

# Delay-Minimized Route Design for Wireless Sensor-Actuator Networks

Edith C.-H. Ngai

Department of Computer Science  
and Engineering  
Chinese University of Hong Kong  
Shatin, NT, Hong Kong  
Email: chngai@cse.cuhk.edu.hk

Jiangchuan Liu

School of Computing Science  
Simon Fraser University  
Vancouver, BC, Canada  
Email: jcliu@cs.sfu.ca

Michael R. Lyu

Department of Computer Science  
and Engineering  
Chinese University of Hong Kong  
Shatin, NT, Hong Kong  
Email: lyu@cse.cuhk.edu.hk

**Abstract**—Wireless sensor-actuator networks (WSANs) have recently been suggested as an enhancement to the traditional sensor networks by employing powerful and mobile actuators. Multiple actuators can patrol along different routes and communicate with the static sensors. To minimize the data collection time, an effective route design is crucial for the actuators to travel in the sensing field. In this paper, we present a mathematical formulation of the route design problem, and show that the general problem is computationally intractable. We then develop a practically efficient algorithm to reduce the waiting time for the sensors. Our algorithm adaptively differentiates the actuator visiting frequencies to the sensors according to their relative weights and data generation patterns. Simulation results demonstrate that our algorithm can effectively reduce the overall data collection time.

## I. INTRODUCTION

Wireless sensor networks (WSNs) have been applied in environment monitoring, battlefield surveillance, chemical attack detection, and target tracking [1][2]. The asymmetric communication patterns from sensors to the sink, however, often overload the nodes close to the sinks and reduce the network lifetime. Moreover, network partitions may occur in sensor networks, which make multihop communications impossible. To alleviate these problems, mobile elements, such as mobile sinks [3] or mobile relays [4], have been suggested for collecting data in WSNs. Actuator nodes, which have much stronger computation and communication power than uni-purpose micro-sensors, have also been introduced [5][6]. In a Wireless Sensor-Actuator Network (WSAN), an actuator (*e.g.*, a robot) can move around to cover the sensing field and interact with static sensors. Each static sensor has a limited buffer, which stores locally sensed data until some actuator approaches. It can then upload the data to the actuator with short-range communications and free its buffer.

Note that the amount and frequency of data generation in the sensor nodes are non-uniform [7]. The actuators thus should visit locations with higher importance more frequently. This can be achieved by either increasing the number of actuators passing through important locations, or reducing the length of the routes of passing-through actuators. More formally, there is a Route Design Problem (RDP) for the actuators to minimize their average inter-arrival time to the static sensors.

In this paper, we present the first formal study on the RDP problem. We demonstrate that the general problem is NP-hard. We then develop an efficient algorithm based on the minimum spanning tree construction. Our simulation results show that the algorithm can effectively reduce the overall data collection time.

The remainder of the paper is organized as follows: The related work is presented in Section II, followed by the formulation and NP-hardness proof of the RDP problem in Section III. The proposed approximation algorithm is described in Section IV. Simulation results are provided and discussed in Section V. Finally, Section VI concludes the paper.

## II. RELATED WORK

Mobile elements have been proposed to carry data in wireless networks. Shah. et al. [8] presented an architecture using moving entities (Data Mules) to collect sensor data. There have been studies on mobile sinks with predictable and controllable movement patterns [9][10]. Mobile sinks can find the optimal time schedule to stay at appropriate sojourn points [3]. Apart from the above, Zhao et al. [11] proposed a message ferrying (MF) approach to address the network partition problem in sparse ad hoc networks. Luo et al. [4] investigated a joint mobility and routing algorithm with mobile relays to prolong the lifetime of wireless sensor networks. Gu et al. [12] proposed a partitioning-based algorithm to schedule the movement of mobile element (ME) to avoid buffer overflow in sensors and reduce the minimum required ME speed.

Our work is motivated by the above studies. The key difference is that we focus on the route design for multiple mobile components in WSANs, specifically, actuators moving along independent routes. We also address issues regarding the non-uniform weights of the static sensors.

