Robot-Assisted Sensor Network Deployment and Data Collection

Yu Wang Changhua Wu

***Abstract*— Wireless sensor networks have been widely used in many applications such as environment monitoring, surveillance systems and unmanned space explorations. However, poor deployment of sensor devices leads (1) bad network connectivity which makes data communication or data collection very hard; or (2) redundancy of coverage which wastes energy of sensors and causes redundant data in the network. Thus, in this paper, we propose using a mobile robot to assist the sensor deployment and data collection for unmanned explorations or monitoring. We assume that the robot can carry and deploy the sensor devices, and also have certain communication capacity to collect the data from the sensor devices. Given a set of interest points in an area, we study the following interesting problems: (1) how to decide minimum number of sensor devices to cover all the interest points; (2) how to schedule the robot to place these sensor devices in certain position so that the path of the robot is minimum; and (3) after the deployment of sensors, how to schedule the robot to visit and communicate with these sensor devices to collect data so that the path of the robot is minimum. We propose a complete set of heuristics for all these problems and verify the performances via simulation.**

I. Introduction

Wireless sensor networks [1] have tremendous prospects due to their relatively lower cost and capability of obtaining valuable information from locations that are beyond human reach. A sensor network consists of a set of sensor nodes[1] that spread over a geographical area. These sensors are able to perform processing as well as sensing and are additionally capable of communicating with each other. Due to its wide-range potential applications such as battlefield, emergency relief, environment monitoring, surveillance system, space explorations, and so on, wireless sensor network has recently emerged as a premier research topic. Most of current research on wireless sensor networks assume the cost of each sensor is cheap thus the number of sensors in a network could be sufficient large (hundreds or thousands) to cover the target area and maintain the network connectivity. However, in many real applications (such as space exploration), certain kind of sensor devices could be very expensive, and it is impossible to have thousands of them to deploy. In addition, since the sensors would have relatively weak radios, inter-node separation is very common in sensor networks. On the other hand, even if the number of sensor is sufficient and the radio is strong enough, poor deployment of sensor devices could also lead to (1) bad network connectivity which makes data communication or data collection very hard; or (2) redundancy of coverage which wastes energy of sensors and causes redundant data in the network. Thus, in this paper, we propose using a mobile robot to assist the sensor deployment and data collection for unmanned explorations or monitoring. We assume that the robot can carry and deploy sensor devices, and also have certain communication capacity to collect the data from these sensor devices.

Recent years have seen the growing interest in mobile sensor networks [2]–[7] or robot-assisted sensor networks [8]–[11]. In mobile sensor networks, all or partial of the sensor nodes have motion capability endowed by robotic platforms. Mobile sensor networks have more flexibility, adaptively and even intelligence compared with stationary wireless sensor networks. Mobile sensors can dynamically reposition themselves to satisfy certain requirements on monitoring coverage, network connectivity, or fault tolerance. However, to make every sensor have motion capability increases the cost of each sensor and maybe not feasible in most applications. On the other hand, robots are large complex systems with powerful resources and can interact with sensor nodes. The new paradigm of robot assisted sensor networks is of ubiquitous sensors embedded in the environment with which the robot interacts: to deploy them, to harvest data from them, and to task them. In turn, the sensors can provide the robot with models that are highly adaptive to changes in the environment and can re-task the robots with feedback from sensors. Therefore, we believe that robotics will have a profound effect on sensor networks.

Most previous research on robot-assisted sensor networks [8]–[11] study using the robot to achieve coverage, localization and navigation. In this paper, we focus on coverage and path planning. Given a set of interest points in an area, we study the following interesting problems: (1) how to decide minimum number of sensor devices to cover all the interest points; (2) how to schedule the robot to place these sensor devices in certain position so that the path of the robot is minimum; and (3) after the deployment of sensors, how to schedule the robot to visit and communication with these sensors to collect data so that the path of the robot is minimum. An illustration of this scenario is depicted in Figure 1 where the rover using one path to deploy the sensors and the other path to collect data from deployed sensors. We propose a complete set of heuristics for all these problems and verify the performances via simulations.

