

# Energy-efficient Task Allocation for Data Fusion in Wireless Sensor Networks

Li Zhigang / Li Shining  
School of Computer  
Northwestern Polytechnical University  
Xi'an, China  
lizhigang@nwpu.edu.cn, dtlsn@yahoo.com.cn

Zhou Xingshe / Yang Zhiyi  
School of Computer  
Northwestern Polytechnical University  
Xi'an, China  
zhouxs@nwpu.edu.cn, yangzy@nwpu.edu.cn

**Abstract**—Data fusion or In-network processing methods were often adopted in Wireless Sensor Networks (WSNs) to reduce data communication and prolong network lifetime, which made WSNs application can be described as a set of tasks (sensing, processing) and dependencies among them. Task assignment has become an important problem which needed to be resolved for different tasks assignment caused different energy consumption. Based on the task graph of WSNs, an energy-efficient task assignment framework was proposed. As application task can be decomposed into sensing subtasks and processing subtasks, we present the task assignment as sensing subtask assignment and processing subtask assignment. We formulate the processing subtask assignment as a 0-1 quadratic programming problem and evaluate it through experiments.

**Keywords**—wireless sensor networks, task graph, task assignment, 0-1 quadratic programming

## I. INTRODUCTION

The rapid advances in MEMS and wireless communication technologies have enabled the integration of sensing, actuation, processing and wireless communication capabilities into tiny sensor devices. These sensors can then be deployed in large numbers to self-organize into networks that serve a wide range purposes, including environmental monitoring, infrastructure management, industrial sensing, medical, and military [1].

Some of sensor network applications often require sensor networks must work for several months, even a few of years. But in wireless sensor networks, energy constraints are paramount since nodes can often not be recharged, which makes energy consumption become the most important factor for determining the lifetime of the network. We must fully tap the system potential usage of energy, and design system in an energy efficient way to prolong the system lifetime.

When sensor nodes working, wireless communications consume the majority of energy. Early experimental results showed that the energy cost of transmitting 1Kb a distance of 100 meters is equivalent to energy for a general-purpose processor executes 3 million instructions[2]. In order to reduce communication traffic, data fusion and In-network processing was desired in sensor networks[3][4]. The raw data collected from sensor nodes was fused locally to reduce redundancy, and only the processed data was submitted to users.

After applying data fusion in sensor networks, application can be described as a set of processing tasks and the data dependences among these different tasks[5]. Different task allocation schemes (which node performs which processing task) lead to different communication traffics, and then cause different energy consumption when performing application. In order to reduce energy consumption and prolong the lifetime of sensor networks, we should research energy efficient task allocation schemes in sensor networks.

The rest of the paper is organized as follows. In Section II, we provide an overview of task allocation problems and the related work in WSNs. Section III formulates the task allocation in WSNs and section IX presents a centralized energy-efficient task allocation algorithm for WSNs. In Section X, the proposed algorithm is evaluated. Our conclusions are given in Section XI.

## II. RELATED WORK

Task allocation is an important research problem in parallel and distributed computing system. In the distributed system, a job can be seen as a task set, and task allocation is to assign tasks of a program among the processors of a distributed system in order to reduce turnaround time and increase the throughput of the system under the constraint of time and task priorities[6]. The optimal task allocation is to maximize and balance the utilization of the resources while minimizing the communication between processors. From the perspective of resource allocation, task allocation in distributed systems is the assignment of computing resources.

In multi-sensor systems, there also exists task allocation problem[7]. In these systems, the allocation of perceptual tasks (system-level tasks, such as target tracking or classification) is a process of assigning different sensing tasks among sensors in a dynamic, uncertain environment to improve the performance of perception. From the perspective of resource allocation, perceptual tasks allocation is the assignment of sensing resources. Therefore, it was known as sensor selection or sensor action planning.

