

2010

Energy-efficient task assignment of wireless sensor network with the application to agriculture

Songyan Xu
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Xu, Songyan, "Energy-efficient task assignment of wireless sensor network with the application to agriculture" (2010). *Graduate Theses and Dissertations*. 11763.

<https://lib.dr.iastate.edu/etd/11763>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Energy-efficient task assignment of wireless sensor network with the application
to agriculture**

by

Songyan Xu

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Wensheng Zhang, Major Professor
Ratnesh Kumar
Ying Cai

Iowa State University

Ames, Iowa

2010

Copyright © Songyan Xu, 2010. All rights reserved.

DEDICATION

I would like to dedicate this dissertation to my parents and my sister without whose encouragement, love, understanding and support I would not have been able to keep working on this research and finish it.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
ABSTRACT	vii
CHAPTER 1. Introduction	1
1.1 Overview	1
1.2 Related Work	2
1.3 Our Contribution	3
1.4 Organization of Thesis	6
CHAPTER 2. Notions and Preliminaries	7
2.1 Modeling of Sensors	7
2.2 Modeling of Tasks	8
CHAPTER 3. Formulation of Task Assignment Problem	10
3.1 Problem Statement	10
3.2 Formulation of Task Assignment Problem	11
3.3 Size of Proposed Formulation	18
CHAPTER 4. Realization of Task Assignment System	20
4.1 System Design	20
4.1.1 Assignment Computation Module	21
4.1.2 Assignment Illustration Module	22
4.1.3 Assignment Execution Module	23
4.2 Implementation Issues	24

4.2.1	Development Language	24
4.2.2	ILP Solver	25
4.2.3	Assumptions	25
4.3	Task Assignment System GUI	27
CHAPTER 5. Simulation and Evaluation		30
5.1	Simulation Environment and Parameters	30
5.2	Simulation Results	31
5.2.1	Simulation On A Simple Wireless Sensor Network	32
5.2.2	Simulation On Multiple Wireless Sensor Networks	36
CHAPTER 6. Conclusion And Future Work		42
6.1	Summary Of Dissertation	42
6.2	Future Work	43
ACKNOWLEDGEMENTS		46
BIBLIOGRAPHY		47

LIST OF TABLES

Table 4.1	Size of task assignment system	25
Table 5.1	Simulation parameters	31
Table 5.2	Configuration of sensors	32
Table 5.3	Configuration of tasks	33
Table 5.4	Optimal task assignment	35
Table 5.5	Random task assignment	36
Table 5.6	User requests	37

LIST OF FIGURES

Figure 3.1	Illustration of sampling times and data combinations	15
Figure 4.1	Architecture of task assignment system	21
Figure 4.2	Interface of assignment computation module-configuration	27
Figure 4.3	Interface of assignment computation module-assignment	28
Figure 4.4	Interface of assignment illustration module	28
Figure 5.1	Structure of command/data packet	32
Figure 5.2	Configurations of Sensors and User Requests	34
Figure 5.3	Number of variables vs nodes	38
Figure 5.4	Number of constraints vs nodes	38
Figure 5.5	Running time vs nodes	39
Figure 5.6	Energy consumption vs nodes	40

ABSTRACT

Wireless sensor networks have attracted considerable attention from academia as well as industry. The applications of wireless sensor networks encompass the domains of industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, traffic control, etc. In this research we focus on the application of wireless sensor networks to agriculture in which sensors are distributed in a field to monitor the environment and soil of certain interested areas in the field. Given a set of measurement requests and tasks, it is critical to develop a formal, automatic and energy-efficient approach to assign the set of measurement tasks among the given wireless sensor network to fulfill the measurement requests subject to the restrictions such as sensor locations, sensing abilities and the expected number of samplings. In this work, we model the measurement requests and tasks as tuples and formulate the task assignment problem of wireless sensor networks with the application to agriculture as an instance of Integer Linear Programming (ILP) problem. We also develop a task assignment system using Java, SAT4J and TinyOS to implement the proposed *formal* and *automatic* task assignment approach. The proposed ILP formulation and developed task assignment system are applied to the simulations on small and middle-sized wireless sensor networks. The simulation results show that the proposed ILP formulation is *correct* and it is *feasible* to apply the proposed ILP formulation to resolve task assignment problems for small and middle-sized (≤ 100 sensors) wireless sensor networks with a small number of measurement requests (≤ 5 requests).

CHAPTER 1. Introduction

1.1 Overview

Wireless sensor networks (WSNs) consist of a large number of low-power, low-cost sensing devices, namely sensors, with local computation, processing, and wireless communication capabilities, in which the distributed sensors work cooperatively to achieve certain tasks. With the evolution of technologies in sensing, computing, Micro-electro-mechanical systems (MEMS) and wireless communications, deploying large scale, low-power, and cost-effective wireless sensor networks has been a pragmatic vision. Compared to the conventional sensing methods, dense deployment of sensors not only extends spatial sensing coverage but also improves fault-tolerance and robustness of the system [26, 11]. Due to these features wireless sensor networks have gained remarkable attention from academia as well as industry and have been widely used in the applications for commercial as well as military purposes. The commercial applications of wireless sensor networks range from environmental applications such as forest fire detection, flood detection, bio-complexity mapping, health applications such as personal health monitoring, home applications such as building automation, smart environment, and other applications such as vehicle tracking and detection, inventory tracking and so on [11]. The military applications of wireless sensor networks involve intrusion detection, perimeter monitoring, information gathering, smart logistics support in unknown areas, etc [18].

In recent years, wireless sensor networks have been applied to *precision agriculture*. Precision agriculture takes advantages of information and control technologies to provide the means of observing, assessing and controlling a wide range of aspects of agricultural practices such as daily herd management, horticulture, and pre- and post-field crop production [13, 14]. There have existed a lot of researches on the applications of wireless sensor networks to pre-

recision agriculture in the literature. See for example, “smart farm” developed by Australian’s Commonwealth Scientific and Industrial Research Organization in [25], test bed for remote monitoring and controlling for agriculture in [12], design of MAC and Network layers in [22], measurement of crop water-content using RF signals in [16], etc. A facet of precision agriculture focuses on site-specific management: monitoring soils, crop, climate in a field so as to provide real-time operations for agricultural production such as fertilizing, pesticide control, tillage and sowing. To support environment and soil monitoring required by precision agriculture, sensors of different types need be deployed in the fields. And the monitoring tasks have to respect the restrictions such as sensing abilities and locations of sensors, and the expected number of samplings on the given sensor network resulting from the monitoring requests. On the other hand, since sensors only carry very limited and possibly irreplaceable power sources, energy conservation is a very important issue to take care of. Therefore an automatic and formal approach to assign the monitoring tasks among a given sensor network with a major concern of energy consumption efficiency is highly desired.

1.2 Related Work

Extensive researches on task assignment/allocation for wireless sensor network have been reported in the literature. The authors of [27] proposed an Integer Linear Programming (ILP) formulation for energy-balanced task allocation onto a single-hop cluster of homogenous sensor nodes. In this initial work [27], tasks are modeled using Directed Acyclic Graph (DAG) and allocations include assigning tasks onto sensors, deciding voltage settings of tasks, assigning communication activities onto channels and scheduling computation and communication activities. The goal is to find an allocation to maximize the lifetime of the cluster. A three-phase heuristic of polynomial time was proposed to help achieving the optimal solution effectively. Energy balance is an importance concern for the applications of wireless sensor networks. However energy efficiency should be considered as well since sensors are equipped with limited resources and it is costly to replace their batteries. Otherwise a sensor network may consist of different types of sensors in order to support monitoring of physical quantities of different types.

