Multi-tour Set Traveling Salesman Problem in Planning Power Transmission Line Inspection

František Nekovář [ID], Jan Faigl [ID], Martin Saska [ID]

Abstract—This paper concerns optimal power transmission line inspection formulated as a proposed generalization of the traveling salesman problem for a multi-route one-depot scenario. The problem is formulated for an inspection vehicle with a limited travel budget. Therefore, the solution can be composed of multiple runs to provide full coverage of the given power lines. Besides, the solution indicates how many vehicles can perform the inspection in a single run. The optimal solution of the problem is solved by the proposed Integer Linear Programming (ILP) formulation, which is, however, very computationally demanding. Therefore, the computational requirements are addressed by the combinatorial metaheuristic. The employed greedy randomized adaptive search procedure is significantly less demanding while providing competitive solutions and scales better with the problem size than the ILP-based approach. The proposed formulation and algorithms are demonstrated in a real-world scenario to inspect power line segments at the electrical substation.

Index Terms—Planning, Scheduling and Coordination

Figure 1. UAV performing visual inspection on a power transmission line.

I. INTRODUCTION

IN this paper, we address the inspection planning of the *Power Transmission Line* (PTL) formulated as a proposed generalization of the well-known combinatorial *Traveling Salesman Problem* (TSP). The presented method is motivated by the inspection of transmission infrastructure using *Unmanned Aerial Vehicles* (UAVs). It is becoming a standard approach advertised as a cost-saving solution compared to either manned inspections or those done using piloted helicopters, such as seen in Fig. 1. Although the development of autonomous inspection is currently in progress, the necessity of having a prepared plan for the efficient utilization of single or multiple vehicles to achieve cost-efficient inspection is desirable for human-piloted UAVs, as well. The problem addressed in this paper is a major aspect of the objectives of the EU-funded AERIAL CORE project.

The transmission line inspection is performed during vehicle flights close to the power lines. It is advantageous to perform such an inspection flight between segments of the power lines defined by the pylons [1]. Hence, the problem is defined as coverage to visit the transmission line segments, where

each segment is a target to be visited. In the inspection, the segments are organized into a sequence of segments; thus, we formulate the sequencing problem as a variant of the TSP.

A solution of the TSP is a sequence of visits to the given targets that is further generalized to targets grouped into sets, where only a single visit to some target in a set is required, but all sets have to be visited. Furthermore, within the context of a small UAV with limited operational time, it might not be possible to inspect all the sets in a single run. Therefore, we generalize the standard TSP formulation to consider limited maximal flight time C_{max}, to consider multiple vehicles to cover all segments or a single vehicle with multiple tours, and with the start and end of the respective tours at the common depot. Such a generalized formulation of the TSP is proposed to describe the herein addressed PTL inspection planning problem, where a UAV has to visit each segment of the PTL exactly once, but in an arbitrary direction. The travel cost to the particular start of the segment inspection is also relevant as it is part of the travel budget C_{max}. Note, the difference between using a single UAV in a multi-tour scenario and multiple UAVs simultaneously is manifested in the problem cost function. Since we aim to minimize the total inspection time with the one-vehicle multiple-tours case, minimizing the min-sum of the tour costs is more relevant rather than minimizing the maximum cost from all the tour costs.

The proposed TSP-based formulation for PTL inspection is addressed by *Integer Linear Programming* (ILP) to find an optimal solution. However, it is computationally demanding and does not scale with the problem size. Therefore, we propose
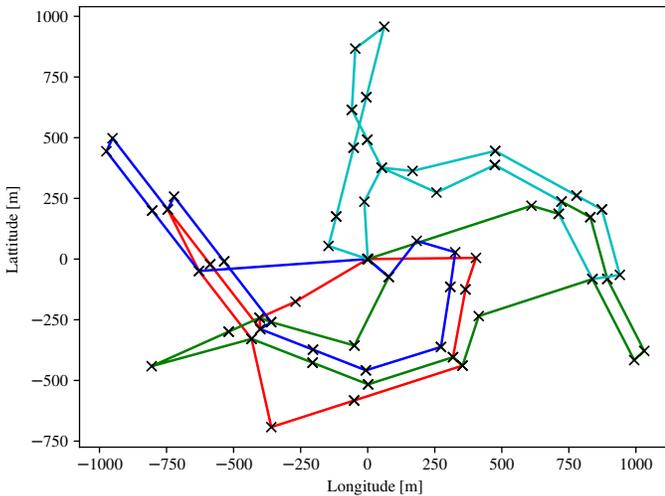
Figure 2. An example of the proposed GRASP-based solution with four UAV routes for inspection of power transmissions lines around Nechranice power line substation, located at 50°20'44.7"N 13°19'30.9"W.

a heuristic solution based on the combinatorial metaheuristic *Greedy Randomized Adaptive Search Procedures* (GRASP) to provide a solution in a reasonable computational time, as it is possible that targets could change and a solution would need to be computed on-site. Based on the performed empirical evaluation, the proposed heuristic solver provides adequate solutions with significantly lower computational requirements than the optimal ILP. An example of the solution found is depicted in Fig. 2. Our contribution is the formulation of the PTL inspection as the ILP problem, solvable using an off-the-shelf solver and introduction of the computationally less demanding GRASP-based solver providing competitive solutions.

The remainder of the paper is organized as follows. An overview of related work on power transmission line inspection and existing related formulations of the TSP is provided in the following section. The addressed problem is defined in Section III. The proposed ILP-based formulation is introduced in Section IV and the proposed heuristic GRASP-based solver in Section V. Results on optimal and heuristic solutions using a real-word test scenario are reported in Section VI. Concluding remarks are detailed in Section VII.

II. RELATED WORK

The use of robotic vehicles in the PTL inspection is widespread, with possible solutions reviewed in surveys [2], [3]. Multi-rotor UAVs in automated inspection utilizing path planning are evaluated in [4], where it is concluded that a high maneuvering precision makes them suitable for efficient vision-based inspection. Various other used, but not herein discussed robots include line-hanging vehicle [5], fixed-wing UAV [6], Vertical Take-Off and Landing fixed-wing UAV [7], or UAV with wire landing capabilities [8]. Vision-based tracking of power-lines is evaluated in [1] and [9] reports on experimentally verified real-time automated tracking of power lines using a multi-rotor UAV.

