**Revision Report of Submission TMC-2015-03-0192**

We thank the comments with cares and insights made by the reviewers, which are helpful for improving the quality and readability of our paper. In our revised paper, we have made detailed explanations and changes in response to all the reviewers’ comments. Here, we explain our revisions based on each of the comments and suggestion.

**Response to Reviewer 1’s Comments**

1. **Comments:** *The authors fail to motivate the reader. After reading Section I, it is not clear what the problem is, why it is important, and what the authors are trying to solve.*

**Response:** Thanks for your careful reading. We have reorganized Section 1 in the new version. Specifically, we have revised the second paragraph in Section 1. In this paragraph, we have demonstrated the scenarios of applications to improve the readability of the motivation and addressed problem in this paper. Then, we have added the third paragraph in Section 1 to illustrate the model of a sampling task, and focused our study on the “continuous interval sampling tasks”. To be clear, we have revised the fourth paragraph to present the vital importance of a strategy for maximizing data sharing among tasks. The fifth paragraph has been revised to present the challenges when maximizing the data sharing among tasks in WSNs.

1. **Comments:** *The paper is on task scheduling; however, there is hardly any discussion on related work on this topic when there are quite a number of papers in this area. There are quite a few unrelated references.*

**Response:** Thanks for your careful reading. We have added Section 2 and presented the reference on this topic. Those unrelated references have been removed. We reorganized Section 2.1 and highlighted the difference between our work and the previous study on the topic of task allocation and the scheduling of *sampling interval* of tasks.

1. **Comments:** *The paper talks about data sampling and data sharing between overlapping tasks. What are the implications?*

**Response:** Thanks for your careful reading. We have reorganized Section 1 and presented the data sampling and data sharing in the third, fourth and fifth paragraphs in a clear way. We have illustrated that data sharing can be used to reduce the redundancy of sampled data. Such sampled data is produced by the sampling tasks. When the *time window* of sampling tasks are overlapping, data sampled in the overlapping region of time may be used by more than one task, i.e., data sharing. That is, data sharing exists in the process of data sampling for overlapping tasks, and we can use it to reduce the volume of sampled data.

1. **Comments:** *The work is concentrated on the improvement of the scheduling algorithm proposed by Fang et al. [9].*

**Response:** Thanks for your careful reading. We have added Section 2 and outlined the difference between our work and the previous work such as Ref. [11] (The Ref. [11] in the new version is the Ref. [9] in the first submission.).

Fang et al. propose an effective sampling approach for interval sampling tasks on a single sensor node. The 2-factor approximation algorithm in Ref. [11] is the state-of-the-art method to maximize data sharing amongst tasks on a single node. However, there are two weak points in Ref. [11]. First, the proposed scheduling method schedules sampling tasks in the descending order of the end time of a task, which, as a result, neglects data sharing between overlapping tasks. Moreover, Fang et al. assume that all tasks have a same length of *sampling interval*, which is too ideal and not practical. By contrast, we observe that multiple tasks may be overlapping, and thereby data sharing exists. Our solution exploits this information by designing a crucial operation, namely, *COMBINE*. *COMBINE* achieves better performance on maximizing data sharing than the scheduling method in Ref. [11].

Second, the solution in Ref. [11] only focuses on task scheduling on a single sensor node, and does not consider the process of task allocation in the scope of a whole wireless sensor network. Unfortunately, the performance of task scheduling is sensitive to the strategy of task allocation. Only does optimizing the process of the former not achieve the final optimization for a deployed system. Our solution is more general and practical because of jointly optimizing the process of task allocation and scheduling sampling interval.

1. **Comments:** *There is too much discussion on proving the complexity of the algorithm and less effort on the experimentation.*

**Response:** Thanks for your careful reading. We have revised Section 3.3 and the proof of Theorem 3 in Section 5.2 in the new version. The discussion on proving the complexity of the algorithm is presented as briefly as possible. Furthermore, we have polished Section 6 and revised the last paragraph of Section 6.3 in the new version. The necessary discussion of the difference of performance between *PRUNE* and *RANDOM* has been added.