Therefore the difference of sensing capabilities among the sensor nodes should be taken into account. Mapping of the data-driven tasks onto sensors can be found in [20], in which a mixed Integer Programming (MIP) formulation was proposed for obtaining an optimal allocation of data-driven sensing, processing and actuation tasks that minimizes the total energy with the concern of energy balance. Due to the complexity of the problem, the formulation is nonlinear, leading to a mixed integer programming. A greedy heuristic is provided to solve the proposed MIP problem. Similar problem with the application of wireless sensor networks for healthcare systems was studied in [10]. The authors of [10] focused on *adaptive* runtime task assignment problem and aimed at improving the battery lifetime of the overall network subject to task dependency and deadline. The proposed Dynamic Task Assignment for Wireless Healthcare System (DynAHeal) quickly adapts dynamic changes in workload. Jointly mapping and scheduling which incorporates channel modeling, task mapping and sensor failure handling in single-hop cluster was investigated in [24, 23]. Topology-aware energy efficient task assignment for multi-hop sensor networks has been addressed in [28], in which an ant-based meta-heuristic algorithm was developed to solve the optimization issue of task assignment. Simulated annealing method was applied to search optimal task transformation and assignment so as to minimize total energy consumption and latency in [19]. An auction theory based approach in which sensing missions are modeled as noncooperative games and sensors are considered as intelligent agents was reported in [15]. Recently a matrix based centralized discrete event supervisor has been used for coordination of sensors for task assignment and resource allocation in mobile wireless sensor networks [17].

1.3 Our Contribution

In this work we focus on the task assignment problem of assigning independent measurement/monitoring tasks among wireless sensor networks. In some applications of wireless sensor networks to agriculture, we expect to monitor environment such as temperature and humidity at certain areas. Unlike the tasks such as “the average temperature of 20 uniformly distributed sensors among 100 sensors”, the tasks we consider are independent, and can be performed and

reported to the server without the cooperation with the other sensors. In such scenarios it is not necessary to consider the dependency/precedence orders among the tasks. And so a simplified modeling of tasks, which is specific for such applications, is needed. In this dissertation, we propose a modeling of tasks to capture the functionalities of the independent measurement tasks. The modeling of tasks is an important issue since the description of tasks would effect the complexity of the task assignment problem [19]. In most of the prior work, people focused on the cooperative tasks and formalized the tasks using graphs such as directed acyclic graph (DAG)/data flow graph (DFG), in which the description of tasks formalizes the dependency among the tasks, whereas the user requests such as the areas to monitored, the expected number of samplings of the variables are not modeled. These user requests should be taken into consideration in certain applications of wireless sensor networks to satisfy user specific requirements of measurements. Moreover care should also be taken for the differences of sensing abilities among sensors. In our work we model the tasks in form of a tuple to capture such user requirements. The proposed modeling of tasks can be easily extended to include other aspects of user requirements such as measurement latency, measurement precision.

We formulate the task assignment problem of wireless sensor networks as an instance of Integer Linear Programming problem. The objective function of the proposed ILP formulation captures the overall energy consumption of a given sensor network under certain task assignment. Energy consumption is a key concern in the applications of wireless sensor networks [27], which requires a systematic energy-aware methodology for resource management. In this dissertation we focus on the energy consumption of the overall sensor network so as to achieve *energy-efficiency* in the given sensor network. The constraints of the proposed ILP formulation model the restrictions on a task assignment strategy resulting from the user requirements, sensor locations, sensor measurement capabilities, etc. The energy consumption we consider includes the energy cost arising due to measurement as well as communications at each sensor. In order to save the communication energy, i.e., the energy for delivering the measurement results, at each sensor node, we combine as many measurement results as possible into one packet and thus make it possible to deliver as few as possible packets among

the sensor network. The communication energy consumption due to the cooperation of the tasks among the sensors has been taken care of in the previous work such as [27]. However the communication energy consumption due to the measurement tasks assigned at each sensor is not considered and formalized in the aforementioned work. In this dissertation we formally formalize such communication energy consumption using binary variables and enumeration of all the nonempty *combinations* of the given tasks, which we denote as *data combinations* in this dissertation. Energy-balance is another important issue in wireless sensor networks. We also propose an objective function for restricting the maximum energy consumption at each single sensor so as to maintain *energy-balance* among the sensor network. Our proposed ILP formulation can be modified to accommodate different concerns of energy.

We also design and implement a task assignment system to solve the task assignment problem based on the proposed ILP formulation. The tasks are then assigned the given sensor network according to the optimal solution to the ILP-based energy-efficient task assignment problem. (Note ILP problem is NP-complete. Therefore we may not be able to get an optimal solution to the given ILP problem. In such case a task assignment is performed using a satisfying solution to the given ILP problem.) The designed task assignment system consists of three modules: assignment computation, assignment illustration, and assignment execution modules and is developed using Java, nesC, TinyOS, and an open-source ILP solver SAT4J. We apply the proposed ILP formulation and developed task assignment system to small as well as middle-sized sensor networks (consisting of ≤ 100 sensors) with respect to a small amount of user measurement requests (consisting of ≤ 5 requirements). The simulation results show that the proposed ILP formulation is *correct* and it is *feasible* to apply the proposed ILP based task assignment approach to small as well as middle-sized sensor networks. The proposed approach is *formal, automatic* (only requiring the users to provide the configurations of the sensor network and measurement requests), and general enough to be applied to wireless as well as wired networks.

1.4 Organization of Thesis

The organization of the rest of the thesis is as follow. Chapter 2 presents the notions and preliminaries that are used for formalizing the task assignment problem of wireless sensor networks. Chapter 3 presents the proposed Integer Linear Programming formulation of the task assignment problem. Chapter 4 presents the architecture and implementation issues of the task assignment system. Chapter 5 shows the simulation results of the proposed Integer Linear Programming formulation on multiple wireless sensor networks. And Chapter 6 concludes the work.

CHAPTER 2. Notions and Preliminaries

The notions and preliminaries that are used for modeling the energy-efficient task assignment problem for wireless sensor networks are given in this chapter.

2.1 Modeling of Sensors

In the application of wireless sensor networks to agriculture, people are interested in monitoring the climatic condition such as humidity, temperature, wind velocity, precipitations, and the soil condition such as soil salinity, PH value, moisture of a crop field. We use a set $H = \{1, \dots, k\}$ to represent the physical quantities to be monitored among a given wireless sensor network, and $P = \{p_i\}, \forall i \in H$ to denote the sampling period of a physical quantity i .

We model a sensor node using a tuple $s_i := (l_i, Q_i, E_i, c_i)$, where

- l_i denotes the location of sensor i ,
- Q_i denotes the set of physical quantities that can be measured at sensor i ,
- E_i denotes the set of energy consumptions arising due to one single sampling at sensor i ,
- c_i denotes the energy consumption due to a single communication (i.e. a single transmission of data) at sensor i ,

and we use $S = \{s_i\}$ to denote a set of sensors.

In the above model, $Q_i = \{q_{ik}\}$ for $k \in H$, where $q_{ik} = 1$ if sensor i is able to measure the physical quantity k . Otherwise, $q_{ik} = 0$. $E_i = \{e_{ik}\}$ for $k \in H$, where e_{ik} denotes the energy consumed due to measurement of a physical quantity k at sensor i . For any variable which can not be measured at sensor i , i.e., $q_{ik} = 0$, it should hold that $e_{ik} = 0$.

2.2 Modeling of Tasks

Next we present the model of the tasks.

