DATA COLLECTION SCHEME FOR WIRELESS SENSOR NETWORK WITH MOBILE COLLECTOR

Khaled Almi'ani, Muder Almi'ani, Ali Al_ghonmein and Khaldun Al-Moghrabi

Al-Hussein Bin Talal University, Ma'an, Jordan

ABSTRACT

In this paper, we investigate the problem of designing the minimum number of required mobile elements tours such that each sensor node is either on the tour or one hop away from the tour, and the length of the tour to be bounded by pre-determined value L. To address this problem, we propose heuristic-based solution. This solution works by directing the mobile element tour towards the highly dense area in the network. The experiment results show that our scheme outperform the benchmark scheme by 10% in most scenarios.

1. INTRODUCTION

Typically, in multi-hop data communication scenario, the sensors closed to the sink are the first to run out energy. This is due to fact that these sensors are required to forward all other nodes data. Once these nodes are disconnected, the operational lifetime of the networks ends, since the entire network become unable to communicate with the sink. This raise reducing the energy consumption as a major challenge.

To address this problem, many proposals [1][2][3] in the literature, have investigated the use of the Mobile Elements (MEs). A Mobile Element serves as a data carrier and travel the entire network to collect the data of each sensor via single-hop communication. By using Mobile Elements, we can avoid the energy consumption due to multi-hop communication. However, the speed of the mobile element is typically low[4][5], and therefore reducing the data gathering latency is important to ensure the efficiency of such approach.

In this paper, we assume that are given more than one mobile element and the objective is to design minimum number of required tours to cover the network. Such that each sensor node is either on the tour or one hop away from the tour, and the length of the tour to be bounded by pre-determined value *L*. We term this problem as the Length-constrained Tours Minimization. To address the presented problems we present heuristic-based solution, which uses cluster-based mechanism to build it solutions.

The rest of the paper is organized as follows. Section 2 presents the related work in this area. Section 3 provides formal definitions of the problem presented in this work. Section 4 presents the heuristic solution of the presented problem. In Section 5 the evaluation for the proposed heuristics is presented. Finally, Section 6 concludes the paper.

2. RELATED WORK

In the literature, several proposals[6][7] have investigated the use mobile sink(or mobile elements) to increase the lifetime of the network. By varying the data gathering path, , the residual energy in the nodes becomes more evenly balanced throughout the network, leading to a higher network lifetime.

Zhao et al. [8][9] investigated the problem of maximizing the overall network utility. Accordingly, they proposed two algorithmic-based solutions, where the mobile element visit each gathering point for a period of time and gather the data from nearby sensors via multi-hop communications. They considered the cases where the sojourn time is fixed as well as variable. Guney et al. [10] formulated finding the optimal sink trajectory and the data flow routs problem as mixed integer programming formulations. Accordingly, they presented several heuristic to address the presented problem.

The problem presented in this work share some similarities with the problem proposed by Xu et al.[11]. In their problem, the objective is to plan the mobile element tour to visit set of nodes in the network named gathering points. The upper-bound of the obtained tour must be also bounded by pre-determined time-deadline. In addition, they also restrict the depth of the routing trees.

The problems presented in this paper inherits some characteristic from the well-known Vehicle Routing Problem (VRP) [12]. In this problem, given a group of vehicles assigned to an initial point, the goal is to determine routes for these vehicles from the initial point to the customers to deliver goods while minimizing the travelling time for these vehicles. Also, our problem can be recognized as variation of the Deadline Travelling Salesman Problem (Deadline-TSP)[13]. In this problem the objective is determine the shortest tour for a salesman to visit a set of cities, where each city must be visited before a time deadline.

