

Article

# A Revisiting Method Using a Covariance Traveling Salesman Problem Algorithm for Landmark-Based Simultaneous Localization and Mapping

Hyejeong Ryu 

Department of Mechatronics Engineering, Kangwon National University, Chuncheon KR24341, Korea; hjryu@kangwon.ac.kr; Tel.: +82-33-250-6376

Received: 15 October 2019; Accepted: 7 November 2019; Published: 10 November 2019



**Abstract:** This paper presents an efficient revisiting algorithm for landmark-based simultaneous localization and mapping (SLAM). To reduce SLAM uncertainty in terms of a robot's pose and landmark positions, the method autonomously evaluates valuable landmarks for the data associations in the SLAM algorithm and selects positions to revisit by considering both landmark visibility and sensor measurement uncertainty. The optimal path among the selected positions is obtained by applying the traveling salesman problem (TSP) algorithm. To plan a path that reduces overall uncertainty, the cost matrix associated with the change in covariance between all selected positions of all pairs is applied for the TSP algorithm. From simulations, it is verified that the proposed method efficiently reduces and maintains SLAM uncertainty at the low level compared to the backtracking method.

**Keywords:** mobile robot; simultaneous localization and mapping; traveling salesman problem; path planning; loop-closing; exploration

---

## 1. Introduction

Environmental mapping is essential when mobile robots are used to perform higher-level tasks. Many researchers have explored how to present the environment and localize a robot using sensor data. Simultaneous localization and mapping (SLAM) is used to simultaneously estimate the positions of a robot and landmarks from noisy sensor data acquired as the robot moves [1,2]. Many SLAM algorithms focus on the accuracy of estimated states; various filters and frameworks have been applied to bound the associated uncertainties.

Feature- and landmark-based SLAM approaches usually employ extended Kalman filters (EKF) to estimate the status of the robot and landmarks and employ a single Gaussian for each state. In such approaches, line or point landmarks are extracted from data derived by laser range finders, sonar sensors, and visual landmarks in images [3,4]. The performance of SLAM algorithms depends on process and measurement noise covariance matrices. To obtain accurate measurement noise covariance matrices without the trial and error methods of EKF-SLAM algorithms, an adaptive neuro-fuzzy inference system was proposed [5]. To reduce the computational complexity of EKF-SLAM, which is affected by the number of landmarks, local submaps are constructed and they are integrated into a global map [6]. Microphone observations can be applied to landmark-based EKF-SLAM algorithms [7]. The time delay estimation between sound sources and microphones by generalized cross-correlation was used to establish the observation model.

Rao-Blackwellized particle filters (RBPFs) are appropriate for landmark-based maps [8], and also efficiently optimize gridmaps and estimate robot poses [9,10]. Radio frequency identification (RFID) phase measurements were employed to estimate the robot pose and passive tag coordinates by using

the landmark-based RBPF SLAM approach [11]. Whereas most previous gridmap-based RBPF SLAM algorithms have modeled each cell as one Bernoulli distribution [12,13], the occupancy probability of each cell is modeled by using a beta-distribution in  $\beta$ -SLAM to distinguish between different causes of uncertainty [14]. It also presents an uncertainty measure to quantify the resulting gridmap by the Shannon entropy for path planning.

Recently, graph-SLAM algorithms have found many applications in large-scale environments. Graph-SLAM [15] optimizes the robot pose graph using constraints that connect the pose nodes via sensor observations. The sensor observations of the graph-SLAM algorithms can be the scan-matching results between laser point clouds [16,17] and the feature-matching/tracking results between keyframe images [18,19]. Keyframe-based graph-SLAM was applied not only to a single robot, but also to multi-agent systems [20]. The graph-SLAM algorithm performed by multiple robots can overcome the inefficiency caused by the growing graph size in large environments. To increase the performance of the graph-SLAM algorithm for low-cost Light Detection And Ranging (LiDAR) sensors and vision sensors, laser scan-matching results and visual bag of words (BoW) were combined [21].

When using SLAM algorithms, loop-closing (i.e., revisiting a previously mapped/detected area) is essential to bound global errors and to improve global consistency. Revisiting minimizes SLAM uncertainty in terms of both the robot pose and landmark positions [22,23]. In general, when a robot returns to the initial position (the startpoint of the SLAM algorithm), a loop trajectory is naturally completed and the SLAM state is optimized by applying the loop-closing constraint [24,25]. This is because, when the SLAM algorithm commenced, the robot pose uncertainty was very low and landmarks were accurately registered. Robot and landmark uncertainties become large when the robot has traveled a long distance, necessitating return of the robot to a better-known area to re-observe accurately estimated landmarks; this increases the likelihood that data associations will be correct. Most loop-closing strategies have focused on detection of loop-closing landmarks and loop optimization [21,26,27]. Several approaches simplify loop-detection issues using special landmarks [11,28]. Appropriate loop-closing positions and planning of loop-closing paths have received less attention.

When a SLAM algorithm must reduce state uncertainties via loop-closing, the robot can easily return to the initial position if it is not remote from that position. However, if the robot is distant from the initial, accurately mapped area, the robot has to backtrack along the previous trajectory or plan a revisiting path that takes the robot in a direction in which uncertainty decreases. It is difficult to determine the extent to which uncertainty is reduced during backtracking, because the same situation inducing growing uncertainty is repeated along the backtracking path.

In this paper, the revisiting method using a solution of the traveling salesman problem (TSP) to devise a landmark-based EKF SLAM is proposed. Given the uncertainties in the positions of SLAM-registered landmarks when revisiting is to commence, we construct a visible landmark uncertainty map and select several revisiting positions to ensure that the obtained measurements adequately reduce uncertainty. The traveling order among the selected positions is chosen via TSP optimization using the predicted covariance changes. We show that robot and landmark uncertainties are reduced more efficiently when the robot travels along such a path (compared to backtracking).

This paper is organized as follows. In Section 2, we propose the visible landmark uncertainty grid map that is used to identify landmarks that must be re-observed to reduce overall uncertainty and to select appropriate revisiting positions. In Section 3, we present the covariance TSP algorithm to derive the revisiting path that includes all selected positions. Section 4 shows simulation results of both backtracking and our method. Section 5 is the conclusion.

## 2. Visible Landmark Uncertainty Portrayed on a Gridmap

To choose positions at which a robot can make loop-closing observations of landmarks registered by the SLAM algorithm, we construct a visible landmark uncertainty (VLU) map that shows the uncertainties of visible or observable landmarks at each grid position.

### 2.1. The VLU Map

The VLU map is a gridmap; the value of each grid is the extent of uncertainty associated with landmark detection at each grid location. We use the covariance matrix of the EKF-SLAM to represent the uncertainty of the SLAM state. This includes correlations between interrelated landmarks. Every observation of a landmark affects the estimation of every other landmark [1]. Therefore, it is optimal to proceed to a position at which the maximum number of possibly known or registered landmarks is visible to the SLAM algorithm, and where new landmarks are also detectable as exploration proceeds. To build the VLU map, we assume that the robot is equipped with a range sensor with a 360° sweep but a defined maximum range. We first calculate the number of visible landmarks in each grid cell,  $G_{i,j}$ :

$$G_{i,j} = \sum_{n=1}^N v_n \quad (1)$$

$$v_n = \begin{cases} 1, & \text{if } \sqrt{\{g_{i,j}(x) - m_n(x)\}^2 + \{g_{i,j}(y) - m_n(y)\}^2} < r_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where  $N$  is the total number of landmarks registered to the SLAM state,  $g_{i,j}(x)$  and  $g_{i,j}(y)$  are the metric coordinates  $x$  and  $y$  of the grid index  $(i, j)$ ,  $m_n(x)$  and  $m_n(y)$  are the metric coordinates of the  $n$ -th landmark  $m_n$ , and  $r_{\max}$  is the maximum range of the sensor. After calculating the number of visible landmarks, the VLU of each grid cell is derived as follows:

$$VLU_{i,j} = (N - G_{i,j}) tr_{\max} + \sum_{k=1}^{G_{i,j}} trace(P_k) \quad (3)$$

$$tr_{\max} = \max \{trace(P_k) : k = 1, \dots, N\} \quad (4)$$

where  $N$  is the total number of landmarks registered to the SLAM state,  $G_{i,j}$  is the number of visible landmarks at grid index  $(i, j)$ ,  $P_k$  is the covariance matrix of the visible landmark  $m_k$ , and  $trace(P_k)$  is the trace of the  $P_k$  matrix.  $P_k$  is obtained from the SLAM covariance matrix;  $P_k$  is a sub-block of the entire covariance matrix.  $tr_{\max}$  is the maximum trace (derived by reference to the most uncertain landmark among all registered landmarks).