A closely related problem to RDP is the vehicle routing problem (VRP), which considers scheduling vehicles stationed at a central facility to support customers with known demands, targeting at minimizing the total distance travelled [13]. There are a number of variations to VRP, including the Capacitated VRP (CVRP)[14] and VRP with time windows (VRPTW)[15]. While these investigations have studied the routes of mobile

components, the unique features of the actuators and the heterogeneous sensor networks have yet to be explored.

### III. PROBLEM FORMULATION

We consider a Wireless Sensor-Actuator Network consisting of multiple mobile actuators and a set of static sensors. The actuators move in the sensing field along independent routes. Each static sensor has a limited buffer to accommodate locally sensed data. When an actuator approaches, the sensor can upload the data to the actuator and free its buffer.

In our design, sensors are assigned with different weights  $W_j$ , according to their types and importance of data. Normally, the sensors which detect more urgent events will be assigned with higher weights. For example, sensors detecting gas leakages would have higher weights than those sensing humidity. The objective of our work is to design the distinct routes for the actuators to walk through the sensors.

We now give a formal description of the system parameters, which is summarized in Table I. Our objective is to minimize the overall length of routes, while guaranteeing locations with higher weights can achieve lower actuator inter-arrival time.

TABLE I  
SYSTEM PARAMETERS

$s$	Sensor node
$R_s$	Communication range of sensor node
$c_{ij}$	Cost from sensor location $i$ to $j$
$x_{ijk}$	Boolean indicating whether link $(i, j)$ is on route $k$
$W_j$	Weight of sensor location $j$ (a value between 0.0 and 1.0)
$N_j$	Number of sensors with weight $W_j$
$A_j$	Average actuator inter-arrival time for sensor location $j$
$T_k$	Period of route $k$
$N$	Number of sensor locations
$M$	Number of actuators

#### Route Design Problem (RDP):

$$\text{Minimize } \sum_j A_j * W_j * N_j, \quad (1)$$

where  $T_k = \sum_{i=1}^N \sum_{j=1}^N x_{ijk} * c_{ij}$ ,  $A_j = \mathbb{F}(T_1, T_2, \dots, T_{M'})$  is the function of all  $T_k$  that pass through the sensor location  $j$ .

Subject to:

$$\sum_{k=1}^M \sum_{i=1}^N x_{ijk} \geq 1, \forall j = 1, \dots, N \quad (2)$$

$$\|s, j\| \leq R_s, \forall s, \exists j \quad (3)$$

$$\sum_{i=1}^N x_{ipk} - \sum_{j=1}^N x_{pjk} = 0, \forall k = 1, \dots, M, p = 1, \dots, N \quad (4)$$

$$y_j - y_i + N \sum_{k=1}^M x_{ijk} \leq N - 1, i \neq j = 1, \dots, N \quad (5)$$

$$x_{ijk} \in \{1, 0\}, \forall i, j, k; y_i \text{ arbitrary}, \quad (6)$$

where our objective (Eq. 1) is to minimize weighted average actuator inter-arrival time among all sensor locations.  $\mathbb{F}(T_1, T_2, \dots, T_{M'})$  is a function for calculating the  $A_j$  of a sensor location that is passed by the routes with period  $T_1, T_2, \dots, T_{M'}$ . Constraint (2) suggests that each sensor location should be on at least one route. Constraint (3) ensures every sensor can communicate with the actuators, provided that at least one sensor location is within the communication range  $R_s$  of the sensor. Constraint (4) states that if an actuator visits a sensor location, it must also depart from it. Constraint (5) is the subtour-elimination condition derived for the traveling salesman problem [16]. The route design problem differs from the traditional vehicle routing problem in the following aspects:

1) The sensors are of different weights, according to their data generation rates and importance. Sensor locations with higher weights will achieve lower average actuator inter-arrival times.

2) Sensors upload data to actuators through wireless communications. Data transmission is possible only when the distance between the sensor and actuator is within a communication range  $R_s$ .

3) It is not necessary for each route to pass through the depot (or the base station), as the actuators generally can communicate with the base station directly with its stronger power.

We now offer some observations on the general route design problem.

*Definition 1:* Route is a path that is walked through repeatedly by an actuator.

*Property 1:* Route with a Hamiltonian cycle achieves shorter maximum inter-arrival time  $A_{max}$  than that without.