A potential application of our proposed robot-assisted sensor network design is for unmanned space explorations. Unmanned space explorations have tremendous prospects

Yu Wang is with Department of Computer Science, University of North Carolina at Charlotte, USA. Email: `yu.wang@uncc.edu`. The work of Wang is supported, in part, by funds provided by Oak Ridge Associated Universities and the University of North Carolina at Charlotte.

Changhua Wu is with Department of Science and Mathematics, Kettering University, USA. Email: `cwu@kettering.edu`

[1]In this paper the term *node* often represents a *sensing device* or called a *sensor*. We often interchange them here.

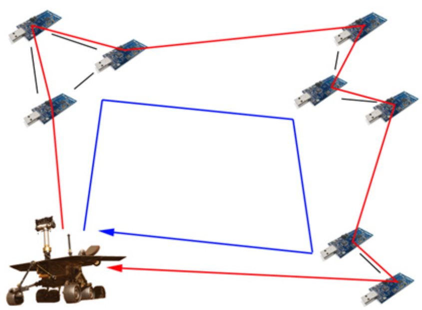


Fig. 1. Illustration of the scenario where a robot (a rover) deploys sensor nodes around interest points and assists to collect data from the unconnected sensor network. Here, the red path is the deployment path, while the blue one is the data collection path.

due to their relatively lower cost and capability of obtaining valuable information from locations that are beyond human reach. The impact of unmanned missions and the use of automated remote monitoring stations and robotic platforms in space have been observed from several successful ventures in the past. Examples include the NASA Mars rovers that are designed to negotiate unpredictable surface conditions and provide valuable data, video samples as well as physical samples through remote control. Our proposed approach can allow the robot (rover) efficiently deploy and maintain the sensor networks which enable data collection over large areas over extended periods of time. The proposed coordinated remote data deployment and collection approach can extend the reach and lifetime of both space rovers and smart sensors.

II. Related Work

Sensor Coverage: Since each sensor covers a limited area, adequate coverage of a large area requires appropriate placement of sensors based on collective coverage and cost constraints. The previous research on sensor coverage mainly focuses on studying how to determine the minimum set of sensors for covering every location or certain objects (interest points) in the target field. Different coverage models and methods are surveyed by Cardei and Wu [12].

Robot-Assisted Sensor Networks: Mobile or robot-assisted sensor networks have been studied recently. Most previous research concentrate on using the robot or mobile sensors to help sensor network to achieve coverage [2], [5], localization [6]–[9], [11], target detection [3], fault-tolerance [4], [10], and navigation [9]. In this paper, we study how to use a robot assisting the sensor deployment and data collection, with a focus on efficient path planning.

Path Planning: One of the most important problems in robotics is path planning (or called motion planning) [13], [14], which is aimed at providing robots with the capability of deciding automatically which motions to execute in order to achieve certain specific goals. It arises in a variety of forms. The common form requires finding a short geometric collision-free path for a single robot in a known static workspace. In this paper, we do not focus on such kind of path planning. The problem we concentrate on is similar to a well-known graph theoretic problem, the *traveling salesman problem* [15]. We assume that the 2D space does not have any obstacles and the robot can move towards any direction freely. The objective of our path planning is to minimize the total length of the path which the robot travels. We study how to deploy sensors and schedule the robot path such that the total travel distance is minimized and the coverage is guaranteed.

