A Novel Shortest Path Routing Algorithm for Wireless Data Collection in Transportation Networks

Izzet Fatih Senturk
Dept. Computer Engineering
Bursa Technical University
Bursa, TURKEY
izzet.senturk@btu.edu.tr

Gaoussou Youssouf Kebe
Dept. Computer Engineering
Bursa Technical University
Bursa, TURKEY
kebegaoussou@gmail.com

***Abstract*—This paper presents a heuristic solution to the shortest path routing problem in a road network where source and target locations are defined in a continuous plane. The problem arises when a wireless mobile entity is required to communicate with wireless nodes deployed apart from each other with a distance greater than their communication range and the mobility is limited with the transportation network layout. The goal is constructing a route which minimizes the mobility cost. Travelling salesman problem (TSP) is a classic discrete optimization problem to identify optimal routes between particular cities. Unlike TSP, our problem defines a neighborhood based on the node location and wireless communication range and seeks the optimal path between the neighborhoods. This paper defines mobility cost in terms of distance and delay separately. However, based on the defined cost function, the solution is applicable to other metrics as well including energy consumption.**

***Keywords*—travelling salesman problem, covering salesman problem, transportation network, graph, open street map, osmnx**

I. Introduction

The Travelling Salesman Problem (TSP) defines a set of cities with pairwise distances. The problem seeks a route for the salesman that visits each city exactly once and returns to the origin city while minimizing the total length of the cycle. TSP is an NP-hard combinatorial optimization problem [1] which dates back to 1800s [2] as a related mathematical problem named Hamiltonian cycle. The solution of the TSP problem has several applications for complex optimization problems including planning, scheduling, and logistics.

Over the years, various special cases have been defined for TSP. In the Euclidean TSP (eTSP), cities are defined as points in the plane and the distance between two cities is the Euclidean distance. Distances satisfy triangle inequality in eTSP. The Generalized Travelling Salesman Problem (GTSP) is an extension of TSP. In GTSP, cities are divided into subsets (i.e. clusters) and the salesman visits exactly one city from each cluster. GTSP is TSP if every cluster consists of a single node. The problem is also known as set TSP and Covering Salesman Problem.

Another variant of the TSP is Close-enough TSP (CETSP). In CETSP, the salesman does not have to visit the exact location of the city. An intuitive application area of CETSP involves problems employing wireless communication. Consider a utility company which equips its workers with radio frequency identification (RFID) technology for meter reading from residential customers. Instead of visiting the exact location of the meter, it is sufficient to approach the house close enough to read the meter. CETSP provides an opportunity to minimize the travel distance but also complicates the TSP problem [3]. CETSP is also studied as TSP with neighborhoods (TSPN) [4].

The problem that we consider in this paper is a variant of TSPN which also considers clusters. We regard the problem as Generalized Travelling Salesman Problem with Neighborhoods (GTSPN). GTSPN arises in partitioned wireless sensor networks (WSNs) when the data is to be collected from partitions with a mobile data collector (*MDC*). WSNs comprise wireless sensors to monitor an area of interest and track certain events. Sensors are typically equipped with limited on-board batteries and employ one of the low-power wireless communication solutions [5]. Due to the short-range communication, sensors must maintain multi-hop network-wide connectivity with a base station (*BS*) at all times. *BS* is less restricted in terms of its resources and provides remote connection to the application area. Contrary to clusters, which are not necessarily connected, partitions consist of connected nodes.

WSNs are prone to node failures due to various reasons including battery depletion, hardware malfunction, and external damage inflicted by the inhospitable surroundings [6]. Depending on the failure location and its scope, the network may be subject to severe consequences including coverage and connectivity loss. Failure of a sensor that serves on a routing path exclusively (i.e. cut-vertex) divides the network into partitions isolated from the rest of the network. The lack of data exchange and coordination among partitions impairs the fidelity of the collected data. One of the reactive solutions to tolerate node failures is employing an *MDC*. *MDC* periodically visits partitions to collect data and provides intermittent connectivity between partitions and the *BS*. However, delay between successive visits impairs data latency [7]. Furthermore, mechanical motion incurs excessive energy consumption [8]. Therefore, it is imperative to optimize the *MDC*'s route in order to extend the network lifetime while minimizing the data latency.