The inspection planning can be formulated as a variant of the TSP [10], e.g., applied in pylon components inspection using a Genetic Algorithm (GA) [11]. The TSP-based formulation with a GA solution is proposed in [12] to address the PTL inspection with the maximum flight distance limit. The authors of [13] propose the TSP-based Two-Layer Point-Arc Routing problem for the coordinated inspection using ground and aerial vehicles solved by "Cluster First, Route Second" and "Route First, Split Second" based heuristics.

The Set TSP, also called the Generalized TSP, has been addressed by heuristics [14] and ILP [15]. The Cost constrained TSP [16] is a suitable formulation for tasks where each target has associated value, and the problem is to visit targets with the maximal sum of the values within constrained tour cost, which is also known as the Orienteering Problem in the literature [17]. A further generalization is addressed in [18]; however, these approaches are for single tour problems.

Multi-route inspection can be formulated as a variant of the Multiple Traveling Salesmen Problem (MTSP) [19] that can be solved using the ILP [20], which is known to be very computationally demanding, however. Alternatively, less demanding GRASP heuristics have been employed in [21] for a problem similar to the herein addressed PTL problem. The problem is referred to as the multi-product multi-vehicle inventory routing problem, which is a generalization of the inventory routing problem [22]. Heuristics and integer programming formulations of the MTSP for UAV planning already exist, such as [23] for UAV-based delivery in combination with trucks. Besides, soft computing techniques have been used for the MTSP, such as GA [24], simulated annealing [25], and neural networks [26], [27] to name a few. However, the proposed Multi-tour Set TSP formulation differs from the existing variants of the MTSP in the set generalization and travel budget. Therefore, a novel solution is required, and the reader is referred to [28] for an overview of methods for the existing variants of the MTSP.

Based on the literature review and the best of the authors' knowledge, the problem of Multi-tour PTL wire inspection with a limited travel budget has not been addressed by the existing approaches. Therefore, we propose to formulate the problem using the ILP to find an optimal solution. Furthermore, we address the computational challenges of the studied PTL inspection planning by the proposed heuristic inspired by the successful deployment of the GRASP-based approach to solving a similar problem in [21]. GRASP is fine-tuned with the guidance of work discussing the specifics of tabu-search [29], which is a part of the proposed novel meta-heuristic algorithm.

III. PROBLEM STATEMENT

In the addressed PTL inspection, the goal is to traverse all given PTL segments in the shortest time possible. The segments are defined as the power lines between two pylons, where each must be traversed in a single run to complete its inspection. However, the direction in which the segment is traversed is arbitrary. The visit to each segment can be formulated as a vertex visit in a set of possible traverse

directions, where every set should be visited exactly once. It implies the problem can be formulated as a variant of the Generalized TSP [30], also called the Set TSP [31]. However, the formulation needs to include the practical limits of UAVs.
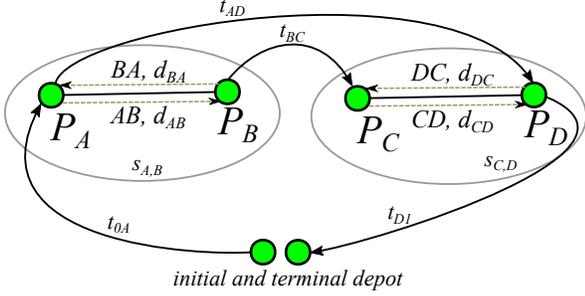
(a) Power-line segments between the pylons
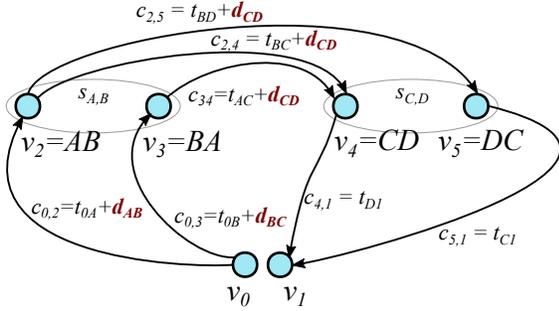
(b) Graph representation of the problem

Figure 3. Representation of the PTL inspection planning problem, where the pylons are denoted by alphabet letters for improved readability. (**a**) Each power line segment can be inspected in two possible directions, with the inspection speed being lower than overflight speed from one segment endpoint to another. (**b**) The vertices represent the power line segment visits in a specific direction; therefore, the edge cost is a sum of the travel cost from one segment to another and the cost of the target segment inspection.

First, the inspection plan must satisfy the UAV's limited flight time. Additional constraints are the maximal flight speed and acceleration addressed in the estimation of flight times using a trajectory planner, making the planned routes physically feasible. Multiple inspection tours are allowed as long as they begin and end at the same specific depot. Based on the problem description, the notation used in the formal problem definition is as follows.

We model the problem domain as a weighted oriented graph $G = (V, E, c)$ with a set of vertices V representing visits to the PTL segments and a set of edges E. A segment can be visited in an arbitrary direction, and therefore, the visits to the segments are partitioned into n_s sets corresponding to n_s physical segments. A segment $s_{A,B} = \{v_{AB}, v_{BA}\}$ corresponds to two directions of visits to the segment between the pylons P_A and P_B, see Fig. 3a. Each inspection tour is requested to start and terminate at the defined locations, and therefore $v_0 \in V$ and $v_1 \in V$ correspond to starting and termination depot, respectively. An example valid route in Fig. 3a is $v_0 \rightarrow P_A \rightarrow P_B \rightarrow P_C \rightarrow P_D \rightarrow v_1$ in, which in the graph formulation Fig. 3b translates to $v_0 \rightarrow v_2 \rightarrow v_4 \rightarrow v_1$.

The graph G contains $n = 2 + 2n_s$ vertices $V = \{v_0, \ldots v_{2n_s}, v_{2n_s+1}\}$ that are grouped into $2 + n_s$ sets, where the first two sets contain vertices corresponding to the initial and termination locations. Hence, the vertices can be referred to as integer values, and the edges can be referred to using integer indices of V, e.g., $e(v_2, v_5) = e_{2,5}$. Furthermore, the first vertex index corresponding to the visit of a power line segment starts at 2, and two directions of the possible segment visits are vertices v_{2i} and v_{2i+1}. Thus, $V = \{v_0, v_1\} \cup \bigcup_{i=1}^{n_s} S_i$, where the set $S_i = \{v_{2i}, v_{2i+1}\}$ represents two possible visits of the i-th segment.