*Proof:* Let  $p_1, p_2, \dots, p_n$  be the sensor locations on a route,  $c_{12}, c_{23}, \dots, c_{n1}$  be the cost between consecutive sensor locations. We have  $T = \sum_{\forall c} c/v$  is the period of the route, where  $v$  is the moving speed of the actuators. If the route is a Hamiltonian cycle,  $A_{max}$  of any sensor location  $p$  is always equal to  $T$ . On the contrary, considering a route that is not a Hamiltonian cycle, an actuator will need to walk from one end to another end, then back to the beginning sensor location to complete a cycle. The  $A_{max}$  of the sensor locations at two ends will then be  $2T$ , which is much longer than  $T$ . ■

*Property 2:* The average actuator inter-arrival time  $A_j$  of a sensor location  $j$  on multiple routes can be calculated as

$$\mathbb{F}(T_1, T_2, \dots, T_{M'}) = \frac{\prod_{k=1}^{M'} T_k}{\sum_{k=1}^{M'} (\prod_{i=1, i \neq k}^{M'} T_i)}$$

*Proof:* Find a common multiple  $\mathbb{Q}$  among the period  $T_1, T_2, \dots, T_{M'}$  of all routes that pass through sensor location  $j$ . One of the simplest common multiple is  $\mathbb{Q} = T_1 * T_2 * \dots * T_{M'}$ . If  $\mathbb{Q}$  is a time interval, the number of times that

the actuator on route  $k$  passes through sensor location  $j$  is  $\prod_{i=1, i \neq k}^{M'} T_i$ . The total number of times that the actuators visit  $j$  is the summation of the number of visits by each actuator,  $\sum_{k=1}^{M'} (\prod_{i=1, i \neq k}^{M'} T_i)$ . The average inter-arrival time  $A_j$  is then equal to the total time over the total number of visits, that is,

$$A_j = \frac{\prod_{k=1}^{M'} T_k}{\sum_{k=1}^{M'} (\prod_{i=1, i \neq k}^{M'} T_i)}. \quad \blacksquare$$

**Theorem 1:** The route design problem (RDP) is NP-hard.

*Proof:* To prove the route design problem (RDP) is NP-hard, we show that  $\text{HAM-CYCLE} \leq_p \text{RDP}$ . Let  $G = (V, E)$  be an instance of HAM-CYCLE. We construct an instance of RDP as follows. We form a complete graph  $G_k = (V_k, E_k)$  for each actuator  $k$ , where  $E_k = (i, j) : i, j \in V_k, i \neq j$ ,  $G' = (V, E') = \bigcup_{\forall k} G_k, V = \bigcup_{\forall k} V_k, E' = \bigcup_{\forall k} E_k$ , for  $k = 1, \dots, M$ , and we define the cost function  $c$  by

$$c_{i,j} = \begin{cases} 0 & \text{if } (i, j) \in E, \\ 1 & \text{if } (i, j) \notin E. \end{cases}$$

Considering  $W_j = \text{constant}$  for all  $j \in V$ , all sensor locations  $j$  will be visited once by any of the actuators, such that the instances of RDP is  $(G', c, C[1, \dots, k, \dots, M] = 1)$ , where  $C[k]$  is the sum of  $c_{i,j}$  for all  $(i, j) \in E_k$ .

We now show that the graph  $G$  has a Hamiltonian cycle if and only if  $G'$  has  $M$  tours, where the cost  $C[k]$  at each cycle  $k$  is at most 1. Suppose that graph  $G$  has a Hamiltonian cycle  $h$ . Each edge in  $h$  belongs to  $E$  and thus has cost 0 in  $G'$ . Select  $M$  edge in  $h, (i_1, j_1), (i_2, j_2), \dots, (i_M, j_M)$  and remove them. Then, reconnect these nodes in sequence  $(j_1, i_2), (j_2, i_3), \dots, (j_M, i_1)$ . Since the edges  $(j_1, i_2), (j_2, i_3), \dots, (j_M, i_1)$  do not belong to  $E$ , each of them has cost 1. As a result,  $h$  is broken into  $M$  cycles, where the cost  $C[k]$  of each cycle  $k$  is at most 1.

