Predictive Scheduling for Spatial-dependent Tasks in Wireless Sensor Networks

Hua Huang, Shan Lin, Anwar Mamat and Jie Wu
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122
{hua.huang, shan.lin, anwar, jiewu}@temple.edu

***Abstract*—Recent advances in wireless energy transfer technology propels the developments of renewable sensor networks. To sustain the operation of a sensor network, a mobile charger is used to recharge each node. Consider each node's recharge request as a task with a soft due time, the mobile charger needs to dynamically schedule these tasks. This scheduling problem is very challenging, since both the nodes' due times and their locations have to be considered. Existing solutions to this problem usually assume fixed paths generated by the travelling salesman or Hamilton cycle algorithms. These solutions suffer from high deadline miss ratios. In this paper, we investigate maximum response ratio based scheduling algorithms that can dynamically select the path of chargers for higher network coverage ratio. To further improve the performance, we exploit the dependencies of different recharge tasks and predict the impacts of executing one charging task to the others. With extensive trace-driven analysis, our algorithms significantly outperform existing algorithms.**

***Keywords*—Vehicle scheduling, spatial dependency, wireless charging**

I. Introduction

Sensor networks are deployed for various military surveillance [1], scientific exploration [2], and first responder applications [3]. Such applications may use visual or acoustic sensors, which have high energy consumption rates. To sustain long-term system operations, sensor nodes need to be recharged periodically after their deployments. With recent advances of wireless energy transfer technology, a mobile vehicle equipped with wireless recharge device can be used to recharge sensor nodes automatically [4], [5]. In small scale systems with low energy consumption rates, the recharge schedule is trivial. We can generate a fix schedule to recharge nodes using travelling salesman algorithms. However, in large scale networks with high energy consumption rates, the travelling time of the mobile charger is comparable to the lifetime of the sensor nodes. Under this condition, using fix schedules will result in low coverage ratio. This is because fix schedules can not adapt itself to the dynamics of sensor nodes' energy levels.

The intrinsic spatiotemporal constraints make this dynamic mobile charge scheduling problem unique and challenging. To solve this problem, both the mobile charger's travelling time and the sensor nodes' due time must be considered. For example, if the travelling time is much shorter than sensors' lifetime, the mobile charge scheduling algorithm boils down to the real-time scheduling problem on a single machine. This can be solved by classic algorithms such as Earliest Deadline First (EDF) algorithm. On the other hand, if the travelling time is much longer than sensors' lifetime, mobile charger basically needs to select the most efficient route to pass every node. This becomes the travelling salesman problem, which is well-known to be NP-hard. When the travelling time is comparable to the sensors' lifetime, this problem is not studied before. There lacks efficient solutions: classic scheduling algorithms [8] focus on task deadlines, while computational geometry algorithms [9] mainly seek efficient routes in multidimensional spaces regardless of deadlines. There are a few related works [6], [7] that deal with the mobility scheduling in the context of sensor networks. Different from these works, we focus on scheduling algorithm designs dealing with spatiotemporal dynamic tasks.

To address this problem, we take each node's recharge request as a task with a soft due time. We note that the due times of tasks are independent to each other, since the energy consumption rate of each node is determined by its own workloads. However, these tasks are dependent in space. Given the same due time, recharging a node with high spatial proximity requires much less travelling overhead than recharging a node far away. To characterize this special feature of mobile charge, we introduce the spatial dependency task model. This model quantifies how the execution of one task influences all others. The spatial dependency model allows nodes lying close to the travelling path to have higher priorities than the other tasks. However, this alone is not enough to avoid falling into local optimal, partially due to the lack of global information in a distributed large scale networks. Therefore, we further design predictive heuristics to look ahead with rough grained global task estimations. Our solution demonstrates a good tradeoff between scheduling coverage and cost. We evaluate our solutions in extensive simulations. The simulation results show that significantly higher coverage with little travelling overhead can be achieved.

II. Task and System Models

We consider a sensor network of n nodes deployed in a two-dimensional space. These sensor nodes have independent energy consumption rates. A Mobile Charger (MC) is a vehicle equipped with a wireless power recharge device. It travels to the sensor nodes one by one to deliver energy.