III. Models and Problems

A. Models

We assume that a set of m interest points (or called targets), denoted by $P = \{p_1, p_2, \cdots, p_m\}$, are distributed in a 2-dimensional plane. The objective of our mission is to deploy a set of sensor devices, denoted by $S = \{s_1, s_2, \cdots, s_n\}$ to form a sensor network to monitor or track these interest points. Each sensor node s_i is equipped with a sensor which can monitor a disk region centered at s_i with radius r_S, i.e., if the distance between p_l and s_i is less than r_S then sensor s_i can monitor the interest point p_l. We assume that single sensor can monitor multiple points inside its sensing region. Each sensor node s_i has an omnidirectional antenna so that it can talk to all sensor nodes or the robot within a disk region centered at s_i with radius r_T. Hereafter, we call r_S and r_T the *sensing range* and the *transmission range* respectively. We assume all sensor nodes are equipped with same hardware devices, thus, they have the same[2] fixed r_S and r_T. We assume the robot R has a larger transmission range than the sensor node, i.e., it can talk with sensor node s_i if it is inside the transmission range of s_i. The robot parks at point v_0 initially and need to return v_0 after all operations. It can travel to any point in the 2-dimensional plane during the operations.

B. The Problem

The problem we study is how to efficiently schedule a robot to (1) deploy a set of sensor nodes S to guarantee the coverage of all interest points P and (2) collect data from these sensor nodes. Here, the efficiency of the path scheduling means the scheduled path for the robot to travel is shortest. We treat this problem as two sub-problems separately: *deployment problem* and *data collection problem*.

For the deployment problem, given the set of interest points P, we study how to find the positions $V = \{v_1, v_2, \cdots, v_n\}$ of sensor nodes S where they will be deployed by the robot, such that (1) the sensor network guarantees the full coverage of all interest points P and uses the minimum number n of sensor nodes S; and (2) the path $\Pi^D = v_0 v_1 v_2 \cdots v_n v_0$ which the robot will travel to deploy sensors at those positions has the minimum total length.

For the data collection problem, given the set of deployed sensors V, we study how to find the turning positions (or called pause points) $U = \{u_1, u_2, \cdots, u_k\}$ where the robot pauses and collects data from sensors, such that (1) the robot

[2] However, our proposed methods can be easily extended to the case with heterogeneous sensing and transmission ranges.

can communicate with every sensor during the round trip and make the minimum number k of stops; and (2) the path $\Pi^C = v_0 u_1 u_2 \cdots u_k v_0$ which the robot will travel to collect data on those pause points has the minimum total length.

Notice that the deployment problem and the data collection are essentially the same except the range of coverage is different (one uses the sensing range, the other uses the transmission range), thus we use the same set of heuristics to solve these two problems.

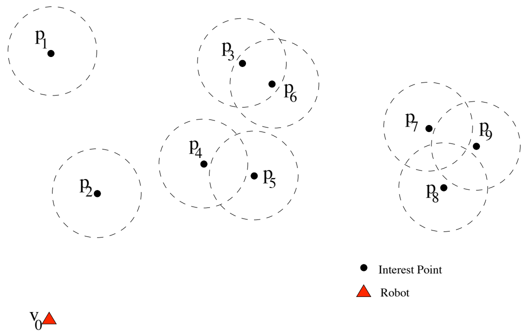


Fig. 2. The set of interest points P (black nodes) and the initial position of the robot (red triangle). Here, 15 areas can be defined as A by the 9 sensing disks and their intersections.

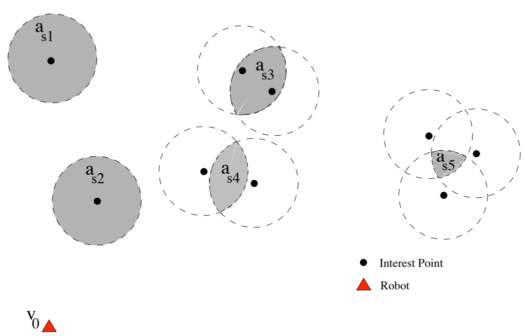


Fig. 3. Grey areas are the subareas selected by Algorithm 1 where a sensor needs to be deployed.

