Indoor Traveling Salesman Problem (ITSP) Path Planning

Jinjin Yan 1,*,†, Sisi Zlatanova 2,†, Jinwoo (Brian) Lee 2,† and Qingxiang Liu 1

1 Qingdao Innovation and Development Center, Harbin Engineering University, Qingdao 266400, China; lavender.liu@hrbeu.edu.cn
2 School of Built Environment, Faculty of Arts, Design and Architecture, University of New South Wales, Kensington, Sydney, NSW 2052, Australia; s.zlatanova@unsw.edu.au (S.Z.); brian.j.lee@unsw.edu.au (J.L.)
* Correspondence: jinjin.yan@hrbeu.edu.cn
† These authors contributed equally to this work.

Abstract: With the growing complexity of indoor living environments, people have an increasing demand for indoor navigation. Currently, navigation path options in indoor are monotonous as existing navigation systems commonly offer single-source shortest-distance or fastest paths. Such path options might not always be attractive. For instance, pedestrians in a shopping mall may be interested in a path that navigates through multiple places starting from and ending at the same location. Here, we name it as the indoor traveling salesman problem (ITSP) path. As its name implies, this path type is similar to the classical outdoor traveling salesman problem (TSP), namely, the shortest path that visits a number of places exactly once and returns to the original departure place. This paper presents a general solution to the ITSP path based on Dijkstra and branch and bound (B&B) algorithm. We demonstrate and validate the method by applying it to path planning in a large shopping mall with six floors, in which the QR (Quick Response) codes are utilized as the indoor positioning approach. The results show that the presented solution can successfully compute the ITSP paths and their potentials to apply to other indoor navigation applications at museums or hospitals.

Keywords: pedestrian; indoor navigation; path planning; traveling salesman problem

1. Introduction

Navigation is a process or activity of accurately ascertaining locations of users, planning a path, and leading the users to follow the route to the goal location [1–3]. Indoor navigation has been studied by researchers and practitioners for many years, but indoor navigation services in most cases are unavailable in practice just yet. This dilemma reflects the fact that people have urgent needs for this service, but effective, robust, and practical indoor navigation solution is challenging. This could have been attributed by increasingly complex indoor environments, such as the growing large-scale offices, shopping malls, and other public buildings with complex, irregular interior arrangements, and people with diverse navigation requirements for living, studying, shopping, or working in indoor spaces.

A smart path planning needs to consider users’ requirements and customize the path accordingly [4]. However, path planning of current navigation systems primarily take the (shortest) travel distance or (fastest) time as the main criteria for optimal path computation [5,6]. Path with the shortest distance or fastest travel time could be a habitual choice for humans [7], but it does not mean this kind of path is always the desired choice under all circumstances. For example, pedestrians may be interested in the shortest path that can navigate them from the departure and come back to the original point after visiting specific places. In this paper, such kind of shortest path is named the indoor traveling salesman problem (ITSP) path in acknowledging that it is a variation of the conventional Traveling Salesman Problem (TSP). The conventional TSP is a well-known algorithmic...
problem in the fields of computer science and operations research, which was defined as the challenge of finding the shortest yet most efficient route for a person to take given a list of specific destinations [8]. In short, in the TSP a traveler must visit each destination once and return to the origin.

At first glance, the ITSP is the same as the conventional TSP. On a closer inspection however, it is not identical. In the conventional TSP, all the nodes are destinations and all of them will be included in the navigation path, while the navigation path of ITSP may only include a part of all possible nodes. In other words, if there are \( n \) nodes in the conventional TSP graph, the navigation path for this issue will include all the nodes. Yet, if there are \( n \) nodes in navigation network for ITSP, the navigation path will include \( m \) \((m \leq n)\) nodes. This is to say, the conventional TSP should be applied at a different set of nodes each time a navigation is requested.

We illustrate the difference between conventional TSP and ITSP with an indoor example. Figure 1 shows an indoor map and corresponding indoor navigation network. There are seven rooms (R1–R7), one Point of Interest (POI)—Automated Teller Machine (ATM) (the black dots in the navigation network), and one obstacle/atrium. The rooms and POI could be desired destinations. The other twenty-three nodes \((N0–N22)\) marked by hollow circles are other indoor locations; here, we name them as dummy nodes. If a person desires to start from R7 and wants to come back after visiting ATM, R1, R2, and R5, the conventional TSP will fail to compute the shortest path, because it generally would include all thirty nodes. Such path is not reasonable for this indoor application, as one shortest path that contains the above five desired nodes \((R7, ATM, R1, R2, and R5)\) at once and forms a ring-like path is what the pedestrian needs. Therefore, we argue that it is necessary to adapt the conventional TSP for indoor applications. This paper proposes a solution to ITSP path planning by combining the Dijkstra algorithm [9] and branch and bound (B&B) algorithm [10,11].

![Figure 1. An indoor map and its navigation network. (a) Indoor map used for illustration. (b) Navigation network corresponding to the indoor map.](image)

The remaining sections are organized as follows. Section 2 reviews the current personalized/agent-tailored navigation paths. Section 3 presents the concepts, modeling, procedures, as well as the illustration of the ITSP. Section 4 demonstrates the ITSP with case studies. Section 5 concludes with next steps related to future work, refinements, and tests.