Our problem is similar to TSP in the sense that a set of cities (i.e. partitions) must be visited exactly once by the salesman (i.e. *MDC*). Unlike TSP, our problem requires the *MDC* to visit each partition periodically. Therefore, the *MDC*

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Grant No. EEEAG-117E050.

follows the same cycle multiple times in an iterative manner until its battery is depleted. On the other hand, each partition will be visited only once at each iteration. The cycle starts and ends at the partition which comprises the *BS*. Fig. 1 illustrates a sample solution for the GTSPN problem. Note that the application area in Fig. 1 considers an Euclidean plane where mobility is possible in any direction. However, this paper assumes a smart city application where the sensors are deployed on the transportation infrastructure to improve public safety [5]. Therefore, mobility of the *MDC* is limited with the transportation network layout.

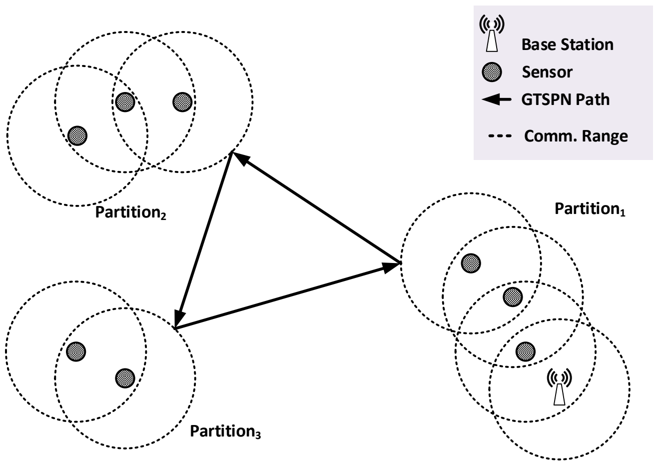

Fig. 1. Disks centered at respective sensor locations represent the wireless communication range. Unit disk graph model is assumed for wireless communication. *MDC* follows the route obtained from the GTSPN solution to provide intermittent network connectivity.

We employ one of the volunteered geographic information (VGI) systems, OpenStreetMap [9], to obtain the road network of various metropolitan cities in Turkey. OpenStreetMap (OSM), models the physical world with three basic components, namely node, way, and relation. Node signifies a particular point on the earths surface with the respective latitude and longitude coordinates. Way, on the other hand, defines a polyline with an ordered list of nodes. In OSM, way represents a road segment in the road network. Points for sensor deployment are determined based on the respective coordinates of nodes constituting ways.

Considering the road network for mobility complicates the problem even further. GTSPN problem arises on a transportation network even for two sensors. Assume a scenario where two sensors are located as given in Fig. 2. The disks centered at the respective sensor location represent the wireless communication range. The radius of the disk is set according to the employed communication range which is same for both sensors. In order to collect data from a particular sensor, it is sufficient to stop at corresponding road segment-circle intersections denoted with red nodes in Fig. 2.

In this paper, we investigate the GTSPN problem between two sensors deployed on the transportation infrastructure. The road network is modeled as a graph structure. Road segments are denoted by directed edges. Based on the desired cost function we set the edge weights accordingly. This paper considers distance and travel time separately as the mobility cost. However, the solution can be extended to cover other metrics (i.e. energy consumption) as well. Based on the employed communication range, several locations can be identified to collect data from the respective sensor. A brute-

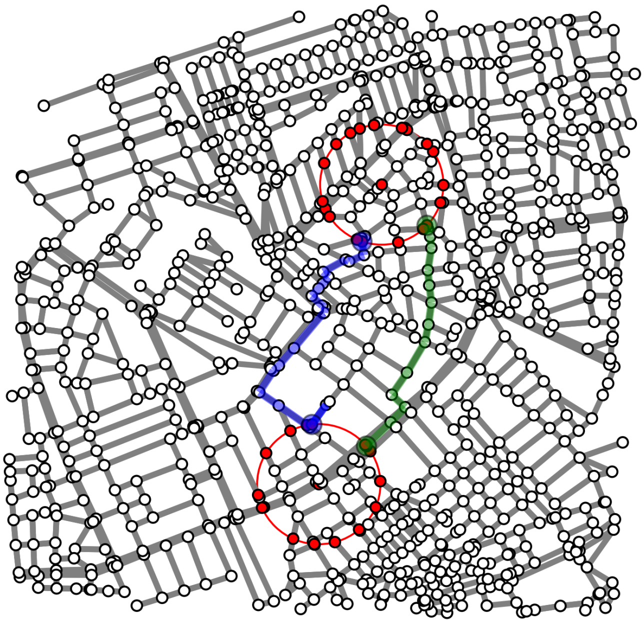

Fig. 2. Partial road network from the metropolitan area of Antalya, TURKEY. From the city center, the nodes within a bounding box of 1500 meters are included. Green and blue colors denote the paths obtained from *TSP-BF* and *TSP-FD* respectively. Please refer to Section IV-C for the details.

