2020 IEEE 16th International Conference on Control & Automation (ICCA)
Oct 9-11, 2020 (Virtual) Sapporo, Hokkaido, Japan

Multi-Regional Coverage Path Planning for Robots with Energy Constraint

Junfei Xie[1] and Jun Chen[2]

Abstract— **Coverage path planning (CPP) has been extensively studied in the literature, which is a key step to realize robotic applications that require complete coverage of a region, such as lawn mowing, room cleaning, land assessment, search and rescue. However, CPP for multiple regions has gained much less attention, which arises in many real scenarios. This multi-regional CPP problem can be considered as a variant of the traveling salesman problem (TSP) enhanced with CPP, namely TSP-CPP. In this paper, we extend our previous investigation on the TSP-CPP problem to further consider the energy constraint of the robots. As the constrained TSP-CPP problem has an NP-Hard computational complexity, a stepwise selection based heuristic algorithm is developed to solve the problem. Simulation experiments and comparison studies show the good performance of the proposed algorithm in balancing optimality and efficiency.**

I. INTRODUCTION

In the past twenty years, robotics or unmanned vehicles have experienced an unprecedented level of growth. Many of their applications, such as lawn mowing [1], room cleaning [2], land surveying [3], to name a few, involve the task of planning the path for the robots to scan a region completely, while satisfying certain safety and mission-related constraints. This task is known as the coverage path planning (CPP) problem.

Two types of approaches have been developed to solve the CPP problem [4]: the randomized approaches and the complete approaches. The randomized approaches, adopted by many floor-cleaning robots, generate a random path that can fully cover a region if long enough. This type of approach is easy to implement but produces long paths. The complete approaches, which are more frequently researched, generate shorter paths that can guarantee full coverage, by decomposing the region into a collection of small cells and then finding the coverage path for each cell.

Although the CPP problem for a single region has been extensively studied, the practical scenario where multiple regions are to be fully covered has been largely ignored. This multi-regional CPP problem can be considered as a variant of the vehicle routing problem (VRP) [5] or the traveling salesman problem (TSP) [6] enhanced with CPP, where the determination of the region visiting order is a TSP and the coverage for each region is a CPP problem. We hence name it the TSP-CPP problem. Despite the existence of many approaches for TSP, such as Branch-and-Bound [7], dynamic programming [8], nearest neighbor [9], to name a few, the TSP-CPP problem cannot be solved by a direct extension to these TSP or CPP approaches. This is because the TSP-CPP problem requires the determination of the entrance and exit locations, which impact both the region visiting order and intra-regional coverage paths and are not considered in either TSP or CPP problems.

A related problem, called the tour polygon problem (TPP) [10], also considers multiple regions, but it does not require full coverage of each region. In particular, the TPP seeks the optimal path to merely visit multiple regions. In cases when the robot is only allowed to visit the edge of each region without entering the region, the TPP is also known as the zookeeper problem [11]. Otherwise, if the robot can freely cross the regions, the TPP is often referred to as the Safari problem [12].

In our previous studies [13]–[15], we have investigated the TSP-CPP problem for scenarios where a single robot with sufficient power supply is tasked to fully cover multiple non-overlapping convex polygonal regions. To solve this problem, we developed a dynamic programming based exact approach [14], [15] that can find (near) optimal solutions and a nearest neighbor (NN) and 2-Opt based heuristic approach, called Fast NN-2Opt [15], that can generate high-quality tours very efficiently. To the best of our knowledge, we were the first to consider the TSP-CPP problem with mathematical formulation provided.

Very recently, a growing interest has been shown in the TSP-CPP problem [16], [17], driven by the popularity of unmanned aerial vehicles (UAVs), which have been adopted by many applications that involve the TSP-CPP problem, such as land assessment [18] and precision agriculture [19]. In particular, a two steps path planning (TSPP) approach was introduced in [16], which first determines the region visiting order using regions' centroids and then plans the path to cover each region. Although simple, this heuristic approach produces longer tours, compared with our methods [15], as it ignores the interaction between the region visiting order and intra-regional coverage paths. It is also less efficient than our Fast NN-2Opt algorithm. Another paper [17] considers the path planning for multiple UAVs. A heuristic procedure was developed to first assign regions to the UAVs and then optimize the region visiting order. This study, however, oversimplifies the coverage problem. How to enter or exit each region and what is the path to fully

*This work was supported partially by the National Science Foundation under Grant CI-1953048/1730589, and partially by San Diego State University under the University Grants Program.