2. Related Work

In the past decades, many studies have been devoted to personalized/agent-tailored in indoor/outdoor navigation paths, such as simplest or minimum turns path [12,13], least or most-space-visited, least-obstruction [14], safe path [15–18], health-optimal routing (e.g., specific level of calories burn) [19], minimum traffic related air pollution exposure [20], path with better-illuminated walking environments [21], Most-top-covered path (MTC-path) [2], and path to the nearest semi-indoor space (NSI-path) [22].
The (shortest) travel distance or (fastest) time has been the main optimization objective for path planning in current navigation systems [5,6,23], although diverse travel modes are considered, such as walking [24], driving [25], transit [26], and cycling [27]. Various algorithms can be used for computing the shortest path. For instance, the Dijkstra algorithm [9], A* [28], and Ant colony [29].

The conventional TSP has attracted much attention from mathematicians, and computer scientists specifically, as it was first formulated in 1930 because it is so easy to describe and so difficult to solve. It belongs to the class of such problems known as non-deterministic polynomial-time complete (NP-complete) [8,30,31]. As the real situation becomes more complicated, new variants of the existing basic TSP developed, such as there is a time limit for delivery (Traveling Salesman Problem with Time Window (TSPTW) [32,33]), one-way road (Asymmetric Traveling Salesman Problem (ATSP) [34]), more than one salesman (Multiple Traveling Salesman Problem (MTSP) [35,36]), one TSP with constrains (Precedence Constrained Generalized Traveling Salesman Problem [37], Partially Ordered Traveling Salesman Problem [38]), etc. However, although the path finding and planning have been widely studied in the literature, to the best knowledge of the authors, no study has been done on ITSP because of its special characteristics. Furthermore, all the algorithms or suggested approaches in the literature for path planning are not directly applicable to this problem.

3. Indoor Traveling Salesman Problem (ITSP)

3.1. Concepts and Modeling

The ITSP can be best described as the following scenario: a pedestrian plans to visit an indoor scene with several \( n \in \mathbb{Z}^+ \) indoor nodes (they could be rooms, indoor POIs, specific functional areas, etc.), in which the distance between every two nodes \( d(s_i, s_j) \) may be the same or different. The pedestrian starts from a space and returns to the same space after visiting all the desired indoor nodes. The solution to this problem is to seek the optimal order of visiting all desired indoor nodes that makes the path to be the shortest.

The mathematical model of ITSP is as follows:

All indoor spaces are represented as nodes:

\[
S = \{s_1, s_2, s_3, ..., s_n\} \tag{1}
\]

The ITSP is symmetric [30], the distances between any two nodes are

\[
d(s_i, s_j) = d(s_j, s_i) \in \mathbb{R}^+, 1 \leq i < j \leq n \tag{2}
\]

where \( i \) and \( j \) mean the \( i \)th and \( j \)th of the indoor nodes.

The goal is to find the optimal order of visiting all desired indoor nodes to minimize the cost:

\[
\min\{ \sum_{i=1}^{n-1} d(s_{k_i}, s_{k_{i+1}}) + d(s_{k_n}, s_{k_1}) \} \tag{3}
\]

where \( s_{k_i} \) denotes the \( k_i \)th indoor node.

3.2. Procedures of ITSP Path Planning

The path planning of ITSP is based on an indoor navigation network, which is modeled as a graph \( G_{\text{original}}(V, E) \). In the graph, vertices \( V \) are abstracted from indoor spaces \( s_i \) and edges \( E \) from the relationships between spaces (the theoretical basis is Poincaré duality [39]). After choosing one node as the departure location \( s_{\text{departure}} \) and several other nodes (all the other nodes can be included in a set \( s_{\text{intermediate}} \) as specified intermediate destinations, a shortest path that includes all intermediate destinations once and goes back to the destination \( s_{\text{departure}} \) must be computed. The procedures of ITSP path planning include the following five steps:

- Step 1: Select intermediate destinations (nodes).
Select departure location \(s_{\text{departure}}\) and specified intermediate places \(s_{\text{intermediate}}\). For instance, \(s_{\text{departure}} = s_1\) and \(s_{\text{intermediate}} = \{s_4, s_7, s_9\}\).

- **Step 2:** Compute navigation paths between every two selected nodes. This step takes the departure location and specified intermediate places as nodes to compute navigation paths between any two nodes based on Dijkstra. The lengths of the navigation paths are used as the distances between two nodes. In this paper, we set the \(d(s_i, s_j) = d(s_j, s_i)\), i.e., the paths in this example are symmetric and we simply assume the route between the two intermediate destinations is bi-directional. In reality, \(d(s_i, s_j)\) may be not equal to \(d(s_j, s_i)\), e.g., one-way corridors such as escalators. In all cases, \(d(s_i, s_i) = d(s_j, s_j) = 0\). Continue the example in Step 1, this step can get six paths: \(s_1 \rightsquigarrow s_4, s_1 \rightsquigarrow s_7, s_1 \rightsquigarrow s_9, s_4 \rightsquigarrow s_7, s_4 \rightsquigarrow s_9, s_7 \rightsquigarrow s_9\), and corresponding reverse paths. The symbol \((\rightsquigarrow)\) means there are zero to several navigation nodes between two nodes. The travel distances are \(d(s_1, s_4) = d(s_4, s_1), d(s_1, s_7) = d(s_7, s_1), d(s_1, s_9) = d(s_9, s_1), d(s_4, s_7) = d(s_7, s_4), d(s_4, s_9) = d(s_9, s_4), d(s_7, s_9) = d(s_9, s_7)\).