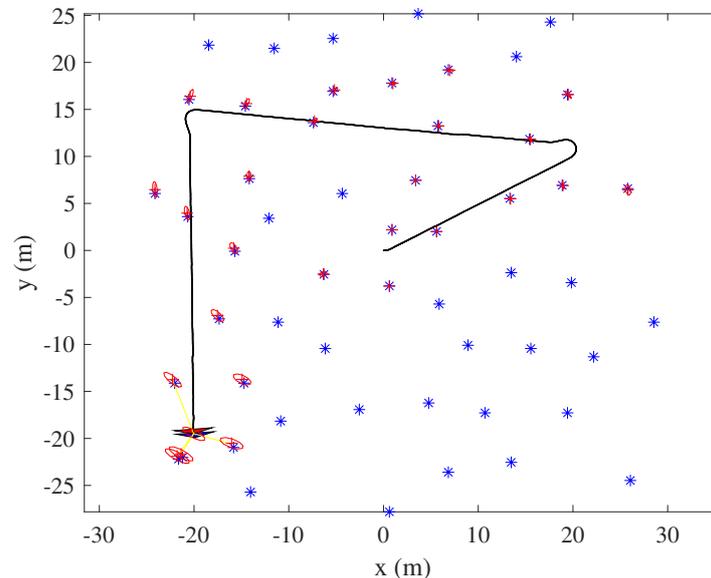
The Shannon entropy (the Shannon information measure) was used in [29] to obtain a scalar reflecting the diversity of a probability distribution represented by a covariance matrix. The entropy  $H$  of a Gaussian SLAM state is obtained by calculating the determinant of the covariance [30]. Building on this concept, the authors of [31] proposed the use of an *a-optimal* measure of information gain to increase map quality during exploration. The *a-optimal* measure is calculated using the trace of the covariance matrix, thus the product of the eigenvalues. When moving in a direction that optimizes the *a-optimal* measure, a more accurate map is obtained (the mean squared errors of the landmarks are minimized).

In Equation (3), we apply this trace of the state covariance matrix to the visible landmarks. The first term  $(N - G_{i,j})$  reflects the number of invisible landmarks, and the second term reflects the uncertainties of the visible landmarks. The maximum trace for each invisible landmark is added, and each visible landmark assumes its corresponding trace.

The higher the grid cell VLU value, the greater the SLAM uncertainty; fewer landmarks with large covariances are visible. The lower the grid cell VLU, the better the robot position; covariance is reduced via correct association of data with visible landmarks.

Figure 1 shows a sample EKF-SLAM state (of both the robot and the landmarks) when uncertainty has increased after traveling a long distance. The initial robot location was (0, 0 m). The estimated

positions of landmarks that were observed early exhibit low uncertainties (represented by the volumes of the ellipses). We can find that landmark uncertainty increases as the robot travels. Revisiting is necessary to enhance the positional accuracies of both registered landmarks and the robot; the VLU grid map is obtained using the landmark state.

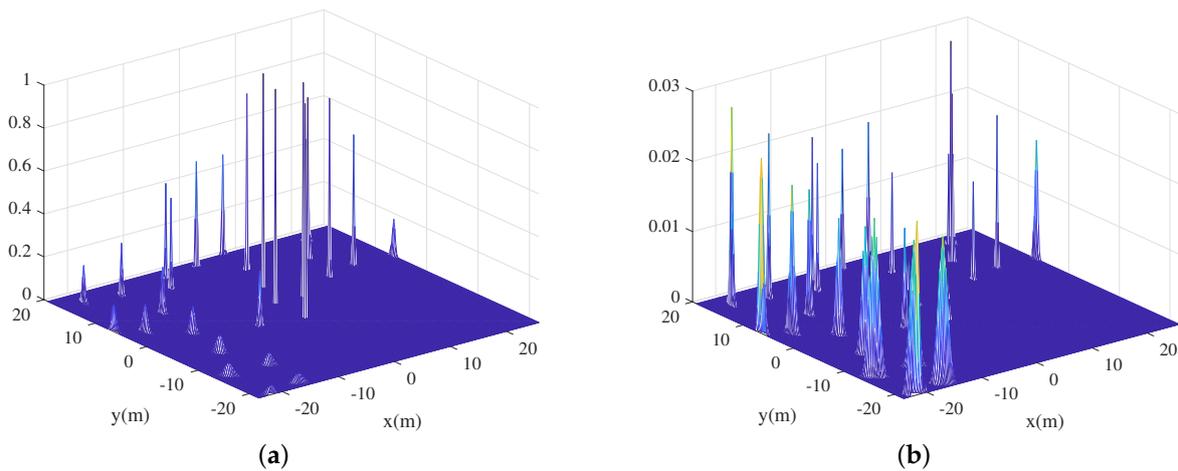


**Figure 1.** An example of extended Kalman filter (EKF)-simultaneous localization and mapping (SLAM) robot/landmark state when revisiting commences. The red ellipses reflect the covariances of registered landmarks; the red stars show the estimated positions of such landmarks; and the blue stars indicate the true positions of unregistered (unmapped) landmarks.

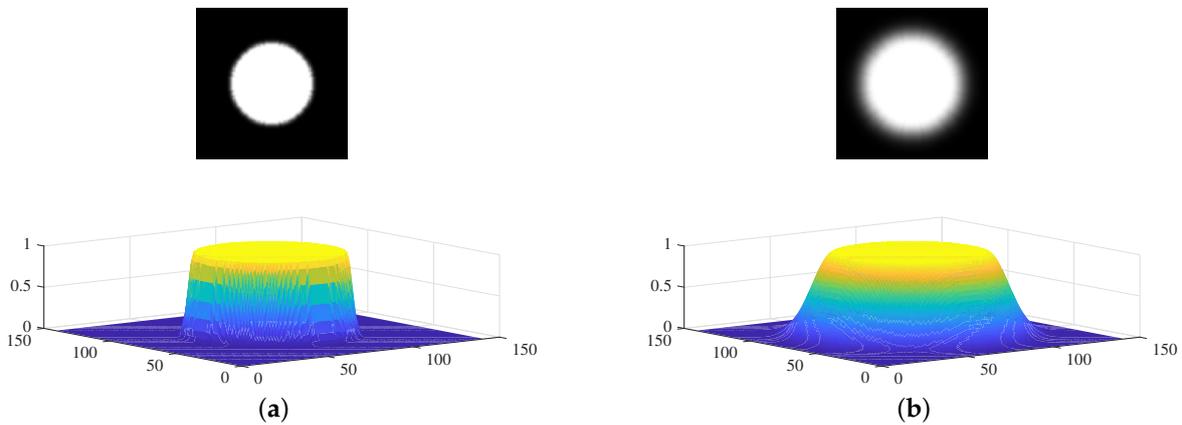
### Visibility Convolution

We now present an efficient method to calculate Equation (2) using an image convolution process. In Equation (2), the visibility of the  $n$ -th landmark from each grid cell is the estimated mean value of the  $n$ -th landmark  $[m_n(x), m_n(y)]$ . We improve the calculation of Equation (2) using the Gaussian distribution of each landmark and an efficient image convolution method that reflects uncertainties in the estimated landmark states. After initializing the gridmap using the size of the currently mapped area, a Gaussian distribution is inserted at each registered landmark position using both the mean value and the covariance matrix. Figure 2a shows the Gaussian distribution of each landmark depicted in Figure 1. Next, each distribution is multiplied by the trace value of its covariance matrix. Figure 2b shows the trace-multiplied Gaussian distribution map.

The two distribution gridmaps are subjected to simple image convolution. The shape of the kernel depends on the working distance, the field of view, and the sensor accuracy (the kernel is the “measurement model” of an EKF-SLAM). Figure 3 shows two kernels used to calculate convolutions. We assume that the robot features a  $360^\circ$  field of view sensor; the kernel can thus be circular. If the robot features a  $120^\circ$  sensor, the kernel can be fan-shaped. Within the maximum range, the sensor uncertainty kernel value is 1 because the sensor can detect the landmarks that are within the maximum range. At the maximum range, near the end of the sensor ray, the kernel value follows a Gaussian distribution that considers errors in sensor measurements. Figure 3a shows a convolution kernel with a small sensor error. Figure 3b shows a kernel reflecting a large sensor error; the deviation is large at the maximum range.