The single hope data gathering problem proposed by Ma et al.[14] can be recognized as an extended version of the problems presented in this work. In[14] , the authors investigated the problem of designing the mobile element tour(s), such that each node is either on the tour or one hop away from the tour. The only difference between this problem and the problems presented in this is the introduction of the "polling points" by the authors. Ma et al.[14] defines the polling point as a point in the network, where the mobile element can communicate with one or more sensor node via a single hop transmission. In situations, where the network is spars, where the polling points are the sensor nodes, the problems presented in this work become exactly similar to the problem presented by Ma et. al.[14].

3.PROBLEMS DEFINITIONS

In the Length-constrained Tours Minimization problem, we are given an undirected complete graph $G = \langle V, E \rangle$. V is the set that represents the sensor nodes in the network ($v_s \in V$, v_s in the sink node) . E is the set of edges that represents the travelling time between the node in the network, i.e. (v_i, v_j) is the time that the mobile element takes to travel between nodes v_i and v_j. Also we are given M and L. M is the set that represents the available mobile elements, and L is the time-deadline constraint.

A solution for this problem consists of multiple tours, where the objective is to minimize the number of obtained tours, such that the travelling time for each tour is less than or equal to L, and each node is on one of the tours or one-hop away from a node included in one of the tours.

4.ALGORITHMIC SOLUTION

To address the presented problem, we present heuristic approach that works by partitioning the network, then in each partition, a mobile element will be assigned. This partitioning takes into consideration the distribution of the nodes, to avoid long distance travelling by the mobile element. The presented approach uses ideas from [15][16][17].

Our approach start by identifying the set of nodes that will be used to construct the mobile elements tours. Nodes does not belong to this set must be at most one hope away from nodes belong to this set. To obtain this set we use the Set-Coverd-based algorithm proposed by Almi'ani et al[15]. Once the nodes of the tours are identified, the presented approach start by partitioning the network into two partitions. The partitioning step employs the well-known k-mean algorithm. Once these two partitioned are obtained, the process proceed by constructing a single tour for each partition. The tours inside each partition is constructed using Christofides algorithm[18]. For each partition, if the obtained tour satisfies the time-deadline constraint, this tour will be assign to a mobile element. Otherwise, this partition will be re-partitioned and the tour construction step will be retriggered.

The k-mean algorithm aims to partition the network graph into k number of clusters such that the distance between the nodes inside each clusters is minimized. At first, k number of nodes are selected at random as the initial center nodes of the clusters. Then, in each iteration, each node is assigned to its nearest cluster (center node). Once all nodes have been thus assigned, the center node for each cluster is recalculated, and the process is repeated from the beginning based on the identity of the new center nodes. The clustering step stops when the obtained centre nodes still the same in two consecutive iterations.[16]

5.EXPERIMENT RESULTS

To validate the performance of our algorithm, we have conducted an extensive set of experiments. We aim to investigate the performance of the presented heuristic in terms of the number of obtained tours and length of the tours. We consider varying the following parameters in our experiments:

(1) varying the number of nodes in the network and
(2) varying the value of the tour length constraint.

Each of the presented experiment is an average of 10 different random topologies. We consider the following deployment scenarios:

- Uniform density deployment: the nodes are uniformly deployed in a square area of $400 \times 400 \ m^2$.
- Variable density deployment: the area of the network is divided into 25 square, where the size of each square is $80 \times 80 \ m^2$. Then, ten of these squares are selected randomly, and the density of the selected squares are set be 5 times the density in the remaining squares.

In this section we will refer to our algorithm as the Partition-Based algorithm. To benchmark the presented algorithms performances, we compare the Partition-Based algorithm against the Data Gathering Algorithm with Multiple M-collector, which also proposed by Ma et al.[14], we will refer to this algorithm as the M-Covering algorithm.