- **Step 3:** Set up graph of desired intermediate destinations. Setting up a graph of all desired intermediate destinations is to take the travel distances of every two specified places as weights and all these places as nodes to make an undirected graph. For the example in Step 1, the undirected graph can be organized as a table (Table 1).

- **Step 4:** Select and sort the navigation paths based on B&B algorithm. Taking the undirected graph as the input, the orders of departure and specified intermediate places can be computed based on the B&B algorithm. Then, the orders are further used to select and sort navigation paths. For instance, the order could be \(<s_1, s_7, s_9, s_4, s_1>\), which means the following navigation paths will selected and sorted as \(s_1 \rightsquigarrow s_7, s_7 \rightsquigarrow s_9, s_9 \rightsquigarrow s_4, s_4 \rightsquigarrow s_1\).

- **Step 5:** Combine the navigation results of Dijkstra as the ITSP path. The last step is to combine the navigation results as the ITSP path. For instance, the ITSP path of the example is \(s_1 \rightarrow s_7 \rightarrow s_9 \rightarrow s_4 \rightarrow s_1\). Then, the final navigation path becomes \(s_1 \rightarrow \ldots \rightarrow s_7 \rightarrow \ldots \rightarrow s_9 \rightarrow \ldots \rightarrow s_4 \rightarrow \ldots \rightarrow s_1\). The symbol \((\rightarrow)\) means there is no other navigation node between the two nodes.

**Table 1. Example of the undirected graph.**

|          | \(s_1\) | \(s_4\) | \(s_7\) | \(s_9\) |
|----------|---------|---------|---------|---------|
| \(s_1\)  | 0       | \(d(s_1,s_4)\) | \(d(s_1,s_7)\) | \(d(s_1,s_9)\) |
| \(s_4\)  | \(d(s_4,s_1)\) | 0       | \(d(s_4,s_7)\) | \(d(s_4,s_9)\) |
| \(s_7\)  | \(d(s_7,s_1)\) | \(d(s_7,s_4)\) | 0       | \(d(s_7,s_9)\) |
| \(s_9\)  | \(d(s_9,s_1)\) | \(d(s_9,s_4)\) | \(d(s_9,s_7)\) | 0       |

### 3.3. Illustration

An indoor example is employed to demonstrates the procedures of ITSP path planning. The navigation network of the indoor scenario is shown in Figure 2. In the first step, we select room (‘R7’) as departure location and specified intermediate places are ‘R1’, ‘R2’, ‘R5’, and ‘ATM’, i.e., \(s_{\text{departure}} = \text{‘R7’}\) (marked by red star) and \(s_{\text{intermediate}} = \{\text{‘R1’, ‘R2’, ‘R5’, ‘ATM’}\}\) (marked by blue dots).

By following the second step in the procedures, the navigation paths of every two desired destinations can be computed based on Dijkstra. Utilizing each pair of nodes as the temporary departure and destination, we can get all the navigation paths and travel distances (Table 2). Then, the distances are utilized to make up an undirected graph (Figure 2 and Table 3).
Figure 2. (a) The navigation network of the indoor scene. (b) Illustration of the undirected graph of all selected desired destinations.

Table 2. Navigation path between any two selected places.

| Start | End | Path | Distance |
|-------|-----|------|----------|
| R7    | R1  | R7 → N0 → N19 → N16 → N1 → R1 | 12.19     |
| R7    | R2  | R7 → N0 → N7 → N13 → N2 → R2 | 13.24     |
| R7    | R5  | R7 → N0 → N7 → N10 → N6 → N5 → R5 | 18.43     |
| R7    | ATM | R7 → N0 → N19 → ATM | 8.19      |
| R1    | R2  | R1 → N1 → N22 → N13 → N2 → R2 | 10.45     |
| R1    | R5  | R1 → N1 → N22 → N13 → N10 → N6 → N5 → R5 | 19.81     |
| R1    | ATM | R1 → N1 → N16 → ATM | 7.70      |
| R2    | R5  | R2 → N2 → N10 → N6 → N5 → R5 | 14.50     |
| R2    | ATM | R2 → N13 → N22 → N21 → N16 → ATM | 11.04     |
| R5    | ATM | R5 → N5 → N6 → N10 → N7 → N20 → N19 → ATM | 18.73     |

Table 3. Undirected graph of all selected places.