1) Task Model: the recharge requests of sensor nodes are modelled as tasks. A task τ_i is represented by processing time, waiting time and soft due time. These three parameters are denoted by $\{p_i, t_i^w, d_i\}$. Since these parameters are dependent on time, the MC has to regenerate the task list each time when it is to make recharge schedule decisions. These parameters are described as follows:

Processing time p_i is modelled as the travelling time for the mobile charger to move to the next target. If we use v to represent MC's speed and l_{next} to represent the distance between current and target nodes, then p_i can be calculated as bellow:

$$p_i = \frac{l_{next}}{v} \tag{1}$$

Waiting time t_i^w is modelled as the amount of time the node has been waiting since its last recharge.

Soft due time d_i is modelled as the amount of time before the corresponding node's energy depletes. In some cases, tasks may be past due. d_i is defined to be negative under these conditions. If we denote the node's maximum energy storage by e^{max} and energy consumption rate by r_i, then the due time of a task can be calculated by the following equation:

$$d_i = e^{max}/r_i - t_i^w \tag{2}$$

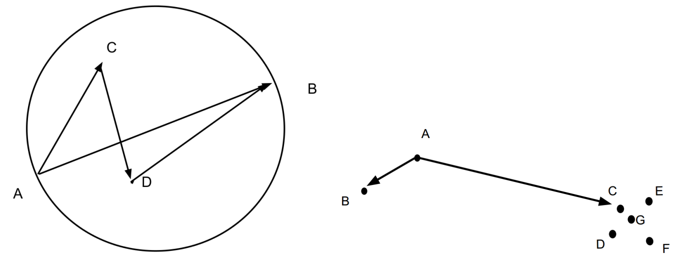
When $t_i^w > e^{max}/r_i$, it means that the node has been waiting too long so that it has already run out of energy.

2) Spatial Task Dependency: Different from traditional task scheduling problems [8], [10], [11], the execution of one task can affect the processing time of all other tasks in our Dynamic Mobile Charger scheduling problem. This is because when the MC executes one task, it moves and changes its relative locations to all other nodes. This will in turn affect all their processing time by equation 1.

An example is shown in Figure 1a. Assume MC is located at node A. Then the processing time of the task to recharge node B is $p(B) = d(\overline{AB})/v$. But if MC is located at node C, then the processing time of the same task is $p(B)' = d(\overline{CB})/v$. In this case, $p(B) > p(B)'$. In other words, the time it takes to recharge node B depends on the MC's current location. This indicates that potential performance gain could be attained by exploring the sensor nodes' spatial dependencies. We discuss two motivating examples as follows.

Nodes near MC's path: In Figure 1a, suppose the MC's path is from node A to node B. Processing time of τ_B is $p(B) = d(\overline{AB})/v$. If $p(C) + p(D) + p(B)' = d(\overline{ACDB})/v < d_B$, then MC can take advantage of this laxity and recharge node C, D before B without missing B's due time. The additional cost for this new scheduling can sometimes be small if C and D are located near path $\overline{AB}$. The extra cost introduced for τ_B is $p(C) + p(D) + p(B)' - p(B)$.

Node clusters: When a group of closely located nodes have similar approaching deadline, higher priority should be given to them. As illustrated in Figure 1b, say MC is located at node A, the processing time of node B is shorter than the other nodes ($p(b) < p(d) \approx p(e) \approx p(f) \approx p(g) \approx p(h)$). However, if MC recharges B first, other nodes will die before being recharged. A solution for lower due time miss ratio is to first charge the node clusters (C, D, E, F, G) with a higher density of approaching due times.



(a) Nodes near path (b) Spatial Cluster

Fig. 1: Task Spatial Dependency

3) Model Metrics: Based on this task model, we can define the following metrics to evaluate any scheduling algorithms.