Conversely, suppose that  $G'$  has  $M$  cycles  $h_1, \dots, h_k, \dots, h_M$  of cost at most 1. We can easily merge these  $M$  cycles into one Hamiltonian cycle. Since the cost of each edge in  $E_k$  are 0 and 1, at most one edge of each cycle is 1, and all other edges are 0. Remove the edge in each cycle that is 1,  $(i_1, j_1), (i_2, j_2), \dots, (i_M, j_M)$ . Then, reconnect the nodes involved in a new order,  $(j_1, i_2), (j_2, i_3), \dots, (j_M, i_1)$ , such that a single hamiltonian cycle can be formed.  $\blacksquare$

#### IV. MST-BASED ROUTE DESIGN ALGORITHM FOR MULTIPLE ACTUATORS

We now develop a practical algorithm for the route design problem with multiple actuators in WSANs. Our algorithm utilizes multiple minimum spanning trees (MSTs). Intuitively, we want to construct  $M$  routes with equal period  $T$ , in which sensors with weights  $W_i$  will be visited more frequently. Notice that each actuator is in charge of one route. A sensor location with weight  $W_i$  will be passed by  $W_i * M$  actuators. In other words, it will be found on  $W_i * M$  routes. For example, a sensor with  $W_i = 0.75$  in a network with 4 actuators will be visited by 3 different actuators. If all actuators have the same

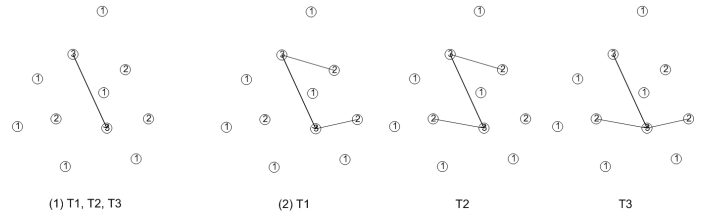


Fig. 1. Step 1: Sensor locations with  $N_i = 3$  are involved in all trees; Step 2: Sensor locations with  $N_i = 2$  are involved in any two of the trees.

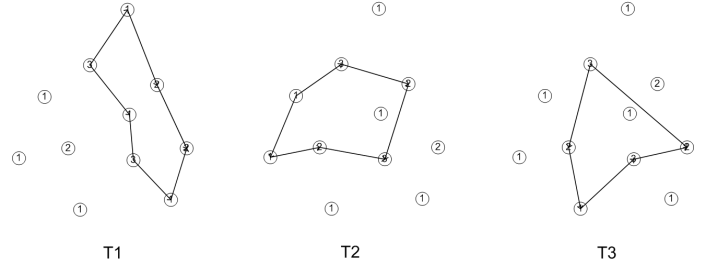


Fig. 2. Final routes formed by the route design algorithm.

period  $T$ , from Property (2), its average actuator inter-arrival time  $A_{avg}$  will be  $T/3$ .

In the following, we describe the details of the MST-based route design algorithm.

##### A. Clustering with MSTs

Sensor locations are assigned with a specific number of routes  $N_i$  according to their weights  $W_i$ , where  $N_i = \lceil W_i * M \rceil$ . The locations with the same  $N_i$  belong to the set  $S_i$ . Our algorithm builds  $M$  spanning trees  $T_k$ , which include the sensor locations on the  $M$  routes respectively, where  $k = 1, \dots, M$ . Firstly, the sensor locations with the highest  $N_i$ , say  $N_i = M$ , will be assigned to all trees. Then, the locations with the next highest  $N_i$  will be assigned to  $N_i$  trees with the lowest costs (Figure 1). The process is repeated until there is no remaining locations, as shown in Algorithm 1. Figure 2 shows the final routes formed by three trees, which have similar length. It demonstrates that sensor locations with higher weights will be visited by more actuators. Since the routes have similar lengths, they also achieve lower actuator inter-arrival time  $A_{avg}$ .