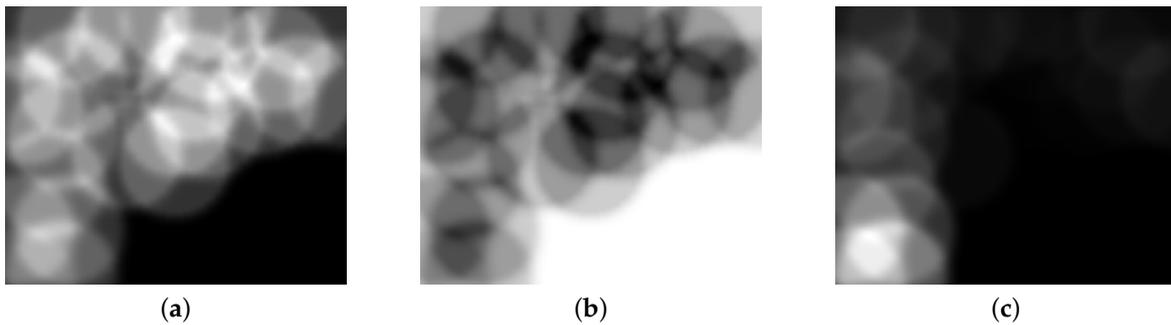


**Figure 2.** The distribution of each landmark on the gridmap: (a) The Gaussian distribution of each landmark. (b) The Gaussian distribution multiplied by the trace value of each landmark.

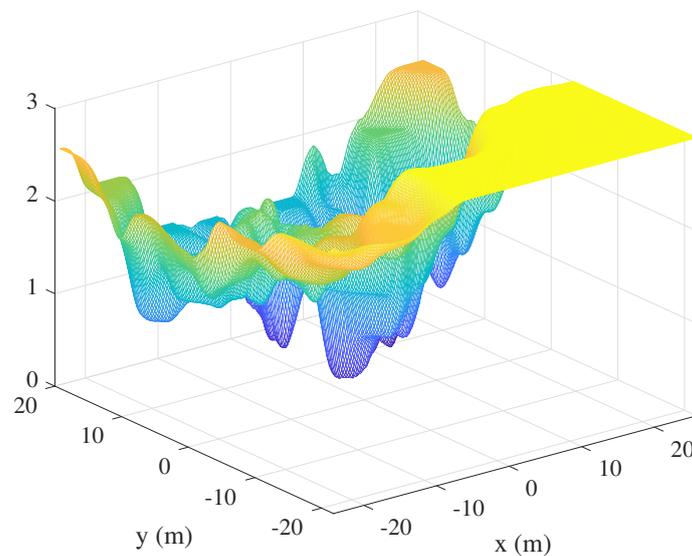


**Figure 3.** The sensor uncertainty kernels used for visibility convolution: (a) A convolution kernel with small sensor errors at the maximum range grid cells. (b) A convolution kernel with large sensor errors at the maximum range grid cells.

Using the Gaussian distribution map of the landmarks, convolution is performed with the sensor uncertainty kernel. This yields the probabilistic number of visible landmarks with consideration of the measurement uncertainty. The visibility convolution of the EKF-SLAM state of Figure 1 is shown in Figure 4. Figure 4a is the convolution between the Gaussian distribution of each landmark in Figure 2a and the sensor uncertainty kernel. Lighter grids indicate that more landmarks are observed at that position. Figure 4b shows the probabilistic numbers of invisible landmarks, obtained by subtracting the probabilistic numbers of visible landmarks from the total number of registered landmarks when revisiting commences. This value is used to calculate the first term of Equation (3) via multiplication by the maximum trace value of all landmarks (thus that of the most uncertain landmark). This assigns higher values to VLU map grid positions at which the robot observes fewer landmarks. Figure 4c shows the convolution between the trace-multiplied Gaussian distribution map of Figure 2b and the sensor uncertainty kernel; this corresponds to the second term of Equation (3). The fewer the number of landmarks detected, and the more uncertain their positions, the lighter the color and the higher the uncertainty. The final VLU map is shown in Figure 5. Overall, the areas in which less uncertain landmarks are visible have lower values and can be considered as revisiting positions to increase the accuracy of landmark positioning and robot pose by correct data association.



**Figure 4.** Visibility convolutions derived using sensor uncertainty kernels: (a) The probabilistic number of visible landmarks. (b) The probabilistic number of invisible landmarks. (c) The results of the convolution between the trace-multiplied Gaussian distribution map and the sensor uncertainty kernel.



**Figure 5.** The visible landmark uncertainty (VLU) gridmap.

## 2.2. Selection of Revisiting Positions Using the VLU Map

The VLU grid map is used to select several revisiting positions. Each position must feature landmarks of low uncertainty to enhance the probability of correct data association. Optimally, the robot will detect not only accurate but also somewhat uncertain landmarks. This may appear to be contradictory. However, this reduces the overall uncertainties of state estimation and increases exploration efficiency; map information must be extended. To select positions at which the robot can achieve these aims, we select the less uncertain registered landmarks. Any selected landmark will be visible from several positions, and these positions have relatively lower VLU values than the positions where the selected landmarks are not visible; the position with the highest VLU value among these several positions is chosen for revisiting.

Algorithm 1 shows how to select revisiting positions using the VLU gridmap. The less uncertain landmarks are identified by Equation (5). A landmark for which the covariance matrix exhibits a trace value greater than the mean of the overall landmark covariance matrix is selected as a re-observation target. Next, for each selected landmark  $l_k$ , we define grid cells from which  $l_k$  is visible (lines 2–9). When there are several grid cells where one selected landmark is visible, the grid cell with the highest VLU is selected as a revisiting position for both data association and exploratory utility. If that cell has been already selected for viewing of another landmark, the cell with the next-highest VLU is selected (11–19). Thus, Algorithm 1 runs until a grid cell with a high VLU that is not assigned to another landmark is chosen for each landmark.

**Algorithm 1** Selecting revisiting locations

**Require:** all grid cell indices  $\{g(i, j)\}$ , VLU gridmap  $\{g_{VLU}(i, j)\}$ , the covariance matrices of all  $N$  registered landmarks  $\mathbf{P} = \{P_1, \dots, P_N\}$

**Ensure:** Revisiting positions  $Loc$

- 1: Select the less uncertain landmarks according to the trace of its covariance matrix

$$\text{Landmark } k, l_k, \text{ with } \text{trace}(P_k) < \frac{1}{N} \sum_{i=1}^N \text{trace}(P_i) \quad (5)$$

$N$  : total number of registered landmarks

$P_i$  : covariance matrix of landmark  $i$

- 2: **for** all selected landmarks,  $l_k$  **do**
- 3:     **for** all grid cells,  $g(i, j)$  **do**
- 4:         **if**  $l_k$  is visible at  $\{g_{i,j}(x, y)\}$  **then**
- 5:             Push  $g(i, j) \rightarrow G_k$
- 6:         **end if**
- 7:     **end for**
- 8:     Push  $G_k \rightarrow \mathbf{G}$
- 9: **end for**
- 10:  $Loc \leftarrow \emptyset$
- 11: **for** all elements,  $G_k$  of  $\mathbf{G}$  **do**
- 12:     Select  $g(i, j)$  which has the highest  $g_{VLU}(i, j)$  among all elements of  $G_k$
- 13:     **if** there is not  $g(i, j)$  in  $Loc$  **then**
- 14:         Push  $g(i, j) \rightarrow Loc$
- 15:     **else**
- 16:         Remove  $g(i, j)$  from  $G_k$
- 17:         **goto** Step 11
- 18:     **end if**
- 19: **end for**

### 3. The Covariance Traveling Salesman Problem

In the previous section, we calculated the VLU of each map position, and used this to select revisiting positions. We employ a solution of the TSP algorithm to define the revisiting order. The TSP is an extensively explored optimization problem that deals with a salesman who requires the shortest path between  $N$  cities or nodes [32]. In terms of acquiring the shortest path, the algorithm can be used to optimize revisiting or to explore robotic frontiers [33]. As the odometry error increases as the robot moves, the shorter the distance traveled to re-detect each landmark, the better the SLAM state estimation.