In the M-Covering algorithm, given the polling points and the transit constraint, this algorithm starts by constructing a minimum spanning tree that connects all polling points. Then, for each node in the tree we calculate the total calculate the summation of the lengths of the edges in the sub-tree rooted at this node. Then, in each iteration, we extract the deepest, heaviest sub-tree, where the total edges weight of this sub-tree is less than or equal to L/2. Once a sub-tree is extracted, the weight value for each node left in the original tree is re-calculated. This process stops when the tree become empty. The tour for each extracted sub-tree is calculated by using a TSP-solver. The bound is used because the authors employed a 2-approximation algorithm [19] to

obtain the tour. In this section, to ensure the fairness of the evaluation, we employed Christofides algorithm as a TSP-solver in the M-Covering algorithm. Also we replace the condition <=L/2, by continuously validate that the length of the obtained tour is L[17].

In this section, we compare the Partition-Based (PB) algorithm against the M-Covering algorithm. We are particularly interested in investigating the performance of the algorithms, while varying the number of nodes and the time deadline constraint (L). Unless mentioned otherwise the value of $L = d \times 1\,m/s$, where d is the distance between the sink and the farthest node in the network. Now, we move to investigate the relationship between the number of nodes and the number of tours obtained by each algorithms. Figures 7 and 8 show the results for the uniform density and the variable density deployment scenarios; respectively. From the figures, we can see that the Partition-Based algorithm always obtains lower number of tours, compared to the M-Covering algorithm. Also, from the figures we can see that in the variable density deployment scenario, the M-Covering algorithm obtains more tours compared to the uniform deployment scenario. The factors behind this behavior are main the process of selecting the tours' nodes, and the partitioning step. The M-Covering algorithm uses the process employed by the T-Covering algorithm to identify the nodes involved in the mobile elements tours. In addition, the Partition-Based algorithm uses the same process used by the Set-Based algorithm to identify the nodes included in the mobile elements tours. In this direction, we expected that number of nodes selected to be included in the tours to be significantly bigger in the M-Covering algorithm, compared to the Partition-based algorithm. The partitioning process employed by the M-Covering algorithm works by cutting the minimum spanning tree, which connect the identified nodes, from bottom to top into smaller trees that satisfy the time deadline constraint. Such cutting mechanism works without considering the distance between the nodes, which are packed into one tour. This is expected to reduce performance of such a mechanism; especially when the distances between neighboring nodes vary frequently, since in this situation, traversing the minimum spanning tree may result in having a few nodes in each tour, which very far from each other. In the Partition-based algorithm, the obtained tours' nodes are partitioned using the k-mean algorithm, and this algorithm has the advantage of partitioning the nodes into clusters (groups of nodes) such that the distances among the same cluster nodes are relatively small. This gives the Partition-based algorithm the advantage, since any tour's nodes are not expected to be far from each other.