Coverage: If the energy of a sensor node runs to 0, it will stop functioning. We define the Coverage of the network to be the percentage of sensor nodes functioning at the moment. It can be defined by the following equation:

$$Coverage = \frac{\text{Number of functioning nodes}}{\text{Number of all nodes}} \tag{3}$$

MC Cost: $Cost$ denotes the energy overhead for the MC to recharge the sensor network. In this paper, we consider only the travelling cost of MC. Since MC is travelling at a constant speed, the cost is linearly proportional to the travelling time:

$$Cost = \sum_{\text{all charged nodes}} K \times p_i, \tag{4}$$

where K is the energy consumption per unit time.

III. Algorithm

In this section, we propose several algorithms for MC scheduling.The MC makes scheduling decisions only when it finishes recharging a node. Our goal is to provide a reliable network such that the coverage ratio is maximized, while lowering MC cost and maintaining fairness between nodes and between regions.

A. Minimum Weight First

In Minimum Weight First (MWF), we seek for a tradeoff between processing time and due time. Therefore, we employ the parameter α to represent their relative importance. The priority of MWF is defined as follows:

$$Priority_i = \alpha p_i + (1 - \alpha) max(0, d_i), \tag{5}$$

where p_i is the processing time defined in equation 1. It represents the cost of executing the task. d_i is the node's soft due time by equation 2. The task with minimum $Priority_i$ will be selected as the next target. When $\alpha = 1$, the problem becomes the Greedy TSP. When $\alpha = 0$, only the deadline determines the schedule.

B. Maximum Response ratio First

Maximum Response ratio First (MRF) scheduler defines the priority of each task to be dependent on both its processing time and waiting time. The longer tasks wait, the higher priority they gain. This will prevent infinite postponement (process starvation) [8]. Intuitively, the higher energy consumption rate a node has, the less time it can afford to wait. Therefore we multiply the energy consumption rate r_i with the waiting time t_i^w. MRF's priority is defined as follows:

$$Priority_i = \frac{t_i^w \times r_i + W_d max(0, -d_i)}{p_i} \tag{6}$$

The term $max(0, -d_i)$ will be non-zero only when a task is past due. We assign the weight W_d to this term to reflect the additional penalty of lateness.

C. Spatial Laxity based Heuristics

As described in section II, spatial dependency of nodes exists. Based on these observations, we propose two heuristics: Spatial Laxity Filling (SLF) and Spatial Laxity Clustering (SLC). SLF searches nodes near MC's travelling path. SLC partitions the sensor network into spatial clusters and computes each node's priority based on both its own and its cluster's energy conditions.

1) Spatial Laxity Filling: As illustrated in figure 1a, Assume MC is located at node A. The Spatial Laxity Filling (SLF) scheduler firstly selects the target node using MRF, say B. Then the scheduler will check if there are any nodes located inside the circle whose diameter being line $\overline{AB}$. If so, SLF will select the node that lies closest to line $\overline{AB}$ to be recharged before B.

2) Spatial Laxity Clustering: Spatial Laxity Clustering (SLC) divides the network into different clusters and takes into account the estimation of each cluster's urgency when scheduling. SLC firstly divides the node space into m non-intersecting clusters. SLC defines the priority of each cluster using the following equation:

$$Priority^c = \frac{\sum_{nodes\ in\ cluster} t_i^w \times r_i}{p^c}, \tag{7}$$

where t_i^w is the waiting time and r_i is the energy consumption rate of each node. The term $\sum_{node\ in\ cluster} t_i^w \times r_i$ is an estimation of the cluster's urgency. p^c is modelled as the travelling time for MC to travel from its current location to the geometric centroid of the node cluster.

SLC will select the next cluster with maximum $Priority^c$. After entering a cluster, MC will be scheduled to recharge nodes within this cluster using MRF. After the urgency estimation of the cluster falls below a threshold, the charger will begin to travel to next cluster. Using this algorithm, better fairness among different clusters can be achieved.

IV. Evaluation

We have implemented a simulation framework to evaluate the scheduling algorithms proposed in this paper. In the simulation, the sensor network consists of 15×15 nodes deployed in a grid manner. We assume the charger has the global energy information of all the nodes. A node's recharge is assumed to be finished at the moment MC moves to it. We will present the results of MRF, SLF, SLC and MWF with $\alpha = 0.2, 1$ in this section.