#### 3.1. Traveling Salesman Problem: A Brief Review

When a salesman must visit several cities, starting from and returning home, he needs to minimize his total travel distance. It is assumed that the salesman knows where all the cities are and the traveling costs between them. The constraints are: (i) each city may be visited only once; and, (ii) the salesman must return home (closing a loop). We view the mobile robot revisiting problem as a two-dimensional

(2D) TSP where the traveling distance is Euclidean or reflects the navigation cost between revisiting positions. The classical formulation of the TSP follows. Let:

$$x_{ij} = \begin{cases} 1 & \text{if the tour uses a leg between } i \text{ and } j \\ 0 & \text{if not} \end{cases} \quad (6)$$

This minimizes:

$$Z = \sum_i \sum_{j>i} c_{ij} x_{ij} \quad (7)$$

when:

$$\sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} = 2, \text{ for all } i \quad (8)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} + \sum_{i \notin S} \sum_{j \in S} x_{ij} \geq 2, \quad (9)$$

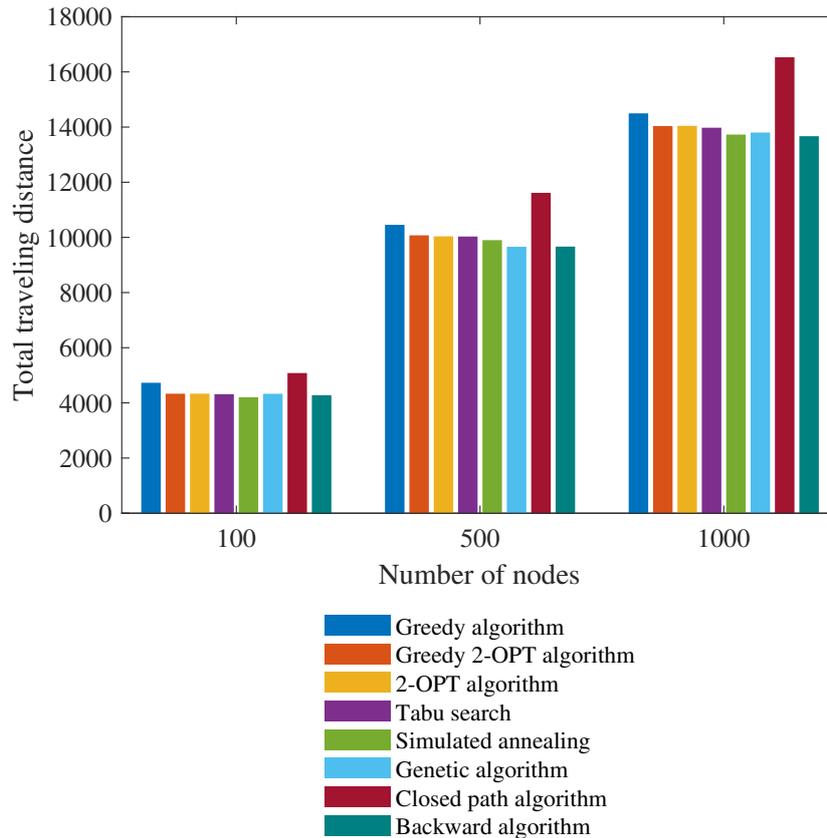
for all proper subset  $S, |S| \geq 3$

$x_{ij} = \text{binary}$ , for all  $i; j > i$ .

Equation (8) deals with constraint (i) and Equation (9) deals with the loop-closing constraint (ii). The TSP is probably the best-known optimization problem but it is unlikely that it can be efficiently solved using a polynomial time algorithm. As all  $O(n!)$  potential permutations of the cities must be examined, computation is difficult when the number of cities is large. Although the TSP cannot be fully solved in polynomial time, a near-optimal solution can be acquired with the aid of heuristics [32,34]. A heuristic algorithm commences by selection of a single city, followed by addition of all remaining cities one-by-one; this is a 'greedy algorithm'. An improved heuristic algorithm is based on the local-search concept, analyzing transitions from one solution to the next in a search for better solutions. We tested various local heuristic algorithms when seeking TSP solutions to mobile robot revisiting. Figure 6 shows the total traveling distances of eight algorithms. The average computational time of seven algorithms (excluding the backward algorithm) was 0.1 s for 100 cities, 4 s for 500 cities, and 7 s for 1000 cities. These values are acceptable when analyzing mobile robot revisiting.

### 3.2. Traveling Costs between Revisiting Positions

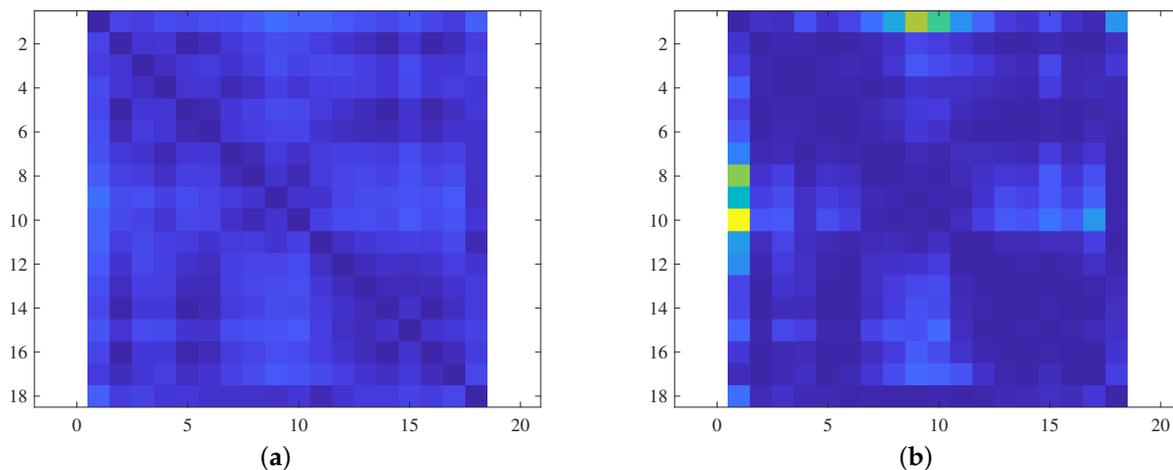
After choosing positions to be revisited, the traveling costs are computed. The positions form the nodes of the TSP and the traveling costs between each pair of positions form the cost matrix of the TSP, thus  $c_{ij}$  of Equation (7). Traveling costs of robotic navigation can be defined in various ways, including the geometric Euclidean distance or the obstacle-free shortest path. If the robot features perfectly accurate sensors (including a perfect wheel odometer), the Euclidean distance can serve as the shortest total revisit distance. However, sensors are never perfect, and SLAM deals with sensor errors. Also, it is important to not only minimize the total travel distance but also to reduce the uncertainty of the overall SLAM state. Such uncertainty depends on both the path and new sensor measurements. We propose the covariance TSP that defines each between-position travel cost as the expected covariance change of the robot after it moves from one revisiting position to another.



**Figure 6.** Comparison of the traveling salesman problem (TSP) solutions using various heuristic algorithms.

SLAM algorithms, especially EKF-SLAM, engage in both prediction and update processes [22]. During the prediction process, the robot state is computed by the motion model and the control inputs. The update process can be performed when sensor observations are available. After an observation is made, the innovation (the difference between the actual and predicted observations as revealed by the observational model) and the robot state evaluated during prediction must be calculated. However, because we are now considering a kind of path planning problem before actual movement, it is impossible to obtain real observations prior to arrival. Therefore, we simulate the observations expected as the robot travels between two revisiting positions. In these simulations, the landmark state is that at the time a revisit is scheduled. The initial covariance of the robot state serves as the zero matrix.

The resulting covariance cost matrix is non-symmetric because the measurements obtained during travel from nodes “A” to “B” and “B” to “A” differ. Although the observed landmarks are the same, the covariance is computed differently because the order of measurements varies. Figure 7 shows examples of cost matrices; Figure 7a is the cost matrix derived using the Euclidean distance between nodes and Figure 7b is the matrix yielded by the expected covariance change. When that expected covariance change serves as the travel cost, we can find a revisiting route that minimizes the covariance change. Therefore, the robot state uncertainty is maintained below a certain, predefined low value during revisiting.



**Figure 7.** Cost matrices of TSP algorithms: (a) A Euclidean cost matrix. (b) A covariance cost matrix.

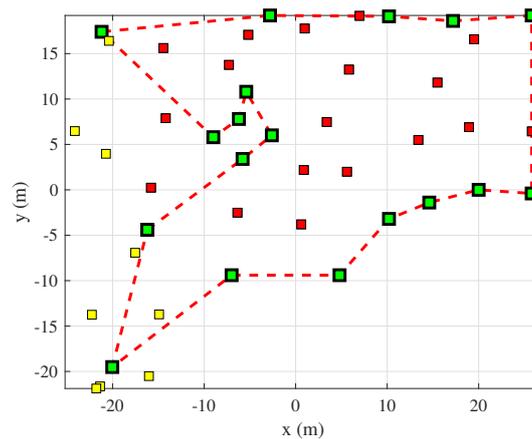
With the covariance cost matrix, a genetic algorithm (GA) was used to solve the TSP in this paper. GA is intuitively very simple and is known to perform very well for small-sized ( $n < 100$ ) TSP instances [35,36]. The proposed covariance cost matrix can be used in combination with any TSP algorithm and GA can be replaced with other heuristics and metaheuristics. The details and benchmark results of leading TSP heuristics such as the Lin–Kernighan (LK) method and the stem-and-cycle (S&C) method can be found in [37]. Metaheuristics have been also widely used to solve TSP and the vehicle routing problem (VRP) [38,39].