The particular flight times are determined by estimating flight trajectories connecting the segment endpoints and trajectories along with the segments. The trajectories are subject to lateral flight dynamics constraints. The constraints include the maximum flight velocity $v_{\max}$, maximum inspection velocity v_{insp}, and acceleration $a_{\max}$. The time of flight between the segment endpoints (pylons) t is limited by the maximum UAV velocity $v_{\max}$, and the time of the segment inspection d is limited by the inspection velocity v_{insp}. The edge cost $c(e_{i,j})$ (also referred as $c_{i,j}$) is the time spent on the travel to the segment endpoint from the endpoint of the previous segment added to the time spent on the inspecting the power line segment, except the edges to the terminal vertex v_1, as seen in Fig. 3b. The practical feasibility that the UAV can track the used trajectories has been experimentally verified and the results are reported in Section VI.

Due to the limited travel budget $C_{\max}$, all the power line segments cannot be visited in a single tour. Therefore, we search for the optimal number of tours $\mathcal{T} = \{T_1, \ldots, T_{n_t}\}$ such that each tour $T_i \in \mathcal{T}$ originates at v_0, terminates at v_1, and each segment is visited exactly once in a union of all tours, i.e., only one of v_{2i} and v_{2i+1} is visited, because the PTL segment is represented by two vertices with the first vertex of the even index and the second incremented by one. The travel cost of the tour $T_i \in \mathcal{T}$ has to satisfy the travel budget constraint $c(T) \leq C_{\max}$.

The PTL inspection planning problem can be formally defined as the *Multi-tour Set Traveling Salesman Problem* (MS-TSP) depicted in Problem 3.1.

Problem 3.1 (Multi-tour Set Traveling Salesman Problem):

$$\text{minimize}_{n_t, \mathcal{T}=\{T_1,\ldots,T_{n_t}\}} \; c(\mathcal{T}) = \sum_{T \in \mathcal{T}} c(T)$$

s.t.

$c(T) \leq C_{\max}$ for $T \in \mathcal{T}$ and for each $S_i = \{v_{2i}, v_{2i+1}\}$, $1 \leq i \leq n_s$, only v_{2i} or exclusively v_{2i+1} is visited by $\mathcal{T}$.

IV. ILP FORMULATION

The optimal solution of the formulated MS-TSP can be obtained by a solution of the ILP problem formulation using the notation of the problem instance on a graph with n vertices and precomputed individual costs between them as two-dimensional asymmetric cost matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$, with the edge costs $c(e_{i,j})$, $0 \leq i, j \leq n$. The solution can be described by the three dimensional matrix of variables $\mathbf{X} \in \{0,1\}^{n_t \times n \times n}$, where the matrix element $x_{m,i,j}$ denotes the edge traversal from i to j by a tour m, and n_t is the

maximal number of the tours. The matrix $\mathbf{X}$ specifies which edges are used by the individual tours. The tours are encoded in traversal matrix $\mathbf{T} \in \mathbb{N}^{n_t \times n}$, where elements $t_{m,i}$ denote the position of the vertex i in the tour m. Thus, $\mathbf{T}$ specifies the order of the edges visits.

The initial and final locations are defined as vertices v_0 and v_1. Furthermore, each segment is represented by two vertices v_{2i} and v_{2i+1} for $1 \le i \le n_s$; therefore, the number of vertices n is always even. The ILP formulation is as follows.

$$\mathcal{C} = \sum_{1 \le m \le n_t} \sum_{1 \le i \le n} \sum_{1 \le j \le n} c_{m,i,j} x_{m,i,j} \quad (1)$$

$$\sum_{i=2}^{n-1} x_{m,0,i} = 1, \text{ for } 0 \le m < n_t \quad (2)$$

$$\sum_{i=2}^{n-1} x_{m,i,1} = 1, \text{ for } 0 \le m < n_t \quad (3)$$

$$\sum_{j=0}^{n-1} \sum_{m=0}^{n_t-1} (x_{m,2i,j} + x_{m,2i+1,j}) = 1, \text{ for } 1 \le i < \frac{n}{2} \quad (4)$$

$$\sum_{j=0}^{n-1} \sum_{m=0}^{n_t-1} (x_{m,j,2i} + x_{m,j,2i+1}) = 1, \text{ for } 1 \le i < \frac{n}{2} \quad (5)$$

$$\sum_{i=0}^{n-1} (x_{m,i,j} - x_{m,j,i}) = 0, \text{ for } 0 \le m < n_t, 2 \le j < n \quad (6)$$

$$\sum_{(i,j) \in \mathbf{X}_m} c_{m,i,j} x_{m,i,j} \le C_{\max}, \text{ for } 0 \le m < n_t \quad (7)$$

$$t_{m,i} - t_{m,j} + x_{m,i,j} \le n - 1, \text{ for } 2 \le i \ne j \le n, \\ 0 \le m < n_t \quad (8)$$

The objective (1) is to minimize the total solution cost $\mathcal{C}$. Constraints (2) and (3) ensure that each tour starts at v_0 and terminates at v_1. The requirement that each set (corresponding to two possible directions of how the power-line segment can be inspected) is entered and exited exactly once is ensured by (4)–(6). Constraints (4) state that for each pair of segment vertices i, the sum of all variables (representing the arrival to destination vertex j) over all routes m is equal to 1. Constraints (6) ensure that for each tour m, once a vertex j is entered, the tour also exits it, where i indexes all possible points of entry and exit for the vertex. The travel budget $C_{\max}$ is constrained by (7), where $\mathbf{X}_m \in \{0,1\}^{n \times n}$ denotes the square sub-matrix of $\mathbf{X}$ containing variables for the tour m. Finally, the sub-tour elimination constraint (Miller-Tucker-Zemlin [20]) is included for each tour using elements of the matrix $\mathbf{T}$ in (8).

The presented formulation can be used to solve the problem using off-the-shelf ILP solvers. The CPLEX [32] solver was used to obtain our baseline solution.

V. PROPOSED HEURISTIC SOLVER

Even though the solution of Problem 3.1 can be found for a particular value of n_t by the presented ILP formulation, it requires high computational resources for finding the optimal solution that might not be practical when the inspection plan needs to be updated on-site in real-time. It may happen that finding the optimal solution requires more time than

performing an inspection with a suboptimal solution. Thus, we propose a GRASP-based heuristic algorithm to find a feasible inspection plan quickly to follow the requirements of PTL operators from the AERIAL CORE consortium.