Since our method is a general solution, we indeed paid more attention to the proof on the theoretical performance bound of the proposed algorithms. Inspired by your suggestion, we have reorganized the paper by presenting the theoretical analysis about algorithms in a simple manner. Especially, we have polished the process of proving the complexity of the algorithms (Lemma 1, Theorem 2 and Theorem 3) in a more clear and way.

1. **Comments:** *Scheduling and allocation problem should be formally defined. What is the purpose of MAX-DHP problem in relationship with MIN-SA (assume it is used to simplify the proof of Lemma 1)? Definition of MAX-DHP problem should also be given.*

**Response:** Thanks for your careful reading. We have added Define 4 and Define 5 in Section 3.1 for the problem of task allocation and the scheduling of *sampling interval* of tasks. Besides, we have added Define 12 for Maximal Directed Hamilton Path (MAX-DHP).

Data sharing between overlapping tasks exists in the processes of task allocation and scheduling of *sampling interval*. Both problems are NP complete due to the complexity of computing the minimal volume of sampled data. Since the strategy of task allocation has a great impact on the performance of the scheduling problem, solving either scheduling problem or allocation problem is meaningless. Thus, we have studied both problems as a whole and designed an effective solution for the joint optimization problem. MAX-DHP problem is NP complete, which has been proved by previous study. We have transformed it to the problem of computing the minimal volume of sampled data (See Lemma 1 in Section 3.3).

1. **Comments:** *4.85% is not a significant energy saving as claimed over GA. Also, similar comments on data loss rate.*

**Response:** Thanks for your careful reading. We have revised the discussion of experiment in Section 6.2.

First of all, 4.85% means that our algorithm *COMBINE* outperforms *GA* and saves more energy of battery by 4.85% than *GA* over a time slot on average. Here, a time slot is a period of 50 seconds. Similarly, comparing to *GA*, *COMBINE* leads to less data loss rate by about 4% over each slot on average. Since a wireless sensor node works over decades of days, such improvements are appreciable and worthy to being exploited.

Second, Section 6.2 evaluates the performance of data sharing when running *COMBINE* on a single node, which is not the total improvement of our solution. Section 6.3 further presents the improvement of our solution. Since it is unknown how to use *GA* for task allocation, when adopting a random strategy of task allocation for 300 tasks in a *k*-coverage and *r*-redundant network with *k*=5 and *r*=2, *GA* consumes more energy and leads to more data loss rate than our solution by over 35% and 30% due to poorly exploiting data sharing among tasks. Such benefit becomes more obvious when the number of tasks increases and these improvements are appreciable for WSNs.

1. **Comments:** *There are some unresolved references.*

**Response:** Thanks for your careful reading. We have reorganized Section 2 and revised the references. The unresolved references have been removed.

1. **Comments:** *Poor choice of words (e.g. 'novel' instead of 'new'; 'initiate' instead of 'initial' 'apartment'; 'quintessential')*

**Response:** Thanks for your careful reading. These typos have been revised in the new version.

1. **Comments:** *There are a number of grammatical errors as well.*

**Response:** Thanks for your careful reading. The paper has been revised and polished carefully in this version. Grammatical errors have been checked carefully.

**Responses to Reviewer 2's Comments**

1. **Comments:** *The motivation is not quite strong. Difference between Ref. [9] is not quite obvious. A similar 2-factor approximation algorithm is also proposed by [9]. Although [9] uses tasks T\_1, T\_2,.... for a single node, the algorithm can be readily extend to the network given T\_i represents the different tasks. Is it true that - the scheduler only sees different T\_i (s) no matter which nodes they have been assigned? If this perception is wrong, it is better to clearly point out in the paper. After reading this paper, the difference between [9] is vague. The authors should clearly state, and elaborate the difference between the two works.*

**Response:** Thanks for your careful reading. We have added Section 2 and reorganized the related work. The difference between Ref. [11] (Ref. [11] in this version is the Ref. [9] in the first submission.) and our work has been highlighted and discussed in Section 2.1 specifically.

Fang et al. propose an effective sampling approach for interval sampling tasks on a single sensor node. Although the 2-factor approximation algorithm in Ref. [11] is the state-of-the-art method to maximize data sharing amongst tasks on a single node, it has many shortages for a wireless sensor network.