#### 4. Simulation Results

We simulated our revisiting method to verify performance in terms of SLAM covariance reduction. We evaluated the robot pose and the positions of registered landmarks. We did not consider the problem of obstacles in the environment. This is the problem of local planners and there are various algorithms to deal with this. The robot’s laser range finder had a scanning range of  $360^\circ$  and a maximum range of 7 m. Robot locomotion featured a differential drive constraining movement within each time step. Using range data and wheel odometry, the EKF-SLAM algorithm estimates the state of the robot and landmarks. The environment is completely unknown before the SLAM algorithm commences.

The initial robot position was (0, 0 m) and the robot moved to (20, 10 m), (−20, 15 m), and (−20, −20 m). On arrival at the third position, the SLAM covariance became large. To reduce uncertainty, the robot revisited the initial position using backtracking or planned the revisiting path by the proposed method. During backtracking, the revisited positions were placed every 2 m along the past trajectory. Using our method, the positions to be revisited were chosen by reference to the VLU map and the covariances of registered landmarks; the revisiting sequence was calculated with the aid of the proposed covariance TSP algorithm with GA.

Figure 8 shows the landmarks selected by, and the revisiting path proposed by, our method. The red and yellow squares are registered landmarks of the EKF-SLAM state at the time revisiting was planned. The yellow squares are non-selected landmarks. The red squares are landmarks selected by Equation (5). These are “less uncertain” landmarks that must be re-observed to reduce the overall SLAM uncertainty. The green squares are the revisiting positions at which the robot can detect the selected landmarks. These positions have low VLU values; non-mapped areas can be scanned to expand map information.



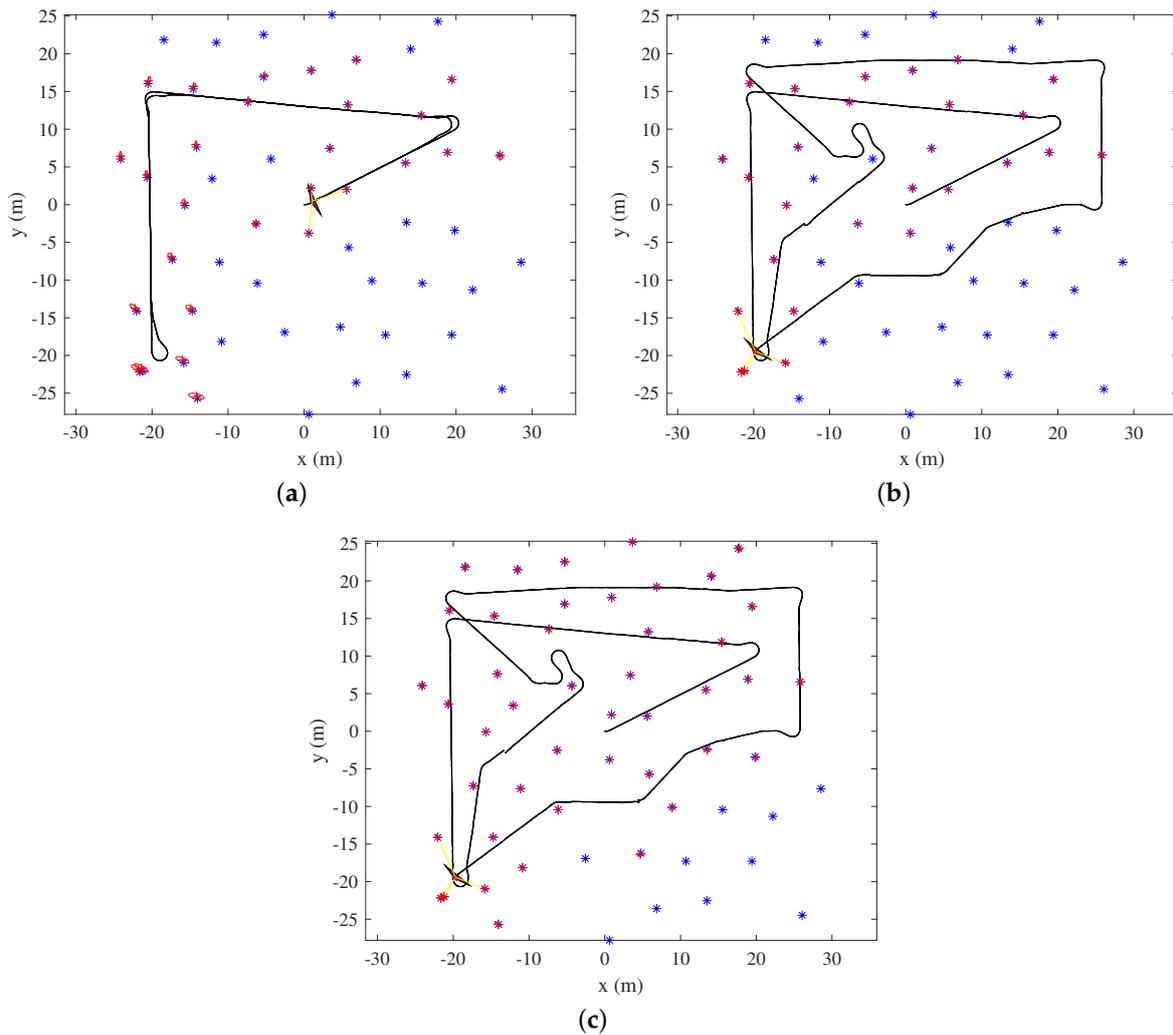
**Figure 8.** A revisiting path constructed using the proposed covariance TSP algorithm. The yellow squares are non-selected landmarks; the red squares are landmarks selected for revisiting; the green squares are the revisiting positions dictated by the VLU grid map; the red dotted line is the optimal revisiting path defined by the covariance TSP algorithm.

Figure 9 shows the final EKF-SLAM state after revisiting either via a backtracking algorithm (Figure 9a) or with the aid of the proposed covariance TSP algorithm. For revisiting by the covariance TSP, two approaches were applied. In the first covariance TSP algorithm, which is denoted by Covariance TSP<sub>fixed</sub>, adding new landmarks to the SLAM state was not permitted during the revisiting process (Figure 9b). The number of registered landmarks was kept the same from the beginning of revisiting to the end. Therefore, the number of landmarks during Covariance TSP<sub>fixed</sub> revisiting was almost the same as the number of landmarks during backtracking. In the second covariance TSP algorithm, Covariance TSP, new landmarks were registered to the SLAM state when the robot obtained new observations (Figure 9c) during revisiting. The volume of the ellipse at each landmark reflects the uncertainty. As the robot obtained almost the same measurements for each registered landmark during backtracking, the uncertainties of landmarks near the position at which revisiting commenced are larger than those of early-registered landmarks. Using the proposed method, the robot moved along a closed TSP path and obtained different measurements for the re-observed landmarks and the newly detected landmarks that were able to make the correct data associations for the SLAM algorithm. Consequently, the overall uncertainty of the landmark was effectively reduced.

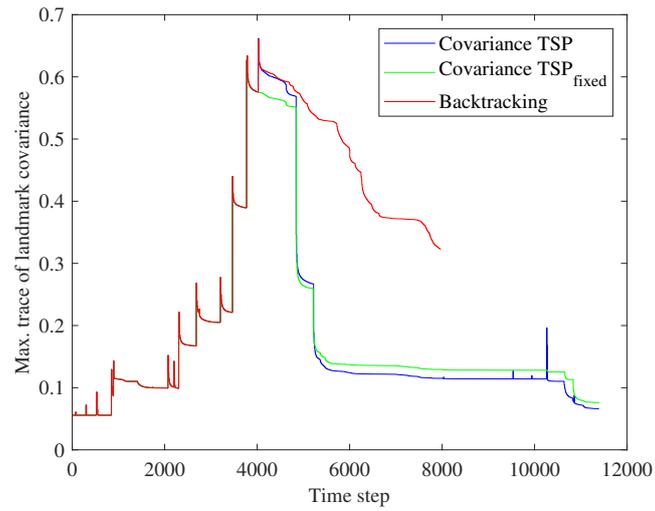
Figures 10 and 11 compare the results of backtracking and use of the proposed covariance TSP algorithms. Revisiting commenced at step 3972. Figure 10a shows the maximum trace value among all landmarks at each time step, and we can compare the uncertainty of the most uncertain landmark identified by backtracking and the covariance TSP algorithms from SLAM commencement to completion of revisiting. As the landmark state was identical for three algorithms until revisiting commenced, the maximum trace values are also the same. After revisiting began, we explored how effectively the three methods reduced landmark uncertainty. As the overall distance traveled during backtracking was shorter than that of covariance TSP revisiting, the robot stopped the algorithm at step 7975 of backtracking and step 11,402 of Covariance TSP and Covariance TSP<sub>fixed</sub>. The ultimate, maximum trace values of Covariance TSP and Covariance TSP<sub>fixed</sub> are much smaller than that of backtracking; our algorithm thus more effectively reduced uncertainty. Moreover, even at the end of backtracking (step 7975), the maximum trace values of our algorithms were smaller because they fell dramatically over time. At about step 10,000, the trace of Covariance TSP rose but then immediately decreased. This was because the robot obtained not only measurements of the landmarks registered when planning the revisiting path but also measurements of new landmarks that were relatively uncertain in terms of SLAM registration. These new landmarks became less uncertain via the EKF-SLAM updating process. As loop-closing by re-observing the already registered landmarks occurred and any new landmarks were not added in Covariance TSP<sub>fixed</sub> revisiting, the maximum

trace consistently decreased until completion of revisiting. However, because the number of landmarks increased and it affected the overall uncertainty in Covariance TSP, the maximum trace was smaller than that of Covariance TSP<sub>fixed</sub>.