##### B. Forming a TSP Solution in each Tree $T_k$

The  $M$  trees above result in  $M$  groups of nodes that should be walked through by actuators on distinct routes. The route design problem can then be reduced to the Travelling Salesman Problem (TSP) for each group of nodes. In the literature, several algorithms to calculate the TSP paths are proposed, such as the nearest neighbor, LKH [17], and some polynomial-time approximation schemes [18]. Among different approximation algorithms, the Approx-TSP-Tour algorithm [19] uses the minimum spanning tree to create a tour whose weight is a lower bound on the length of an optimal traveling-salesman

**Algorithm 1** Forming Spanning Trees for Actuators

```

for  $k = 1$  to  $M$  do
     $T_k = \phi$ ;
end for
for  $i = M$  down to  $1$  do
    for each node  $p \in S_i$  do
        for each spanning tree  $T_k$  do
             $T_{k'} = T_k \cup p$ ;
            Calculate the new cost of the tree  $T_{k'}$ ,  $C(T_{k'})$ 
        end for
        sort the list of  $C(T_{k'})$ 
        for the top  $i$  trees with the lowest new cost  $C(T_{k'})$  do
             $T_k = T_{k'}$ ;
             $C(T_k) = C(T_{k'})$ ;
        end for
    end for
end for
    
```

tour. Its cost is no more than twice of the minimum spanning tree’s weight. After computing the MST, then it performs a preorder traversal on the tree to obtain a Hamiltonian cycle [12]. In our solution, the MST is created using Prim’s algorithm [19], which is in polynomial-time.

*C. Determining the Locations of Actuators*

It is more efficient for a sensor to have short waiting time to the actuators, so the maximum inter-arrival time  $A_{max}$  may also be an important consideration other than  $A_{avg}$ . We focus on the sensor locations with the highest  $W_i$  and provide a simple method for deciding the starting point of the actuator on its route. A location with the highest  $W_i$  can be selected as a reference point  $p_r$ . Then, each actuator  $k$  will be assigned to the point after travelling for time  $T * k/M$  from  $p_r$  on its own route. This is to encourage more even inter-arrival time of the actuators. More advanced methods can be studied as the future work.

V. PERFORMANCE EVALUATION

We have conducted simulations for our proposed route design algorithm for multiple actuators. The simulation settings are mainly drawn from [20], which are summarized in Table II.

TABLE II  
SIMULATION PARAMETERS

Network size	200m x 200m
Node placement	Uniform Random or Cluster
No. of sensor locations ( $N$ )	50 or 100
Weight of sensors ( $W_j$ )	0.0-1.0
No. of actuators	$M$
Speed of actuators	$v$
Radio range	40m
MAC layer	IEEE 802.11

*A. Average Actuator Inter-Arrival Time under Uniform Random Sensor Distribution*

In the first set of experiments, we evaluate the average actuator inter-arrival time  $A_{avg}$  with uniformly and randomly

distributed sensors.

Note that the average inter-arrival distance  $D_{avg}$  is shown in our results, instead of the inter-arrival time  $A_{avg}$ . It is because the moving speed  $v$  of actuators may vary in different environments. Indeed, the average inter-arrival time of actuator  $A_{avg}$  can be calculated easily by  $D_{avg}/v$ , where  $v$  is the moving speed of the actuators. We also compare our results with the baseline algorithm, which divides the nodes into different groups according to their weights, and provides a route for each group to be walked through by an actuator. More specifically, the nodes are divided into  $M$  weight range and form groups accordingly. Then, a TSP route is formed to contain the nodes in its group. Finally,  $M$  routes are formed in the network and each of them is patrolled by a distinct actuator.

Figure 3 shows the average inter-arrival distance  $D_{avg}$  for an actuator to visit the sensor locations periodically under uniform random sensor distribution with  $N = 50$  and  $M = 5$ . It evaluates distance  $D_{avg}$  to the sensor locations with weights in the ranges 0.0-0.2, 0.2-0.4, 0.4-0.6, 0.6-0.8, and 0.8-1.0, respectively. It demonstrates that the sensor locations with higher weights achieve shorter inter-arrival distance  $D_{avg}$ , and hence shorter inter-arrival time  $A_{avg}$  in our algorithm. Similarly, Figure 4 shows  $D_{avg}$  in the same network with  $N=100$  and  $M=7$ . Again, sensor locations with higher weights are visited more frequently, which follows our design objective. In comparison with the baseline algorithm, our algorithm guarantees much shorter inter-arrival time for highly weighted sensors. It differentiates the inter-arrival time among the sensors effectively, while a reasonable low overall inter-arrival time is realized simultaneously.