force approach is considering all possible permutations and selecting the pair of locations with the least mobility cost. Despite providing the optimal solution, this approach does not scale well. Given the high computational cost of this approach, we present a heuristic which performs very close to the optimal solution.

The rest of the paper is organized as follows. Related work is summarized in Section II. The proposed solution is discussed in Section III. Approaches are evaluated in Section IV. The paper is concluded in Section V.

II. Related Work

Fault-tolerance is considered as a form of topology management in WSNs [6]. The main goal is sustaining network connectivity while ensuring a certain level of coverage. Fault-tolerance solutions can be classified into three broad categories according to the applied technique for recovery. The first group of solutions assumes possible intervention to the application area and deploys additional relay nodes to link the partitions [10]. Relay placement solutions pose two different challenges. First, it may be infeasible to intervene the application area. Second, recovery will not be successful unless sufficient number of relays are deployed. The solutions in the second group assumes inherent sensor mobility and restructures the network topology through controlled mobility [11]. Despite the flexibility of reactive schemes, this type of solutions increases the cost of initial deployment. In the third category, a mobile entity (i.e. *MDC*) visits partitions periodically and collects the sampled data. The mobile provides intermittent network connectivity by forwarding the collected data to the *BS*. This scheme not only enables a reactive solution without intervening the application area but also avoids the higher initial deployment cost of mobile sensors. On the other hand, intermittent connectivity creates increased data latency based on the tour length and the velocity of the *MDC*. Minimizing the tour length alleviates both the energy cost of mobility and the data latency. Therefore, it is of paramount importance to minimize the *MDC*'s tour length.

TSP is a well-studied optimization problem with a wide array of applications [12] and various extensions including asymmetric TSP [13], multiple travelling salesmen problem [14], generalized TSP [15], close-enough TSP [3], and TSP with neighborhoods [4]. TSP is NP-hard even for points in the Euclidean plane [1]. Several exact algorithms have been proposed for the TSP [16]. Exact solutions typically model TSP as an integer linear programming (ILP) problem. However, ILP solutions do not scale well and heuristics are pursued for large datasets instead.

TSP has been applied to WSNs as well [17], [18]. [17] assumes a single *MDC* to collect data from WSNs similar to this paper. However, *MDC* collects data from each sensor in a single-hop communication in [17]. Recall that, we consider partitions with one or more sensors and it is sufficient to collect data from the whole partition upon contacting one of the sensors in the partition through multi-hop communication. Determining partitions' contact nodes complicates the problem. [18] presents a partitioning-based *MDC* scheduling algorithm to collect data from partitions. Unlike our paper, partitions do not denote connected network segments in [18] and nodes are grouped into partitions according to their data generation rate and location. [18] also assumes mobility is possible in any direction. On the other hand, we assume mobility dictated by the transportation network layout.

III. Proposed Solution

Finding an optimal route from a start location to a target location (i.e. route planning) in a road network is a challenging task which has attracted considerable attention due its wide applicability in the real-world scenarios [19]. In this paper, we obtained the geospatial data from from OpenStreetMap [9] by employing OSMnx [20]. Note that the road network can be modeled by using the graph data structure. Graphs are mathematical abstractions to model pairwise relations between objects. A graph is comprised of nodes connected with edges. In the obtained road network, edges represent road segments and nodes signify intersections. All the spatial characteristics including geographic and metric information is preserved. The edges are weighted according to the employed cost function (e.g. length of the road segment, travel time on the road segment). The edges are directed. Self-loops and multiple edges between nodes are possible (i.e. multidigraph). Note that the resulting spatial network is not planar since the edges may not only intersect on nodes considering tunnels and overpasses.