Due to the characteristics of sensor networks, there exist not only the sensing resources in multi-sensor systems, but also the computing resources in distributed systems. In sensor networks, user's request (such as querying the temperature of a

region, or detecting enemy tanks in an area and tracking them) can be executed like the task allocation in multi-sensor systems, which means only the sensing resources are assigned to task. Sensors are selected according to the task requirements, selected sensors monitor environment, collect data, and transmit a large number of raw data to the sink node. For wireless communication is the main consumer of energy, this method leads to a large amount of data transmissions and causes excessive energy consumption. In order to perform the user's task in an energy-efficient way, the computing capabilities in sensor nodes should be used to process and fuse the raw collected data locally in the network. Finally only the fused data is delivered to users. Through the trade-off of communication and computing, user's tasks are successfully performed. At the same time, network traffic is greatly reduced and much energy is saved.

So we can use task allocation methods to design wireless sensor networks in an energy-efficient way. For WSNs having sensing resources and computing resources, we can decompose user's task into sensing subtasks and computing subtasks, and allocate sensing subtasks and computing subtasks in turn. Sensing task allocation involves sensor selection in WSNs and some works have been done for this problem. Wang proposed an entropy-based sensor selection heuristic for localization application[8]. Given a prior probability distribution of the target location, the locations and the sensing models of a set of candidate sensors for selection, the algorithm greedily selects an informative sensor such that the prior target location distribution would yield on average the greatest or nearly the greatest reduction in the entropy of the target location distribution. In a ground surveillance system for target localization and environmental monitoring, Molnar argues sensor is selected only if the quality of its raw data promises a significant improvement to the localization results[9]. He proposed a self-organized control system that allows the sensors to select the algorithm complexity which balances the requirements for good localization performance and energy conservation. The sensors make their selection autonomously based on their own sensor data, information that they receive from other sensors in the region, and the amount of energy they have left.

In this paper, we consider the problem of how to assign the computing tasks after the selection of sensors to make the application performed using minimum energy.

### III. TASK ALLOCATION IN WSNs

#### A. Network Model

We assume that a set of wireless sensor nodes  $S = \{N_1, N_2, \dots, N_n\}$  are deployed randomly in a two-dimensional area. After deployment, nodes no longer change their locations. For simplicity, we assume that these nodes have the same maximum transmission range and we normalize it to one unit. Consequently, all wireless nodes in  $S$  together construct a unit disk graph  $UDG(S)$ , which has an edge  $(N_i, N_j)$  if and only if the Euclidian distance between  $N_i$  and  $N_j$  is less than one unit.

Special node  $N_n$  is defined as sink node. Due to the characteristics of WSNs, each sensor node has certain attributes, such as node location, sensing mode, the minimum hop count to sink, the current available energy and so on. So we define the capacity vector  $p_i$ , associated with the node  $N_i$ , which describes the node's attributes.

Then the sensor network can be described as a weighted graph  $SN = \{S, E, P\}$ , in which  $S = \{N_1, N_2, \dots, N_n\}$  is the nodes set,  $E \subseteq S \times S$  is the connection sets (the edge set of unit disk graph), and  $P = \{P_1, P_2, \dots, P_n\}$  is the node's attribute set.

#### B. Task Model

The expression power of the task description would affect the complexity of task assignment algorithm[5]. There are two issues in describing tasks. One is about description of functionality of the task. The other is to specify the distribution (or locations) of sensors which will participate in the task.

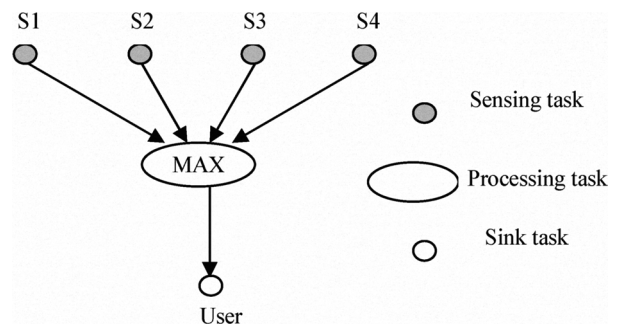


Figure 1. task representation using DAG

Reference to the task graph in distributed computing, we can use DAG to describe the task in sensor networks[6]. Nodes in DAG represent the subtasks needed to be carried out by sensors and edges denote the data transfer between tasks. So we can define task in WSNs as  $ST = \{T, C\}$ , in which  $T = \{T_1, T_2, \dots, T_m\}$  is the subtask set and  $C \subseteq T \times T$  denotes the priority relations between subtasks. Each edge  $C_i = (T_u, T_v)$  has weight  $w_i$ , representing the amount of data transferred from subtask  $T_u$  to  $T_v$ . In particular, the nodes in  $ST$  having no input edges represent the source subtasks, which are also called sensing subtasks (monitoring the environment and collecting the environmental data). The node in  $ST$  having no output edge represents the sink subtask. In the DAG task graph, the sink subtask in an empty task which does not perform any operations and only indicates the convergence of information. The sink subtask has only one input, which represents the information application needed. The input to the sink subtask was produced by the final computing subtask. For example, in an environmental temperature monitoring application, assuming the given task is reading the maximal