A. Coverage

Firstly, we compare the performance of different mobile charger algorithms in terms of network $Coverage$ defined in equation 3. In each experiment, the charger totally recharges 10000 times before it stops. Then we compute the network's average coverage ratio during the experiment. For each algorithm, we repeat this experiment 50 times and compute the average of these two statistics. The results are shown below.

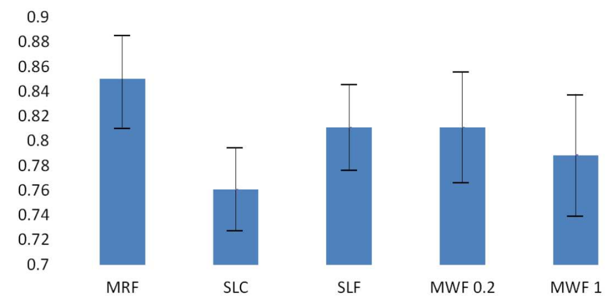


Fig. 2: Coverage

From figure 2 we can see that MRF achieves the highest coverage ratio among all the five algorithms tested. This is due to its ability to achieve a balance between waiting time and processing time in scheduling. On the other hand, SLC has a lower performance. The reason is that a large travelling cost can occur during cluster switching.

When MWF algorithm has the parameter $\alpha = 0.2$, it has a better performance than when $\alpha = 1$. However, as can be seen in later analysis, this is at the expense of higher recharging overhead.

B. Due Time

Each time a sensor node is being recharged, we record that node's soft due time defined in equation 2. The Cumulative Distribution Functions(CDF) of the Due time are plotted below:

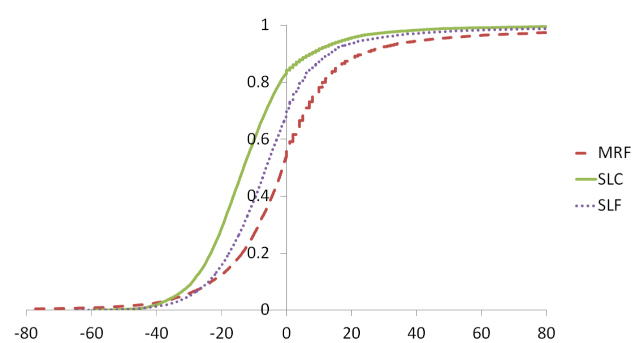


Fig. 3: Due Time

From figure 3 we can see that MRF is able to maintain a smaller deadline miss ratio. This is because MRF focuses more on nodes with low energy. On the other hand, although SLF can ensure the next task's due time will not be missed, a larger number of dead nodes still occur. This is because inserting nearby nodes can increase the waiting time of nodes lying far away.

C. Fairness

In order to evaluate the spatial fairness of these scheduling algorithms, we divide the sensor network equally into 3×3 clusters with rectangle shape. Each cluster has 5×5 nodes. The coverage ratio is computed for each cluster using the following equation:

$$Coverage_i = \frac{\text{Number of functioning nodes in } Cluster_i}{\text{Number of all nodes in } Cluster_i} \tag{8}$$

Then we can compute the standard deviations of $Coverage_i$ between different clusters for each algorithm. The results are shown below.

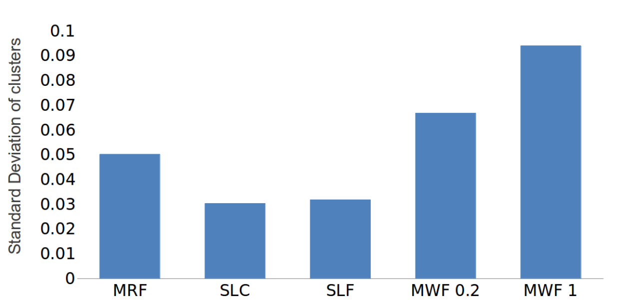


Fig. 4: Fairness

This figure shows that SLC has the lowest standard deviation, which means the coverage ratios between different clusters are similar. This is due to the fact that SLC can adjust its scheduling based on each cluster's need. On the other hand, MRF has high standard deviation. This indicates that MRF achieves higher coverage ratio at the expense of spatial fairness. The fairness of MWF with $\alpha = 1$ is the worst. This is because this scheduler favours too much about nearby nodes, while being unfair to nodes far away.