[1]Junfei Xie is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, 92182. jxie4@sdsu.edu.

[2]Jun Chen is with the Department of Aerospace Engineering, San Diego State University, San Diego, CA, 92182. jun.chen@sdsu.edu.

cover each region are not addressed. Paper [20] considers the distributed motion planning problem for multiple robots to cover multiple rectangular regions. In this study, each robot determines its next motion based on a set of rules.

In this paper, we extend our previous study on the TSP-CPP problem [13]–[15], [21] to further address the energy constraint of the robot. In particular, we consider a more realistic scenario where a robot with limited energy is sent to scan multiple regions. During the mission, the robot is allowed to return to the depot to change its battery. An alternative scenario is to send multiple robots with limited energy to scan the regions collaboratively. This constrained TSP-CPP problem is much more challenging than the original TSP-CPP problem, which is NP-Hard, in that it further requires the optimization of the number of sub-tours (or UAVs) and the assignment of regions to each sub-tour (or UAV). To solve this problem efficiently, we propose a heuristic algorithm based on the Fast NN-2Opt. The proposed algorithm adopts a stepwise selection procedure to minimize the number of sub-tours (or UAVs) and optimize the region assignment, and adopts NN- and 2Opt-based approaches to find the best tour. To demonstrate the performance of the proposed algorithm, we conduct various comparative simulation studies. The results show that the proposed algorithm achieves a good tradeoff between optimality and efficiency.

In the rest of this paper, we first formulate the constrained TSP-CPP problem in Section II. We then briefly review the Fast NN-2Opt algorithm in Section III. In Section IV, we solve the constrained TSP-CPP problem using a stepwise selection based heuristic approach, whose performance is then evaluated through simulation studies in Section V. Section VI finally concludes the paper.

II. Problem Formulation

A. Problem Description and Notations

In this study, we consider the scenario where a single robot (e.g., ground vehicle or UAV) with arbitrary radius of curvature is tasked to scan or survey $N \in \mathbb{Z}^+$ non-overlapping convex polygonal regions. It departs from a depot located at v_0 and returns to the same depot after mission completed. Due to limited power supply, the robot may return to the depot v_0 to change its battery during the mission, but only after completing the scan for at least one region, assuming that the robot has sufficient power to fully cover the largest region. For simplicity, we assume the robot moves at a constant speed and the maximum distance it can travel at this speed and with its battery fully charged is D. Note that D characterizes the robot's energy capacity. The constrained TSP-CPP problem then aims to find the optimal tour[1] for the robot to fully cover all regions with the minimum total cost, while satisfying the energy constraint.

To formulate the constrained TSP-CPP problem mathematically, we first introduce some notations. Let $\boldsymbol{b}_i = \{b_{ik}\}$ be a feasible path to fully cover region $i \in [N] = \{1, 2, \ldots, N\}$, where b_{ik} is the k-th waypoint in the path, $k \in [n_i]$ and $n_i = |\boldsymbol{b}_i|$ is the total number of waypoints. To describe the region visiting order, we introduce a binary variable x_{ij}, $i, j \in [N] \cup \{0\}$, where 0 is the index of the depot. x_{ij} equals to 1 if the robot moves to region (or depot) j after visiting region (or depot) i. Otherwise, x_{ij} equals to 0. To denote the number of sub-tours, where each sub-tour (except the last one) indicates a battery change, we use symbol $m \in \mathbb{Z}^+$.

[1]A tour is a feasible solution to the (constrained) TSP-CPP problem.