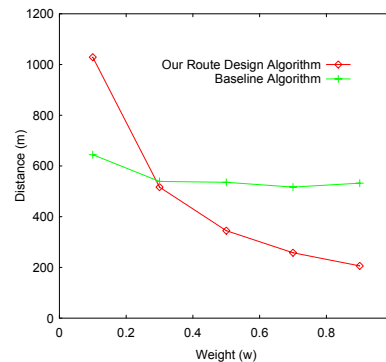


Fig. 3. Average inter-arrival distance under uniform sensor distribution with  $N=50$  and  $M=5$ .

*B. Average Actuator Inter-Arrival Time under Non-Uniform Sensor Distribution*

We next evaluate our algorithm with a non-uniform sensor distribution. Sensor nodes are deployed unevenly, which may result in clusters and network partitions. Specifically, we place the sensors into three clusters randomly in this experiment.

Figure 5 shows the average actuator inter-arrival distance  $D_{avg}$  with  $N=50$  and  $M=5$ . Under a cluster-based sensor

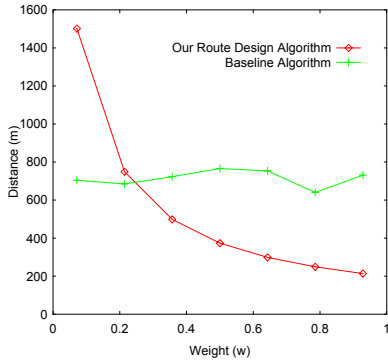


Fig. 4. Average inter-arrival distance under uniform sensor distribution with  $N=100$  and  $M=7$ .

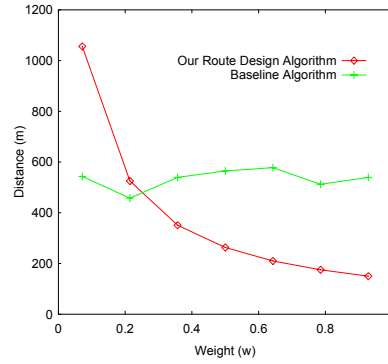


Fig. 6. Average inter-arrival distance under non-uniform sensor distribution with  $N=100$  and  $M=7$ .

deployment, our algorithm also achieves better performance than the baseline algorithm. An interesting observation is that the  $D_{avg}$  under cluster-based sensor deployment is lower than that under uniform random deployment for both algorithms. The reason is that the nodes are less distributed under cluster-based deployment, so they have shorter Euclidean distances and lead to the routes with shorter lengths. Similarly, Figure 6 shows the result for the same network with  $N=100$  and  $M=7$ . Again, shorter inter-arrival distance  $D_{avg}$  is achieved by our algorithm for sensors with higher weights in comparison with the baseline algorithm. Our algorithm performs generally better than the baseline algorithm for all sensors except those with the lowest weights.

they are visited by all actuators. Similarly, Figure 8 shows the results of the same experiment with  $N=100$  and  $M=7$ . An interesting observation is that the sensor locations with lower weights suffer from greater variance on  $D_{avg}$ . The reason is that these locations are visited by fewer actuators, and the difference on the route lengths may lead to more significant variations on their final inter-arrival distance  $D_{avg}$ .

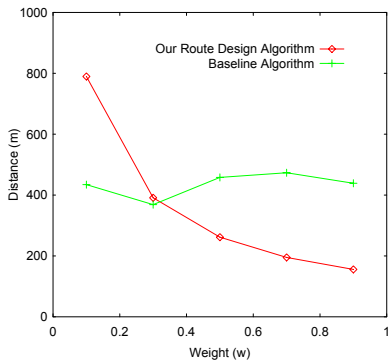


Fig. 5. Average inter-arrival distance under non-uniform sensor distribution with  $N=50$  and  $M=5$ .

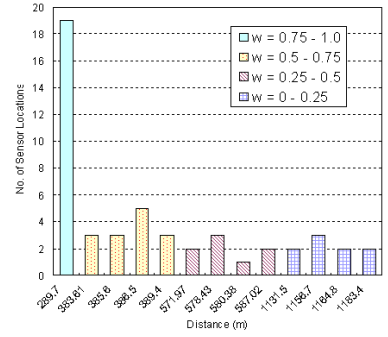


Fig. 7. Distribution on average inter-arrival distance with  $N=50$  and  $M=4$ .