IV. Robot-Assisted Sensor Deployment

In this section, we describe our algorithm for how to deploy the sensors with assistance from the mobile robot. As shown in Figure 2, we first use the sensing range r_S to draw a disk D_i^S for each interest point p_i. To guarantee all interest points are covered by sensors, we need at least one sensor node inside each disk D_i^S to monitor p_i. However, one sensor can sit in the intersection of multiple disks to monitor multiple targets. Thus, we define the areas formed by the disks and their intersections, denoted by $A = \{a_1, a_2, \cdots, a_l\}$, putting a sensor in an area a_i covers one or multiple interest points. The first optimization problem is how to select the minimum number of areas to deploy sensor nodes to guarantee the coverage. This problem is actually the minimum set cover problem which aims to find the minimum number of subsets to cover the whole space. The minimum set cover problem is a NP-hard problem [15]. However, there exists many heuristics for it. The simplest and most classical method is a greedy method, in which you always greedily select the subset which can cover the maximum number of uncovered elements. This greedy algorithm can achieve an approximation ratio of $O(\ln s)$ where s is the size of the largest subset. Inapproximability results [16], [17] show that the greedy algorithm is essentially the best-possible polynomial time approximation algorithm for set cover under plausible complexity assumptions. Algorithm 1 shows our greedy algorithm and Figure 3 illustrates the results from Algorithm 1 on the example shown in Figure 2.

Algorithm 1 Greedy algorithm to select the minimum number of areas where to deploy sensors

Input: A set of areas $A = \{a_1, a_2, \cdots, a_l\}$ and a set of interest points $P = \{p_1, p_2, \cdots, p_m\}$.
Output: A subset of areas $A_S = \{a_{s_1}, a_{s_2}, \cdots, a_{s_n}\}$ to place the n sensors S.

Initially, set all interest points *uncovered* and the uncovered counter $k = m$. Let the potential coverage c_i of each area a_i equal to the number of disks D_j^S intersecting with this area. Here, each interest points p_j could be covered by a sensor placed in area a_i. We call p_j can be covered by a_i.
while $k! = 0$ **do**
  Select area a_j with the largest potential coverage c_j (using IDs to break a tie) and add it into the selected subset A_S;
  Mark all interest points covered by a_j *covered*;
  $k = k - c_j$;
  Update the c_k for all adjacent areas a_k.
end while

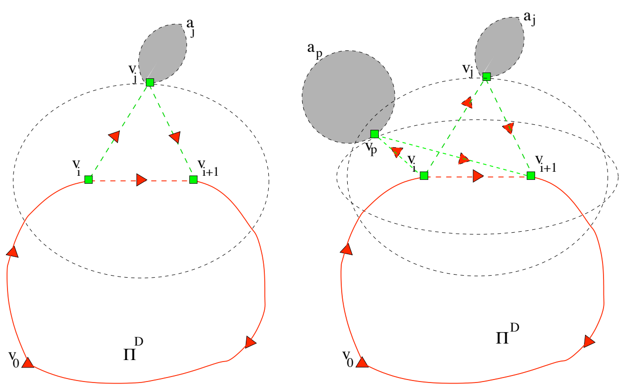
After selecting the area to place the sensors, we need to decide their exact positions. Since the positions can affect the total length of the path that the robot needs to visit, we consider the position problem joint with the path schedule problem. In other words, we propose an algorithm to schedule the robot to deploy the sensors in each selected area a_{s_i}, so that the total length of the path travelled by the robot is minimum. This problem is actually the *traveling salesperson problem with neighborhoods* (TSPN) which is also a NP-hard problem [18]. The classical TSP studies what is the shortest round-trip route that visits each point exactly once and then returns to the starting point, given a set of point in a plane. TSPN studies what is the shortest round-trip route that visits each area exactly once and then returns to the starting area, given a set of areas. There are several approximation algorithm exists for TSPN, however most of them are very complex and not practical at all. Our algorithm is an iterative algorithm in which each step we add a new turn point inside one of the unvisited areas such that the distance added to the robot path is minimum. Assume, we have n areas needed to be visited (deploying the sensor) and initially all areas are unvisited, the algorithm will terminate after n rounds, since each round it adds a new turn point in the path and covers an unvisited area. Algorithm 2 shows the detailed algorithm.