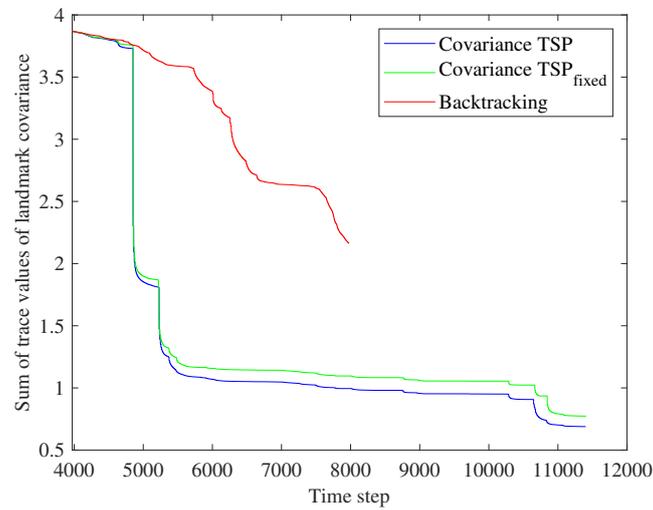
In Figure 10b, we also compared the sum of the trace values of the landmarks registered to the SLAM state at the time when revisiting was planned, step 3972. The proposed algorithms outperformed backtracking in terms of reducing and maintaining landmark uncertainty. Although the proposed revisiting path was longer than that of backtracking, our methods reduced the overall landmark uncertainty earlier than backtracking. Therefore, it is possible to cease revisiting in the middle of the planned path if the overall uncertainty falls to the desired value.



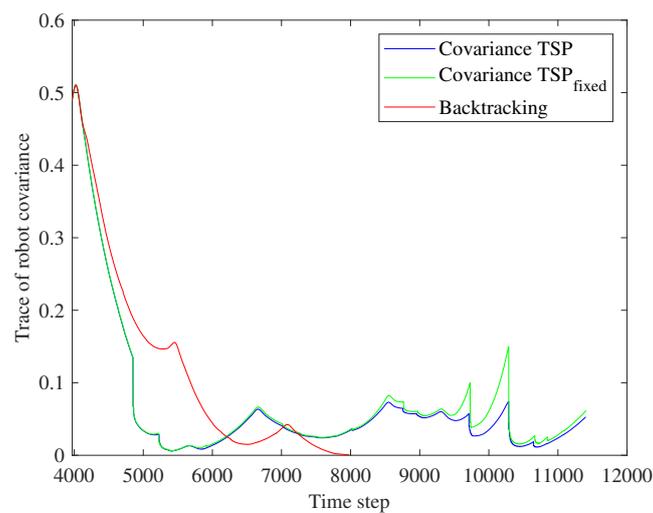
**Figure 9.** The simulation results of three revisiting methods: (a) Backtracking. (b) The covariance TSP<sub>fixed</sub> algorithm that restricts adding new landmarks during revisiting. (c) The covariance TSP algorithm.



(a)



(b)

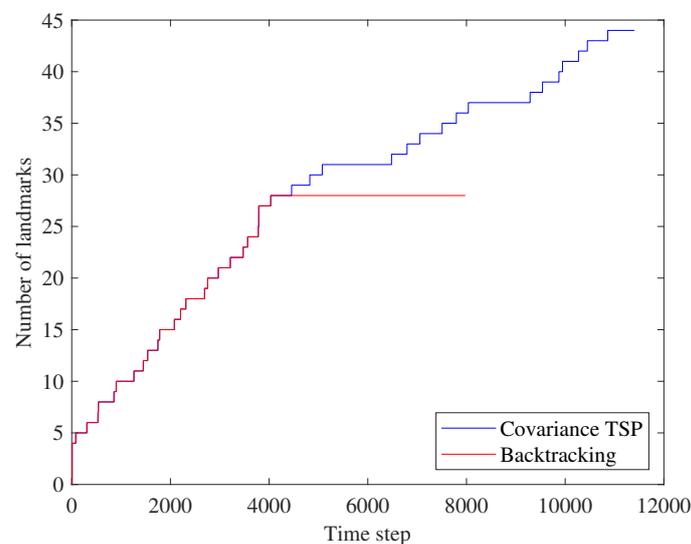


(c)

**Figure 10.** Comparisons of the proposed methods and backtracking: (a) The maximum trace value of landmark covariance matrix. (b) The sum of the trace values of the landmarks registered until the time when revisiting was planned. (c) The trace of the robot covariance matrix.

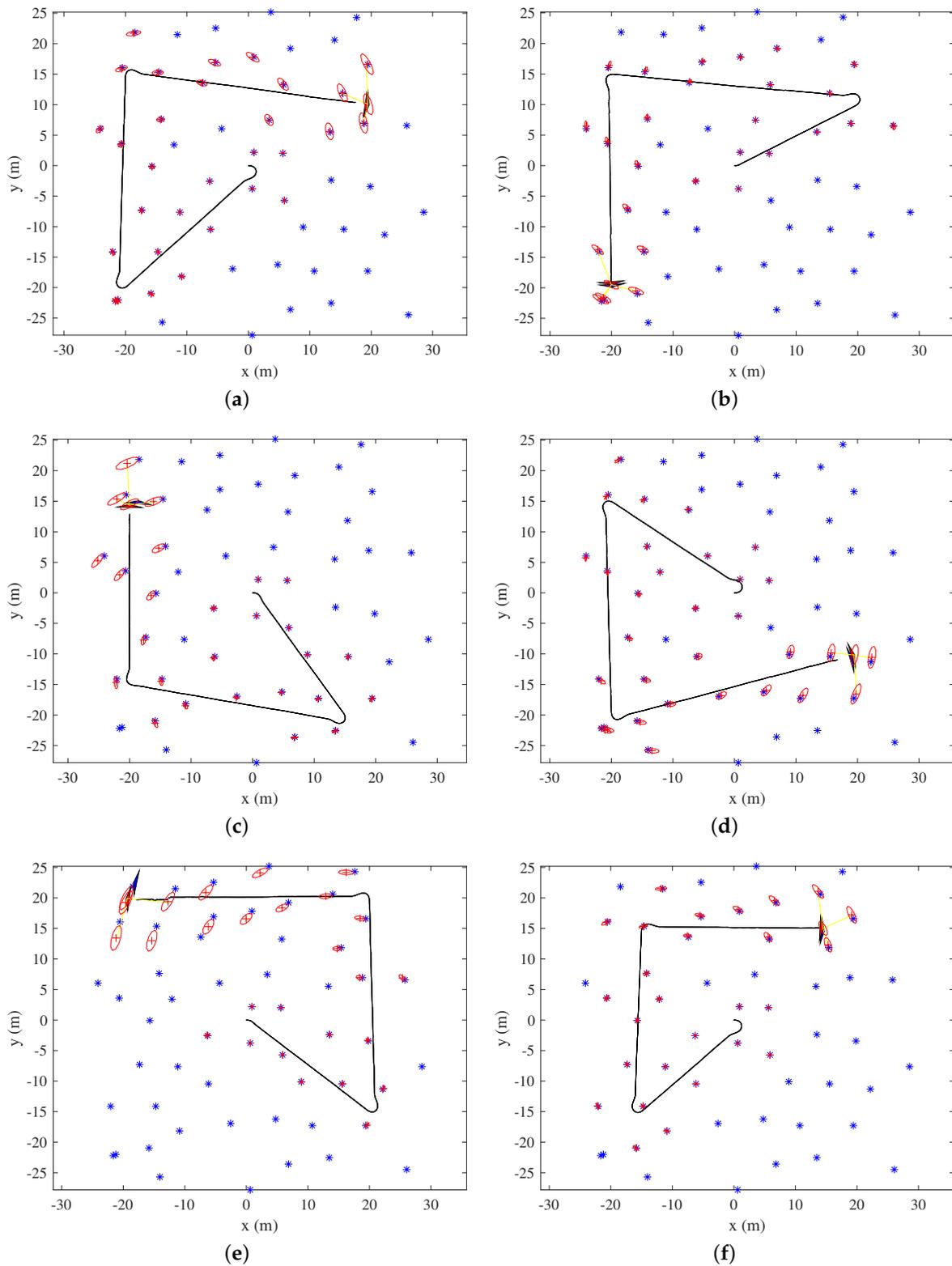
Figure 10c shows the covariance trace of the robot during revisiting. In the result of backtracking, when the robot was at the initial position (0, 0 m), the robot covariance became almost the zero matrix, that is the initial covariance matrix. With our methods, the robot pose uncertainty decreased more rapidly than during backtracking at early revisiting steps. Although the robot pose uncertainty of the proposed revisiting algorithms was larger than that of backtracking, the value was maintained at relatively low values compared to that when revisiting commenced. This was because the revisiting path of the covariance TSP solution does not contain the initial position (0, 0 m) as a node; the nodes are the positions from which less uncertain landmarks are highly observable.