Now, we study the impact of the time deadline constraint (L) on the number of tours obtained by each algorithm. Figures 7 and 8 show the results for both deployment scenarios, for 500-nodes networks. Here, the horizontal axis shows the value of L normalized as a fraction of $d \times 1\,m/s$. We observe that, reducing the value of L reduces the gap between the presented algorithms performances. This is expected, since reducing the value of L reduces the solution space and therefore reduces the effect of each algorithm main advantages.
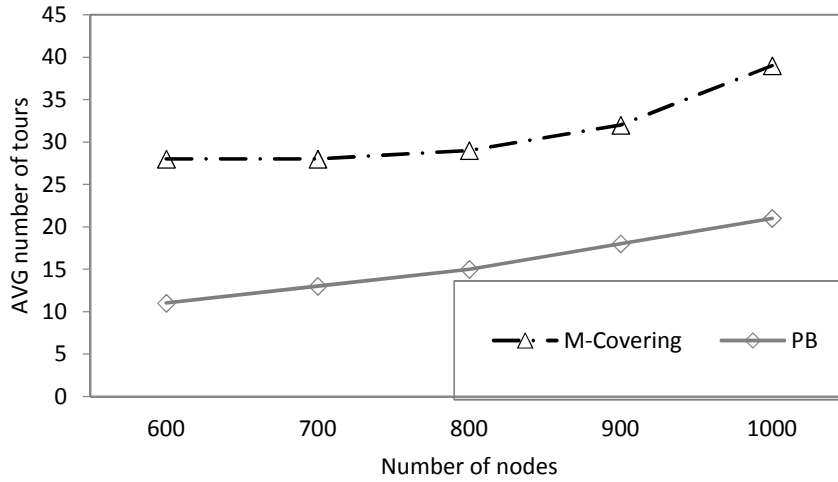
Figure 5: Number of nodes against total the average number of tours, for the uniform density deployment scenario.
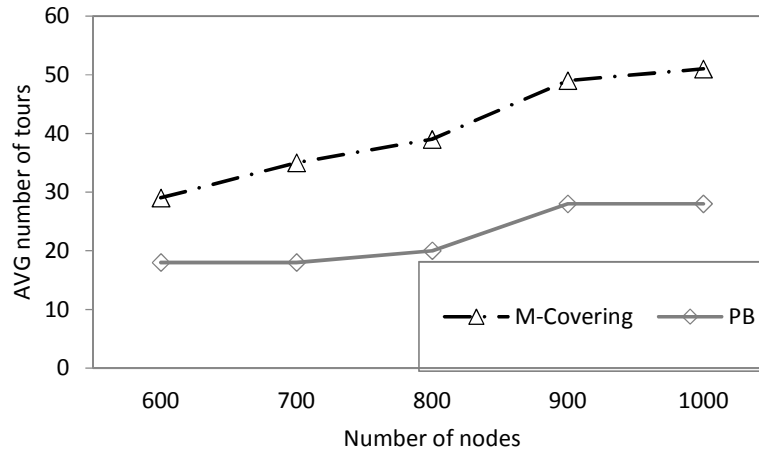

Figure 5: Number of nodes against total the average number of tours, for the variable density deployment scenario.
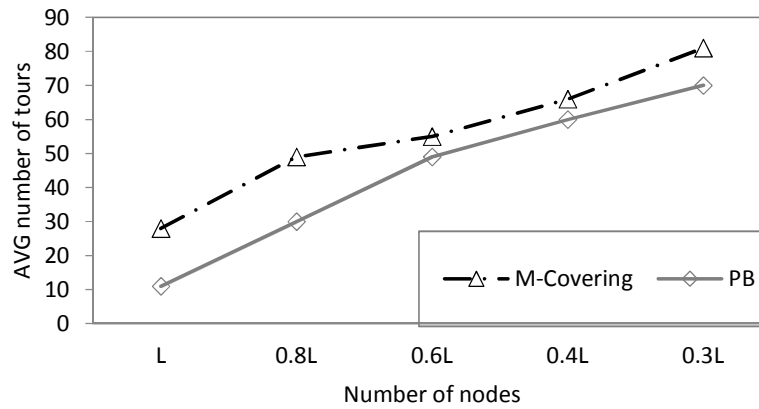

Figure 5: Normalized value of *L* against the average number of tours, for the uniform density deployment scenario.
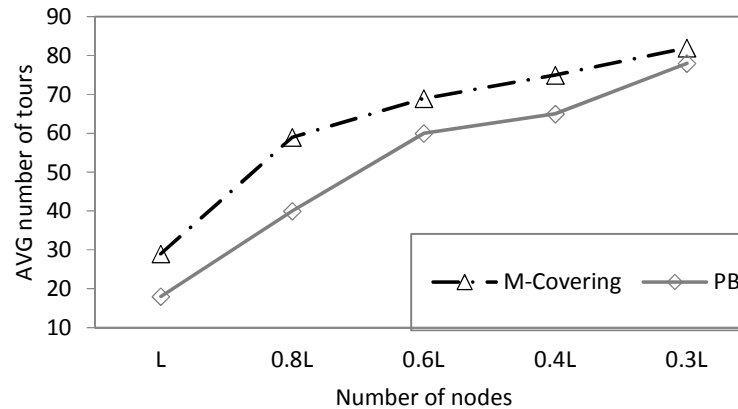
Figure 5: Normalized value of *L* against the average number of tours, for the variable density deployment scenario.