### C. Variance on Average Actuator Inter-Arrival Time

We have observed  $D_{avg}$  for the sensor locations with the same weight in the previous experiments. We are also interested to know whether a sensor location can achieve similar  $D_{avg}$  to the others with the same weight. Figure 7 shows the distribution of  $D_{avg}$  in a network with  $N=50$  and  $M=4$  under uniform sensor distribution. It is clear that the sensor locations on the same weight range obtain quite close  $D_{avg}$ , so they have similar waiting time to the actuators. Note that the locations with the highest weight range achieves the same  $D_{avg}$ , since

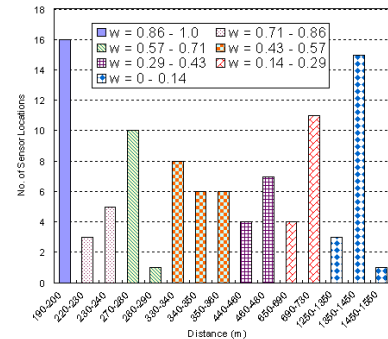


Fig. 8. Distribution on average inter-arrival distance with  $N=100$  and  $M=7$ .

## VI. CONCLUSION AND FUTURE WORK

In this paper, we focused on wireless sensor networks with multiple actuators and their route design. We demonstrated that

the problem is NP-hard and proposed an effective MST-based approximation. Our algorithm aims at minimizing the overall inter-arrival time of actuators, while differentiating the visiting frequency to the sensor locations with different weights. Simulation results suggested that our algorithm remarkably reduces the average inter-arrival time. Our future work is to further improve the performance of the route design algorithm and to consider cooperations among the actuators.

## REFERENCES

- [1] I. F. Akyildiz, W. Su, and T. Sandarasubramaniam, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 5, pp. 393–422, 2002.
- [2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. of ACM MobiCom*, Seattle, Washington, U.S., 1999.
- [3] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proc. of the 38th Hawaii International Conference on System Sciences (HICSS)*, 2005.
- [4] J. Luo and J. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proc. of the 24th IEEE INFOCOM*, Mar 2005.
- [5] I. F. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Elsevier Ad Hoc Networks Journal*, Oct 2004.
- [6] E. C.-H. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "Reliable reporting of delay-sensitive events in wireless sensor-actuator networks," in *Proc. of The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'06)*, Vancouver, Canada, Oct 2006.
- [7] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *Proc. of 25th IEEE International Realtime Systems Symposium (RTSS'04)*, 2004, pp. 296–305.
- [8] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proc. of the IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.
- [9] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design fo sensor networks," in *Proc. of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, Apr 2003.
- [10] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2004.
- [11] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. of ACM MobiHoc'04*, Mar 2005.
- [12] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C.-G. Lee, "Partitioning-based mobile element scheduling in wireless sensor networks," in *Proc. of SECON*, Santa Clara, U.S., Sep 2005, pp. 386–395.
- [13] N. Christofides, A. Mingozzi, and P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Mathematical Programming*, no. 1, pp. 255–282, Dec 1981.
- [14] T. Ralphs, L. Kopman, W. Pulleyblank, and L. Trotter, "On the capacitated vehicle routing problem," *Mathematical Programming*, no. 2-3, pp. 343–359, Jan 2003.
- [15] L. Lee, K. Tan, K. Ou, and Y. Chew, "Vehicle capacity planning system: A case study on vehicle routing problem with time windows," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, pp. 169–178, 2003.
- [16] C. Miller, A. Tucker, and R. Zemlin, "Integer programming formulation of the travelling salesman problem," *Journal of the Association for computing machinery* 7, 1960.
- [17] J. Bentley, "Fast algorithms for geometric traveling salesman problem," *ORSA Journal on Computing*, pp. 387–411, 1992.
- [18] S. Arora, "Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems," *Journal of ACM*, vol. 45, no. 5, pp. 753–782, Sep 1998.
- [19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and et al., *Introduction to Algorithms*. The MIT Press, 2002.
- [20] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A real-time routing protocol for sensor networks," in *Proc. of the IEEE ICDCS*, Providence, RI, U.S., May 2003, pp. 46–55.