value from four sensors. The corresponding task graph is shown in Figure 1.

From the Figure, we can see that DAG task graph is actually a data flow graph (DFG), which represents the data flow relationships between subtasks. Although DAG task graph can describe the data flow relationships between subtasks, the distributed properties of the nodes performing the subtask cannot be expressed. We should append additional constraints to the nodes of the DAG, so each node  $T_i$  in DAG has an associated attributes  $A_i$ , which represents the executing location of subtask, the needed sensing data and computing power. So we can use weighted graph to describe the task in WSNs, then  $ST$  is transformed  $ST = \{T, C, A\}$ .

### C. Task Allocation

For the task  $ST = \{T, C, A\}$ , we can use an attribute vector to describe the attribute  $A_i$  of subtask  $T_i$ . For example, the source subtask often has the attributes of the data type, sample rate, coverage area and so on. The main attribute of the computing subtask is the computing power.

For the WSN  $SN = \{S, E, P\}$ , a capability vector was also can be defined to describe the attribute  $P_i$  of sensor  $N_i$ . The common sensor attributes is the location of the sensor, sensor mode, the hop distance to the sink, and the currently available energy.

For simplicity, we assume the task attribute vector is the same type of sensor capability vector, that is:

$$T.Attr = (T.Att1, T.Att2, \dots, T.Attn)$$

$$S.Cap = (S.Att1, S.Att2, \dots, S.Attn)$$

Given the task attribute T.Atti and sensor capability S.Atti, if S.Atti can meet the requirements of T.Atti, we call "S.Atti covers T.Atti", which denoted as  $S.Atti \geq T.Atti$ . If every element in capability vector  $P_i$  of sensor  $N_i$  covers the corresponding element of attributed vector  $A_j$  of task  $T_j$ , we call "sensor  $N_i$  covers task  $T_j$ ", which denoted as  $P_i \geq A_j$ .

**Definition 1:** Given task  $ST = \{T, C, A\}$  and WSN  $SN = \{S, E, P\}$ , task allocation TA is the mapping from nodes in ST to nodes in SN. That is  $TA: T \rightarrow S$ ,  $T$  is the subtask set of  $ST$ , and  $S$  is the sensor set of  $SN$ .

**Definition 2:** Given task  $ST = \{T, C, A\}$  and WSN  $SN = \{S, E, P\}$ , a task allocation is termed feasible if:

1) Source subtask in ST is only mapped to one node in SN. That is, for source subtasks  $x$  and  $y$ , if  $x \neq y$ , then  $fa(x) \neq fa(y)$ .

2) The sink subtask in ST is mapped to the sink node in SN. That is, for sink subtask  $x$  in ST,  $fa(x) = N_n$

3) If  $N_i = fa(T_j)$ , thus  $P_i \geq A_j$ , which means the attribute  $P_i$  of the mapped node in ST meets the requirements of attribute  $A_j$  of the mapping subtask.

**Definition 3:** For a feasible task allocation  $fta$ , we define its utility:

$$U(fta) = \partial G(fta) - (1 - \partial)E(fta)$$

$G(fta)$  represents the information gain produced by execution of task, and  $E(fta)$  denotes the energy consumption for performing the task.  $\partial$  is a relative weight. In this paper, we make  $\partial = 0$ , which means only the energy consumption is considered in this paper.

**Definition 4:** The optimal task allocation  $ofta$  is one of the feasible task allocation achieved maximum utility. That is:

$$U(ofta) = \text{Max}U(fta), \forall fta$$

For  $\partial = 0$ , maximizing the utility of task allocation is equivalent to minimizing energy consumption for performing the task, which means the goal of optimization of task allocation is to  $\text{Min}E(fta)$ .