Figure 11 shows the overall numbers of landmarks in the SLAM state. The proposed method, Covariance TSP, selects revisiting positions that ensure both data association (by re-observing registered landmarks) and exploratory utility (by adding new landmarks). The number of landmarks does not change during backtracking. The number of landmarks rises as new positions are visited and new measurements obtained along the TSP path of the proposed method.



**Figure 11.** The number of registered landmarks.

For further validation of the proposed method, multiple simulations using different SLAM states in the same environment were performed. Figure 12 shows six SLAM states when revisiting commenced. After traveling through three positions, the uncertainty of the robot pose and the landmark positions have increased. It is worth noting that revisiting paths can be different even with the same SLAM state since GA incorporates randomness. Ten simulations for each SLAM state were performed by using Covariance TSP<sub>fixed</sub> and Covariance TSP, respectively; total 120 simulations were performed by using the proposed algorithms. The average, minimum, and maximum values for each ten simulations were calculated. Because the backtracking path was determined along the previous trajectory, one simulation by backtracking was performed for each SLAM state.



**Figure 12.** Examples of EKF-SLAM robot/landmark state when revisiting commenced: (a–f) The EKF-SLAM state used for simulations. After traveling through three positions, uncertainty has increased. Ten simulations for each SLAM state were performed using Covariance TSP<sub>fixed</sub> and Covariance TSP, respectively.

Figure 13, and Tables 1 and 2 show the reduction of the trace sum of the landmarks registered in the SLAM state at the time when revisiting was planned and started. The sum of trace values of the landmarks for a single simulation is shown in Figure 10b, and the reduction was calculated as follows:

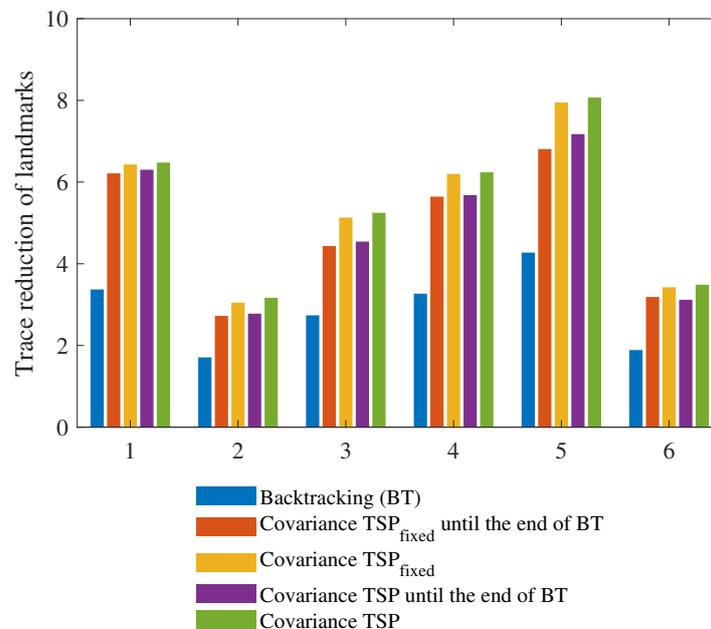
$$\sum_{n=1}^N \text{trace} (P_n^{\text{Start}}) - \sum_{n=1}^N \text{trace} (P_n^{\text{End}}) \quad (10)$$

where  $N$  is the number of the landmarks in the SLAM state at the time when revisiting was planned,  $P_n^{\text{Start}}$  is the covariance matrix of the  $n$ -th landmark at the time of starting revisiting, and  $P_n^{\text{End}}$  is the covariance matrix of the  $n$ -th landmark at the end of revisiting. The reduction of the trace sum means reduced uncertainty by each revisiting algorithm. Tables 3 and 4 show the ratio of the trace sum at the start of revisiting to the end of revisiting, and it was calculated as follows:

$$\frac{\sum_{n=1}^N \text{trace} (P_n^{\text{End}})}{\sum_{n=1}^N \text{trace} (P_n^{\text{Start}})}. \quad (11)$$

The ratio of the trace sum means how efficiently each revisiting algorithm reduces overall uncertainty in the registered landmarks. As more uncertainty has been reduced during the revisiting process, the ratio is smaller. The results of Tables 1 and 3 are obtained by the simulations from the beginning of revisiting to the end of revisiting by the proposed algorithms. The numbers in parentheses are the results from the beginning of revisiting to the time step when backtracking finished.

The average reduction of the trace sum by Covariance<sub>fixed</sub> TSP and Covariance TSP in Table 1 for all 120 simulations is larger than that by backtracking in Table 2. The average ratio of the final trace sum to the starting trace sum by the proposed methods in Table 3 is smaller than that by backtracking in Table 4. A few results in parentheses are larger than the backtracking results. However, these values have been reduced when revisiting is completed.



**Figure 13.** The reduction of the trace sum of the landmarks by backtracking and the average reduction of the trace sum of the landmarks by Covariance TSP<sub>fixed</sub> and Covariance TSP.

**Table 1.** The reduction of the trace sum of the landmarks registered in the SLAM state at the time when revisiting was planned and started.

		1	2	3	4	5	6
Covariance TSP (fixed)	Min.	6.40 (5.49)	2.96 (2.47)	5.04 (2.00)	6.14 (4.34)	7.82 (3.61)	3.40 (3.11)
	Avg.	6.43 (6.21)	3.05 (2.72)	5.13 (4.43)	6.20 (5.64)	7.95 (6.81)	3.43 (3.19)
	Max.	6.48 (6.41)	3.12 (2.80)	5.24 (4.98)	6.27 (6.08)	8.00 (7.23)	3.46 (3.25)
Covariance TSP	Min.	6.46 (5.75)	3.15 (2.61)	5.17 (2.33)	6.22 (4.19)	7.96 (4.89)	3.47 (1.33)
	Avg.	6.48 (6.30)	3.17 (2.78)	5.25 (4.54)	6.24 (5.68)	8.07 (7.17)	3.49 (3.12)
	Max.	6.50 (6.40)	3.19 (2.87)	5.35 (5.27)	6.28 (6.02)	8.18 (7.94)	3.57 (3.36)

**Table 2.** The reduction of the trace sum of the landmarks registered in the SLAM state at the time when revisiting was planned and started.

	1	2	3	4	5	6
Backtracking	3.37	1.71	2.74	3.27	4.27	1.89

**Table 3.** The ratio of the trace sum at the start of revisiting to the end.

		1	2	3	4	5	6
Covariance TSP (fixed)	Min.	0.09 (0.10)	0.19 (0.28)	0.11 (0.15)	0.10 (0.13)	0.09 (0.18)	0.14 (0.20)
	Avg.	0.09 (0.12)	0.21 (0.30)	0.13 (0.25)	0.11 (0.19)	0.10 (0.23)	0.15 (0.21)
	Max.	0.10 (0.23)	0.24 (0.36)	0.14 (0.66)	0.12 (0.38)	0.11 (0.59)	0.16 (0.23)
Covariance TSP	Min.	0.08 (0.10)	0.18 (0.26)	0.09 (0.10)	0.10 (0.13)	0.07 (0.10)	0.12 (0.17)
	Avg.	0.09 (0.11)	0.18 (0.28)	0.11 (0.23)	0.10 (0.18)	0.09 (0.19)	0.14 (0.23)
	Max.	0.09 (0.19)	0.19 (0.33)	0.12 (0.60)	0.11 (0.40)	0.10 (0.45)	0.14 (0.67)

**Table 4.** The ratio of the trace sum at the start of revisiting to the end.

	1	2	3	4	5	6
Backtracking	0.52	0.56	0.53	0.53	0.52	0.53

## 5. Conclusions

We developed a revisiting method reducing the overall uncertainty of the SLAM state. Existing revisiting approaches simply select revisiting positions from previous trajectories or return to the initial positions, whereas the proposed method selects revisiting positions with consideration of both landmark and sensor uncertainty. To do this, we proposed VLU grid maps showing the landmark visibility from each grid cell position. Less-uncertain landmarks were then defined, and revisiting positions from which the robot could not only re-observe selected landmarks but also detect new landmarks (to expand map information) were selected. Travel among revisiting positions was optimized using the TSP algorithm. As our covariance TSP algorithm employs covariance changes between pairs of nodes, the revisiting path efficiently reduces the overall covariance, not the total distance traveled.