B. Mathematical Formulation

Based on the notations defined above, a tour can be described jointly by 1) the region visiting order captured by x_{ij}, $i, j \in [N] \cup \{0\}$, 2) the path to cover each region $\boldsymbol{b}_i$, $i \in [N]$ and 3) the number of sub-tours $m \in [N]$. Mathematically, the constrained TSP-CPP problem can be formulated as follows:

$$\min_{\substack{x_{ij}, \forall i,j \in [N]\cup\{0\} \\ \boldsymbol{b}_i, \forall i \in [N] \\ m \in [N]}} J = \sum_{i=1}^{N} [x_{0i}\mathtt{d}(v_0, b_{i1}) + x_{i0}\mathtt{d}(b_{in_i}, v_0)] + \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}\mathtt{d}(b_{in_i}, b_{j1}) + \sum_{i=1}^{N} \mathtt{g}(\boldsymbol{b}_i) + \beta m \tag{1}$$

subject to:

$$\sum_{i=1}^{N} x_{0i} = m \tag{2}$$

$$\sum_{i=1}^{N} x_{i0} = m \tag{3}$$

$$\sum_{i=0}^{N} x_{ij} = 1, \quad \forall j \in [N] \tag{4}$$

$$\sum_{j=0}^{N} x_{ij} = 1, \quad \forall i \in [N] \tag{5}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1, \forall S \subseteq [N], |S| > 0 \tag{6}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in [N] \cup \{0\} \tag{7}$$

$$\sum_{i\in S\setminus\{0\}} [x_{0i}\mathtt{d}(v_0, b_{i1}) + x_{i0}\mathtt{d}(b_{in_i}, v_0)] + \sum_{i\in S\setminus\{0\}}\sum_{j\in S\setminus\{0\}} x_{ij}\mathtt{d}(b_{in_i}, b_{j1}) + \sum_{i\in S\setminus\{0\}} \mathtt{g}(\boldsymbol{b}_i) < D, \quad \forall \sum_{i\in S}\sum_{j\in S} x_{ij} = |S|, |S| > 1 \tag{8}$$

where J is the total cost to be minimized, which includes the cost of inter-regional paths (first two terms), the cost of intra-regional coverage paths (third term), and the cost for the robot to change its battery at the depot (last term). $\mathtt{d}(a, b)$ denotes the travel cost from location a to location b, which here is measured by the Euclidean distance. $\mathtt{g}(\boldsymbol{b}_i)$ computes the cost of path $\boldsymbol{b}_i$, particularly, $\mathrm{g}(\boldsymbol{b}_i) = \sum_{k=1}^{n_i-1} \mathrm{d}(b_{ik}, b_{i(k+1)})$. $\beta \geq 0$ is the unit cost for changing the battery of the robot.

In the above formulation, (2)-(8) are introduced to ensure the validity of the tour. In particular, (2) and (3) limit the number of sub-tours to m and ensure that each sub-tour starts and ends at the depot. (4) and (5) ensure that each region is scanned just once. (6) is used to prevent invalid sub-tours. (7) ensures that x_{ij} takes valid values. (8) prevents long sub-tours that exceed the energy capacity of the robot. Note that this problem formulation is also suitable for scenarios where m robots with the same energy capacity are tasked to scan N regions, under the goal of minimizing the total cost and the number of robots m. From this point of view, the proposed problem can be considered as a variant of the multiple TSP (MTSP) enhanced by CPP with energy constraint.

As it is infeasible and exceedingly time-consuming to directly solve the constrained TSP-CPP problem formulated in (1), considering the infinite number of possible tours to cover all regions, in this paper, we propose a heuristic approach to solve this problem efficiently.

III. Fast NN-2Opt Algorithm for TSP-CPP without Energy Constraint

Before solving the proposed problem, we first briefly review the Fast NN-2Opt algorithm [15] for the original TSP-CPP problem without energy constraint, which was developed in our previous studies and can generate high-quality tours very efficiently. Based on this algorithm, we will then address the constrained TSP-CPP problem in the next section.