In this paper, we assume the attribute vector of subtask and capability vector of nodes include only two elements: area and energy. Area attribute of sensor represents the sensing coverage area, and sensor energy attribute is the available energy of it, denoted as  $P_i.av$ . While for sensing subtask, area attribute represents the needed monitoring area, and energy attribute is the energy consumption for collecting environmental data, which denoted as  $A_i.re$ . For computing subtask, the area attribute is 0, which means the computing subtask has no location constraint. The energy attribute of computing subtask represents the energy consumption needed to perform the subtask.

## IV. ENERGY-EFFICIENT TASK ALLOCATION

For a feasible task allocation  $f$ , the total energy consumption of task execution after allocation is the sum of energy consumption of all the nodes in the networks, including energy consumption of sensing nodes, computing nodes and forwarding nodes. For simplicity, we assume the transmitting power is equal to receiving power because nodes use short range communication. Due to the route selection, forwarding nodes are dynamic and cannot be determined in advance. So we transform the total energy consumption of task execution into the sum of energy consumption of performing sensing subtasks, processing subtasks and communication.

Given sensing subtasks  $T_1, T_2, \dots, T_a$ , the energy consumption of sensing subtasks is:

$$E_{sens} = \sum_{i=1}^a T_i.re \quad (1)$$

Processing subtasks meet the principle of additivity. Given processing subtasks  $T_{a+1}, T_{a+2}, \dots, T_{m-1}$ , if  $T_i, \dots, T_j$  are assigned to the same node, the energy consumption of the node executing  $T_i, \dots, T_j$  is  $T_i.re + \dots + T_j.re$ . In addition, if the node also executes a sensing subtask, the execution of processing subtask cannot affect the energy consumption of sensing subtask. Therefore, the energy consumption of execution processing subtasks is:

$$E_{proc} = \sum_{i=a+1}^{m-1} T_i.re \quad (2)$$

For  $C_i = (u, v) \in C$ , if subtasks  $u$  and  $v$  are executed by nodes, the energy consumption of communication in the task execution is  $w_i \times e_{path}\{f(u), f(v)\}$ .  $w_i$  is the communication traffic from task  $u$  to task  $v$ .  $f(u)$  and  $f(v)$  represent the corresponding sensors carrying out subtask  $u$  and  $v$  after task allocation.  $e_{path}\{f(u), f(v)\}$  is the energy consumption for transmitting a bit of data from  $f(u)$  to  $f(v)$ . We assume sensor network use shortest path routing algorithm, so

$$e_{path}\{f(u), f(v)\} = hopCount(f(u), f(v)) \times (e_r + e_t) \\ = 2e' \times hopCount(f(u), f(v))$$

$hopCount(f(u), f(v))$  denotes the minimum hop count between  $f(u)$  and  $f(v)$ . If  $f(u) = f(v)$  (subtasks  $u$  and  $v$  are assigned to the same node), then  $hopCount(f(u), f(v)) = 0$ .  $e_r$  represents energy consumption for node receiving a bit of data, and  $e_t$  represents energy consumption for node transmitting a bit of data. Because we assume the transmitting power of nodes is equivalent to receiving power, then  $e_r + e_t = 2e'$ .

The energy consumption of communication for exchanging information between subtasks is:

$$E_{comm} = \sum_{C_i=(u,v) \in C} 2e'w_i \times hopCount(f(u), f(v)) \quad (3)$$

Then, the total energy consumption for executing task is:

$$E_{total} = E_{sens} + E_{proc} + E_{comm} \quad (4)$$

As in (1), for different feasible task allocations, energy consumption of sensing subtasks and processing subtasks do not change. The only changing thing is the energy consumption of communication. Different task allocations lead to different nodes performing the subtasks, making the energy consumption of communication different.