First, Fang et al. aim to propose method for optimizing data sampling on a single node. But Fang et al. do not clarify the effectiveness of the method in Ref. [11] when taking the scope of a wireless sensor network. Since the strategy of task allocation dominates the performance of the method of scheduling *sampling interval*, the method in Ref. [11] cannot work well in a physical network due to neglecting the process of task allocation. For example, if the *time window* of tasks which are allocated to a sensor node are not overlapping, the method of scheduling *sampling interval* cannot exploit data sharing to reduce redundant data sampling. Such the weakness limits the performance of method in Ref. [11] for a wireless sensor network.

Second, the proposed scheduling method schedules sampling tasks in the descending order of the end time of a task, which, as a result, neglects data sharing between overlapping tasks. Moreover, Fang et al. assume that all tasks have a same length of *sampling interval*, which is too ideal and not practical. By contrast, we observe that multiple tasks may be overlapping, and thereby data sharing exists. Our solution exploits this information by designing a crucial operation, namely, *COMBINE*. *COMBINE* achieves better performance on maximizing data sharing than the scheduling method in Ref. [11].

Third, the solution in Ref. [11] only focuses on task scheduling on a single sensor node, and does not consider the process of task allocation in a wireless sensor network. Unfortunately, the performance of algorithms of task scheduling is sensitive to the strategy of task allocation. Only does optimizing the process of the former not achieve the final optimization for a deployed system. Our solution is more general and practical because of jointly optimizing the process of task allocation and that of scheduling *sampling interval*.

When a sampling task is generated, it will be allocated to *r* sensor nodes. Here, a sensor node is a scheduler. All these tasks are different, and thus a scheduler only sees different tasks indeed.

1. **Comments:** *On the contribution of algorithm - the task scheduling, allocation problems are well-studied topics in general. The algorithm presented in the paper works in WSNs whereas their performance in the general setting is unknown. At least, the literature review doesnt provide any information. For example, is the 2-factor approximation algorithm the best one ? Are there any inapproximation ratio exists for the problem to derive polynomial-time algorithms? What are other algorithms proposed and what are their ratios and complexity ? Answers to these questions are expected as they will give readers a background how to evaluate your algorithm design in a general setting.*

**Response:** Thanks for your careful reading. We have reorganized Section 2 and highlighted the difference between our solution and the state-of-the-art method in Ref. [11] in Section 2.1. We have revised the last paragraph of Section 3.1, and presented that some other more advanced but complicated strategies for the joint optimization is not practical for a resource-constrained node. Besides, we have revised Section 3 in order to present the analysis of our solution in a clear way.

We do not present our idea clearly in the first submission. We aim to provide a general solution for jointly optimizing the allocation of tasks and the scheduling of *sampling interval*. The performance of our solution is analyzed theoretically and the rigorous bound has been presented in the general setting. We have proved that the joint optimization problem is NP-hard in the general setting and there is no polynomial-time algorithm. Inspired by your suggestion, we have investigated the joint optimization problem in other fields. Several methods, including branch and bound techniques, require high computational complexity and are not applied to a wireless sensor node.

Ref. [11] provides a state-of-the-art approximation algorithm for the scheduling problem. However, the proposed scheduling method in Ref. [22] schedules sampling tasks in the descending order of the end time of a task, which, as a result, neglects data sharing between overlapping tasks. Moreover, Fang et al. assume that all tasks have a same length of *sampling interval*, which is too ideal and not practical. By contrast, we observe that multiple tasks may be overlapping, and thereby data sharing exists. Our solution exploits this information by designing a crucial operation, namely, *COMBINE*. *COMBINE* achieves better performance on maximizing data sharing than the scheduling method in Ref. [11].

1. **Comments:** *It seems the allocation and scheduling algorithms are centralized. However, how to disseminate decisions from the central controller and expected message overhead are not mentioned in the paper. These are equally important information since the main claim of this paper is the design of these algorithms in a network.*

**Response:** Thanks for your careful reading. We have added Section 7 and highlighted the details of the design of our algorithm in a wireless sensor network in the first paragraph of Section 7.