The Fast NN-2Opt algorithm shown in Algorithm 1 consists of two phases. In the first phase, the tour is initialized using a nearest neighbor (NN) based algorithm. The key idea is to first determine the region visiting order using the NN algorithm based on the regions' centroids (Line 4), and then find the complete tour that fully covers all regions (function FindTour() in Line 5). The pseudocode for function FindTour() is provided in Algorithm 2. In this function, the tour is generated by connecting the best intra-regional coverage path for each region one by one in the region visiting order, where the candidate paths for covering each region (Line 2) are generated by a back-and-forth CPP algorithm [14], [15]. This CPP algorithm generates a set of paths with back-and-forth pattern, which have line sweep directions[2] perpendicular to the region's edges and have nice properties in terms of full coverage guarantee, optimality and complexity.

In the second phase, the tour is improved using a 2-Opt based algorithm (function ImproveTour() in Line 6). The pseudocode for function ImproveTour() is provided in Algorithm 3. In this function, a 2-Opt move is performed in each iteration to update the region visiting order, if the cost of the resulting tour is reduced.

[2]The line sweep direction of a back-and-forth path points towards the moving direction of the robot.

Algorithm 1: Fast NN-2Opt Algorithm

```
Input: v_0, vertices of N regions, sensing range of the robot
Output: tour τ*, cost J*
1 for i ← 1 to N do
2     B_i ← candidate paths for covering region i;
3     c_i ← centroid of region i;
  /* Initialize the tour */
4 o ← order in which regions and the depot are covered or
    visited, where o(0) = 0;
5 τ ← FindTour(o, v_0, {c_i}, {B_i});
  /* Improve the tour */
6 τ* ← ImproveTour(τ, o, v_0, {c_i}, {B_i});
7 J* ← g(τ);
8 return τ*, J*
```

Algorithm 2: FindTour($\boldsymbol{o}$, v_0, $\{c_i\}$, $\{\boldsymbol{B}_i\}$)

```
τ ← {v_0};
for t ← 1 to N do
    i ← o(t);
    if t < N then
        j ← o(t + 1);
    else
        j ← 0;
    b_i ← coverage path for region i that minimizes
        g(b_i) + d(τ(end), b_i1) + d(b_in_i, c_j), where τ(end)
        is the last location in path τ, c_0 = v_0 and b_i ∈ B_i;
    τ ← {τ, b_i};
τ ← {τ, v_0};
return τ
```

Algorithm 3: ImproveTour($\boldsymbol{\tau}$, $\boldsymbol{o}$, v_0, $\{c_i\}$, $\{\boldsymbol{B}_i\}$)

```
cost ← ∞;
while g(τ) < cost do
    cost ← g(τ);
    for i ← 0 to N − 1 do
        for j ← i + 2 to N + 1 do
            if d(c_o(i), c_o(i+1)) > d(c_o(i), c_o(j)) then
                o' ← o with links (o(i), o(i + 1)) and
                  (o(j), o(j + 1)) replaced with
                  (o(i), o(j)) and (o(i + 1), o(j + 1)),
                  respectively;
                τ' ← FindTour(o', v_0, {c_i}, {B_i});
                if g(τ') < g(τ) then
                    τ ← τ';
return τ
```

IV. A Heuristic Approach for TSP-CPP with Energy Constraint

Similar as the Fast NN-2Opt algorithm, the proposed heuristic approach solves the constrained TSP-CPP problem also through two phases: *tour initialization* and *tour improvement*. In particular, the first phase determines the minimum number of sub-tours and the regions to be covered in each sub-tour. It also finds a feasible path to fully cover the regions in each sub-tour. The second phase performs a local optimization to improve the sub-tours. Algorithm 4 summarizes the procedures of the proposed algorithm. Next, let's describe each phase in more detail.