We can formulate the task allocation as a 0-1 quadratic programming problem. The notations used in the formulation are listed as follows:

$\{x_{ij}\}$ : a set of 0-1 variables. If task  $T_i$  is assigned to node  $N_j$ , then  $x_{ij} = 1$

$e'$ : the energy consumption of node transmitting or receiving a bit of data

$T_i.re$ : the energy consumption for executing subtask  $T_i$

$N_i.av$ : the current available energy of node  $N_i$

$w_i$ : the weight of edge  $E_i = (u, v)$  in DAG task graph, represents the data traffic from subtask  $u$  to  $v$

$d_{ij}$ : the distance of shortest path between node  $N_i$  and  $N_j$ , represented by the minimum hop count between nodes  $N_i$  and  $N_j$

The distance  $d_{ij}$  of shortest path between two nodes can be derived by Floyd-Warshall algorithm[10].

Objective function:

$$Min \sum_{T_i \in T} \{T_i.re\} + \sum_{C_i=(u,v) \in C} \{ \sum_{kl} w_i x_{uk} x_{vl} d_{kl} \} \\ = Min \sum_{C_i=(u,v) \in C} \{ \sum_{kl} w_i x_{uk} x_{vl} d_{kl} \}$$

Constraint set:

$$\forall T_i \in T, \sum_j x_{ij} = 1$$

$$\forall N_j \in S,$$

$$\sum_{T_i \in T} (x_{ij} T_i.re) + \sum_{C_i=(u,v) \in C} \{ e' w_i | x_{uj} - x_{vj} | \} \leq N_j.av$$

$$x_{11} = 1, x_{22} = 1, \dots, x_{aa} = 1, x_{mn} = 1$$

This is a typical 0-1 quadratic programming problem with linear constraints. We can convert it to a linear programming problem using linear technology and solve it [11].

The objective of the above model is to minimize the energy consumption  $E_{total}$  for executing the task. In sensor networks, we should also consider the balance of energy consumption in addition to energy efficiency, to avoid nodes died for excessive energy consumption. So we can set the weight of edge in  $SN$  as:

$$w(N_i, N_j) = Max\left(\frac{1}{N_i.av}, \frac{1}{N_j.av}\right)$$

After the distance of shortest path  $d'_{ij}$  between nodes  $N_i$  and  $N_j$  has been derived using Floyd-Warshall algorithm, the objective function is converted to:

$$\text{Min} \sum_{C_i=(u,v) \in C} \{ \sum_{kl} w_i x_{uk} x_{vl} d'_{kl} \}$$

## V. EVALUATION

According to the assumptions made in this paper, we design a simulation program using Matlab and evaluate the proposed centralized 0-1 quadratic programming algorithm. At present, we chose the simulating parameters for only considering the energy consumption of the application. From the analysis of section IV, the energy consumption is only affected by the different communication traffic caused by different task allocation. Therefore, we use the communication traffic to represent the energy consumption of application execution after task allocation.

The experiment uses the task graph shown in Figure 1. The network simulation scenario is a 100m x 100m square area, where randomly deployed 40 sensors. The communication range of the sensor is set 30m, and sink node is located at the center of the square region. We firstly assume the nodes of ID=1,2,3 as the sensing nodes after sensor selection. Node of No.1 executes subtask T1, Node of No. 2 executes subtask T2, and Node of No. 3 executes subtask T3.

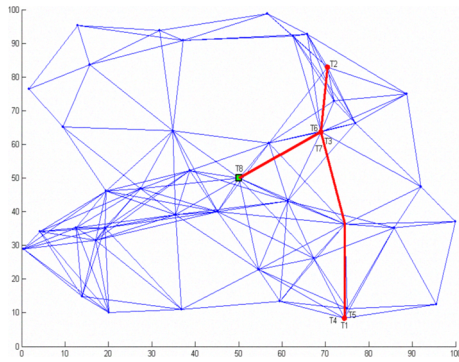


Figure 2. result of task assignment

By solving the 0-1 quadratic programming problem, the optimal result of task allocation is shown in Figure 2. From the figure, we can see that the subtasks T4 and T5 are assigned to the node of No. 1, subtasks T6 and T6 are assigned to node of No.3 and T8 are allocated to sink node. After the task allocation, the overall communication traffic of application execution is 140.

Reselect the location of sensing nodes and make it more dispersed. By solving the problem, the optimal result of task allocation is shown in Figure 3. We can see from it, subtasks T4 and T5 are assigned to node of No.1, subtasks T6 and T7 are assigned to sink node. After task allocation, execution of the task produces communication traffic of 140.

