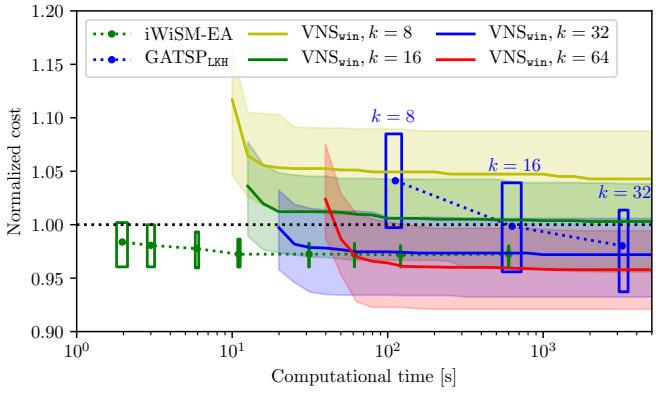
The computational benefit of the windowing technique is fully seen for VNS$_{\text{win}}$ with reusing of individual subtours queries, and the overall achieved speedup is about two orders of magnitude; see also Table I. The hash map is being filled with the increasing number of examined sequences, and thus

it is utilized more and more often as it can be seen in Fig. 6.

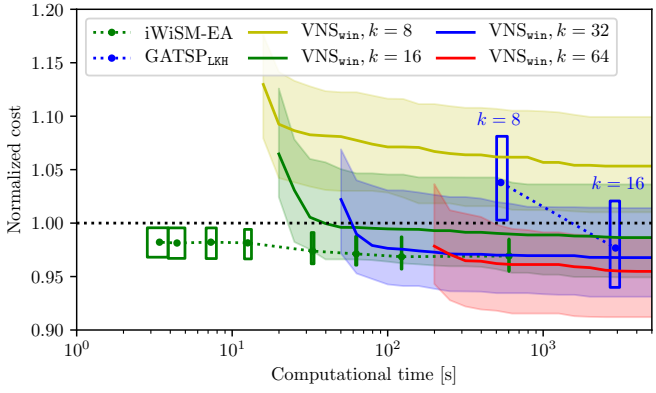
The impact of the windowing technique on the solution quality of the DTSP has been further studied, and the performance of the proposed VNS_{win} algorithm has been compared to the existing algorithms. The results are reported in the following section.

B. Solution of the Dense Instances of the DTSP

The proposed sequence rejection employed in the VNS-based solution of the DTSP has been compared with two state-of-the-art DTSP solvers[1]. In this evaluation, we exploit the benefit of the windowing technique to learn surrogate approximator using short Dubins tours employed in an evolutionary algorithm denoted as iWiSM-EA [25]. Besides, the proposed VNS_{win} algorithm is also compared with a regular sampling-based approach [13] with $k = 16$ that is transformed into an instance of the GATSP further transformed to the Asymmetric TSP solved heuristically by LKH [30] because the optimal solution would be too computationally demanding. The sampling-based transformation approach is denoted $\text{GATSP}_{\text{LKH}}$.



(a) $n = 50$

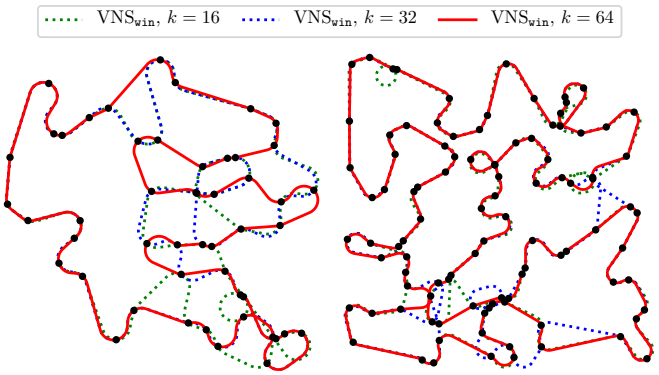


(b) $n = 100$

Fig. 7. Evolution of the median solution quality with 80 % non-parametric confidence interval computed from solutions of ten trials for each of ten random instances with n target locations, $k = 16$, and $d = 0.1$.