| Start | R7 | R1 | R2 | R5 | ATM |
|-------|----|----|----|----|-----|
| R7    | 0  | 12.19 | 13.24 | 18.43 | 8.19 |
| R1    | 12.19 | 0   | 10.45 | 19.81 | 7.70 |
| R2    | 13.24 | 10.45 | 0   | 14.50 | 11.04 |
| R5    | 18.43 | 19.81 | 14.50 | 0   | 18.73 |
| ATM   | 8.19 | 7.70 | 11.04 | 18.73 | 0    |

With the undirected graph, the optimal orders of the nodes can be computed by using B&B algorithm: <‘R7’, ‘R5’, ‘R2’, ‘R1’, ‘ATM’, ‘R7’>. It means that, having ‘R7’ as the departure and ‘R1’, ‘R2’, ‘R5’, ‘ATM’ as the specified intermediate places during the traveling, the sorted navigation path is: R7 ← R5 ← R2 ← R1 ← ATM ← R7 (Figure 3).

The last step is to combine the results of Dijkstra to the sorted path as the ITSP path. For the case, the optimal path is R7 → N0 → N7 → N10 → N6 → N5 → R5 → N5 → N6 → N10 → N2 → R2 → N2 → N13 → N22 → N1 → R1 → N1 → N16 → ATM → N19 → N0 → R7 (Figure 3). The cost of this path is 59.27.
Figure 3. The result of TSP and the optimal navigation path of ITSP. (a) Results of sorted paths. (b) The navigation path of ITSP.

4. Implementation and Case Study

4.1. Case Description and Data Preparation

In this section, we modeled one of the largest shopping centers in Sydney, Australia, for testing the ITSP path planning. Two software packages were employed in this implementation: Quantum GIS (QGIS) and Rhinoceros (with Grasshopper). In the whole process, QGIS is used to edit the indoor maps, including editing footprints and attributes of indoor elements, and setting the coordinate system. Rhinoceros (with Grasshopper) is used to process the entire procedures of ITSP path planning, and the whole data process is developed in Python script. We edited the floor plan into four shapefile layers, including rooms, corridors, atriums, and POIs (Figure 4), in which except for the POIs, all other spaces are polygons. The shops, toilets, escalators, stairs, and lifts are included in the layer named Rooms, while the ATM and Bench are edited in POIs layer. Atriums and corridors are edited in two separated layers named Atriums and Corridor. Each element in each layer has three attributes: {ID, name, category}. ID records the id of each element, name keeps the semantic of the element (e.g., “Coles”, “EscalatorL5_L4”, “ToiletA_L5”, “ATM_A_L5”), and category records the supplement information. For instance, the category of the shop named “Kidstuff”, is “Books, stationary and gifts, Entertainment and activities, Sporting goods stores, Toys and hobbies stores”.

4.2. Navigation Network Derivation

Navigation paths are computed based on navigation network. In this paper, we follow a space-based approach for constructing a network [40]. The indoor spaces are polygons, and the theory that is used for automatically generating navigation networks is the Poincaré duality [39].

The indoor localization technique this implementation uses is Quick Response (QR) codes. Compared to QR codes, other techniques (e.g., Bluetooth beacons, Wi-Fi) suffer from some limitations, such as extra installation, maintenance cost, or infrastructure cost. QR codes are matrix codes, which are similar to two-dimensional bar codes, but more efficient than the barcodes with their capacity of encoding, faster reading speed, and readability from any direction from 360 degrees [41]. During the navigation, pedestrians can scan the nearest QR codes to have their current locations. In this case study, QR codes are placed on the ground in front of the doors (the left figure of Figure 5). Then, locations of QR codes and the vertices of the corridor and atrium polygons are utilized as the generators of Voronoi Tessellation with envelope [42] to subdivide the corridor spaces. This approach can make sure each subspace only contains one QR code, and consequently, the subdivided spaces can indicate the topological relationship between QR codes, see the middle figure of Figure 5. Finally, on the basis of Poincaré duality, the topological relationships are automatically derived as indoor navigation networks (the right figure of Figures 5 and 6). In Figure 6, the distances between two floors are enlarged to 50 m to make the spaces and navigation network more visible. In the navigation path planning, we reduced this distance to 3 m. It is necessary to explain that there are two reasons why the QR codes are
placed on the ground: (i) QR codes should be placed to the places where pedestrians can walk, and (ii) we consider it is easier for users to find and scan them, especially the indoor open spaces.

Figure 4. Floor plan of the floor utilized for testing in the shopping mall. (a) The floor plan edited in QGIS. (b) The attributes of indoor elements.

Figure 5. The floor plan, QR codes, space subdivision, and navigation network derivation of the selected floor. Black squares are QR codes, green lines in the left figure are split lines, blue lines in the right figure are edges of navigation network.
4.3. ITSP Path Planning

Two navigation cases in the shopping mall are utilized to demonstrate the ITSP path planning. The first case is a pedestrian who plans to search for some items for kids on the fifth floor. From his/her current place ('LiftC_L5' or 'EscalatorB_L5'), he/she would like to have a shortest navigation path that can help him/her to visit all the shops selling kids and babies products and then come back to the original place. This navigation can be translated as an ITSP path planning case, in which the shops with products for babies or kids are desired destinations.