D. Cost

We sum up all the travelling distance of the MC for each experiment. Then we normalize these distances by dividing them by the travelling cost of MWF with $\alpha = 1$. This is because MWF with $\alpha = 1$ is a classical greedy solution to Travelling Salesman Problem. Besides, it achieves minimum travelling cost in all the algorithms we designed.

As shown in 5, we can see that SLC has a large travelling cost, due to its long distance travel when changing clusters. SLF achieves a lower cost than MRF because it selects to recharge nodes which are located close to the recharging route. This proves to be efficient in terms of travelling cost.

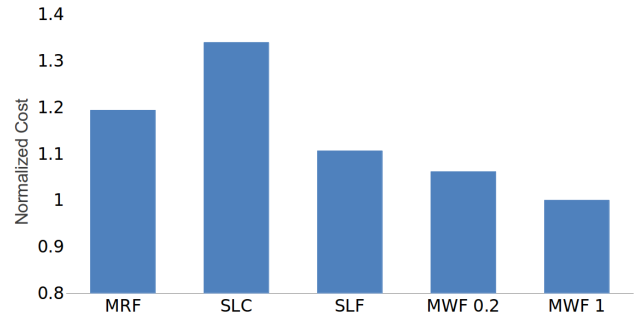


Fig. 5: Cost

To summarize, we can see that MRF can achieve a higher coverage ratio. The advantage is more significant when the recharge workload is heavy. When we hope to minimize the travelling cost of MC, we can choose SLF that can reduce travelling cost, while not incurring serious performance degradation. If the geometric condition of the sensor network is so complex that energy consumption rates of nodes from different regions differ greatly, we can choose SLC scheduler so that fairness between different regions can be ensured automatically.

Acknowledgement

This research was supported in part by the NSF CNS 1218718, NSF CNS 1218718 , CNS 1065444, CCF 1028167, CNS 0948184 and CCF 0830289.

References

[1] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, *et al.*, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks*, 2006.

[2] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, *et al.*, "Envirotrack: Towards an environmental computing paradigm for distributed sensor networks," in *Distributed Computing Systems, 2004.*

[3] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu, "Automatic and robust breadcrumb system deployment for indoor firefighter applications," ser. MobiSys '10.

[4] R. Sugihara and R. K. Gupta, "Speed control and scheduling of data mules in sensor networks," *ACM Transactions on Sensor Networks*, 2010.

[5] S. Zhang, J. Wu, and S. Lu, "Collaborative mobile charging for sensor networks," in *Proc. of the 9th IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2012.

[6] G. Xing, J. Wang, K. Shen, Q. Huang, X. Jia, and H. C. So, "Mobility-assisted spatiotemporal detection in wireless sensor networks," in *28th IEEE International Conference on Distributed Computing Systems*, 2008.

[7] L. Luo, C. Huang, T. Abdelzaher, and J. Stankovic, "Envirostore: A co-operative storage system for disconnected operation in sensor networks," in *26th Annual IEEE Conference on. Computer Communications*, 2007.

[8] P. Brucker, *Scheduling Algorithms*, 2004.

[9] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, 1992.

[10] B. Sprunt, L. Sha, and J. Lehoczky, "Aperiodic task scheduling for hard-real-time systems," *Real-Time Systems*, 1989.

[11] J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, and K. Kennedy, "Task scheduling strategies for workflow-based applications in grids," in *Cluster Computing and the Grid, 2005.*

[12] A. K. Wong and A. M. Goscinski, "The impact of under-estimated length of jobs on easy-backfill scheduling," in *Parallel, Distributed and Network-Based Processing*, 2008.