For analysis of certain physical quantity such as soil moisture in a field, usually multiple measurements at several sampling areas are expected. (Note in our work we consider independent measurement tasks. The cooperative tasks such as “the average of temperature” and “the maximum moisture” are not considered here.) To capture such requirements of sensing locations and measurement amounts, we model a (measurement) task as a tuple $t_i := (L_i, D_i, M_i)$, where

- L_i denotes the area to be monitored,
- D_i denotes the set of physical quantities to be measured,
- M_i denotes the set of measurement amounts,

and we use $T = \{t_i\}$ to denote the set of the measurement tasks

In the above model, $D_i = \{d_{ij}\}$, where d_{ij} denotes whether a physical quantity of type j is required to be measured by task i : $d_{ij} = 1$ if a physical quantity j is queried by the users; Otherwise, $d_{ij} = 0$. $M_i = \{m_{ij}\}$, where m_{ij} denotes the expected number of measurements of a physical quantity j by task i . For any variable of type j which is not queried by task i , i.e., $d_{ij} = 0$, it should hold that $m_{ij} = 0$. Moreover, for simplification, we assume that a sampling area is always circular: the center of a sampling area i is denoted by o_i , and the radius of a sampling area i by r_i .

Given a set of measurement tasks T and a set of sensors S , we need determine whether a node is located within certain sampling areas. This can be captured by a set $N = \{n_{ij}\}$, where $n_{ij} = 1$ if and only if a sensor i is inside of an area j , i.e., $|o_i - l_j| \leq r_j$. Here the operation $|\cdot|$ represents the geographical distance between two locations.

Remark 1 In the previous work such as [27, 20, 10, 24, 23], tasks are modeled using a Directed Acyclic Graph (DAG) in which the nodes represent the set of tasks and the edges represent the set of communications (i.e., the output of the source node of the edge need be transmitted

to the destination node of the edge as its input). However the features of a measurement task such as the requirement of sensing locations, measurement amounts of the interested physical quantities have not been defined. Such modeling of a task is suitable for cooperative tasks but not suitable for the independent measurement tasks in certain applications of wireless sensor networks to agricultures since the user requirements on the measurements are missing. In this work we model a task as a tuple to capture the user requirements of sampling. Also we focus on the independent, not corporative, measurement tasks, and so the dependency/precedence of the executions of measurement tasks (input-output ordering) need not be taken into account.

CHAPTER 3. Formulation of Task Assignment Problem

In this chapter, we formulate the energy-efficient task assignment problem of wireless sensor network with the application to agriculture as an instance of Integer Linear Programming (ILP) problem. The objective function of the proposed ILP formulation models the overall energy consumption of the given sensor network under a task assignment. And the constraints of the proposed ILP formulation model the restrictions on a task assignment strategy resulting from the user requirements, sensor locations, sensor measurement capabilities, etc. The energy considered include the energy consumption arising due to measurement as well as communications among the network. We first address the statement of the task assignment problem in below.

3.1 Problem Statement

Energy consumption due to data sampling as well as transmitting is a major concern in the applications of wireless sensor networks. It is crucial to assign the measurement tasks among wireless sensor networks in a smart way so that the assignment could fulfill the user requests subject to the restrictions on the given sensor networks while consuming as few energy as possible. In particular, in the application of wireless sensor networks to agriculture, since sensors are buried underneath the ground, the replacement of sensor batteries becomes more inconvenient, which additionally increases the maintenance cost of a wireless sensor network. Therefore an energy-aware task assignment algorithm for wireless sensor networks with the application to agriculture is highly expected.

The energy-efficient task assignment problem of wireless sensor network with the application to agriculture can be defined as follows: Given a wireless sensor network deployed in a

field and a set of measurement tasks, assign the given set of tasks among the sensor network subject to the constraints arising due to the user requirements and geography of the field so that the energy consumed resulting from samplings and communications is efficient.

The goals of this research are to

1. Formulate the energy-efficient task assignment problem of wireless sensor networks with the application to agriculture, and
2. Propose the corresponding formal and automatic task assignment approach, and
3. Develop and implement the framework of energy-efficient task assignment system.

3.2 Formulation of Task Assignment Problem

We formulate the energy-efficient task assignment problem as an instance of Integer Linear Programming problem as follows:

$$\begin{aligned}
 P : \quad & \min \sum_{i \in S, j \in H} e_{ij} \lfloor U_T/p_j \rfloor x_{ij} + \sum_{i \in S, B_l \subseteq B} c_i b_l y_{il} \\
 & \text{s.t.} \\
 & 1. \quad w_{ikj} \leq q_{ij}, \quad \forall i \in S, k \in T, j \in H \\
 & 2. \quad w_{ikj} \leq l_{ik}, \quad \forall i \in S, k \in T, j \in H \\
 & 3. \quad w_{ikj} \leq d_{kj}, \quad \forall i \in S, k \in T, j \in H \\
 & 4. \quad \sum_{i \in S} w_{ikj} \geq m_{kj}, \quad \forall k \in T, j \in H \\
 & 5. \quad x_{ij} \leq \sum_{k \in T} w_{ikj}, \quad \forall i \in S, j \in H \\
 & 6. \quad x_{ij} \geq w_{ikj}, \quad \forall i \in S, k \in T, j \in H \\
 & 7. \quad y_{il} \leq \sum_{j \in B_l} x_{ij}, \quad \forall i \in S, l \in [1, \dots, 2^{|H|}] \\
 & 8. \quad y_{il} \geq x_{ij}, \quad \forall i \in S, l \in [1, \dots, 2^{|H|}], j \in B_l \\
 & 9. \quad w_{ikj}, x_{ij}, y_{ij} \in \{0, 1\}
 \end{aligned}$$

where $\lfloor \cdot \rfloor$ denotes the operation of “floor”: $\lfloor U_T/p_j \rfloor$ equals to the maximum integer which is not greater than U_T/p_j .

The parameters and decision variables that are used in the above Integer Linear Programming formulation for the energy-efficient task assignment problem are listed below.

- parameters:
 - U_T : time interval
 - B : the set of data combinations, where $B = \{B_l\}$ for $l \in [1, 2^{|H|} - 1]$, $B_l \subseteq H$ denotes the l -th data combination, and $|H|$ (with an abuse of the symbol $|\cdot|$) denotes the size of a set H
 - b_l : the number of the occurrences of data combination B_l during U_T
- binary decision variables:
 - w_{ikj} : 1 if and only if the measurement of physical quantity j required by task k is assigned to sensor i
 - x_{ij} : 1 if and only if certain task which requires the measurement of physical quantity j is assigned to sensor i
 - y_{il} : 1 if and only if the measurement of certain physical quantity of B_l is assigned to sensor i

Remark 2 In the above formulation, U_T is used to denote the least common multiple of the sampling periods of the variables to be measured. It represents the sampling period of the overall wireless sensor network. In case that periodic samplings are not needed, U_T is used to denote the user-interested time interval.

Note given a set of physical quantities to be monitored H , there exist $2^{|H|} - 1$ nonempty subsets of H , which represent all the nonempty combinations of the physical quantities in H . In the above formulation, B is used to denote the set of such combinations. For each element of B , $B_l = \{j_1, \dots, j_k\} \subseteq H$, where $l \in [1, 2^{|H|} - 1]$. The index of a combination B_l is defined by $l = \sum_{j_k \in B_l} 2^{j_k - 1}$, for $j_k \in B_l$. That is, we consider the index l as a decimal number that consists of $|H|$ bits. Each bit of l corresponds to a physical quantity in H . If a physical quantity j (represented by the number j in H) is contained by the combination B_l , then we set the bit j of l to be 1. Otherwise, the bit j of l is set to be 0. For instance, given $H = \{1, 2, 3\}$, we have $B_5 = \{1, 3\}$. Then by this means we are able to order the combinations

of the physical quantities to be monitored arbitrarily, and to determine the physical quantities included by a data combination based on its index and vice versa. Let us revisit the example given before. Suppose $H = \{1, 2, 3\}$. Then we can order all the nonempty subsets of H : $B_1 = \{1\}$, $B_2 = \{2\}$, $B_3 = \{1, 2\}$, $B_4 = \{3\}$, $B_5 = \{1, 3\}$, $B_6 = \{2, 3\}$, $B_7 = \{1, 2, 3\}$. The notion of data combinations is introduced to the Integer Linear Programming formulation since in order to save the data transmission energy, a sensor should deliver its sampling results efficiently. I.e., at a certain time instance, if the measurement results of several physical quantities are available, then these data should not be delivered to the server separately, but should be fit in one packet and sent back together. Therefore at each sensor, we need to count how many such combined data transmissions have occurred. For this, we introduce the notion of data combinations as well as auxiliary binary y_{il} to the formulation. Later in this chapter, we will describe how to compute the energy consumption caused by the combined data transmissions in details.