The proposed algorithm is based on the existing GRASP-based planner for the multi-product multi-vehicle routing problem [21] that we generalized to the Set TSP. In particular, the proposed planner employs two meta-heuristics. The initial (but not necessarily feasible) solution is found by the *Greedy Random search Procedure* (GRP) and is followed by the adaptive *Tabu Search* (TS) to explore the solution neighborhood using a set of specific moves. A soft constraint is used to limit the maximum tour cost during the search. It is not guaranteed that the best solution found will be feasible with respect to $C_{\max}$. Hence, multiple iterations with an increasing number of tours n_t might be required.

The soft constraint is violated if the tour cost $c(T)$ exceeds the maximum allowed cost $C_{\max}$, as shown in Equation 9. The constrained cost is referred to as $c_{con}(T)$.

$$c_{con}(T) = \begin{cases} c(T) & \text{if } c(T) \le C_{\max}, \\ c(T) + (c(T) - C_{\max})k_c & \text{if } c(T) > C_{\max}. \end{cases} \quad (9)$$

The constant k_c is chosen to be a sufficiently large number with regard to the tour costs. Similarly to the ILP formulation, all costs between the vertices are pre-computed using the same trajectory planner to obtain feasible trajectories. The real computational requirements of finding all trajectories are negligible compared to the combinatorial part of the problem. The two parts of the GRASP-based solver are described in the following parts of this section.

A. Greedy Random search Procedure (GRP)

The initial solution $S_{initial}$ of is obtained using the GRP where the segments are iteratively placed into $S_{initial}$ from the set of unused segments $S_{available}$. Possible insertions are stored in $I_{proposed}$, where the insertion $i_{s,m,p,d} \in I_{proposed}$ encodes a possible insertion of some segment s into the tour m at the position p of the tour in the direction d. Since the costs are pre-computed, the solution cost after insertion is known and denoted as $\text{Cost}(i)$ and is the cost $c_{con}(T)$ of the route T that is being inserted into. It partially equalizes individual tour costs to reach a valid solution that does not violate $C_{\max}$ and leads to better solutions in the following TS.

The Boolean variable indicates the inspection direction d, however it can be an integer value in the case of further generalization. The *Restricted Candidate List* (RCL) of the best insertion candidates denoted I_{RCL} is utilized in the randomized search of the GRP. The I_{RCL} is empirically chosen to contain 25% of all proposed insertions to get varied initial solutions. The GRP is summarized in Algorithm 1.

B. Adaptive Tabu Search

After the GRP determines the initial solution, the adaptive TS is applied over several iterations to the routes. The solution space is explored by repeatedly selecting from four distinct moves using the weighted roulette wheel [21]. Similar moves

are used to specify the solution neighborhood with one move for direction-switching added, which hastens the convergence to a lower solution cost.

- Move 1 - *Random-shift*: Remove a random segment from a random route and insert it into a random position in the same or other route. The move adds randomness to the search, increasing its weight in every iteration.
- Move 2 - *Best-shift*: Remove a random segment from the route and insert it into the best position in the same or other route. The cheapest (best) solution is determined by the exhaustive search from all possible insertions.
- Move 3 - *Best-swap*: Swap a random segment from a random route with the best segment in the same or other route. The best swap is, like in the previous move, found through an exhaustive search from all swaps.
- Move 4 - *Best-direction-switch*: Switch a traversal direction for the best segment that is also evaluated over all segments.

Both directions of segments are evaluated for the first three moves and the best is/are chosen.

The score weight of each move ($w_i, i \in 1 \ldots 4$) is initialized to the value w_0 and is increased every iteration according to the performance of moves. Two prize values are used in the employed adaptive TS: $p1$ and $p2$. The price $p1$ is added to the best move in the neighborhood, and $p2$ is added to the move that improved the best overall TS solution. The value of $p1$ is added to the weight w_1 every iteration to promote randomness in neighborhood exploration. For every R_t iterations, the score weights are reinitialized to the value w_0. The neighborhood of the size n^2 is explored using the tabu list T_{list} of the size $n/4$.

The TS procedure is outlined in Algorithm 2, where S denotes some existing solution, T_{list} is the tabu list of explored solutions, and $N(S)$ is the neighborhood of some solution S. The stopping criterion in Algorithm 2 is when there is no improvement of S_{best} in the last 50 iterations of creating a solution neighborhood. This empirically selected criterion is chosen to acquire the best possible solution to be compared with the CPLEX solution. Specific application of the GRASP procedure might prefer a different criterion, such as a time limit in real-time operation.

Algorithm 1 Greedy Random search Procedure (GRP)

1: $S_{initial} \leftarrow \emptyset$;
2: $S_{available} \leftarrow S$;
3: **while** size($S_{available}) \neq 0$ **do**
4:      Fill $I_{proposed}$ with all possible insertions;
5:      Order elements of $I_{proposed}$ by Cost(i);
6:      Fill I_{RCL} with a subset of $I_{proposed}$;
7:      $i_{selected} \leftarrow$ Random i from I_{RCL};
8:      Insert $i_{selected}$ into the route m and update $S_{initial}$;
9:      Remove $i_{selected}$, and other insertions referring to same segment set from $S_{available}$;
10: **end while**
11: **return** $S_{initial}$;

Algorithm 2 Adaptive Tabu Search

Require: Some existing solution S.
1: $S_{best} \leftarrow S$;
2: $S_{current} \leftarrow S$;
3: Insert $S_{current}$ into T_{list};
4: **while** Stopping criterion is not met **do**;
5:      Create a solution neighborhood $N(S_{current})$;
6:      $S_{current} \leftarrow$ Best$\{S \in N(S_{current})$ and $S \notin T_{list}\}$;
7:      **if** Cost($S_{current}$) < Cost(S_{best}) **then**
8:          $S_{best} \leftarrow S_{current}$;
9:      **end if**
10: **end while**
11: **return** S_{best};

VI. RESULTS

The feasibility of the proposed PTL inspection planning has been empirically validated on a real-world dataset [33] of the PTLs originating from the substation Nechranice, courtesy of ČEPS, a.s. (Czech power transmission infrastructure institution, a member of the AERIAL CORE advisory board). A map showing an overview of the PTL layout surrounding the substation is depicted in Fig. 4.
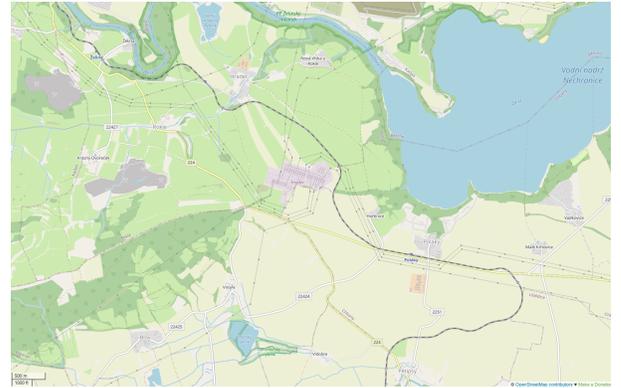
Figure 4. Map of Nechranice substation and its surroundings used to create realistic PTL inspection planning problem instances. The map was acquired from Open Street Map [34] database.