We first search the shops that have products for babies/kids as the desired destinations. Considering the name or category of shops can give the clues that they are selling goods for kids or babies, we conduct this process in the Query Builder of QGIS, in which we set a specific filter expression: “name” LIKE '%baby%' or “name” LIKE '%kid%' or “category” LIKE '%Baby%' or “category” LIKE '%kid%'. The search results are twelve shops (Table 4). Thus, the desired destinations are the twelve shops. Meanwhile, we suppose the costumer starts from LiftC_L5 and EscalatorB_L5 (Figure 7).
Table 4. The twelve shops with products for babies or kids.

| Id   | Name            | Category                                                                 |
|------|-----------------|--------------------------------------------------------------------------|
| L5_6 | Sheriden        | Baby and nursery, Clothing and accessories, Home                         |
| L5_16| Adairs          | Baby and nursery, Home                                                   |
| L5_21| Bed Bath N’ Table | Clothing and accessories, Baby and nursery, Home                         |
| L5_22| Priceline       | Baby and nursery, Discount and variety, Health and fitness               |
| L5_25| Adairs Kids     | Clothing and accessories, Home, Toys and hobbies stores                  |
| L5_26| Kidstuff        | Books, stationary and gifts, Entertainment and activities, Sporting goods stores, Toys and hobbies stores |
| L5_28| Stokke          | Baby and nursery                                                         |
| L5_31| Bonds Kids      | Clothing and accessories                                                 |
| L5_36| Seed Kids       | Books, stationary and gifts                                               |
| L5_37| Purebaby        | Baby and nursery, Clothing and accessories                                |
| L5_38| Cotton on Kids  | Clothing and accessories                                                 |
| L5_39| Target          | Baby and nursery, Toys and hobbies stores                                 |

Having the twelve desired destinations and the navigation network, the two procedures that (i) compute navigation paths between every two destinations based on Dijkstra algorithm and (ii) set up a graph of destinations are conducted automatically. Then, taking the undirected graph as the input, the orders of departure and specified intermediate places are computed based on the B&B algorithm (Table 5).

Table 5. TSP results with the two departure locations.

| Departure     | TSP Results                                                                 |
|---------------|-----------------------------------------------------------------------------|
| LiftC_L5      | LiftC_L5 ⇝ Seed Kids ⇝ Target ⇝ Kidstuff ⇝ Adairs Kids                      |
|               | ⇝ Stokke ⇝ Bonds Kids ⇝ Priceline ⇝ Bed Bath N’ Table                      |
|               | ⇝ Sheriden ⇝ Adairs ⇝ Purebaby ⇝ Cotton on Kids ⇝ LiftC_L5                 |
| EscalatorB_L5 | EscalatorB_L5 ⇝ Sheriden ⇝ Adairs ⇝ Seed Kids ⇝ Purebaby                  |
|               | ⇝ Cotton on Kids ⇝ Target ⇝ Kidstuff ⇝ Adairs Kids                         |
|               | ⇝ Stokke ⇝ Bonds Kids ⇝ Priceline ⇝ Bed Bath N’ Table                      |
|               | ⇝ EscalatorB_L5                                                             |

The final step is combining the results as a navigation path (Figure 8). The length of the path starts from ‘LiftC_L5’ is 1002.41 m and that of the path from ‘EscalatorB_L5’ is 1050.34 m. These two navigation paths truly simulate the process of a customer coming out of the elevator or stairs to go shopping in stores that have products for kids or babies. Therefore, we believe ITSP path is useful for such kinds of people.

The second navigation case is to show that the presented solution also can support ITSP path planning in multi-floors. The scenario is a pedestrian who wants to visit several shops in the shopping mall (Figure 9). He/she starts from the supermarket named “Coles” on the first floor, which is the departure and final destination location of this indoor travel. The other desired destinations are a shop named “Bondi Hair” that also on the first floor, “MacDonald’s” on the third floor, and two shops (“Target” and “Myer L5”) on the fifth floor.
Figure 8. The ITSP path tests on the selected floor. The red star and circle represent the departure location, the blue location tags mark the shops that have stuff for kids/babies, the blue lines are the navigation paths, and the pink dots in the navigation path are the vertex of path segments. (a) Departure is ‘LiftC.L5’; (b) Departure is ‘EscalatorB.L5’.

Figure 9. Departures and intermediate places in multi-floors.

In the path planning, we added a condition on the traveling ways of between different floors, i.e., lifts only, or all the ways (stairs, lifts, and escalators). The navigation paths computed by presented approach can be seen in Figure 10, in which the orders of the shops that will be visited in the two navigation paths are coincidentally the same: “Coles” → “Bondi Hair” → “Target” → “Myer L5” → “McDonald’s” → “Coles”. But, the two paths have different travel distance, 1202.06 m for the using lifts (Figure 10) only while 955.41 m for vitalizing all types of ways (Figure 10).
Figure 10. The ITSP path tests on multi-floors. The red star represent the departure location, the blue location tags mark the shops that the customer is interested. (a) Only Lifts; (b) Stairs, escalators, and lifts.