The surrogate approximator is learned by high-quality solutions (with up to $k = 1024$ heading samples) of the DTP instances with $w = 3$ found by the informed sampling algorithm IRIS [7]. The approximator is a multi-layer perceptron [31] with three hidden layers, each with 256 rectified linear units, trained by Adam [32]. The learned approximator is employed in a generic evolutionary algorithm for the TSP with simple inversion mutation [33] and OX1 crossover [34]. The initial population is created using a solution of the ETSP found by Concorde [29], see [25] for further details.

[1]See [25] for a comparison of the iWiSM-EA and $\text{GATSP}_{\text{LKH}}$ with more existing approaches.



(a) $n = 50$, $\mathcal{L}_{16} = 141.2$, $\mathcal{L}_{32} = 137.8$, $\mathcal{L}_{64} = 135.0$

(b) $n = 100$, $\mathcal{L}_{16} = 300.2$, $\mathcal{L}_{32} = 292.0$, $\mathcal{L}_{64} = 285.2$

Fig. 8. Example of found solutions for $n = 50$ (left) and $n = 100$ (right) with $k \in \{16, 32, 64\}$ after one hour of computation. The path length is reported as $\mathcal{L}_k$ where the subscript denotes the number of heading samples per each target.

The solution quality is measured as the relative tour length to the reference solution of the DTP with 1024 samples for the sequence found as the ETSP [25] and the evolution of the normalized cost is depicted in Fig. 7. For VNS_{win}, motivated by low computational requirements, we further increase the number of heading samples up to $k = 64$ and we report performance of VNS_{win} for $k \in \{8, 16, 32, 64\}$. Selected found solutions are depicted in Fig. 8.

The reported results indicate that the windowing technique positively influences computational requirements and supports finding better solutions in lower computational time. The iWiSM-EA algorithm is the fastest among the evaluation methods because it takes the benefits of learned surrogate approximator. Although the reported times are for a single core deployment, iWiSM-EA can further exploit multi-core (or massively parallel graphics card) computations. However, the approximator learned for $w = 3$ is not sufficient to improve the solution that stucks in a local extreme. On the other hand, the VNS-based algorithm enabled by the proposed rejection schema can take advantage of many heading samples $k = 64$, which allows improving the solution quality. Finally, the transformation method $\text{GATSP}_{\text{LKH}}$ can provide similar solutions to the iWiSM-EA and VNS_{win} with $w_m = 32$, but it is significantly more demanding. Furthermore, both iWiSM-EA and VNS_{win} provide the first solutions very quickly, while $\text{GATSP}_{\text{LKH}}$ needs to solve the problem as a whole.

Note that iWiSM-EA does not need to compute Dubins path between all target locations for all sampled headings, and it only needs to find the initial sequence by the solution of the ETSP. That is why plots for VNS_{win} and $\text{GATSP}_{\text{LKH}}$

start from about ten seconds in Fig. 7. Hence, a further combination of the iWiSM-EA based initialization of VNS_{win} with the lazy evaluation of Dubins paths between the target locations can provide additional computational benefits.

VII. Conclusion

In this paper, we present an estimation of the Dubins tour assessment in the solution of the DTSP based on the windowing technique utilized in the quick rejection of possible candidate sequences of visits to the target locations. The windowing technique improves real computation requirements by rejecting sequences based on small sizes of the windows. However, it mostly supports organizations of the computation using short Dubins subtours that can be stored and reused. Although theoretical complexity is not improved, the real computational requirements are about two orders of magnitude lower. Thus, the proposed fast sequence rejection enables to employ a relatively demanding VNS-based metaheuristic to the solution of the DTSP. The windowing-enabled algorithms provide better solutions than the transformation method, which thus seems to be surpassed by the proposed and recent iWiSM-EA algorithms in both criteria, the computational requirements, and the solution quality. Therefore, we can conclude that the proposed windowing-based sequence rejection for routing with Dubins vehicle can improve the scalability of algorithms for large instances of curvature-constrained multi-goal planning problems.

Acknowledgments

The authors acknowledge the access to the computational grid provided under OP VVV MEYS RCI project CZ.02.1.01/0.0/0.0/16_019/0000765.

References

[1] S. Alatartsev, S. Stellmacher, and F. Ortmeier, "Robotic Task Sequencing Problem: A Survey," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 2, pp. 279–298, 2015.

[2] C. Wurll and D. Henrich, "Point-to-point and multi-goal path planning for industrial robots," *Journal of Robotic Systems*, vol. 18, no. 8, pp. 445–461, 2001.