The S-JTSK coordinates of pylons were transformed into the Cartesian system with the origin at the substation. The benchmark problem instances were sampled from the area around the Nechranice substation using the perimeter D_{max}, starting with a small number of segments that increases with D_{max}, as indicated in Table I. Furthermore, several limits on the

Table I
SIZE OF THE BENCHMARK INSTANCES

D_{max}[m]	500	600	700	800	900	1 000	1 200	2 000	5 000
n_s	15	23	28	33	40	43	51	76	172

flight time C_{max} have been chosen for each particular scenario to evaluate the influence of the inspection area defined by D_{max} and the maximal flight time per a single tour. The travel budget C_{max} is set proportionally to D_{max} to obtain feasible solutions for $n_t \in \{3, \ldots, 6\}$ that corresponds to tens of minutes of

flight time for D_{max} in hundreds of meters and reaches up to several hours for finding a feasible solution of the largest instances.

The travel costs have been determined by estimating flight trajectories of a real UAV constrained by its flight dynamics. The cost of travel between the segments is limited by the maximum flight velocity $v_{max} = 5\,\mathrm{m\,s^{-1}}$ and the traversal cost during the segment inspection is limited by the maximum inspection velocity $v_{insp} = 1\,\mathrm{m\,s^{-1}}$. The maximum acceleration is limited to $a_{max} = 2.5\,\mathrm{m\,s^{-2}}$. Experimental validation of the costs is reported in Section VI-B.

Each particular instance is solved optimally using the proposed ILP formulation for the lowest n_t, for which the ILP solver provides the solution within the maximal computational time of up to $24\,\mathrm{h}$. Note that the ILP solver finds the optimal solution for the increasing number of tours, which is usually less demanding. However, such a solution is not the optimal solution of the original problem with respect to the minimal number of tours as stated in Problem 3.1.

Both proposed methods, the ILP-based and GRASP-based approaches, have been implemented in C++. The ILP is solved using CPLEX 12.8.0 [32] implemented in C++ using the ILOG Concert technology library. Computations were performed on the Intel Xeon processors (2.2–3.3GHz) limited to utilization of a single core. Each CPLEX instance has been limited by $100\,\mathrm{GB}$ of RAM. Without a prior initial solution, the computational requirements of solving the ILP are exceptionally high, and most of the herein reported results would not be computed in a limited time. Therefore, the CPLEX solver has been initialized a solution found by the proposed GRASP-based solver, which computational cost is negligible compared to the cost of the optimal solution. The GRASP-based solver has been parameterized as follows.

The candidate list size is $25\,\%$ of the possible insertion list size. The initial move weight in the TS is $w_0 = 5$, the prize weights $p_1 = 1$, $p_2 = 5$, and number of iterations after which the weights are reset is set to $w_r = 5$. As outlined in [29], the size of tabu lists should be kept small. We therefore opted for $n/4$, where n is the number of segments manipulated. The size of the solution neighborhood has been fixed at n. The stopping criterion is 50 iterations without improvement of the best solution. Because the GRASP is stochastic and relatively inexpensive to compute, each instance has been solved 30 times. Both the best solution found and mean solution values among the trials were reported.

A. Computational Evaluation

The performance of both proposed solvers including a basic Greedy Random (GR) search has been studied for the size of the instances defined by D_{max}, the travel budget C_{max}, and the number of inspection tours n_t. The evaluation results are depicted in Table II, where the lowest optimal solution costs are in bold. The best solution cost provided by the GR algorithm is also included in case it managed to provide a valid solution among the performed trials. The $\%SR$ denotes success rate of this search. In all examined benchmark instances, the best solutions are found by the ILP problem formulation for the
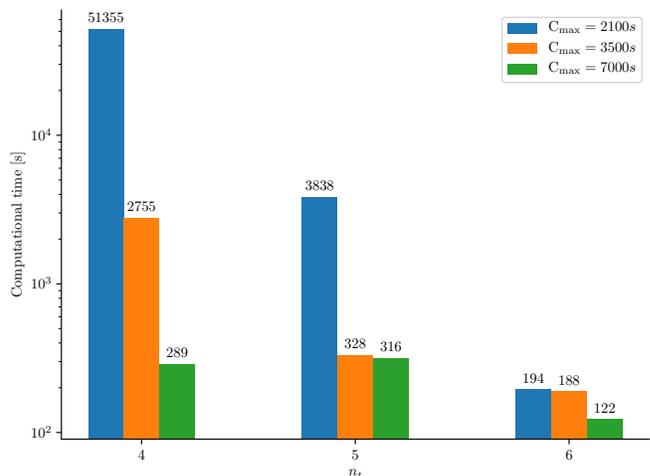

Figure 5. Computational requirements to solve the ILP formulation for instances with $D_{max} = 700\,\mathrm{m}$, travel budget C_{max}, and n_t tours. The $y-$axis is in log-scale. Solution costs are above their respective bars.