We aim to provide a general solution: *CATS* for jointly optimizing the allocation of tasks and the scheduling of *sampling interval*. *CATS* is a centralized algorithm and works well on the sink node. In a wireless sensor network, the sink node runs *CATS* and gets the strategy of task allocation. It then allocates the sampling tasks to the sensor nodes. The allocation message will be added into packets and disseminated to sensor nodes with sampling tasks together. Comparing to the volume of sampled data of sampling tasks, such expenditure on the allocation solution is rather little. For example, as illustrated in Section 6.3, adopting the strategy of task allocation can reduce redundancy of sampling data by more than 30%. Therefore, it is worthy to trading little expenditure for appreciable data sharing.

Besides, some other details such as the protocol of communication and the strategy of routing do not impact the performance of our algorithms. We do not adopt optimization on such communication protocols or routing strategy. We aim to propose an effective solution for the joint optimization problem in general setting of a network.

1. **Comments:** *A little bit too many symbols - some self-defined, weird ones too which are not commonly used. This degrades the readability greatly. The reviewer suggests the authors minimize the usage of these symbols rather than inherit from [9] directly.*

**Response:** Thanks for your careful reading. We have revised the table of symbols in Section 3.1. Rare and weird symbols have been removed in the new version. To be clear, we use as few symbols as possible to present the definitions and the proofs in this version.

1. **Comments:** *Theorem 3 - proofs, please give the general case rather than proving by examples and you may always explain the proofs through an example later. Some other proofs in this paper may have the same problem.*

**Response:** Thanks for your careful reading. We have revised Section 3.3 and Section 4 and Section 5.2. Specifically, we have revised and polished the proofs of Theorem 3. The example in Theorem 3 has been removed from the proof. We present the example to clarify the proof instead. Moreover, other proofs have been checked in the new version.

1. **Comments:** *On presentation problems - [?], reference not added, several spots; t.count++ -> t.count = t.count + 1; Fig. 7 doesnt display so well on black-white print-outs; the procession of construction -> the process of construction; replaced by the novel task-> replaced by the new task.*

**Response:** Thanks for your careful reading. The presentation problems have been revised. All the references have been checked and added. We have replaced “*t.count++*” with “*t.count + 1*” in Algorithm 3 and 4. Figures (including Fig. 7, Fig. 8 and all the experimental figures) have been presented well if they are printed. The weird expressions such as “*the procession of construction*” and “*replaced by the novel task*” have been revised in the new version.

**Responses to Reviewer 3's Comments**

1. **Comments:** *Since the allocation and scheduling problems are tightly coupled, it is more meaningful to consider the joint optimization (as defined in Definition 5) rather than considering the two problems separately. That is why "random" and "prune" allocation always result in poor performance.*

**Response:** Thanks for your careful reading. Inspired by your suggestion, we have formulated the allocation and scheduling problems as a joint optimization problem in Section 3.1. The task allocation and scheduling of *sampling interval* are indeed to be considered as a whole in the joint optimization problem.

We have reorganized Section 5 to present our solution: *CATS* in a natural way. *RANDOM* neglects the data sharing between two overlapping tasks, and allocates tasks randomly. That is the reason its performance is always worst. *PRUNE* allocates tasks by pruning the unreasonable choices when considering the data sharing between tasks. Since each pruning step in *PRUNE* is not the optimal, its performance is not as good as *CATS*.

1. **Comments:** *The scheduling algorithm is simply a greedy one, which lacks of novelty. Its performance is also straightforward.*

**Response:** Thanks for your careful reading. We have reorganized Section 3.3 and Section 4 and presented a crucial ingredient of our solution: *COMBINE* in the new version, instead of the scheduling algorithm in the first submission. We do not present the method of scheduling *sampling interval* well in the previous version. Since the problem of scheduling *sampling interval* is NP-complete, there is no a polynomial-time algorithm for the scheduling problem. Moreover, the greedy scheduling algorithm is designed for a resource-constrained sensor node. Our method of scheduling *sampling interval* outperforms the state-of-the-art method in Ref. [11].