Algorithm 4: Proposed Heuristic Algorithm

Input: v_0, vertices of N regions, sensing range of the robot, K, D
Output: tour $\boldsymbol{\tau}^*$, cost J^*
1 **for** $i \leftarrow 1$ **to** N **do**
2     $\boldsymbol{B}_i \leftarrow$ candidate paths for covering region i;
3     $c_i \leftarrow$ centroid of region i;
 /* Initialize the tour */
4 **for** $k \leftarrow 1$ **to** N **do**
5     $\boldsymbol{\tau}_k \leftarrow \emptyset$; $p \leftarrow k$; $U \leftarrow [N]$;
6     $C \leftarrow$ indices of the top K regions that are closest to region p;
7     $\boldsymbol{o}_s \leftarrow$ order in which regions in set C and the depot are covered or visited;
8     $\boldsymbol{\tau}_s \leftarrow$ FINDTOUR$(\boldsymbol{o}_s, v_0, \{c_i\}, \{\boldsymbol{B}_i\})$;
9   **repeat**
10   **if** $g(\boldsymbol{\tau}_s) > D$ **then**
11   Remove regions in set C that are farthest from region p one by one until $g(\boldsymbol{\tau}_s) < D$;
12   **else**
13   Move regions that are closest to region p from set $U \setminus C$ to set C one by one until no more regions can be added to keep $g(\boldsymbol{\tau}_s) < D$ satisfied;
14         $U \leftarrow U \setminus C$;
15         $j \leftarrow$ index of the region closest to region p, where $j \in U$;
16         $C \leftarrow$ indices of the top $\min\{K, |U|\}$ regions that are closest to region j, where $C \subseteq U$;
17         $p \leftarrow j$;
18         $\boldsymbol{\tau}_k \leftarrow \{\boldsymbol{\tau}_k, \boldsymbol{\tau}_s\}$;
19   **until** $U = \emptyset$;
20     $\boldsymbol{\tau} \leftarrow \arg\min_{\boldsymbol{\tau}_k} g(\boldsymbol{\tau}_k)$;
 /* Improve the tour */
21 **foreach** *sub-tour* $\boldsymbol{\tau}_s$ *in* $\boldsymbol{\tau}$ **do**
22     $\boldsymbol{\tau}^* \leftarrow \emptyset$;
23     $\boldsymbol{\tau}_s^* \leftarrow$ IMPROVETOUR$(\boldsymbol{\tau}_s, \boldsymbol{o}_s, v_0, \{c_i\}, \{\boldsymbol{B}_i\})$;
24     $\boldsymbol{\tau}^* \leftarrow \{\boldsymbol{\tau}^*, \boldsymbol{\tau}_s\}$;
25 $J^* \leftarrow g(\boldsymbol{\tau}^*)$;
26 **return** $\boldsymbol{\tau}^*$, J^*

A. Tour Initialization

The tour initialization phase aims to achieve three goals: 1) determine the minimum number of sub-tours m; 2) determine the regions to be covered in each sub-tour; and 3) find a feasible tour efficiently.

To achieve the first two goals, we adopt the rules of stepwise selection [22] and nearest neighbor [23] techniques. The key idea is to maximize the number of regions covered by each sub-tour and minimize the cost of each sub-tour by grouping regions that are closely located. To realize this idea, we perform a greedy search to find the sub-tours one by one, and apply the stepwise selection rule to determine the maximum number of regions that can be covered in each sub-tour. In particular, starting from a randomly selected region p, we first find the top $K \in [N]$ regions (including region p) that are closest to region p, denoted as set C (Line 8). We then generate a feasible sub-tour to cover the K regions in set C, with depot as both the start and end location (Line 9-10). If the cost of the sub-tour exceeds the energy capacity of the robot, we exclude regions in set C that are farthest from region p one by one, until the energy constraint is satisfied (Line 12-13). Otherwise, we add regions in set $U \setminus C$ that are closest to region p into set C one by one, if the inclusion does not violate the energy constraint of the robot, where U is the set of regions that haven't been assigned to any sub-tours (Line 15). The resulting set C then includes all regions to be covered in the first sub-tour. Repeating all above steps, we can then find the other sub-tours. As different starting points (value of p) may lead to different region assignments, we vary the starting points (Line 4) and search for the best region assignment (Line 22).

To measure the closeness between two regions, i and j, we adopt the *cosine similarity*: $\cos(\theta)$, where $\theta = \frac{(c_i - v_0)\cdot(c_j - v_0)}{\|c_i - v_0\|\|c_j - v_0\|}$ is the angle between vector $c_i - v_0$ and vector $c_j - v_0$, and c_i is the centroid of region i. $\|v\|$ measures the Euclidean norm of vector v and $\cdot$ is the dot product. This metric is chosen because that, with each sub-tour starting and ending at the depot, assigning regions of similar orientations with respect to the depot into the same sub-tour is expected to result in shorter tours. We will demonstrate the effectiveness of this metric in the simulation study.