For two given sensor locations, we designate discrete data collection points in the plane by defining disks centered at the respective sensor location with a radius of R_c. Disks denote the areas where wireless communication is possible with the respective sensor. In order to collect data, it is sufficient to stop on the circumference of the disk. Since the *MDC* can only follow the road network, we determine the points where the road network intersects the disk. For two sensors, we identify two sets of data collection points. The next step is to identify the optimal pair of data collection points as well as the routing path on the road network. The intuitive approach is applying Dijkstra's algorithm for the possible permutations and selecting the one with the shortest path. This is regarded as the brute-force approach and employed as one of the baselines as discussed in Section IV-C. However, this approach does not scale well and we present various heuristics as detailed next.

A. Vector-based Heuristic (VH)

Our first attempt to solve the routing problem considers the fact that regions with higher road density are more likely to offer better routes. Assume a region with a forest or a lake. Topographic features can dictate the pattern of the road network causing extended paths. We can identify and avoid regions with obstructing topographic features by evaluating the road density towards the movement direction. The idea is defining a vector originating from the respective data collection point with a direction towards the outbound of the disk on the current road segment. Recall that the data collection point already resides on the road segment. The challenging part is the randomness of the road network. Despite the deterministic part of the road network where one can define the source and the target, it is highly likely to change the direction from time to time while following the path. Therefore, we define a constant (i.e. l) to denote the number of consecutive road segments to be considered. For multiple vectors, we compute the resulting direction based on the vector addition operation. We have varied l between 1 and 4 to investigate its impact on the performance and reported the results in Table I. We have employed the *Point-to-Point* baseline defined in Section IV-C. *Point-to-Point* does not consider the communication range and provides the shortest path between sensors. As can be noticed from Table I, VH performs worse than the baseline and increasing l does not improve its performance. Therefore, we present another heuristic next.

TABLE I
Total tour length (meters) for *VH* with respect to l. Communication range is set to 200 meters.

Point-to-Point	***VH (l=1)***	***VH (l=2)***	***VH (l=3)***	***VH (l=4)***
1444,76	1719,78	1744,08	1783,48	1761,28

B. Shortest Path Routing with Data Collection (SPR-DC)

SPR-DC defines a disk for each sensor to denote the area for wireless data collection from the corresponding sensor. The disk is centered at the respective sensor location and has a radius of R_c. The circumference of the disk signifies the boundary where data collection can start. Therefore, we obtain the circle at the edge of the disk and compute the road segment-circle intersections using the road network. Then we compute the shortest path between two sensors by exploiting Dijkstra's algorithm. In this step, we do not consider the communication area. Finally, we identify the data collection points which intersect the computed shortest path. It is guaranteed that the shortest path will intersect each circle at exactly one the data collection point for each sensor. *MDC* follows the obtained shortest path between the identified data collection points.

IV. Experimental Evaluation

A. Experiment Setup

In the experiments, we assumed unit disk graph model for wireless communication. According to this model, two nodes can communicate if and only if the distance between them is at most R_c which is the common communication range of the nodes. To assess the impact of the network size and the communication range on performance metrics, we considered two different scenarios. In the first scenario, we fixed R_c and changed the size of the application area. In the second scenario, R_c is constant while the size of the application area changes. In order to control the size of the application area, we use the bounding box distance in each direction (north, south, east, and west) from city centers obtained from OSM. In the experiments, we considered 30 metropolitan cities in Turkey [21] and reported the average result for significance.

To determine the sensor locations, we employed k-means algorithm. We set k=2 and designate 2 clusters by minimizing the variance of the clusters being merged. After designating clusters, we deployed the sensors at the centroid of each cluster. Note that the sensor does not have to be on the road network. While computing travel time, we use the speed limits for road segments. The default speed is set to 50 kilometers per hour if the speed limit is not defined for a particular road segment in the obtained path.

B. Performance Metric

We have employed the following metrics to evaluate the performance of the proposed solutions.

- *Total tour length*: This metric measures the total travel distance of the salesman (i.e. *MDC*).
- *Travel time*: This metric measures the total time required to complete the tour.

C. Baselines

We have considered three different baselines to construct the tour, namely Point-to-Point tour (*Point-to-Point*), TSP with brute-force (*TSP-BF*), and TSP with the minimum flying distance (*TSP-FD*). As the name suggests, *Point-to-Point* designates the shortest path between the respective sensor locations without considering the communication range. *TSP-BF* defines a disk, for each sensor, to denote the communication range. The disk is centered at the respective sensor location and has a radius equal to R_c. Afterwards, the road segments that intersect the circumference of the respective disk are identified. Line-circle intersections provide one or more points that *MDC* can stop and collect data from the corresponding sensor. *TSP-BF* examines all possible permutations of such points among different sensors and selects the one with the shortest path. *TSP-FD* also determines the data collection points through line-circle intersections but unlike *TSP-BF*, selects the pair of data collection points based on the flying distance between them. In Fig. 2, the green color signifies the path obtained from *TSP-BF*. The blue color denotes the path obtained from *TSP-FD*.