6.CONCLUSION

In this paper, we consider the problem of designing the mobile elements tours such that total size of the routing trees is minimized. In this work, we present an algorithmic solution that creates its solution by partitioning the network, then in each partition; a caching node is selected based on the distribution of the nodes. The algorithms presented here are based only on the static network topology. All nodes are considered to be identical in terms of energy consumption and communication capabilities. In order to further maximize the network lifetime, our algorithms could be extended to take into account the residual energy of individual nodes, during the design of the tours.

REFERENCES

[1] K. Almi'ani, S. Selvadurai, and A. Viglas, "Periodic Mobile Multi-Gateway Scheduling," in Proceedings of the Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2008, pp. 195–202.

[2] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C. G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on, 2005, pp. 386–395.

[3] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in wireless sensor networks with mobile elements," IEEE Trans. Mob. Comput., vol. 7, no. 12, pp. 1430–1443, Dec. 2008.

[4] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN), 2005, pp. 404 – 409.

[5] R. Pon, M. A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. J. Kaiser, and others, "Networked infomechanical systems: a mobile embedded networked sensor platform," in Proceedings of the 4th international symposium on Information processing in sensor networks, 2005, pp. 376 – 381.

[6] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in Proceedings of IEEE Globecom, 2003, vol. 1, pp. 377–381.

[7]   Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in Proceedings ofthe 38th Annual Hawaii International Conference on System Sciences (HICSS), 2005, vol. 9, pp. 03–06.

[8]   M. Zhao and Y. Yang, "Optimization-Based Distributed Algorithms for Mobile Data Gathering in Wireless Sensor Networks," IEEE Trans. Mob. Comput., vol. 11, no. 10, pp. 1464–1477, 2012.

[9]   M. Zhao and Y. Yang, "efficient data gathering with mobile collectors and space-division multiple access technique in wireless sensor networks," IEEE Trans. Comput., vol. 60, no. 3, pp. 400–417, 2011.

[10] E. Güney, I. K. Altmel, N. Aras, and C. Ersoy, "Efficient integer programming formulations for optimum sink location and routing in wireless sensor networks," in Proceedings of the 23rd International Symposium on Computer and Information Sciences, 2008, pp. 1–6.

[11] Z. Xu, W. Liang, and Y. Xu, "Network Lifetime Maximization in Delay-Tolerant Sensor Networks With a Mobile Sink," in Proceedings of the 8th IEEE International Conference on Distributed Computing in Sensor Systems, 2012, pp. 9–16.

[12] P. Toth and D. Vigo, "The Vehicle Routing Problem," Soc. Ind. Appl. Math., 2001.

[13] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Approximation algorithms for deadline-TSP and vehicle routing with time-windows," in Proceedings of the thirty-sixth annual ACM symposium on Theory of computing (STOC), 2004, pp. 166 – 174.

[14] M. Ma, Y. Yang, and M. Zhao, "Tour Planning for Mobile Data-Gathering Mechanisms in Wireless Sensor Networks," IEEE Trans. Veh. Technol., 2013.

[15] K. Almi'ani and A. Viglas, "Designing connected tours that almost cover a network," in Proceedings of the 2013 14th International Conference onParallel and Distributed Computing, Applications and Technologies(PDCAT'13), 2013, pp. 281 – 286.

[16] K. Almi'ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in Proceeding of the IEEE 35th Conference on Local Computer Networks (LCN), 2010, pp. 582–589.

[17] K. Almi'ani, M. Aalsalem, and R. Al-Hashemi, "Data gathering for periodic sensor applications," in in Proceedings of the 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2011, pp. 215–220.

[18] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," 1976.

[19] S. S. Skiena, "Traveling Salesman Problem," Algorithm Des. Man., pp. 319–322, 1997.