To approximate the cost for covering a set of regions and achieve the third goal, we apply the NN-based algorithm, the one used in the first phase of the Fast NN-2Opt algorithm (Line 4-5 in Algorithm 1).

B. Tour Improvement

Although the NN-based algorithm used in the tour initialization phase is efficient, it ignores the correlation between region visiting order and intra-regional paths and hence may generate low-quality tours. The tour improvement phase further improves the tour by applying the IMPROVETOUR() function (Algorithm 3), which is also used in the second phase of the Fast NN-2Opt algorithm.

V. SIMULATION STUDIES

In this section, we conduct a series of simulation studies to evaluate the performance of the proposed algorithm from various aspects.

A. Experiment Settings

As the problem considered in this paper hasn't been studied in the literature, there are no existing approaches we can directly use as benchmarks. To evaluate the performance of the proposed algorithm, we implement the following three methods alternative to the proposed algorithm as the benchmarks.

- **Benchmark Method 1:** This method uses the Euclidean distance, instead of cosine similarity metric, to measure the closeness between two regions.
- **Benchmark Method 2:** This method adopts the procedure of forward selection to determine the sub-tours,

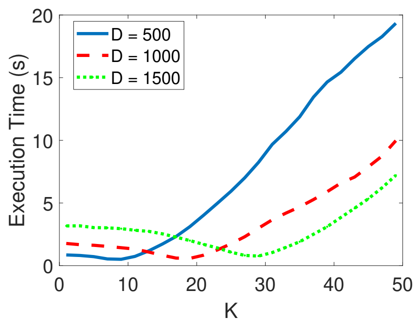


Fig. 1: The mean execution time of the proposed algorithm at different values of K in the three scenarios.

which is a special case of the proposed algorithm with $K = 1$.

- **Benchmark Method 3:** This method directly applies the Fast NN-2Opt algorithm (Algorithm 1), instead of the NN-based algorithm, to generate sub-tours during the search. As tour improvement is performed at each iteration, the tour improvement phase in the proposed algorithm is not implemented in this method.

In all experiments, we consider $N = 50$ randomly generated convex polygonal regions spatially distributed around the depot. The sensing range of the robot is set to 1.5×3. To better understand the performance of above methods, we vary the energy capacity of the robot and consider three scenarios: $D = 500$, $D = 1000$, and $D = 1500$. To reduce experimental uncertainties, we repeat each experiment for 10 times, and measure the mean execution time of each method and the mean cost of the derived tour.

B. Parameter Impact Analysis

In the proposed algorithm, there is a parameter $K \in [N]$ that needs to be configured. To understand the impact of this parameter, we vary its value from 1 to $N = 50$. As K mainly impacts the efficiency, we show in Figure 1 the mean execution time of the proposed algorithm at different values of K. Note that the optimal K that leads to the highest efficiency appears at a larger value when the energy capacity of the robot increases. This is because K directly determines the number of iterations required to find the maximum number of regions that can be covered in a sub-tour, and no iterations will be executed when K equals to the number of regions allowed in a sub-tour. As larger energy capacity permits the robot to cover more regions in a sub-tour, the optimal value of K increases correspondingly. This analysis provides us with guidelines for selecting a proper value of K. In the following experiments, we set K to 8, 16 and 24 in the three scenarios, respectively.

C. Optimality Study

Figure 2 shows the mean cost of the tour generated by each method in different scenarios. We also selectively visualize the tour generated by each method when $D = 500$ in Figure 3. All methods find the same number of sub-tours in the three scenarios, in particular $m = 7$, $m = 3$ and $m = 2$, respectively. One exception is that the Benchmark Method 3 only finds $m = 6$ sub-tours when $D = 500$, as shown in Figure 3.