lowest possible n_t that is considered the optimal solution of the instance with the particular cost $\mathcal{C}^*$. Therefore, we measure the quality of the ILP-based solution for the increased number of tours n_t, and also for the heuristic GRASP-based solutions as the percentage deviation of the solution cost to the optimal solution cost $\%\mathrm{PDB} = (\mathcal{C} - \mathcal{C}^*)/\mathcal{C}^* \cdot 100\%$. The computational time of the optimal ILP-based solution and the solution of all 30 trials by the GRASP is denoted t and reported in seconds. Thus, the times can be compared with the solution cost $\mathcal{C}$ that is reported as the required total flight time in seconds. Additionally for the GRASP, the mean computational time t_{mean} per a single trial is reported together with the mean solution quality $\mathcal{C}_{mean}$, which is reported as the percentage deviation of the mean solution cost to the optimal solution cost $\%\mathrm{PDM} = (\mathcal{C}_{mean} - \mathcal{C}^*)/\mathcal{C}^*$ that measures the robustness of the solution among the performed trials.
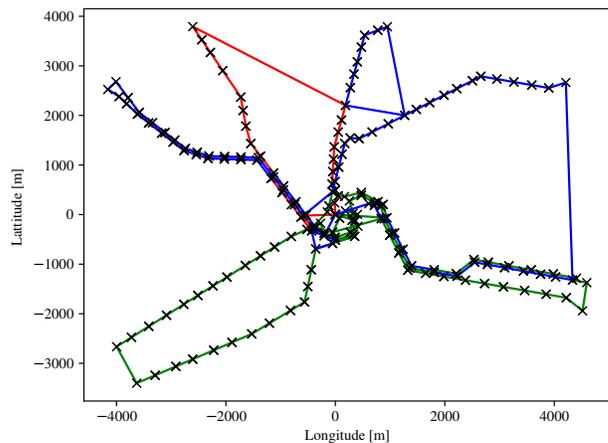

Figure 6. Example of the solution found by the GRASP for the largest examined instance for $D_{max} = 5\,\mathrm{km}$ with $n_s = 172$ segments and $n_t = 3$.

The result indicates that the ILP is very demanding. Moreover, the computational time spikes unpredictably depending on the mixture of constraints and quality of the initial solution.

Table II
COMPUTATIONAL RESULTS

D_{max} [m]	C_{max} [s]	n_t	ILP			GRASP						GR†			
			t [s]	$\mathcal{C}$ [s]	%PDB	t_{mean} [s]	t [s]	$\mathcal{C}_{mean}$ [s]	%PDM	$\mathcal{C}$ [s]	%PDB	t [s]	$\mathcal{C}$ [s]	%PDB	%SR
500	1 000	4	7.4	**3 178.6**	0.0	0.05	1.64	3 506.5	10.3	3 499.8	10.1	0.05	-	-	0
500	1 000	5	2.6	3 221.4	1.3	0.08	2.27	3 397.4	6.9	3 371.0	6.0	0.06	-	-	0
500	1 000	6	0.9	3 328.4	4.7	0.04	1.34	3 566.6	12.2	3 478.4	9.4	0.06	-	-	0
500	1 500	3	1.9	**3 053.2**	0.0	0.06	1.77	3 281.7	7.5	3 262.4	6.9	0.04	-	-	0
500	1 500	4	1.4	3 087.9	1.1	0.12	3.47	3 321.7	8.8	3 260.1	6.8	0.04	5 228.8	71.3	62
500	1 500	5	1.2	3 194.9	4.6	0.05	1.49	3 458.4	13.3	3 448.1	12.9	0.05	5 063.1	65.8	100
500	1 500	6	0.8	3 296.1	8.0	0.05	1.58	3 492.2	14.4	3 452.7	13.1	0.05	5 637.1	84.6	100
600	1 800	3	74.1	**5 143.5**	0.0	0.07	2.15	5 582.9	8.5	5 497.5	6.8	0.13	-	-	0
600	1 800	4	46.3	5 176.8	0.7	0.15	4.63	5 645.5	9.8	5 590.7	8.7	0.14	-	-	0
600	1 800	5	35.5	5 244.6	2.0	0.11	3.25	5 764.9	12.1	5 705.2	10.9	0.15	8 608.0	67.4	19
600	1 800	6	39.4	5 356.3	4.1	0.13	4.01	5 843.6	13.6	5 701.6	10.9	0.18	8 501.0	65.3	81
600	3 000	3	12.1	**5 090.4**	0.0	0.18	5.36	5 562.5	9.3	5 460.4	7.3	0.13	8 473.0	66.5	57
600	3 000	4	11.9	5 162.0	1.4	0.08	2.49	5 595.4	9.9	5 464.2	7.3	0.14	8 750.1	71.9	100
600	3 000	5	10.9	5 229.8	2.7	0.13	3.78	5 698.6	11.9	5 661.9	11.2	0.16	9 098.5	78.7	100
600	3 000	6	12.9	5 344.1	5.0	0.16	4.84	5 878.6	15.5	5 796.4	13.9	0.16	8 994.1	76.7	100
700	2 100	4	51 355.1	**6 268.3**	0.0	0.17	5.21	6 951.6	10.9	6 736.5	7.5	0.26	-	-	0
700	2 100	5	3 838.8	6 321.1	0.8	0.14	4.08	6 986.3	11.5	6 717.4	7.2	0.26	-	-	0
700	2 100	6	194.3	6 416.9	2.4	0.20	6.13	7 042.3	12.3	6 928.9	10.5	0.27	11 247.0	79.4	100
700	3 500	3	6 401.6	**6 161.8**	0.0	0.29	8.61	6 700.0	8.7	6 526.1	5.9	0.24	-	-	0
700	3 500	4	2 755.7	6 222.4	1.0	0.19	5.71	6 748.0	9.5	6 637.1	7.7	0.25	10 744.4	74.4	95
700	3 500	5	328.7	6 289.7	2.1	0.43	12.88	6 880.6	11.7	6 716.0	9.0	0.26	11 434.3	85.6	100
700	3 500	6	188.4	6 415.7	4.1	0.20	6.01	7 118.6	15.5	7 024.6	14.0	0.28	10 676.7	73.3	100
800	2 400	5	15 765.0	**7 510.0**	0.0	0.19	5.77	8 420.8	12.1	8 234.7	9.6	0.40	-	-	0
800	2 400	6	2 807.3	7 602.4	1.2	0.28	8.54	8 421.3	12.1	8 288.1	10.4	0.42	-	-	0
800	4 000	3	24 120.2	**7 369.2**	0.0	0.39	11.64	8 058.1	9.3	7 879.4	6.9	0.38	-	-	0
800	4 000	4	769.0	7 404.0	0.5	0.35	10.56	8 210.2	11.4	8 020.5	8.8	0.39	13 000.5	76.4	100
800	4 000	5	24 379.7	7 488.0	1.6	0.46	13.91	8 285.1	12.4	8 081.8	9.7	0.40	12 807.7	73.8	100
800	4 000	6	1 770.0	7 602.4	3.2	0.49	14.73	8 482.1	15.1	8 289.6	12.5	0.41	13 613.2	84.7	100
900	2 700	6	10 210.1	**9 014.4**	0.0	0.59	17.77	10 210.4	13.3	9 933.2	10.2	0.67	-	-	0
900	4 500	3	35 372.2	**8 782.4**	0.0	0.64	19.05	9 804.2	11.6	9 600.3	9.3	0.63	-	-	0
900	4 500	4	1 004.2	8 817.1	0.4	0.53	15.87	9 906.9	12.8	9 631.8	9.7	0.64	15 503.3	76.5	100
900	4 500	5	26 029.7	8 884.4	1.2	0.87	25.97	10 108.4	15.1	9 809.0	11.7	0.64	15 896.3	81.0	100
900	4 500	6	3 506.5	9 006.6	2.6	0.52	15.47	10 246.7	16.7	9 986.0	13.7	0.67	16 090.8	83.2	100
1000	3 000	6	21 174.5	**9 887.6**	0.0	0.65	19.51	11 130.4	12.6	10 870.0	9.9	0.80	-	-	0
1000	5 000	4	86 370.0	**9 712.7**	0.0	0.62	18.48	10 845.0	11.7	10 544.2	8.60	0.76	16 970.2	74.7	100
1000	5 000	6	13 605.2	9 887.6	1.8	0.63	18.97	11 112.0	14.4	10 917.7	12.4	0.82	17 442.7	79.6	100
1100	5 500	4	86 370.1	**11 113.4**	0.0	1.50	45.14	12 407.4	11.6	12 144.0	9.3	1.02	19 603.2	76.4	100
1100	11 000	3	59 362.5	**10 995.0**	0.0	1.11	33.38	12 196.8	10.9	11 988.4	9.0	0.98	17 879.1	62.6	100
1100	11 000	4	86 370.0	11 078.4	0.8	0.68	20.27	12 372.6	12.5	12 168.2	10.7	0.98	18 963.6	72.5	100
1100	11 000	5	9 371.1	11 179.8	1.7	0.77	23.21	12 565.2	14.3	12 233.3	11.3	0.97	19 021.9	73.0	100
1200	12 000	4	86 370.1	**12 453.0**	0.0	0.76	22.80	13 870.3	11.4	13 567.6	9.0	1.25	21 502.8	72.7	100
1200	12 000	5	86 370.0	12 564.0	0.9	1.03	30.83	14 044.0	12.8	13 502.2	8.4	1.23	22 327.5	79.3	100
1300	13 000	3	86 370.0	**13 143.8**	0.0	1.31	39.33	14 476.4	10.1	14 156.6	7.7	1.32	22 278.2	69.5	100
1500	15 000	4	86 370.1	**15 860.6**	0.0	1.75	9.81	17 637.8	11.2	17 258.8	8.8	2.03	26 821.4	69.1	100