The two navigation paths show that the multi-floor ITSP path planning based on the presented approach is feasible. During the navigation, pedestrians can have desired destinations that are distributed on the same or different floors. Meanwhile, the ways (stairs, lifts, or escalators) that the people using for transferring from one floor to another can be considered in ITSP path planning.

4.4. Discussion

The use cases above demonstrate that the presented approach can solve the ITSP path planning. However, several aspects are still worth mentioning.

The performance of navigation paths. The navigation paths can help pedestrians to have shopping plans, but due to the incalculable walking distance and residence time of pedestrians in each store, there will be a deviation in estimated journey time of the entire itinerary. Moreover, distance is the only factor that is considered when determining the optimal path between two desired destinations. Other than the distance, there are other factors that may be also important, such as the pedestrian number, the crowd congestion, and the traffic time on the road. These factors are not considered in this paper. Moreover, the navigation paths are a little angular, which is the common results of path planned based on the navigation network. Therefore, the navigation path should be further refined to avoid refractions.

Selection of desired destinations. Theoretically, the proposed approach can compute a navigation path for the ITPS no matter how many desired destinations are included. However, the more desired destinations are involved and the more complex the navigation network is, a longer processing time will be required for computing navigation paths. Thus, on one hand, it is necessary to improve the performance of Dijkstra [43] and B&B algorithm [11] or use other algorithms, and on the other hand, set up some rules on the selection of destinations should be investigated to reduce their number, such as setting up a scoring system to rank all the desired destinations and chose the top five in the path planning.

Navigation should be performed based on 3D spaces [2,40,44]. This navigation implementation is based on 2D floor plans; therefore, the vertical information of this floor is missing from the data source. Although it is very unlikely in the shopping center case, some indoor spaces may have hazardous areas due to a low height constraint. Therefore,
extending the floor plan to 3D and possible hazardous areas would be beneficial for such places where the height limit presents [2].

Other combinations of navigation path planning algorithms. The implementations show that combining Dijkstra and B&B algorithm is a feasible solution to ITSP path planning. Note that this combination is only one optional option. Using other algorithms also may achieve the same effect but more efficient, such as using A* [45] or Ant colony to replace Dijkstra while replacing B&B algorithm other algorithms, such as stochastic algorithms [46], approximation algorithms [47,48], other heuristics algorithms [49,50], or even brute force algorithm.

5. Conclusions and Future Work

The motivation of this study is that pedestrians may be interested in the shortest path that can navigate them from a departure and come back to the original location after visiting specific places once. This research calls such an indoor navigation path type as Indoor Traveling Salesman Problem (ITSP) and presents a solution to this path type on the basis of the Dijkstra and branch and bound (B&B) algorithm. The implementation was demonstrated on a large, multi-level shopping mall, which shows that the presented approach can compute a shortest path for pedestrians in indoor spaces successfully.

This path type can be used for other indoor scenarios where visitors knows what kind of indoor locations he/she is going to visit or deliver and needs the shortest path that can connect all the locations. For example, visitors to an hospital may need to visit multiple locations for examinations. Museum is another type of indoor environment where the ITSP path can assist visitors, for example, top twenty exhibits or that in specific themes to help such kind of visitors rather than wandering around the whole museum for more enjoyable visit.

In future work, we will concentrate on further elaboration and testing of the current work as follows: (i) Investigate the rules of selecting desired destinations. (ii) Employ the presented approach to outdoor and test ITSP in seamless indoor and outdoor navigation. (iii) Build a more refined navigation network or certain strategies for path smoothing. (iv) Conduct ITSP tests based on indoor 3D spaces. (v) Test and compare other combinations of path planning algorithms for ITSP path planning and give selection suggestions by comparing the results. (vi) Consider more factors into the ITSP planning, such as the pedestrian number, the crowd congestion, and the traffic time on the road. (vii) Take dynamic POIs as the destinations for ITSP path planning.

Author Contributions: Conceptualization, Jinjin Yan, Qingxiang Liu; methodology, Sisi Zlatanova, Jinwoo (Brian) Lee; software, Jinjin Yan; validation, Jinjin Yan, Sisi Zlatanova and Jinwoo (Brian) Lee; formal analysis, Sisi Zlatanova, Jinwoo (Brian) Lee; investigation, Jinjin Yan, Qingxiang Liu; resources, Jinwoo (Brian) Lee; data curation, Jinwoo (Brian) Lee; writing—original draft preparation, Jinjin Yan; writing—review and editing, Sisi Zlatanova, Jinwoo (Brian) Lee; visualization, Jinjin Yan; supervision, Sisi Zlatanova, Jinwoo (Brian) Lee; project administration, Jinwoo (Brian) Lee; funding acquisition, Jinwoo (Brian) Lee. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors are grateful to the Korea Agency for Infrastructure Technology Advancement: Grant 20CTAP-C151789-02, which funded the developments in this research. We highly acknowledge the anonymous reviewers who helped to improve the work considerably through their questions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jayakanth, K.; AbdelGhani, K.; Somaya, A.; Abdulla, A. Indoor positioning and wayfinding systems: A survey. *Hum.-Centric Comput. Inf. Sci.* 2020, 10, 18.
2. Yan, J.; Diakité, A.A.; Zlatanova, S.; Aleksandrov, M. Top-Bounded Spaces Formed by the Built Environment for Navigation Systems. *ISPRS Int. J. Geo-Inf.* 2019, 8, 224. [CrossRef]
3. Ambareesh, S.; Tejashwini, D.; Deeksha, R.S.; Sangeetha, S. Navigation for Indoor Location Based On QR Codes and Google Maps—A Survey. *Int. J. Innov. Res. Inf. Secur.* 2017, 4, 43–48.
4. Fallah, N.; Apostolopoulos, I.; Bekris, K.E.; Folmer, E. Indoor Human Navigation Systems: A Survey. *Interact. Comput.* 2013, 25, 21–33.