Comparing the four methods, we can see that the Benchmark Method 3 achieves the best performance in terms of optimality, as it generates shorter sub-tours during the search and better estimates the minimum cost required to cover a particular set of regions. However, it quickly loses its advantage as the robot's energy capacity increases, and the performance of the proposed algorithm is comparable to it for large D. Additionally, Benchmark Method 1 has the worst performance, evidencing the effectiveness of the cosine similarity metric. Benchmark Method 2 achieves a similar performance as the proposed algorithm, as they only differ in the value of K, which mainly impacts the efficiency.

D. Efficiency Study

In this study, we demonstrate the efficiency of the proposed algorithm. As shown in Figure 4, the proposed algorithm achieves the highest efficiency. Benchmark Method 3 is the most time-consuming method, though it can generate shorter tours. Moreover, its execution time rises quickly as D increases. This is because higher energy capacity allows more regions to be covered in each sub-tour, and the efficiency of the Fast NN-2Opt degrades more quickly than the NN-based algorithm as the number of regions increases.

Benchmark Method 2 is also less efficient than the proposed algorithm, as it requires more iterations to find the maximum number of regions allowed in a sub-tour. Its performance degrades with the increase of the value D, as higher energy capacity allows more regions to be covered in each sub-tour. The efficiency of the Benchmark Method 1 is similar as the proposed algorithm, as they only differ in the similarity metric used.

VI. Conclusion

This paper addresses a new path planning problem that seeks the optimal tour to fully cover multiple non-overlapping convex polygonal regions, while satisfying the energy constraint of the robot. This problem arises in many robotic, especially UAV-based, applications and has received

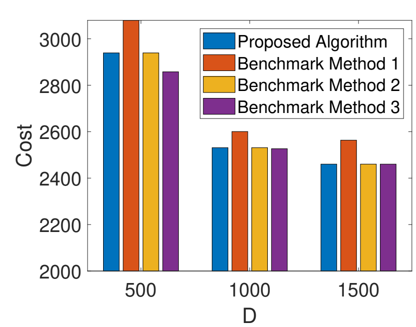
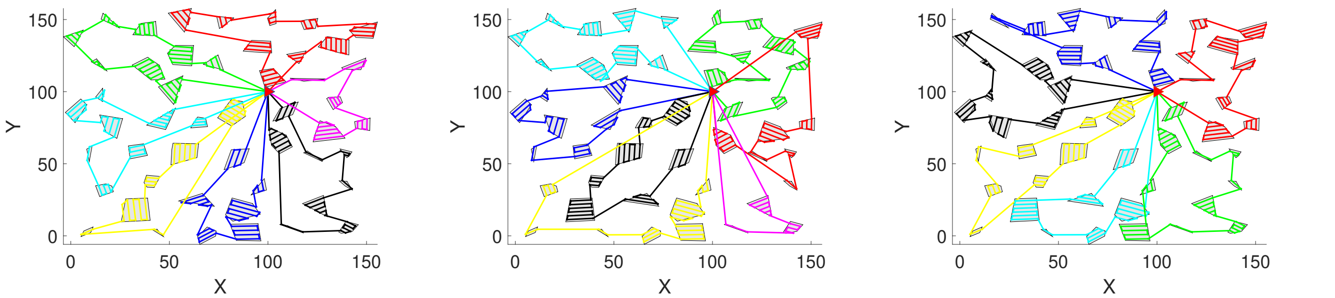


Fig. 2: The mean costs of the tours generated by different methods at different values of D.



(a) (b) (c)

Fig. 3: Tour generated by the a) proposal algorithm and Benchmark Method 2, b) Benchmark Method 1, c) Benchmark Method 3. The grey polygons and the red triangle represent regions and depot, respectively. Sub-tours are differentiated by different colors.

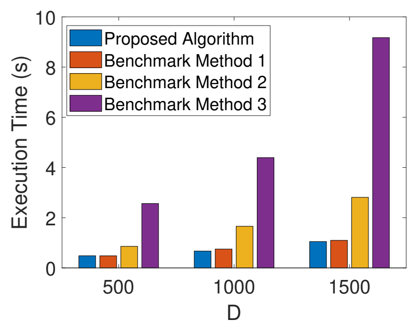


Fig. 4: The mean execution times of different methods at different values of D.

increasing attentions from the researchers. To address this problem, a mathematical formulation was first provided. A stepwise selection based heuristic algorithm was then developed, which achieves a good trade off between optimality and efficiency. Comprehensive simulation studies demonstrated the good performance of the proposed algorithm. In the future, we will extend this study to consider more complicated scenarios, such as the existence of more than one depots.