Moreover since a sensor may be located inside of several sampling areas, the measurement of a physical quantity of type j at sensor i can be applied to satisfy the requests of multiple measurement tasks. Therefore we need to additionally introduce auxiliary binary variable x_{ij} to formulate the energy consumption due to measurements at each sensor.

In the above Integer Linear Programming formulation of the task assignment problem, we expect to minimize both the measurement and transmission energies resulted by a task assignment among the overall wireless sensor network. The first item of the objective function $\sum_{i \in S, j \in H} e_{ij} \lfloor U_T/p_j \rfloor x_{ij}$ computes the energy consumed by data sampling, in which $\lfloor U_T/p_j \rfloor$ computes the number of measurements of a physical quantity j at sensor i during the period U_T , and $e_{ij} \lfloor U_T/p_j \rfloor x_{ij}$ computes the energy consumed at sensor i resulting from the measurements of a physical quantity j during U_T . The second item of the objective function $\sum_{i \in S, l \in B} c_i b_l y_{il}$ computes the energy consumed by data transmitting, in which $\sum_{l \in B} c_i b_l y_{il}$ computes the energy consumption resulting from data transmissions at sensor i during U_T . To obtain the value of the parameter b_l , i.e., the number of the samplings of certain variables in B_l , we first list all the (distinct) sampling times in the time interval U_T according to the sampling periods

of the given physical quantities. Let $K = \{k_1, \dots, k_m\}$ represent the set of the sampling times. For each $k_t \in K$ ($t \leq m$), $k_t \leq U_T$, and $\exists j \in H$ such that $k_t \% p_j = 0$. Here $\%$ denotes the operation of “modulus”, i.e., the remainder after division. We next determine which physical quantity(quantities) can be sampled at k_t . Let $B(k_t) = \{j_1, \dots, j_m\}$ denote the set of physical quantities that can be measured at k_t . For each $j_s \in B(k_t)$ ($s \leq m$), $k_t \% p_{j_s} = 0$. It is easy to check that $B(k_t)$ is a nonempty subset of H , and thus it corresponds to a data combination (defined before) B_l for $l = \sum_{j_s \in B(k_t)} 2^{j_s - 1}$. That is, there exists a one-to-one correspondence between a sampling time k_t and a data combination B_l . Finally the value of b_l is obtained by checking the number of occurrences of data combination B_l during U_T . Note each sampling time corresponds to a unique data combination, and so $\sum_{B_l \subseteq B} b_l$ equals the number of distinct sampling times during U_T . Further note data transmissions at each sensor occur at sampling times (we assume sampling can be done instantaneously), and so in order to know how much energy is consumed due to the data transmissions during U_T , we only need to check if sampling of certain physical quantities occurs at the sampling times of K . This can be done by using the auxiliary binary variable y_{il} . The following example illustrates how to determine the value of b_l .

Example 1 Given a set of physical quantities to be monitored in a field $H = \{1, 2, 3\}$, where the numbers 1,2,3 represent temperature, humidity and soil moisture of the field respectively. Suppose the sampling periods of these physical quantities are 15, 25, and 35 seconds, and the time interval U_T that the users are interested in monitoring be 100 seconds, which is shorter than the least common multiple of the sampling periods of the physical quantities 525 seconds. The sampling times of the given physical quantities during U_T and their corresponding data combinations are as shown in Figure 3.1. As mentioned before, $B = \{B_l\}$ for $1 \leq l \leq 7$, where $B_1 = \{1\}$, $B_2 = \{2\}$, $B_3 = \{1, 2\}$, $B_4 = \{3\}$, $B_5 = \{1, 3\}$, $B_6 = \{2, 3\}$, $B_7 = \{1, 2, 3\}$. The sampling times within U_T are 15, 25, 30, 45, 50, 60, 70, 75, 90 and 100, and the corresponding data combinations are $B_1, B_2, B_1, B_4, B_1, B_2, B_1, B_4, B_3, B_1$ and B_2 . Then we have $b_1 = 5$, $b_2 = 3$, $b_3 = 1$, $b_4 = 2$ and $b_l = 0$ for $l \in \{5, 6, 7\}$, totally 11 sampling times during U_T . It should be noticed that b_l only represents the number of occurrences of B_l during U_T . It can

not be used to compute the energy consumption due to the data transmission of B_l during U_T directly since whether a sampling of a physical quantity of B_l is really performed at a sensor is decided by the task assignment at the sensor.

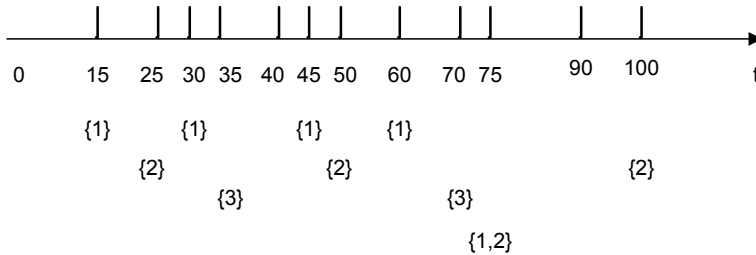


Figure 3.1 Illustration of sampling times and data combinations

Remark 3 The objective function of the above Integer Linear Programming formulation focuses on the overall energy consumption (including sampling and transmitting energy consumptions) in the whole wireless sensor network. Energy-efficiency of a task assignment is a key concern for the applications of wireless sensor networks. See for an example [19, 28]. The object function can also be re-formulated to accommodate the other energy-aware strategies. For instance, if we expect to distribute the tasks among the wireless sensor network evenly so that the energies consumed among the sensors could get balanced, then the object function can be formulated as

$$\min (\max_i \sum_{j \in H} e_{ij} [U_T/p_j] x_{ij} + \sum_{B_l \subseteq B} c_i b_l y_{il}).$$

Such objective function minimizes the maximal energy consumption due to the measurements and data transmissions at each individual sensor. The resulting optimization problem is a *min-max* optimization problem which can be converted to a minimization problem by introducing an auxiliary variable ε together with a new constraint

$$\sum_{j \in H} e_{ij} [U_T/p_j] x_{ij} + \sum_{B_l \subseteq B} c_i b_l y_{il} \leq \varepsilon,$$

and changing the objective function to be *min* ε . If the battery lifetime of a sensor network with respect to a given set of tasks is expected to be maximized, then the objective function

can be re-formulated as

$$\min (max_i \frac{\sum_{j \in H} e_{ij} \lfloor U_T/p_j \rfloor x_{ij} + \sum_{B_l \subseteq B} c_i b_l y_{il}}{Bat_i}),$$

where Bat_i represents the total battery lifetime available at sensor i . Such objective function ensures the rate of energy depletion throughout the sensor network to be balanced so that no sensor mote is over-used by minimizing the maximum fraction of the energy consumed due to a task assignment out of the energy available on a sensor.

By reformulating the objective function and introducing the variables and constraints accordingly, our proposed ILP formulation can formalize different concerns of energy consumption for the applications of wireless sensor networks.