We simulated the proposed method and simple backtracking. The proposed method reliably reduced the overall uncertainty of the EKF-SLAM state (the robot pose and the landmark positions). Although the closed path of the proposed method is longer than that of backtracking, our method reduces overall uncertainty earlier, and more extensively, than does backtracking. This is because the proposed revisiting path directs the robot to a place that allows valuable data association. Therefore, the proposed covariance TSP revisiting method can efficiently reduce and control the overall state uncertainty of landmark-based SLAM problems.

**Funding:** This research was supported by a project of Korean government (No. 10073166), and a grant from Endowment Project of "Development of fundamental technologies on underwater environmental recognition for

long-range navigation and intelligent autonomous underwater navigation” funded by Korea Research Institute of Ships and Ocean engineering (PES3160).

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Dissanayake, M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [[CrossRef](#)]
2. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [[CrossRef](#)]
3. Zhang, X.; Rad, A.B.; Wong, Y.K. Sensor fusion of monocular cameras and laser rangefinders for line-based simultaneous localization and mapping (SLAM) tasks in autonomous mobile robots. *Sensors* **2012**, *12*, 429–452. [[CrossRef](#)] [[PubMed](#)]
4. Sola, J.; Vidal-Calleja, T.; Civera, J.; Montiel, J.M.M. Impact of landmark parametrization on monocular EKF-SLAM with points and lines. *Int. J. Comput. Vis.* **2012**, *97*, 339–368. [[CrossRef](#)]
5. Do, C.H.; Lin, H.Y. Incorporating neuro-fuzzy with extended Kalman filter for simultaneous localization and mapping. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419874645. [[CrossRef](#)]
6. Zheng, B.; Zhang, Z. An Improved EKF-SLAM for Mars Surface Exploration. *Int. J. Aerosp. Eng.* **2019**, *2019*. [[CrossRef](#)]
7. Chen, X.; Sun, H.; Zhang, H. A New Method of Simultaneous Localization and Mapping for Mobile Robots Using Acoustic Landmarks. *Appl. Sci.* **2019**, *9*, 1352. [[CrossRef](#)]
8. He, B.; Zhang, S.; Yan, T.; Zhang, T.; Liang, Y.; Zhang, H. A novel combined SLAM based on RBPF-SLAM and EIF-SLAM for mobile system sensing in a large scale environment. *Sensors* **2011**, *11*, 10197–10219. [[CrossRef](#)]
9. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In Proceedings of the IJCAI, Acapulco, Mexico, 9–15 August 2003, pp. 1151–1156.
10. Grisetti, G.; Tipaldi, G.D.; Stachniss, C.; Burgard, W.; Nardi, D. Fast and accurate SLAM with Rao–Blackwellized particle filters. *Robot. Auton. Syst.* **2007**, *55*, 30–38. [[CrossRef](#)]
11. Martinelli, F. Simultaneous Localization and Mapping Using the Phase of Passive UHF-RFID Signals. *J. Intell. Robot. Syst.* **2019**, *94*, 711–725. [[CrossRef](#)]
12. Rapp, M.; Dietmayer, K.; Hahn, M.; Duraisamy, B.; Dickmann, J. Hidden Markov model-based occupancy grid maps of dynamic environments. In Proceedings of the IEEE 2016 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 1780–1788.
13. Ristic, B.; Angley, D.; Selvaratnam, D.; Moran, B.; Palmer, J.L. A random finite set approach to occupancy-grid SLAM. In Proceedings of the IEEE 2016 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 935–941.
14. Clemens, J.; Kluth, T.; Reineking, T.  $\beta$ -SLAM: Simultaneous localization and grid mapping with beta distributions. *Inf. Fusion* **2019**, *52*, 62–75. [[CrossRef](#)]
15. Grisetti, G.; Kummerle, R.; Stachniss, C.; Burgard, W. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43. [[CrossRef](#)]
16. Dai, J.; Yan, L.; Liu, H.; Chen, C.; Huo, L. An Offline Coarse-To-Fine Precision Optimization Algorithm for 3D Laser SLAM Point Cloud. *Remote Sens.* **2019**, *11*, 2352. [[CrossRef](#)]
17. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the IEEE 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016, pp. 1271–1278.
18. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
19. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 834–849.
20. Schmuck, P.; Chli, M. CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *J. Field Robot.* **2019**, *36*, 763–781. [[CrossRef](#)]

21. Jiang, G.; Yin, L.; Jin, S.; Tian, C.; Ma, X.; Ou, Y. A Simultaneous Localization and Mapping (SLAM) Framework for 2.5 D Map Building Based on Low-Cost LiDAR and Vision Fusion. *Appl. Sci.* **2019**, *9*, 2105. [[CrossRef](#)]
22. Bailey, T.; Nieto, J.; Guivant, J.; Stevens, M.; Nebot, E. Consistency of the EKF-SLAM algorithm. In Proceedings of the IEEE 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 3562–3568.
23. Latif, Y.; Cadena, C.; Neira, J. Robust loop closing over time for pose graph SLAM. *Int. J. Robot. Res.* **2013**, *32*, 1611–1626. [[CrossRef](#)]
24. Williams, B.; Cummins, M.; Neira, J.; Newman, P.; Reid, I.; Tardós, J. A comparison of loop closing techniques in monocular SLAM. *Robot. Auton. Syst.* **2009**, *57*, 1188–1197. [[CrossRef](#)]
25. Cole, D.M.; Newman, P.M. Using laser range data for 3D SLAM in outdoor environments. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 1556–1563.
26. Vlaminck, M.; Luong, H.; Philips, W. Have I Seen This Place Before? A Fast and Robust Loop Detection and Correction Method for 3D Lidar SLAM. *Sensors* **2019**, *19*, 23. [[CrossRef](#)]
27. Wen, J.; Qian, C.; Tang, J.; Liu, H.; Ye, W.; Fan, X. 2D LiDAR SLAM Back-End Optimization with Control Network Constraint for Mobile Mapping. *Sensors* **2018**, *18*, 3668. [[CrossRef](#)]
28. Wang, J.; Takahashi, Y. Slam method based on independent particle filters for landmark mapping and localization for mobile robot based on hf-band rfid system. *J. Intell. Robot. Syst.* **2018**, *92*, 413–433. [[CrossRef](#)]
29. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
30. Makarenko, A.A.; Williams, S.B.; Bourgault, F.; Durrant-Whyte, H.F. An experiment in integrated exploration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, Lausanne, Switzerland, 30 September–4 October 2002; Volume 1, pp. 534–539.
31. Sim, R.; Roy, N. Global a-optimal robot exploration in slam. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 661–666.
32. Applegate, D.L.; Bixby, R.E.; Chvatal, V.; Cook, W.J. *The Traveling Salesman Problem: A Computational Study*; Princeton University Press: Princeton, NJ, USA, 2006.
33. Sim, R.; Little, J.J. Autonomous vision-based robotic exploration and mapping using hybrid maps and particle filters. *Image Vis. Comput.* **2009**, *27*, 167–177. [[CrossRef](#)]
34. Kahng, A.B.; Reda, S. Match twice and stitch: A new TSP tour construction heuristic. *Oper. Res. Lett.* **2004**, *32*, 499–509. [[CrossRef](#)]
35. Choi, I.C.; Kim, S.I.; Kim, H.S. A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem. *Comput. Oper. Res.* **2003**, *30*, 773–786. [[CrossRef](#)]
36. Nagata, Y.; Soler, D. A new genetic algorithm for the asymmetric traveling salesman problem. *Expert Syst. Appl.* **2012**, *39*, 8947–8953. [[CrossRef](#)]
37. Rego, C.; Gamboa, D.; Glover, F.; Osterman, C. Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *Eur. J. Oper. Res.* **2011**, *211*, 427–441. [[CrossRef](#)]
38. Siarry, P. *Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 23.
39. Anbuudayasankar, S.; Ganesh, K.; Mohapatra, S. Survey of methodologies for tsp and vrp. In *Models for Practical Routing Problems in Logistics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 11–42.