†Missing values indicate no feasible solution has been found among the performed trials for a particular problem instance.

Generally, it tends to increase exponentially with tighter C_{max} and decreasing n_t, as is further shown in Fig. 5.

The computational cost of the proposed heuristic GRASP-based solver is significantly lower than the ILP, and scales with the size of the instance defined by D_{max} and tight constraints on C_{max} and n_t. Solutions are provided in about $1\,\mathrm{min}$ for the largest instance depicted in Fig. 6. In the presented results for 30 trials, the solution cost variance is about $17\,\%$ and %PDM is relatively high, but not exceeding $20\,\%$, which indicates a high variance in the found solutions. The measured %PDB of GRASP is not exceeding $15\,\%$.

Discussion: - Based on the reported results, the trade-off between almost instantaneously solving the ILP to improve the initial solution provided by GRASP is mostly unfavorable in the case of limited computational time. The benefit of the optimal solution rapidly decreases with tighter constraints or larger instances. Thus, it might be more suitable to deploy heuristic solutions, which are about tens or even hundreds of seconds longer than the optimal solutions that require significantly longer computational times. On the other hand, the results also indicate that with the increasing number of tours n_t, the ILP-based solution might still be better than a heuristic solution for a lower number of tours with significantly lower computational time to find the optimal solution. Here, it is worth mentioning that the lowest feasible n_t is not necessarily optimal for C_{max}. The coincidence of the lowest n_t in the presented results arises from the problem geometry of closely spaced star-shaped PTLs and the flight cost between the segment inspections being cheaper than the inspection cost. The optimal value of n_t depends on the use-case and which optimality criterion is preferred, i.e., the lowest number of tours or the lowest overall flight time.

B. Experimental Validation

The practical feasibility of the proposed PTL inspection planning approach has been experimentally verified with real UAV in a testing polygon. The experimental validation has been carried out using the control architecture [35] and UAV platforms designed for the AERIAL CORE project [36]. A multi-rotor UAV used during the inspection is depicted in Fig. 7a and a trajectory containing a mix of segment lines is shown in Fig. 7b. The experimental deployment supports the feasibility of the presented approach and verifies the created
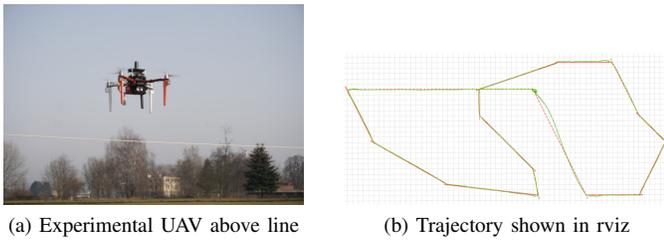
(a) Experimental UAV above line   (b) Trajectory shown in rviz

Figure 7. The multi-rotor UAV used during the experiment **(a)** was equipped with Reach M2 GNSS module for *Real-Time Kinematic* (RTK) positioning. Time sampled trajectory ($\Delta t = 0.2\,\mathrm{s}$) over segments **(b)** (shown in red) was tracked with regard to the UAV constraints (shown in green). Grid resolution is $1\,\mathrm{m}$. Video from the experiment is available at [33].

benchmark instances as realistic. Besides, it also supports feasibility of the estimated trajectory costs found by the trajectory planner with the employed UAV controller.

VII. CONCLUSION

The PTL inspection planning with aerial vehicles has been addressed by the novel generalization of the TSP called the Multi-tour Set TSP. The optimal ILP-based formulation is proposed together with a less computationally demanding GRASP-based heuristic approach. Both solvers have been empirically evaluated and the feasibility of the proposed approach has been experimentally verified with real vehicles. Additionally, a realistic dataset has been created and made available for benchmarking the proposed approach and for eventual further approaches. The main benefit of the presented GRASP-based solver is in low computational requirements. It can provide solutions on-site almost instantaneously, where a change of the inspection plan may be needed as required by the PTL operators and end-users of the proposed approach. The possible future work lies with improving the scalability for solving large instances.