The constraints of our proposed Integer Linear Programming formulation captures the following requirements: (1) The measurement of a physical quantity j required by task k can be assigned to sensor i only if sensor i is able to measure the physical quantity of type j (Constraint 1-measurement capability constraint). (2) The measurement of a physical quantity j required by task k can be assigned to sensor i only if sensor i is located within the sampling area of task k (Constraint 2-location constraint). (3) The measurement of a physical quantity j in task k can be assigned to sensor i only if the measurement of the physical quantity j is required by task k (Constraint 3-measurement request constraint). (4) Enough number of sensors are assigned for the measurement of a physical quantity j for task k (Constraint 4-measurement amount constraint). (5) The measurement of a physical quantity j is performed at sensor i only if there exists a task which requires the measurement of the physical quantity j is assigned to sensor i (Constraint 5 & 6-measurement redundancy constraint). (6) The transmission of the measurement result of certain physical quantity belonging to data combination B_l is performed at sensor i only if the measurement of a certain physical quantity j in B_l is assigned to sensor i (Constraint 7 & 8-transmission redundancy constraint).

As introduced before, x_{ij} is an auxiliary binary variable. It denotes whether an energy consumption due to the measurement of a physical quantity j should be counted at sensor i . For instance, when multiple tasks require the measurement of the physical quantity j , it

is possible that all such measurement tasks are assigned to sensor i (due to its efficiency of sampling). In this case, Constraint 6 guarantees x_{ij} to be 1, and Constraint 5 becomes trivial. Whereas if no measurement of the physical quantity j is needed at sensor i , Constraint 5 guarantees x_{ij} to be 0, and Constraint 6 becomes trivial. And so by using the auxiliary variable x_{ij} and Constraint 5 & 6, the energy consumption at each sensor due to the measurement of a physical quantity under the requests of different tasks will not be counted repeatedly.

Similarly y_{il} is also an auxiliary binary variable. It denotes whether the data transmission of certain physical quantity in B_l should be counted at sensor i . For instance, if there exists a physical quantity j in B_l which is assigned to sensor i for measurement by certain task, then Constraint 7 guarantees y_{il} to be 1, and Constraint 8 becomes trivial. Whereas if no measurement of the physical quantity j is assigned to sensor i , then Constraint 7 guarantees y_{il} to be 0, and Constraint 8 becomes trivial. The following example further explains how the data transmission energy is computed using y_{il} .

Example 2 Let us revisit Example 1. Suppose the measurements of physical quantity 1 & 2 are assigned to sensor i , whereas the measurement of physical quantity 3 is assigned to the other sensors, i.e., $x_{i1,2} = 1$, and $x_{i3} = 0$. Then from Constraint 7 & 8, we have $y_{i1} = 1$ for $B_1 = \{1\}$, $y_{i2} = 1$ for $B_2 = \{2\}$, $y_{i3} = 1$ for $B_3 = \{1,2\}$, and $y_{i4} = 0$ for $B_4 = \{3\}$, $y_{il} = 1$ for $l \in \{5,6,7\}$. From Example 1, we have $b_1 = 5$, $b_2 = 3$, $b_3 = 1$, which correspond to the sampling of physical quantity 1 at 15, 30, 45, 60, 90, physical quantity 2 at 25, 50, 100, and physical quantity 1 & 2 at 75 seconds, and $b_l = 0$ for $l \in \{5,6,7\}$. Then the total number of data transmissions at sensor i that occur during 100 seconds can be computed by $\sum_{B_l \subseteq B} b_l y_{il} = 5 \cdot y_{i1} + 3 \cdot y_{i2} + 1 \cdot y_{i3} = 9$. Note although $y_{il} = 1$ for $l = \{5,6,7\}$, the corresponding b_l equals 0 since the data combination B_l does not occur during 100 seconds. Therefore the transmission of B_l for $l = \{5,6,7\}$ should not be counted. Moreover, since y_{il} is binary, the data transmission of the physical quantities in B_l are not counted repeatedly. For example, $y_{i3} = 1$ for the data transmission of physical quantities 1 & 3 of B_3 . Further, since the transmission of a physical quantity and the transmission of a combination including this physical quantity are counted separately, no redundant transmission will be considered.

For instance, during the period of 100 seconds, physical quantities 1 and 2 are sampled for 6 and 4 times respectively. However since at 75 seconds, the measurement of physical quantities 1 & 2 will be delivered to the server together, only one transmission (for B_3) will happen at this moment. Therefore the total number of data transmissions that occur during the period of 100 seconds is 9, the same as what we have computed before. On the other hand, if it is assumed that only physical quantity 2 is assigned to sensor i for measurement, we have $y_{il} = 1$ for $l \in \{2, 3, 6, 7\}$, and $y_{il} = 0$ for $l' \in \{1, 4, 5\}$. Then it can be checked that the total number of data transmissions at sensor i during the given period is 4. This can also be obtained by $\sum_{B_l \subseteq B} b_l y_{il} = 3 \cdot y_{i2} + 1 \cdot y_{i3} = 4$. With the number of data transmissions in hand, the total energy consumption resulting from the data transmissions at each sensor in the wireless sensor network can then be computed by $\sum_{i \in S, B_l \subseteq B} c_i b_l y_{il}$.

So far we have proposed an Integer Linear Programming formulation to model the energy-efficient task assignment problem of wireless sensor network and given the descriptions of the decision variables, objective function and constraints of the proposed Integer Linear Programming formulation. Next we point out a simplification of the proposed formulation.

Note q_{ij} , l_{ik} and d_{kj} are binary parameters, not decision variables. And so Constraint 1 3 of the full version of the proposed formulation can be combined as

$$w_{ikj} \leq q_{ij} \cdot l_{ik} \cdot d_{kj}, \text{ for } i \in S, k \in T, j \in H.$$

This constraint is still a linear constraint since q_{ij} , l_{ik} and d_{kj} are not decision variables.

Remark 4 q_{ij} , l_{ik} and d_{kj} are binary parameters, and so Constraint 1 3 of the full version of the proposed formulation contain multiple redundant inequations. Such repetitive inequations can be cleaned by the solvers of Integer Linear Programming as well.

3.3 Size of Proposed Formulation

In the following we analyze the size of the proposed Integer Linear Programming formulation of the task assignment problem.

The number of the decision variables and constraints of the proposed Integer Linear Programming formulation is

- w_{ikj} : $|S| \times |T| \times |H|$
- x_{ij} : $|S| \times |H|$
- y_{il} : $|S| \times (2^{|H|} - 1)$
- constraints on w_{ikj} (Constraint 1-4): $3|S| \times |T| \times |H| + |T| \times |H|$
- constraints on x_{ij} (Constraint 5,6): $|S| \times |T| \times |H| + |S| \times |H|$
- constraints on y_{il} (Constraint 7,8): bounded by $|H| \times |S| \times (2^{|H|} - 1) + |S| \times (2^{|H|} - 1)$

Remark 5 Constraints on w_{ikj} can be simplified as proposed before. In the simplified version of the Integer Linear Programming formulation, the number of the constraints on w_{ikj} is $|S| \times |T| \times |H| + |T| \times |H|$.

In the following we estimate the size of the task assignment problem of a middle-sized sensor network. Given a wireless sensor network consisting of 10 sensors, suppose each sensor of the network can measure at most 8 types of physical quantities and the users have submitted 5 measurement tasks. Then from the above analysis, we have 3030 binary decision variables and at most 24670 constraints will be needed in the proposed Integer Linear Programming formulation (the full version) for the given wireless sensor network, which is computable for the existing solvers of the Integer Linear Programming problem. This shows that the proposed Integer Linear Programming formulation is workable for the small and middle-sized wireless sensor network. Whereas for the large-scaled wireless sensor network, solving an Integer Linear Programming optimization problem, which has been proved to be NP-hard, is challenging. In this beginning research of the task assignment among wireless sensor network, we focus on the problem formulation and its implementation. The heuristic algorithm for efficiently solving the proposed Integer Linear Programming formulation will be further investigated in the future work.