Algorithm 2 Path schedule and sensor placement algorithm: to select the turn points of the robot to deploy sensors

---

Input: A set of areas $A_S = \{a_{s_1}, a_{s_2}, \cdots, a_{s_n}\}$.
Output: A path $\Pi^D = v_0 v_1 v_2 \cdots v_n v_0$ which the robot use for sensor deployment.

1: Initially, set all selected areas a_{s_i} *unvisited* and the unvisited counter $k = n$. Let the path $\Pi^D = v_0 v_0$.
2: **while** $k! = 0$ **do**
3: For each edge on $v_i v_{i+1}$ in path Π^D and every unvisited area a_j, we draw an ellipse which uses v_i and v_{i+1} as its foci and is tangent to a_j. Let v_j be the tangent point. See 4(a) for illustration. If select a_j to visit between a_i and a_{i+1}, the distance added to the path Π^D will be $||v_i v_j|| + ||v_j v_{i+1}|| - ||v_i v_{i+1}||$.
4: It is obvious that we want to select the unvisited area which adds the least distance to path Π^D. For example, in Figure 4(b), a_p, hence v_p, is a better choice than a_j. Assume we select a_p which is the best for all edges in Π^D and all unvisited areas, we mark a_p visited, and insert v_p between v_i and v_{i+1} in Π^D. Thus the number of edges in the path increases by one. $k = k - 1$.
5: **end while**



(a) (b)

Fig. 4. (a) For each edge on $v_i v_{i+1}$ in path Π^D and every unvisited area a_j, we draw the ellipse which uses v_i and v_{i+1} as its foci and is tangent to a_j. The distance added to the path Π^D by visiting a_j is $||v_i v_j|| + ||v_j v_{i+1}|| - ||v_i v_{i+1}||$. (b) We select the unvisited area which adds the least distance to path Π^D. In this example, a_p is a better choice than a_j.

Notice that in Step 3 of Algorithm 2 we need to draw an ellipse which is tangent to a_j. This can be done by two ways. We can start with a small ellipse and increase its size until it reaches a_j. However, how to decide the initial size of the ellipse and what size to increase at each step are difficult to answer. The second way to do is using binary search. We first randomly select a point b inside a_j. We use $||v_i b|| + ||v_{i+1} b||$ as the major axis to draw the ellipse which guarantees to intersect with a_j. Then we reduce the major axis by half, if the ellipse does not intersect with a_j, we increase the major axis, otherwise further reduce it. By recursively doing this, we can find the ellipse which is tangent to a_j efficiently. In practice, if the sensing range is small compared with the distance between all areas, we can just use the ellipse via b to estimate the optimal ellipse.

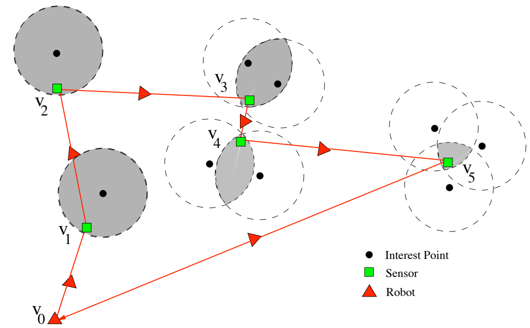


Fig. 5. Path Π^D (red line) generated by Algorithm 2. Here, green squares are the positions to place the sensors (also the turn points of the robot).

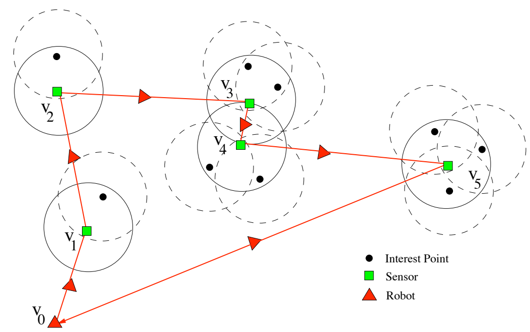


Fig. 6. The deployed sensors (green squares) and their sensor ranges (solid circles) after the sensor deployment phase.