1. **Comments:** *The literature review is far from sufficient. Only one work ([9]) is mentioned a little bit in the paper. The basic ideas of all important related works should be carefully discussed to provide the background of the problem under investigation. It is better to have a separate section for related works.*

**Response:** Thanks for your careful reading. We have added Section 2 in order to outline the related work in the new version. The related work has been investigated carefully and organized as a separate section which includes three domains in Section 2.1, Section 2.2 and Section 2.3.

Our work shares the same background with the related work in Ref. [10] and Ref. [11] (Ref. [11] in the new version is the Ref. [9] in the first submission.).

In Ref. [10], Tavakoli et al. presented an approach for task scheduling on a sensor node to minimize network communication overhead. A task in Ref. [10] is finished by sampling once during its time window. Since a task in Ref. [10] is a special case of our task model, the solution in Ref. [10] cannot be applied to our problem. In Ref. [11], Fang et al. propose an effective sampling approach for interval sampling tasks on a single sensor node. The 2-factor approximation algorithm in Ref. [11] is the state-of-the-art method to maximize data sharing amongst tasks on a single node. However, there are two weak points in Ref. [11]. First, the proposed scheduling method schedules sampling tasks in the descending order of the end time of a task, which, as a result, neglects data sharing between overlapping tasks. Moreover, Fang et al. assume that all tasks have a same length of *sampling interval*, which is too ideal and not practical. By contrast, we observe that multiple tasks may be overlapping, and thereby data sharing exists. Our solution exploits this information by designing a crucial operation, namely, *COMBINE*. *COMBINE* achieves better performance on maximizing data sharing than the scheduling method in Ref. [11].

Second, the solution in Ref. [11] only focuses on task scheduling on a single sensor node, and does not consider the process of task allocation in a wireless sensor network. Unfortunately, the performance of algorithms of task scheduling is sensitive to the strategy of task allocation. Only does optimizing the process of the former not achieve the final optimization for a deployed system. Our solution is more general and practical because of jointly optimizing the process of task allocation and that of scheduling *sampling interval*.

1. **Comments:** *In theorem 1, the problem "MAA-SAA" is not defined. It should be MIN-SAA. In the proof, there are also typos "MAA-SAA" and "MAA-SA". Please check throughout the paper to ensure correcting all such errors.*

**Response:** Thanks for your careful reading. We have revised Section 3 and the error abbreviations such as “MIN-SA” and “MIN-SSA” in the previous version have been removed from the new version. Typos and error abbreviations in the new version have been checked carefully.

1. **Comments:** *In page 4, line 34 (left column), check the sentence "As illustrated in Fig."*

**Response:** Thanks for your careful reading. The error *"As illustrated in Fig."* in the first submission has been revised in the new version. The paper has been checked and polished carefully.

1. **Comments:** *There are citation errors (i.e., some citations are shown as question marks) throughout the paper, such as page 9, line 24 (right column) and page 12, line 17 (left column). Please check and correct carefully.*

**Response:** Thanks for your careful reading. We have checked all the citations in the new version. All the references have been added.

1. **Comments:** *The presentation and organization of the entire paper can be improved to be in a more lucid manner.*

**Response:** Thanks for your careful reading. We have reorganized and polished the paper carefully in the new version.

We have reorganized Section 1. The writing of the motivation of our work has been polished. We have highlighted the vital importance of minimizing the volume of sampled data by using data sharing. Besides, the problem of scheduling *sampling interval* and the problem of task allocation have been described in a more clear and understandable way.

We added Section 2 in order to outline the related work. The related work has been investigated carefully and organized as three domains. Especially, we have highlighted the difference between our work and Ref. [10] and Ref. [11].

We have reorganized Section 3. To be easy to understand our work, we have first presented the task and network model, and then formulated our problem. Finally, we have analyzed the complexity of our problem theoretically.

We have polished Section 4 and Section 5. In Section 4, we have demonstrated a method: *COMBINE* to compute the volume of sampled data. In Section 5, we have proposed a solution: *CATS* to allocate tasks and to schedule the *sampling interval* of tasks in WSNs.

We have added Section 7 to discuss our solution in a physical network and the future work of our work.