5. Dudás, P.M.; Ghafoorian, M.; Karimi, H.A. ONALIN: Ontology and algorithm for indoor routing. In Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, Taipei, Taiwan, 18–20 May 2009; pp. 720–725.

6. Ghafoorian, M.; Karimi, H.A. CAD/GIS Integration Issues for Seamless Navigation between Indoor and Outdoor Environments. In *CAD and GIS Integration*; Auerbach Publications: Boca Raton, FL, USA, 2009; pp. 141–154.

7. Golledge, R.G. Path selection and route preference in human navigation: A progress report. In *International Conference on Spatial Information Theory*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 207–222.

8. Bellmore, M.; Nemhauser, G.L. The traveling salesman problem: A survey. *Oper. Res.* 1968, 16, 538–558. [CrossRef]

9. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* 1959, 1, 269–271. [CrossRef]

10. Pasiliao, E.L.; Pardalos, P.M.; Pitsoulis, L.S. Branch and bound algorithms for the multidimensional assignment problem. *Optim. Methods Softw.* 2005, 20, 127–143. [CrossRef]

11. Morrison, D.R.; Jacobson, S.H.; Sauppe, J.J.; Sewell, E.C. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discret. Optim.* 2016, 19, 79–102. [CrossRef]

12. Nandini, D.; Seeja, K. A novel path planning algorithm for visually impaired people. *J. King Saud Univ.-Comput. Inf. Sci.* 2019, 31, 385–391. [CrossRef]

13. Duckham, M.; Kulik, L. “Simplest” paths: Automated route selection for navigation. In *International Conference on Spatial Information Theory*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 169–185.

14. Liu, L.; Zlatanova, S. A two-level path-finding strategy for indoor navigation. In *Intelligent Systems for Crisis Management*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 31–42.

15. Andreev, S.; Dibbelt, J.; Nöllenburg, M.; Pajor, T.; Wagner, D. Towards Realistic Pedestrian Route Planning. In Proceedings of the 15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2015), Patras, Greece, 17 September 2015; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Wadden, Germany, 2015.

16. Balata, J.; Berka, J.; Mikovec, Z. Indoor-Outdoor Intermodal Sidewalk-Based Navigation Instructions for Pedestrians with Visual Impairments. In *International Conference on Computers Helping People with Special Needs*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 292–301.

17. Cambra, P.J.; Gonçalves, A.; Moura, F. The digital pedestrian network in complex urban contexts: A primer discussion on typological specifications. *Finisterren* 2019, 54, 155–170.

18. Wang, Z.; Zlatanova, S. Safe Route Determination for First Responders in the Presence of Moving Obstacles. *IEEE Trans. Intell. Transp. Syst.* 2020, 21, 1044–1053. [CrossRef]

19. Sharker, M.H.; Karimi, H.A.; Zgibor, J.C. Health-optimal routing in pedestrian navigation services. In Proceedings of the First ACM SIGSPATIAL International Workshop on Use of GIS in Public Health, Redondo Beach, CA, USA, 6 November 2012; ACM: New York, NY, USA, 2012; pp. 1–10.

20. Alam, M.; Furugu, H.; McNabola, A. A comparison of route-choice navigation across air pollution exposure, CO₂ emission and traditional travel cost factors. *Transport. Res. Part D Transp. Environ.* 2018, 65, 82–100. [CrossRef]

21. Fang, Z.; Li, L.; Li, B.; Zhu, J.; Li, Q.; Xiong, S. An artificial bee colony-based multi-objective route planning algorithm for use in pedestrian navigation at night. *Int. J. Geogr. Inf. Sci.* 2017, 31, 2020–2044. [CrossRef]

22. Yan, J.; Zlatanova, S.; Diakite, A.A. Two new pedestrian navigation path options based on semi-indoor space. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2020, VI-4/W1-2020, 175–182. [CrossRef]

23. Kimmel, R.; Amir, A.; Bruckstein, A.M. Finding shortest paths on surfaces using level sets propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* 1995, 17, 635–640. [CrossRef]

24. Karimi, H.A.; Jiang, M.; Zhu, R. Pedestrian navigation services: Challenges and current trends. *Geomatica* 2013, 67, 259–271. [CrossRef]

25. Millonig, A.; Schechtner, K. Developing landmark-based pedestrian-navigation systems. *IEEE Trans. Intell. Transp. Syst.* 2007, 8, 43–49. [CrossRef]

26. Zar, M.T.; Sein, M.M. Finding shortest path and transit nodes in public transportation system. In *Genetic and Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 339–348.