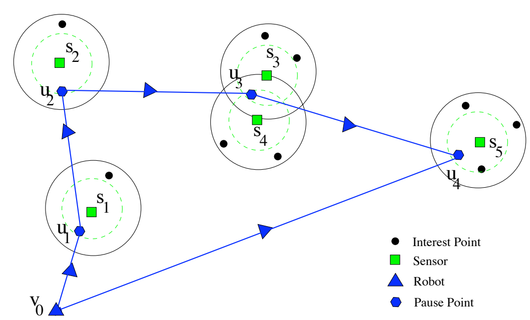
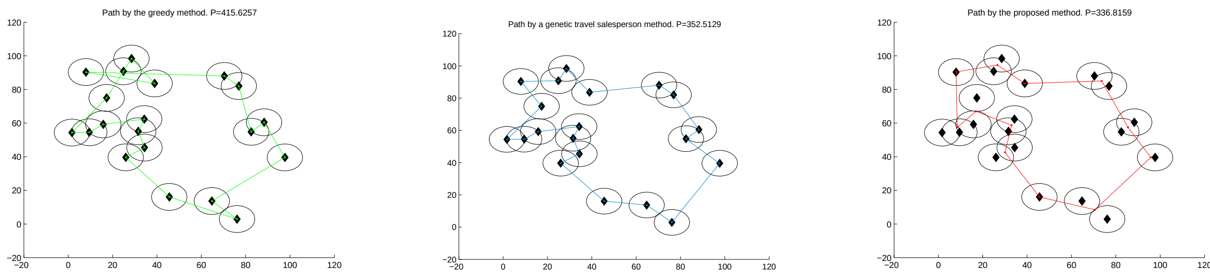


Fig. 7. The robot-assisted data collection: the robot travels via the blue path to collect data from each sensor. Here, the green dash circle is the communication range of the sensor.

Figure 5 shows the path Π^D generated by Algorithm 2. Path Π^D represented by red line is the path that the robot will follow to place the n sensors, while the green squares are the positions to place the sensors (also the turn points of the robot). Figure 6 shows the deployed sensors and their sensing ranges after the deployment phase. It is clear that every interest point is covered at least by one sensor.

V. Robot-Assisted Data Collection

After the robot has deployed the sensors, all sensors begin to collect information about the interest points. All



greedy method | generic method for TSP | proposed method

Fig. 8. Sample paths found by the three methods.

information will be sent to a centralized control center. However, due to the fact that the communication range of sensor is limited, the sensor network may be partitioned to components far away from each other. Adding more sensor nodes can improve the connectivity, however, it is not feasible in many applications, such as space exploration with expensive sensor devices. In such scenario, the mobile robot can help. We assume that the robot is also equipped with communication devices and can collect data from the deployed sensors. The path planning for the robot is again an optimization problem where we try to minimize the total distance traveled by the robot.

Here, given the set of deployed sensors $S = \{s_1, s_2, \cdots, s_n\}$ and their positions $v_1, v_2, \cdots, v_n$, we study how to schedule the robot to visit certain pause points $U = \{u_1, u_2, \cdots, u_k\}$ where the robot can collect data from sensors, such that (1) the robot can communicate with every sensor during the round trip and make the minimum number k of stops; and (2) the path $\Pi^C = v_0 u_1 u_2 \cdots u_k v_0$ which the robot will travel to collect data on those pause points has the minimum total length.