CHAPTER 4. Realization of Task Assignment System

In this chapter we introduce the architecture of the developed task assignment system and its implementation using Java, PBSolver of SAT4J and TinyOS.

4.1 System Design

We formulate the energy-efficient task assignment problem of wireless sensor networks with the application to agriculture as an instance of an Integer Linear Programming optimization problem. In the following we design a task assignment system to realize the assignment of tasks among a given wireless sensor network by solving the proposed ILP formulation.

The task assignment system we developed consists of three modules: assignment computation module, assignment illustration module and assignment execution module. In an real application of the task assignment system, the assignment computation module will run on a computer in the lab so that an optimal task assignment can be calculated and refined according to the user needs. The assignment illustration module will run on the server in the field so that the users can send commands/receive sampling data to/from the sensors deployed in the field. The assignment execution module will run on the sensor nodes so that the commands/sampling data can be processed/delivered in the wireless sensor network and the sampling tasks can be performed at each sensor according to the given assignment. The following figure illustrates the architecture of the task assignment system.

The detailed design of each module is presented next.

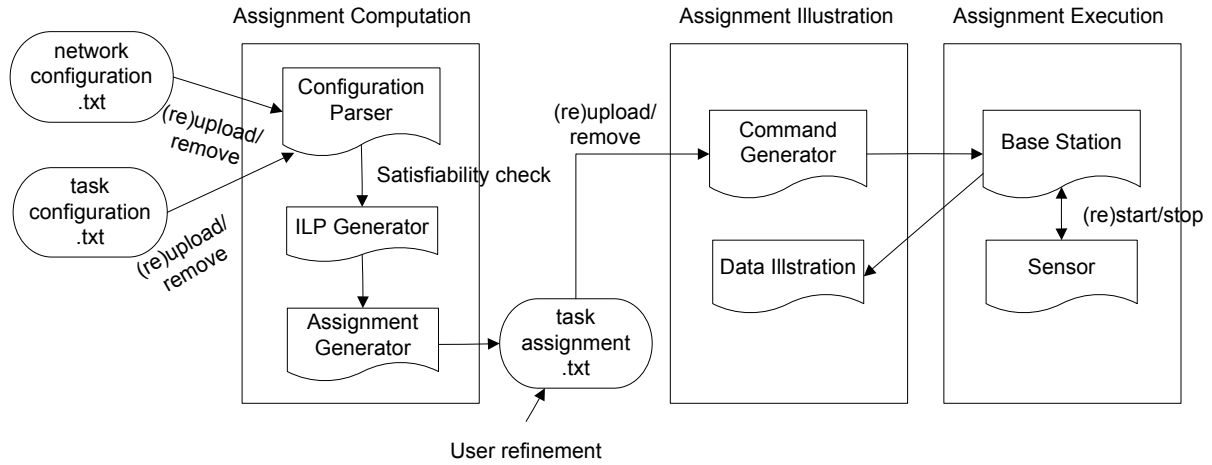


Figure 4.1 Architecture of task assignment system

4.1.1 Assignment Computation Module

The major task of the assignment computation module is to solve the task assignment problem formulated in the previous chapter. In this beginning research, we focus on the problem formulation and its implementation, and so we choose the existing solvers to resolve the Integer Linear Programming optimization problem. The assignment computation module is responsible for preprocessing/postprocessing the solver input/output.

The assignment computation module consists of the following three components: configuration parser, ILP generator, and assignment generator. For configuration parsing, necessary configuration information of the sensors and measurement tasks is inputted to the task assignment system. The configuration of the sensors includes the location, brand, type, and the variables that can be measured, the energy consumed by each sampling and data transmission of a sensor. The configuration of the tasks include the location to be monitored, the variables to be measured, the sampling periods of the interested variables, and the amount of measurements expected. For simplicity, the configuration is written into textual files of predefined formats. By reading these files, the task assignment system can get the needed information of the sensor network and user requirements. For ILP formulation generation, the input file for an Integer Linear Programming solver is created based on the proposed ILP formulation and the solver is initiated to resolve the Integer Linear Programming optimization problem.

For assignment generation, the output file returned by the solver is first parsed. If a (either optimal or satisfiable) solution is found, the value of objective function and solver running time are reported to the users on a graphical user interface, and the assignment of the given tasks among the sensor network is written into a textual file based on the solution found by the solver. The resulting task assignment file will be inputted to the assignment illustration module. Whereas if the solver can not find a solution (either the solver does not know how to solve the given optimization problem or proves the given problem unsatisfiable), no task assignment is created and the users will be informed that no solution is found.

In addition, the assignment computation module also allows the users to re-upload/remove the configuration files to/from the task assignment system so as to keep track of the changes of the network/task configuration. Beside, the assignment computation module supports a graphical illustration of the sensor deployment in the crop field. Furthermore, the task assignment module provides the function of initial satisfiability check. The solver for ILP problem is initiated only if the given ILP is satisfiable.

4.1.2 Assignment Illustration Module

The major task of the assignment illustration module is to assign the tasks to the sensors according to the given assignment and display the returned sampling data.

The assignment illustration module consists of the following two components: command generator, and data illustration. For assignment command generation, the assignment illustration module first parses the task assignment file generated by the assignment computation module. For each sensor that is assigned certain measurement tasks, the sampling (timer) period of the sensor which equals the least common multiple (greatest common divisor) of the sampling periods of the assigned tasks is computed and written into the command packet together with the address of the destination sensor. Then the assignment illustration module sends the task assignment commands to a special sensor mote which is connected to the server, named base station, via serial port. The base station then delivers the received command packets to the destination sensors via the wireless sensor network by following certain

wireless communication protocol. Similarly the sampling data are delivered back to the base station from the sensors via the wireless sensor network and further delivered to the assignment illustration module via serial port. Then the assignment illustration module displays the received data on a graphical user interface. Note the data transmissions in the wireless sensor network are accomplished by the assignment execution module described in the following section.

In addition, the assignment illustration module also allows the users to re-upload/remove the task assignment to/from the task assignment system so as to keep track of the changes of the task assignment. Besides, the assignment illustration module supports the functions of terminating/restarting data samplings in the wireless sensor network.

4.1.3 Assignment Execution Module

The major task of the assignment execution module is to send/recieve task assignment commands and perform sampling according to the task assignment at each sensor.

The assignment execution module consists of the following two components: base station and sensor. A base station is a special sensor which behaves as the bridge between the server and the wireless sensor network. For base station, the packet received from the server containing task assignment commands is sent to the wireless sensor network, whereas the packet received from the wireless sensor network containing the sampling data is sent to the server. For sensor, the assignment execution module is responsible for the execution of sampling tasks and transmission of sampling results. When a task assignment command arrives, a sensor first checks whether a sampling task should be (re)started or stopped. If sampling at a sensor should be (re)started, the timer of the sensor is initiated to perform periodic measurements. When the sampling results are ready, the sampling data are sent back to the server via the wireless sensor network. And if sampling at a sensor is required to be stopped, the assignment execution module stops the sensor timer and resets the sensor for the new coming sampling tasks.

4.2 Implementation Issues

In this section we discuss the implementation issues of the task assignment system.

4.2.1 Development Language

The three modules of the task assignment system work in different environments: the assignment computation module running on a computer in a lab, the assignment illustration module running on a server in the crop field, and the assignment execution module running on a base station/sensors of the wireless sensor network. Considering the differences of the functions and working environments of each module, the task assignment system is implemented using different programming languages. The assignment computation and illustration modules can be implemented using advanced programming languages such as C, C++, Java, etc. since they will run on the computers with fewer limitations. In this work, we develop the assignment computation and illustration modules using Java. The assignment execution module can be implemented using wireless sensor network development tools such as TinyOS since this module will run on sensor motes which have quite limited memory, computation capability and power. In this work, we develop the assignment execution module using TinyOS. TinyOS is an open-source operating system designed for wireless sensor network, of which the component-based architecture enables minimizing the implementation codes as required by sensor resource constraint. It provides the library for network protocol, sensing, data acquisition, and simulation, which greatly simplifies the software development procedure for the applications of wireless sensor network.