27. Howard, C.; Burns, E.K. Cycling to work in Phoenix: Route choice, travel behavior, and commuter characteristics. *Transp. Res. Rec.* 2001, 1773, 39–46. [CrossRef]

28. Zeng, W.; Church, R.L. Finding shortest paths on real road networks: The case for A. *Int. J. Geogr. Inf. Sci.* 2009, 23, 531–543. [CrossRef]

29. Kolavali, S.R.; Bhatnagar, S. Ant colony optimization algorithms for shortest path problems. In *International Conference on Network Control and Optimization*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 37–44.

30. Jünger, M.; Reinelt, G.; Rinaldi, G. The traveling salesman problem. *Handb. Oper. Res. Manag. Sci.* 1995, 7, 225–330.

31. Hoffman, K.L.; Padberg, M.; Rinaldi, G. Traveling salesman problem. *Encycl. Oper. Res. Manag. Sci.* 2013, 1, 1573–1578.

32. López-Ibáñez, M.; Blum, C.; Ohlmann, J.W.; Thomas, B.W. The travelling salesman problem with time windows: Adapting algorithms from travel-time tomakespan optimization. *Appl. Soft Comput.* 2013, 13, 3806–3815. [CrossRef]
33. Montero, A.; Méndez-Díaz, I.; Miranda-Bront, J.J. An integer programming approach for the time-dependent traveling salesman problem with time windows. *Comput. Oper. Res.* 2017, 88, 280–289. [CrossRef]

34. Öncan, T.; Altunel, I.K.; Laporte, G. A comparative analysis of several asymmetric traveling salesman problem formulations. *Comput. Oper. Res.* 2009, 36, 637–654. [CrossRef]

35. Bektaş, T. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* 2006, 34, 209–219. [CrossRef]

36. Kitjacharoenchai, P.; Ventresca, M.; Moshref-Javadi, M.; Lee, S.; Tanchoco, J.M.; Brunese, P.A. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Comput. Ind. Eng.* 2019, 129, 14–30. [CrossRef]

37. Salman, R.; Ekstedt, F.; Damaschke, P. Branch-and-bound for the precedence constrained Generalized Traveling Salesman Problem. *Oper. Res. Lett.* 2020, 48, 163–166. [CrossRef]

38. Werner, M. Selection and ordering of points-of-interest in large-scale indoor navigation systems. In Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference, Munich, Germany, 18–22 July 2011; pp. 504–509.

39. Munkres, J.R. *Elements of Algebraic Topology*; Addison-Wesley Menlo Park: 1984; Volume 2.

40. Yan, J.; Zlatanova, S.; Diakité, A. A unified 3D space-based navigation model for seamless navigation in indoor and outdoor. *Int. J. Digit. Earth* 2021, 14, 985–1003. [CrossRef]

41. Idrees, A.; Iqbal, Z.; Ishfaq, M. An efficient indoor navigation technique to find optimal route for blinds using QR codes. In Proceedings of the 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), Auckland, New Zealand, 15–17 June 2015; pp. 690–695.

42. Watson, D.F. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput. J.* 1981, 24, 167–172. [CrossRef]

43. Xie, D.; Zhu, H.; Yan, L.; Yuan, S.; Zhang, J. An improved Dijkstra algorithm in GIS application. In Proceedings of the World Automation Congress 2012, Puerto Vallarta, Mexico, 24–28 June 2012; pp. 167–169.

44. Yan, J.; Diakité, A.A.; Zlatanova, S. A generic space definition framework to support seamless indoor/outdoor navigation systems. *Trans. GIS* 2019, 23, 1273–1295. [CrossRef]

45. Roth, J. Efficient many-to-many path planning and the Traveling Salesman Problem on road networks. *Int. J. Knowl.-Based Intell. Eng. Syst.* 2016, 20, 135–148. [CrossRef]

46. Boryczka, U.; Szwarc, K. The harmony search algorithm with additional improvement of harmony memory for asymmetric traveling salesman problem. *Expert Syst. Appl.* 2019, 122, 43–53. [CrossRef]

47. Anily, S.; Bramel, J. Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. *Nav. Res. Logist.* 1999, 46, 654–670. [CrossRef]

48. Monnot, J.; Paschos, V.T.; Toulouse, S. Approximation algorithms for the traveling salesman problem. *Math. Methods Oper. Res.* 2003, 56, 387–405. [CrossRef]

49. Alipour, M.M.; Razavi, S.N.; Derakhshi, M.R.F.; Balafar, M.A. A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. *Neural Comput. Appl.* 2018, 30, 2935–2951. [CrossRef]

50. Choong, S.S.; Wong, L.P.; Lim, C.P. An artificial bee colony algorithm with a modified choice function for the traveling salesman problem. *Swarm Evol. Comput.* 2019, 44, 622–635. [CrossRef]