For the first half problem, we first use transmission range r_T of each sensor s_i to draw the areas to be covered, and then run the greedy algorithm (Algorithm 1) to select minimum number of pause points to cover all sensor nodes. The problem is essentially the same as the one in the deployment phase except that the range of coverage is transmission range r_T instead of sensing range r_S. For the second half problem, we need to schedule the robot to visit these selected areas using shortest round trip. By using the same heuristic (Algorithm 2), we can find a solution Π^C and return the turn points $u_1 u_2 \cdots u_k$of the robot, shown as the blue path in Figure 7.

Notice that if some sensors can communicate and transfer data with each other, then it will suffice for the robot to visit only one of these sensors to pick up data. For this situation, we can merge these sensors' transmission ranges to a union area and use it as a single area in the input of Algorithm 1 instead of several individual areas. By asking the sensors to increase their transmission ranges, the connectivity of the sensor network can increase, which will lead to less areas the robot needs to visit. This is a tradeoff between the communication cost plus power consumption at sensors and the power consumption at the robot. For example, if the transmission range of each sensor is infinitely large, then the robot does not need to move to collect the data. If the transmission range is infinitely small, the robot needs to visit each sensor at its position to collect the data.

VI. Simulation Studies

We carried out several simulation experiments to evaluate the proposed method. As we have discussed earlier, the sensor deployment and data collection are actually one problem. Therefore, we only simulate in the context of sensor deployment. Conclusions made from the simulation in sensor deployment can be applied to the data collection problem.

In the simulation, all sensors have the same sensing range. For simplification, the visiting point v_i of each area a_i is chosen to be the center of a_i. However, this simplification does not undermine the virtual of the proposed approach. In the simulation, we compared the travel distance by the proposed approach with two traditional methods: greedy method and near-optimal solution from traveling salesman problem. In the greedy method, the robot starts from the current interest point and goes to the next point which is closest to the current one until all interest points have been visited and returns to the original position. The near optimal solution from traveling salesman problem is obtained by a genetic algorithm [19]. In the simulation, we want to know how the proposed approach performs with regard to the number of interest points and the sensing range compared with the two traditional methods.

Figure 8 shows a case of the simulation, in which 20 interest points, shown as black square dots, are randomly generated within a 100×100 field. In this case, the sensing range is 8. The total travel distances found by the greedy method, the generic TSP method, and the proposed method are 415.6257, 352.5129 and 336.8159. As we can see, when there is overlapping between the disks, the travel distance found by the proposed approach can be considerably smaller than the distances found by the genetic TSP method and the the greedy method on the interest points.

In the next two simulation experiments, we will compare the three methods with regard to the number of interest points and the sensing ranges. Figure 9 shows the experiment in evaluating the proposed method with regard to the number of

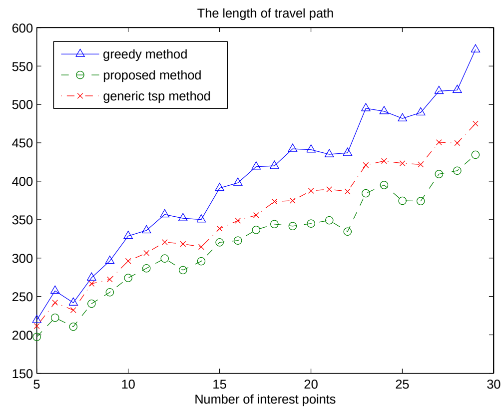


Fig. 9. Travel distance comparison with regard to number of interest points.

Fig. 10. Travel distance comparison with regard to sensing range

interest points and a fixed sensing range. The sensing range in this simulation is 8, and the number of interest points varies from 5 to 29. We can see that the total travel distance increases almost proportionally to the number of interest points. Among the three methods, our proposed approach can always achieve the smallest travel distance. Figure 10 shows travel distances found by the three methods with regard to a fixed number (20) of random interest points and various sensing ranges (from 5 to 19). It shows that for the greedy method and the genetic TSP method, the total travel distance do not vary much with regard to the sensing range, which is caused by the fact that they always go to the interest points instead of the overlapping areas. The travel distance found by the proposed approach, however, steadily decreases with increasing sensing range. This demonstrates that when the sensing range is large, there is high probability of overlapping, therefore traveling only to the overlapping regions will save large amount of time and energy cost. Since this paper does not intend to propose a method for finding near optimal solution for the traveling salesman problem, we did not compare the path finding algorithm, described in Algorithm 2, in the proposed method with the greedy method and the genetic method.