The sizes of Java codes for implementing assignment computation and illustration modules and TinyOS code for assignment execution module are listed in Table 4.1.

The developed TinyOS software running on a base station (sensor mote) is at the size of 2873 Bytes in RAM, 27646 Bytes in ROM (respectively 2911 Bytes in RAM, 28754 Bytes in ROM), which can be implemented on the sensor motes with limited memories.

Table 4.1 Size of task assignment system

Module	Language	Lines	Size
Assignment Computation	Java	3296	86.9KB
Assignment Illustration	Java	1403	34KB
Assignment Execution-Base Station	TinyOS	244	7KB, 2873 Bytes in RAM
Assignment Execution-Mote	TinyOS	456	11KB, 2911 Bytes in RAM
Total:		5399	138.9KB

4.2.2 ILP Solver

In this initial research of task assignment of wireless sensor network, we adopt the existing solvers to resolve the proposed ILP optimization problem. Many ILP solvers have been provided by the academia and industry such as CPLEX [1], LP-Solve [2], MatLab [3], Excel Solver [4], Coin-OR [5], AMPL [6] for solving (Mixed) Integer Linear Programming problems. Integer Linear Programming has a close relation with SAT (satisfiability). An ILP problem can be converted to a SAT problem (and vice versa), and thus can be solved by SAT solvers. SAT has received a lot of attention in the literature of computer science. Today several efficient SAT solvers have been developed, for instance Spear, MiniSat⁺ [7], RSat [8], SAT4J [9], and so on. SAT4J is a mature open-source SAT solver. Its efficiency has been validated during SAT competition 2004, 2005 and SAT Race 2006. Now SAT4J has been applied to many fields such as formal verification, algorithm configuration, software engineering, and semantic web. In this work, we apply PBSolver, part of SAT4J, to optimize the task assignment problem. PBSolver is a solver for Pseudo Boolean (PB) problem, which is a generalization of SAT problem. The interested readers are asked to refer to the references about the relations/conversions between 0-1 Integer Linear Programming, SAT and PB problems. It should be noticed that ILP/SAT is NP-complete. The existing ILP/SAT solvers can not work efficiently in any ILP/SAT problem. Heuristic algorithm especially for solving the proposed Integer Linear Programming formulation of task assignment of wireless sensor network is expected.

4.2.3 Assumptions

We make the following assumptions in while implementing the task assignment system.

1. Each sensor can measure at most 8 physical quantities.
2. The wireless sensor network consists of at most 256 sensors.
3. Sampling data can be encoded using 2 bytes.
4. Average sampling result is sent back to the users.
5. Commands/Sampling data can be delivered in the wireless sensor network successfully .
6. Commands/Sampling data can be delivered to the destinations on time.
7. No failure occurs at each sensor in the wireless sensor network works.

A sensor mote has very limited resources and sensing/computation capabilities. We assume that at most 8 types of variables can be measured at each sensor. This assumption holds in most applications of wireless sensor network. Due to the computation efficiency issue, we prefer to applying the proposed ILP formulation to the task assignment of middle-size wireless sensor network. Therefore it is assumed that the wireless sensor network contains no more than 256 sensors. The sampling results need be encoded (into integers) for transmission. We assume that the sampling data can be stored in 2 bytes. This assumption holds in most applications with no tight precision requirements. Assumption 1~3 are made to determine the size of commands/sampling data packet. In order to save the communication/data transmission energy at each sensor, not every sampling result needs to be reported especially in case that the sampled variable does not change very frequently. Therefore we assume that the average of every 10 (determined by the designer according to the characters of the variables and user needs) samplings need be sent back. Assumption 4 is made to determine when and how a sampling result is reported. For simplicity, the designed task assignment system is a non-fault-tolerant open-loop system without taking sensor feedback into consideration. For example, when a task is assigned to a sensor, the sensor is not asked to report its current status to the server. And the server will not reassign the tasks if the sensors that should perform the tasks have failed. Therefore we assume that the users carefully monitor the working condition of the wireless sensor network and perform the task assignment only when the whole network works.

Remark 6 In the previous chapter U_T is used to denote the sampling period of the overall wireless sensor network or the sampling time interval given by the users. In the former case, while applying the task assignment system to an application, the value of U_T should be modified accordingly. This because we assume that not every sampling, but the average of every certain number of samplings, is reported to the users. Then in order to compute the energy consumption resulting from data transmissions correctly, we need multiply U_T accordingly.

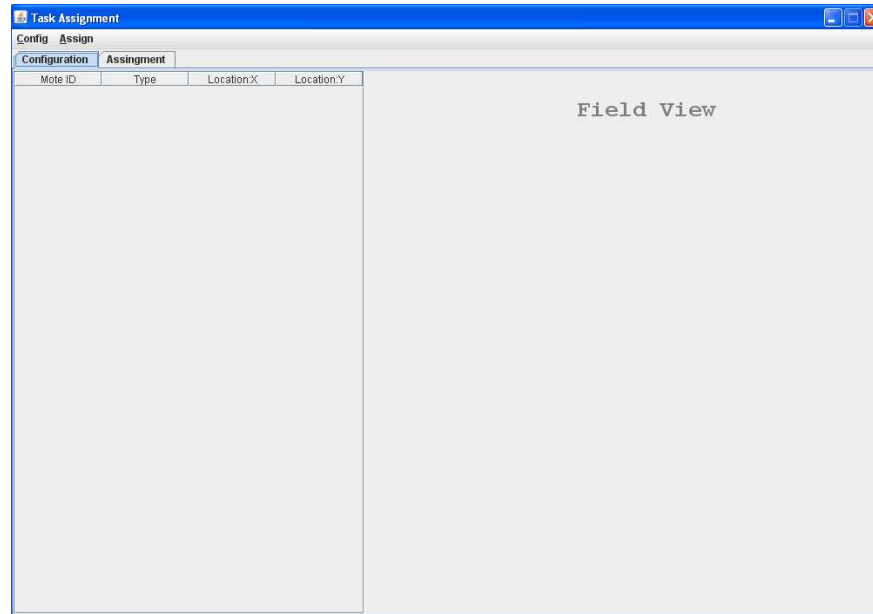


Figure 4.2 Interface of assignment computation module-configuration

4.3 Task Assignment System GUI

To support the functions of the task assignment system, we design graphical user interfaces (GUIs) for the assignment computation and assignment illustration modules. Figure 4.2 and 4.3 illustrate the user interfaces of the assignment computation module, and Figure 4.4 of the assignment execution module.