References

[1] Z. L. Cao, Y. Huang, and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *Journal of Robotic systems*, vol. 5, no. 2, pp. 87–102, 1988.

[2] F. Yasutomi, M. Yamada, and K. Tsukamoto, "Cleaning robot control," in *Proc. of IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1839–1841.

[3] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Uavs for smart cities: Opportunities and challenges," in *Proc. of International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 267–273.

[4] 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.

[5] L. Chen, W.-C. Chiang, R. Russell, J. Chen, and D. Sun, "The probabilistic vehicle routing problem with service guarantees," *Transportation Research Part E: Logistics and Transportation Review*, vol. 111, pp. 149–164, 2018.

[6] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992.

[7] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, no. 1, pp. 60–100, 1991.

[8] C. Chauhan, R. Gupta, and K. Pathak, "Survey of methods of solving tsp along with its implementation using dynamic programming approach," *International Journal of Computer Applications*, vol. 52, no. 4, 2012.

[9] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local Search in Combinatorial Optimization*, vol. 1, no. 1, pp. 215–310, 1997.

[10] M. Dror, A. Efrat, A. Lubiw, and J. S. Mitchell, "Touring a sequence of polygons," in *Proc. of the thirty-fifth annual ACM symposium on Theory of computing*, 2003, pp. 473–482.

[11] C. Wei-Pang and S. Ntafos, "The zookeeper route problem," *Information Sciences*, vol. 63, no. 3, pp. 245–259, 1992.

[12] X. Tan and T. Hirata, "Finding shortest safari routes in simple polygons," *Information processing letters*, vol. 87, no. 4, pp. 179–186, 2003.

[13] J. Xie, L. R. G. Carrillo, and L. Jin, "An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems," *IEEE control systems letters*, vol. 3, no. 1, pp. 67–72, 2018.

[14] J. Xie, L. Jin, and L. R. Garcia Carrillo, "Optimal path planning for unmanned aerial systems to cover multiple regions," in *Proc. of AIAA Scitech 2019 Forum*, 2019, p. 1794.

[15] J. Xie, L. R. G. Carrillo, and L. Jin, "Path planning for uav to cover multiple separated convex polygonal regions," *submitted to IEEE Access*, 2020.

[16] J. I. Vasquez-Gomez, J.-C. Herrera-Lozada, and M. Olguin-Carbajal, "Coverage path planning for surveying disjoint areas," in *Proc. of International Conference on Unmanned Aircraft Systems*. IEEE, 2018, pp. 899–904.

[17] J. Chen, C. Du, X. Lu, and K. Chen, "Multi-region coverage path planning for heterogeneous unmanned aerial vehicles systems," in *Proc. of IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2019, pp. 356–3565.

[18] L. Ma, L. Cheng, W. Han, L. Zhong, and M. Li, "Cultivated land information extraction from high-resolution unmanned aerial vehicle imagery data," *Journal of Applied Remote Sensing*, vol. 8, no. 1, p. 083673, 2014.

[19] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[20] B. Xin, G.-Q. Gao, Y.-L. Ding, Y.-G. Zhu, and H. Fang, "Distributed multi-robot motion planning for cooperative multi-area coverage," in *2017 13th IEEE International Conference on Control & Automation (ICCA)*. IEEE, 2017, pp. 361–366.

[21] J. Xie, W. Zhang, and J. Chen, "Path planning for multiple energy constrained unmanned aerial vehicles to cover multiple regions," in *AIAA AVIATION 2020 FORUM*, 2020, p. 2887.

[22] B. Ratner, "Variable selection methods in regression: Ignorable problem, outing notable solution," *Journal of Targeting, Measurement and Analysis for Marketing*, vol. 18, no. 1, pp. 65–75, 2010.

[23] N. Bhatia *et al.*, "Survey of nearest neighbor techniques," *arXiv preprint arXiv:1007.0085*, 2010.