VII. Conclusion

In this paper, we studied how to use a mobile robot to assist the sensor deployment and data collection for unmanned explorations or monitoring using sensor networks. Given a set of interest points, we proposed a set of heuristics to (1) decide minimum number of sensor devices to cover all the interest points; (2) schedule the robot to place the sensor devices in certain positions so that the path of the robot is minimum; and (3) schedule the robot to visit and communicate with these sensor devices to collect data so that the path of the robot is minimum. We also verified the performances via simulations, and the results demonstrated a nice performance of our method compared with the greedy algorithm and a genetic algorithm for TSP.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. of 6th International Symp. on Distributed Autonomous Robotics Systems (DARS)*, 2002.

[3] T.-L. Chin, P. Ramanathan, K. K. Saluja, and K.-C. Wang, "Exposure for collaborative detection using mobile sensor networks," in *Proc. of IEEE International Conf. on Mobile Adhoc and Sensor Systems*, 2005.

[4] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings of IEEE INFOCOM*, 2005.

[5] Y. C. H. Yongguo Mei, Yung-Hsiang Lu and C. S. G. Lee, "Reducing the number of mobile sensors for coverage tasks," in *Proceedings of International Conference on Intelligent Robots and Systems*, 2005.

[6] C. Wu, W. Sheng, and Y. Zhang, "Mobile self-localization using multi-dimensional scaling in robotic sensor networks," *Int'l J. on Intelligent Control and Systems*, pp. 163-175, vol. 11, no. 3, 2006.

[7] W. Sheng, G. Tewolde, and Y. Guo, "Distributed robot-assisted node localization in active sensor networks," in *Proc. of the 2006 IEEE International Conference on Mechatronics and Automation*, 2006.

[8] H. Lee, H. Dong, and H. Aghajan, "Robot-assisted localization techniques for wireless image sensor networks," in *Proc. of the Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2006.

[9] P. Corke, R. Peterson, and D. Rus, "Localization and navigation assisted by networked cooperating sensors and robots," *International Journal of Robotics Research*, vol. 24, no. 9, pp. 771–786, 2005.

[10] Y. Mei, C. Xian, S. Das, Y. C. Hu, and Y.-H. Lu, "Replacing failed sensor nodes by mobile robots," in *International Conference on Distributed Computing Systems, Workshop on Wireless Ad hoc and Sensor Networks (WWASN)*, 2006.

[11] A. Panousopoulou, E. Kolyvas, Y. Koveos, A. Tzes, and J. Lygeros, "Robot-assisted target localization using a cooperative scheme relying on wireless sensor networks," in *Proceedings of IEEE Control and Decision Conference, (CDC)*, 2006.

[12] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad hoc sensor networks," *Computer Communications Journal (Elsevier)*, vol. 29, no. 4, pp. 413–420, 2006.

[13] J.-C. Latombe, *Robot Motion Planning*. Springer, 1991.

[14] P. C.-Y. Sheu and Q. Xue, *Intelligent Robotic Planning Systems*. World Scientific Pub Co Inc, 1991.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 2001.

[16] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *Journal of the ACM (JACM)*, vol. 41, no. 5, pp. 960–981, 1994.

[17] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[18] A. Dumitrescu and J.S.B. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," in *Proc. of ACM Symposium on Discrete Algorithms*, 2001.

[19] J. Kirk, "Traveling Salesman Problem - Genetic Algorithm," http://www.mathworks.fr/matlabcentral/fileexchange/loadFile.do ?objectId=13680&objectType=file