REFERENCES

[1] O. Menendez, M. Perez *et al.*, "Vision based inspection of transmission lines using unmanned aerial vehicles," in *IEEE MFI*, 2016, pp. 412–417.

[2] J. Katrasnik, F. Pernus *et al.*, "A survey of mobile robots for distribution power line inspection," *IEEE Transactions on Power Delivery*, vol. 25, pp. 485–493, 2010.

[3] T. Disyadej, J. Promjan *et al.*, "High voltage power line maintenance & inspection by using smart robotics," in *IEEE ISGT*, 2019, pp. 1–4.

[4] T. He, Y. Zeng *et al.*, "Research of multi-rotor UAVs detailed autonomous inspection technology of transmission lines based on route planning," *IEEE Access*, vol. 7, pp. 114 955–114 965, 2019.

[5] H. Lee, C. Kim *et al.*, "Vision-based automatic real time inspection of power transmission line," in *2013 44th International Symposium on Robotics (ISR)*, 2013, pp. 1–3.

[6] G. Dong, X. Chen *et al.*, "Inspecting transmission lines with an unmanned fixed-wings aircraft," in *ICARPI*, 2012, pp. 173–174.

[7] W. Filho and A. Pinto, "Vtol aerial robot for inspection of transmission line," in *ICARPI*, 2014, pp. 1–4.

[8] P. Hamelin, F. Miralles *et al.*, "Discrete-time control of linedrone: An assisted tracking and landing UAV for live power line inspection and maintenance," in *ICUAS*, 2019, pp. 292–298.

[9] G. Zhou, J. Yuan *et al.*, "Robust real-time UAV based power line detection and tracking," in *IEEE ICIP*, 2016, pp. 744–748.

[10] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[11] T. Moyo and F. du Plessis, "The use of the travelling salesman problem to optimise power line inspections," in *Robotics and Mechatronics Conference (RobMech)*, 2013, pp. 99–104.

[12] J. Cui, Y. Zhang *et al.*, "Path planning algorithms for power transmission line inspection using unmanned aerial vehicles," in *Chinese Control And Decision Conference (CCDC)*, 2017, pp. 2304–2309.

[13] Y. Liu, J. Shi *et al.*, "Two-layer routing for high-voltage powerline inspection by cooperated ground vehicle and drone," *Energies*, vol. 12, p. 1385, 04 2019.

[14] K. Helsgaun, "Solving the equality generalized traveling salesman problem using the Lin-Kernighan-Helsgaun Algorithm," *Mathematical Programming Computation*, vol. 7, no. 3, pp. 269–287, Sep 2015.

[15] G. Laporte, H. Mercure *et al.*, "Generalized travelling salesman problem through n sets of nodes: the asymmetrical case," *Discrete Applied Mathematics*, vol. 18, no. 2, pp. 185–197, 1987.

[16] P. R. Sokkappa, "The cost-constrained traveling salesman problem," Ph.D. dissertation, University of North Texas, 1990. [Online]. Available: https://digital.library.unt.edu/ark:/67531/metadc1113789/

[17] R. Pěnička, J. Faigl *et al.*, "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.

[18] J. Faigl, J. Deckerová *et al.*, "Fast heuristics for the 3D multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," *IEEE Robotics and Automation Letters*, vol. 4, pp. 2439–2446, 2019.

[19] T. Bektas, "The multiple traveling salesman problem: An overview of formulations and solution procedures," *Omega*, vol. 34, pp. 209–219, 2006.

[20] C. Miller, A. Tucker *et al.*, "Integer programming formulation of traveling salesman problems," *Journal of the Association for Computing Machinery*, vol. 7, pp. 326–329, 1960.

[21] O. Guemri, A. Bekrar *et al.*, "Grasp-based heuristic algorithm for the multi-product multi-vehicle inventory routing problem," *4OR - A Quarterly Journal of Operations Research*, vol. 14, pp. 377–404, 2016.

[22] L. Bertazzi and M. Speranza, "Inventory routing problems: An introduction," *EURO Journal on Transportation and Logistics*, vol. 1, pp. 307–326, 2012.

[23] P. Kitjacharoenchai, M. Ventresca *et al.*, "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach," *Computers and Industrial Engineering*, 2019.

[24] A. Maha and Z. Ahmed, "Genetic algorithms for the multiple travelling salesman problem," *International Journal of Advanced Computer Science and Applications*, vol. 11, pp. 553–560, 07 2020.

[25] C. Song, K. Lee *et al.*, "Extended simulated annealing for augmented tsp and multi-salesmen tsp," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3, 2003, pp. 2340–2343 vol.3.

[26] H. Wacholder and Mann, "A neural network algorithm for the multiple traveling salesmen problem," *Biological Cybernetics*, vol. 61, pp. 11–19, 1989.

[27] J. Faigl, "An application of self-organizing map for multirobot multigoal path planning with minmax objective," *Computational Intelligence and Neuroscience*, pp. 2 720 630:1–2 720 630:15, 2016.

[28] O. Cheikhrouhou and I. Khoufi, "A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy," *Computer Science Review*, vol. 40, p. 100369, 2021.

[29] S. Tsubakitani and J. Evans, "Optimizing tabu list size for the traveling salesman problem," *Computers & OR*, vol. 25, pp. 91–97, 02 1998.

[30] G. Laporte and Y. Nobert, "Generating travelling salesman problem through n sets of nodes: An integer programming approach," *Information Systems and Operational Research*, vol. 21, pp. 61–75, 1983.

[31] S. G. Manyam, S. Rathinam *et al.*, "GPS denied UAV routing with communication constraints," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 691–703, 2016.

[32] Cplex, IBM ILOG, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[33] "Supporting Material for the Multi-tour Set Traveling Salesman Problem in Planning PTL Inspection," 2021, [cited 1 Mar 2021]. [Online]. Available: http://mrs.felk.cvut.cz/mstsp-ptl-ral

[34] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," 2017, [cited 1 Mar 2021].

[35] T. Baca, D. Hert *et al.*, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6753–6760.

[36] G. Silano, T. Baca *et al.*, "Power line inspection tasks with multi-aerial robot systems via signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4169–4176, 2021.