D. Performance Results

In this subsection, we discuss the performance of the proposed solution in terms of the defined performance metrics. The experiments are conducted on networks with varying size and communication range. *SPR-DC* denotes the proposed solution in the rest of the paper.

1) Total tour length: Total tour length of the approaches are presented in Tables II and III with respect to the employed communication range and the bounding box distance respectively. Table II indicates that the total tour length declines for all approaches except *Point-to-Point* with the increased communication range. This can be attributed to the extended disk size representing the area for wireless communication. Recall that *Point-to-Point* does not consider the disk for wireless communication and computes the path between the respective sensor locations. It can be observed from Table II that the minimum tour length can be attained by *TSP-BF*. This is expected since *TSP-BF* is a brute-force approach that examines all possible permutations before identifying the shortest path. *Point-to-Point* incurs the highest cost due to avoiding the wireless communication. *SPR-DC*, on the other hand, performs very close to the *TSP-BF* and outperforms *TSP-FD*.

TABLE II
TOTAL TOUR LENGTH (METERS) WITH RESPECT TO COMMUNICATION RANGE. THE BOUNDING BOX DISTANCE IS SET TO 1000 METERS.

TR	*Point-to-Point*	*SPR-DC*	*TSP-BF*	*TSP-FD*
50	1473,17	1363,60	1333,16	1386,80
100	1473,17	1233,86	1174,65	1301,59
150	1473,17	1110,73	1035,77	1172,03
200	1473,17	968,09	863,50	1023,33

Table III reveals that the total tour length increases for all approaches with the extended bounding box distance. Considering the fixed communication range in the expanded road network, the average distance between sensors are expected to be higher. As earlier, *Point-to-Point* incurs the highest cost while *TSP-BF* provides the shortest path. *TSP-BF* and outperforms *TSP-FD* but the performance gap between them drops from 21% to 5% when the bounding box distance is increased from 500 meters to 1000 meters. The gap declines further with the increased bounding box distance.

TABLE III
TOTAL TOUR LENGTH (METERS) WITH RESPECT TO THE BOUNDING BOX DISTANCE. COMMUNICATION RANGE IS SET TO 1000 METERS.

BBox Distance	*Point-to-Point*	*SPR-DC*	*TSP-BF*	*TSP-FD*
500	899,75	645,46	535,23	694,73
1000	1473,17	1233,86	1174,65	1301,59
1500	2008,30	1774,33	1722,90	1769,58
2000	2612,03	2367,56	2296,13	2417,51

2) Travel time: Tables IV and V signify the travel time with respect to the employed communication range and the bounding box distance respectively. It can be noticed from Table IV that the travel time decreases for all approaches except *Point-to-Point* when the communication range is extended. This is expected since the average distance between sensors are more likely to decrease with the extended communication range in a network of a fixed size. *TSP-BF* offers the best performance again. On the other hand, *SPR-DC* performs very close to the brute-force solution and outperforms *TSP-FD*.

TABLE IV
DATA DELAY (SECONDS) WITH RESPECT TO COMMUNICATION RANGE. THE BOUNDING BOX DISTANCE IS SET TO 1000 METERS.

TR	*Point-to-Point*	*SPR-DC*	*TSP-BF*	*TSP-FD*
50	106,07	103,45	100,61	104,69
100	106,07	94,92	90,16	101,55
150	106,07	87,12	80,66	93,50
200	106,07	76,11	67,11	81,12

Table V indicates that the travel time increases for all approaches when the network size is larger. Given the communication range is fixed, the average distance between sensors is expected to increase when the bounding box distance is increased. As justified earlier, *TSP-BF* provides the best performance. *SPR-DC* performs very close to *TSP-BF* especially in larger networks. The performance gap between *SPR-DC* and *TSP-BF* declines from 22% to 5% when the bounding box distance is increased from 500 meters to 1000 meters. The gap declines even further when the bounding box distance is increased.