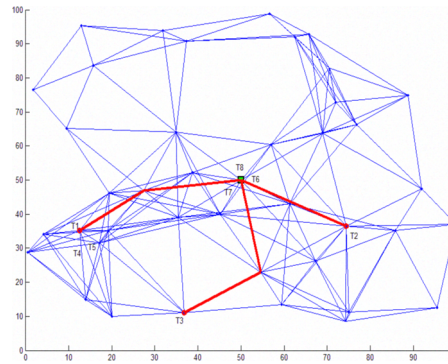


Figure 3. results of task assignment with different location of sensing node

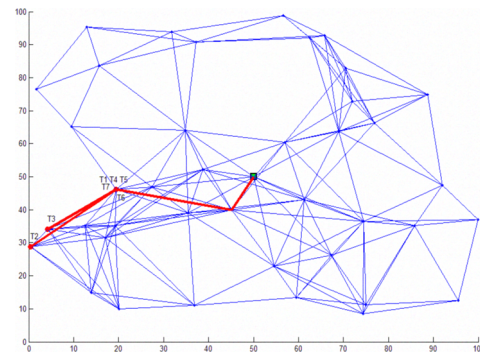


Figure 4. results of task assignment with centralized location of sensing node

Centralizing the location of sensing nodes and solving the problem, the result of optimal allocation is shown in Figure 4. In the figure, subtasks of T4, T5, T6 and T7 are all assigned to the node of No.1, and the overall communication traffic is 90.

## VI. CONCLUSION

For the energy limitation in WSNs, it is important to do data fusion in the network to reduce communication traffic. Data fusion introduces computing into WSNs, and different nodes performing computing cause different communication traffic and different energy consumption. In order to prolong the system lifetime of WSNs, we studied the optimal task allocation method. In this paper, we describe application of WSNs using task graph and propose a task allocation model of WSNs. For the computing subtask allocation, we formulate the problem as 0-1 quadratic programming problem and evaluate the online centralized algorithm. In the future, we will propose the corresponding distributed algorithm for task allocation.

## ACKNOWLEDGMENT

This paper was supported by the National Natural Science Foundation of China under Grant No. 60573161, national scientific and technological support project "Key technologies

of precision agriculture for western predominant crops”, and Natural Science Foundation of Shaanxi Province under Grant No. SJ08-2T06.

#### REFERENCES

- [1] I.F.Akyildiz et al., "Wireless Sensor Networks: a survey," Computer Networks, vol. 38, no. 4, pp. 393-422, 2002.
- [2] G.J. Pottie, W.J. Kaiser, "Wireless Integrated Network Sensors", Communications of the ACM, Vol. 43, No 5, pp. 51-58, May 2000.
- [3] D. Estrin and R. Govindan, "Next century challenges: Scalable coordination in sensor networks", in Proc. Mobicom, pp. 263-270, 1999.
- [4] K. Sohrabi, J. Cao, V. Ailawadhi, G. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network", IEEE Personal Communications Magazine, Vol. 7, No. 5, pp. 16-27, Oct. 2000.
- [5] Heemin Park, Mani B. Srivastava, "Energy-Efficient Task Assignment Framework for Wireless Sensor Networks", CENS Technical Reports, September 2003.
- [6] Jie Wu, "Distributed System Design", CRC Press, 1999.
- [7] N. Xiong, P. Svensson, "Multi-sensor management for information fusion: issues and approaches," Information Fusion 3(2), pp. 163-186, 2002.
- [8] Hanbiao Wang, Pottie G., Kung Yao, Estrin D., "Entropy-based sensor selection heuristic for target localization", The Third International Symposium on Information Processing in Sensor Networks, pp. 36-45, April 2004.
- [9] P. Molnar, E.J. Lockett and L.M. Kaplan, "Self-organized task assignment for distributed sensors", SPIE's Conference on Sensor Fusion and Decentralized Control in Robotic Systems, pp. 189-196, Nov. 2003.
- [10] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", The MIT Press (Second Edition), September 2001.
- [11] Sourour Elloumi, Alain Faye, Eric Soutif, "Decomposition and Linearization for 0-1 Quadratic Programming", Annals of Operations Research, Volume 99, Numbers 1-4, pp. 79-93, 2000.