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

[4] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, pp. 497–516, 1957.

[5] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and traveling salesperson problems for Dubins' vehicle," in *American Control Conference*. IEEE, 2005, pp. 786–791.

[6] S. G. Manyam, S. Rathinam, D. Casbeer, and E. Garcia, "Tightly Bounding the Shortest Dubins Paths Through a Sequence of Points," *Journal of Intelligent & Robotic Systems*, pp. 1–17, 2017.

[7] J. Faigl, P. Váňa, M. Saska, T. Báča, and V. Spurný, "On solution of the dubins touring problem," in *European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–6.

[8] D. G. Macharet and M. F. M. Campos, "A survey on routing problems and robotic systems," *Robotica*, vol. 36, no. 12, pp. 1781–1803, 2018.

[9] P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 4, pp. 70–77, 2010.

[10] D. Guimaraes Macharet, A. Alves Neto, V. Fiuza da Camara Neto, and M. Montenegro Campos, "An evolutionary approach for the Dubins' traveling salesman problem with neighborhoods," in *Annual Conference on Genetic and Evolutionary Computation*. ACM, 2012, pp. 377–384.

[11] X. Zhang, J. Chen, B. Xin, and Z. Peng, "A memetic algorithm for path planning of curvature-constrained UAVs performing surveillance of multiple ground targets," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 622–633, 2014.

[12] J. Le Ny, E. Feron, and E. Frazzoli, "On the Dubins Traveling Salesman Problem," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 265–270, 2012.

[13] K. J. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-based path planning for a visual reconnaissance unmanned air vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 619–631, 2012.

[14] C. Noon and J. Bean, "An Efficient Transformation of the Generalized Traveling Salesman Problem," Department of Industrial and Operations Engineering, University of Michigan, Tech. Rep. 91–26, 1991.

[15] X. Ma and D. A. Castanon, "Receding horizon planning for Dubins traveling salesman problems," in *45th IEEE Conference on Decision and Control*, 2006, pp. 5453–5458.

[16] P. Isaiah and T. Shima, "Motion planning algorithms for the Dubins Travelling Salesperson Problem," *Automatica*, vol. 53, pp. 247–255, 2015.

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

[18] I. Cohen, C. Epstein, P. Isaiah, S. Kuzi, and T. Shima, "Discretization-Based and Look-Ahead Algorithms for the Dubins Traveling Salesperson Problem," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 383–390, 2017.

[19] L. Babel, "New heuristic algorithms for the Dubins traveling salesman problem," *Journal of Heuristics*, 2020.

[20] S. G. Manyam and S. Rathinam, "On tightly bounding the dubins traveling salesman's optimum," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 7, p. 071013, 2018.

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

[22] R. Pěnička, J. Faigl, and M. Saska, "Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants," *European Journal of Operational Research*, vol. 276, no. 3, pp. 816–825, 2019.

[23] R. Pěnička, J. Faigl, M. Saska, and P. Váňa, "Data collection planning with non-zero sensing distance for a budget and curvature constrained unmanned aerial vehicle," *Autonomous Robots*, vol. 43, no. 8, pp. 1937–1956, 2019.

[24] R. J. Kenefic, "Finding good dubins tours for UAVs using particle swarm optimization," *Journal of Aerospace Computing, Information, and Communication*, vol. 5, no. 2, pp. 47–56, 2008.

[25] J. Drchal, J. Faigl, and P. Váňa, "WiSM: Windowing Surrogate Model for evaluation of curvature-constrained tours with Dubins vehicle," *IEEE Transactions on Cybernetics*, 2020. [Online]. Available: https://doi.org/10.1109/TCYB.2020.3000465

[26] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.

[27] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.

[28] J. Bezanzon, S. Karpinski, V. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," *CoRR*, vol. abs/1209.5145, 2012.

[29] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, "Concorde TSP Solver," 2003, [cited 26 Jan 2020]. [Online]. Available: http://www.tsp.gatech.edu/concorde.html

[30] K. Helsgaun, "LKH solver 2.0.9," 2018, [cited 26 Jan 2020]. [Online]. Available: http://www.akira.ruc.dk/~keld/research/LKH/

[31] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[33] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, 1992.

[34] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999.