TABLE V
DATA DELAY (SECONDS) WITH RESPECT TO THE BOUNDING BOX DISTANCE. COMMUNICATION RANGE IS SET TO 1000 METERS.

BBox Distance	***Point-to-Point***	***SPR-DC***	***TSP-BF***	***TSP-FD***
500	64,80	53,07	43,38	57,02
1000	106,07	94,92	90,16	101,55
1500	144,76	133,68	128,70	133,85
2000	187,26	176,31	171,01	183,19

V. CONCLUSION

In this paper, we tackled with the problem of short path routing in road networks where source and target locations are defined in a continuous plane. In this problem, road network dictates the mobility direction and the velocity is set according to the defined speed limits for corresponding road segments on the path. We assumed a smart city application where sensors are deployed to track certain events on the transportation infrastructure to improve public safety. For each sensor, an area can be defined for wireless data collection. To optimize the data collection process, we needed a routing algorithm which provides the best path according to the defined cost function. Unlike travelling salesman problem, our problem considers a continuous plane for each sensor based on the sensor location and the communication range. We presented a heuristic solution to identify data collection points for each sensor on the road network and the best path between such points based on the total tour length and the travel time separately. We conducted extensive simulations to evaluate the proposed scheme. The results indicate that the proposed solution outperforms baselines and performs very close to the optimal solution.

ACKNOWLEDGMENT

Map data copyrighted OpenStreetMap contributors and available from https://www.openstreetmap.org

REFERENCES

[1] C. H. Papadimitriou, "The euclidean travelling salesman problem is np-complete," *Theoretical Computer Science*, vol. 4, no. 3, pp. 237 – 244, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304397577900123

[2] N. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory, 1736-1936*. Oxford University Press, 1986.

[3] X. Wang, B. Golden, and E. Wasil, "A steiner zone variable neighborhood search heuristic for the close-enough traveling salesman problem," *Computers & Operations Research*, vol. 101, pp. 200 – 219, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054818302132

[4] A. Dumitrescu and J. S. Mitchell, "Approximation algorithms for tsp with neighborhoods in the plane," *Journal of Algorithms*, vol. 48, no. 1, pp. 135 – 159, 2003, twelfth Annual ACM-SIAM Symposium on Discrete Algorithms. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0196677403000476

[5] I. F. Senturk and G. Y. Kebe, "A new approach to simulating node deployment for smart city applications using geospatial data (to appear)," in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, Nov 2019.

[6] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Computer Networks*, vol. 58, pp. 254 – 283, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613002879

[7] A. Bari, Y. Chen, D. Roy, A. Jaekel, and S. Bandyopadhyay, "Designing hierarchical sensor networks with mobile data collectors," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 128 – 139, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119210000611

[8] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 4, March 2005, pp. 2302–2312 vol. 4.

[9] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org. [Online]. Available: https://www.openstreetmap.org

[10] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Networks*, vol. 24, pp. 1 – 19, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870514001437

[11] I. F. Senturk, "Partition-aware centrality measures for connectivity restoration in mobile sensor networks," *International Journal of Sensor Networks*, vol. 30, no. 1, pp. 1–12, 2019.

[12] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton university press, 2006.

[13] J. Cirasella, D. S. Johnson, L. A. McGeoch, and W. Zhang, "The asymmetric traveling salesman problem: Algorithms, instance generators, and tests," in *Workshop on Algorithm Engineering and Experimentation*. Springer, 2001, pp. 32–59.

[14] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[15] M. Fischetti, J. J. Salazar González, and P. Toth, "A branch-and-cut algorithm for the symmetric generalized traveling salesman problem," *Operations Research*, vol. 45, no. 3, pp. 378–394, 1997.

[16] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231 – 247, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/037722179290138Y

[17] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1472–1483, May 2013.

[18] Yaoyao Gu, D. Bozdag, E. Ekici, F. Ozguner, and Chang-Gun Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *2005 Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005.*, Sep. 2005, pp. 386–395.

[19] 9th DIMACS Implementation Challenge, "Shortest Paths," http://www.dis.uniroma1.it/challenge9/, 2006, [Online; accessed 27-May-2019].

[20] G. Boeing, "Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126 – 139, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971516303970

[21] Wikipedia, "Metropolitan municipalities in Turkey," https://en.wikipedia.org/wiki/Metropolitan_municipalities_in_Turkey, [Online; accessed 27-May-2019].