The interface of the assignment computation module consists of two tab pages. At Configuration page, the users can upload/reupload/remove the configuration of sensors to/from the task assignment system. And the distribution of sensors in the crop field is displayed

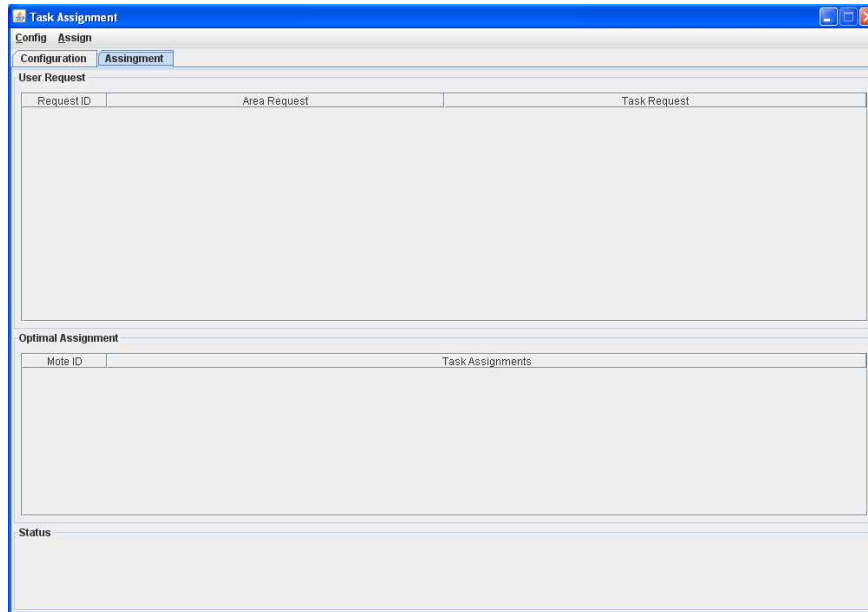


Figure 4.3 Interface of assignment computation module-assignment

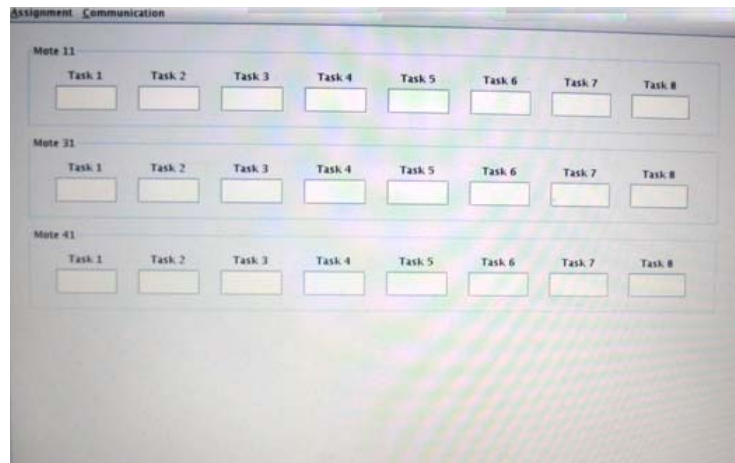


Figure 4.4 Interface of assignment illustration module

automatically in the Field View area.

At Assignment page, the users can (re)upload/remove the configuration of tasks and a task assignment to/from the task assignment system. Feasibility of task assignment among the wireless sensor network for the given tasks is examined before the solver is called to solve the ILP optimization problem. If the ILP problem is satisfiable, then the users are allowed to start the solver by using the Assignment menu. The resulting satisfiable/optimal solution is displayed in the Optimal Assignment area, and the corresponding energy cost and solver running time are displayed in the Status area.

On the interface of the assignment illustration module, the users can upload/reupload/remove the task assignment to/from the task assignment system. Then the sensors can be started/stopped performing the sampling tasks. The sampling results returned from the sensors are displayed in the Data area.

CHAPTER 5. Simulation and Evaluation

In this chapter we evaluate the proposed Integer Linear Programming formulation and designed task assignment system by simulations on real wireless sensor networks.

5.1 Simulation Environment and Parameters

Before demonstrating the simulation results, we first introduce the simulation environment and parameters.

The simulation is performed using a desktop, a laptop and wireless sensor network: the assignment computation module on the desktop, the assignment illustration module on the laptop and the assignment execution module on the wireless sensor network. The desktop (laptop) has 512 MB RAM memory and with Intel Pentium 4(M) CPU of 1.8GHz (1.6GHz). The wireless sensor network consists of Telosb sensor nodes. Telosb nodes, produced by Crossbow Technology INC, are designed for experiments for the research community. Telosb node has integrated onboard antenna, TI microcontroller with 10KB RAM, 250kbps data rate, and IEEE 802.15.4 compliant RF transceiver. It can be integrated with temperature, humidity and light sensor. For the details of Telosb nodes, the interested readers are asked to refer to Crossbow Technology website and Telosb data sheet.

Table 5.1 lists the parameters that are used for simulation.

From the experiment result of [21], each sampling (resp., transmitting) consumes around 8.7mJ (resp., 28.1mW). Note the data rate of the Telosb node is 250kbps and the command/data packet consists of 23 bytes. And so it can be computed that each transmission consumes $23/(125 \cdot 10^3) \cdot 28.1 = 0.0202\text{mJ}$. Base station is responsible for transmission between the server and the sensor network. In order to respond the received commands/data,

Table 5.1 Simulation parameters

Parameter	Value
Energy of each sampling	8.7 mJ
Energy of each transmission	0.0202 mJ
Length of command/data packet	23 bytes
Length of packet buffer	10
Command sending interval	1s

the received packets are first buffered in a queue and sent out when the base station is not busy. A tradeoff between the buffer size and efficiency/performance should be considered. If the buffer is too big, that will waste the limited memory of base station. However if the buffer is too small, more packets have to be dropped. In this work the size of the packet buffer at base station is chosen to be 10. If the simulation is applied to large-scale sensor network with dense measurement requirements, the size of the packet buffer should be increased accordingly. While the server sends out task assignment commands, in order to avoid interference/collision of wireless transmission, we add 1s delay after sending each command.

The structure of command/data packet is shown in Figure 5.1. A command packet consists of 23 bytes that represent mote ID, task assignment, sampling periods, gcd (greatest common divisor) and lcm (least common multiple) of the assigned sampling tasks. Each bit of assigned tasks denotes whether the task is assigned to the sensor. It is set as zero when the sampling tasks performed on the sensor are requested to be stopped. A data packet reuses the structure of a command packet. The sampling period bytes of a command packet are used to store the sampling results, and the gcd/lcm bytes are used to store the index of the sampling results (to determine the sampling times at the sensor).

5.2 Simulation Results

We evaluate the performance of the proposed Integer Linear Programming formulation and designed task assignment system by simulation on small-sized as well as middle-sized wireless sensor works. In these simulations we assume that the field where the sensors are deployed is rectangular, with a length of 200 meters and a width of 100 meters. In the following we first

2 bytes	1 byte	2*8 bytes	2 bytes	2 bytes
Mote ID	Task Assignment	Sampling Period of Each Task	GCD (Timer Period)	LCM (Sampling Period)

(a) command packet

Mote ID	Task Assignment	Sampling Result of Each Task	Sampling Index
---------	--------------------	------------------------------	----------------

(b) data packet

Figure 5.1 Structure of command/data packet

give the simulation result on a small-sized sensor network to evaluate the correctness of our ILP formulation.

5.2.1 Simulation On A Simple Wireless Sensor Network

In this simulation a simple wireless sensor work consisting of twelve sensors is selected. The sensors are evenly distributed in the field. The configuration of the sensors, including the index, location, and sensor type of each sensor is as listed in Table 5.2.

Table 5.2 Configuration of sensors

Mote Index	Location	Mote Type
1	(0,10)	2
2	(0,50)	1
3	(0,90)	2
4	(60,10)	1
5	(60,50)	2
6	(60,90)	1
7	(120,10)	2
8	(120,50)	2
9	(120,90)	2
10	(180,10)	1
11	(180,50)	2
12	(180,90)	2

where the location of each mote consists of its coordinates in $X - Y$ plane, in which X -axis

corresponds to the width of a field, and Y -axis corresponds to the length of a field.

As mentioned before, it is assumed that there are at most 8 physical quantities to monitored. For sensor of type 1, we assume that it can measure the physical quantities of type 1, 2, 3, 5, 6, 8 and for sensor of type 2, it can measure the physical quantities of type 1, 2, 4, 5, 7, 8. And we assume that the sampling period of each physical quantity is 15, 25, 40, 35, 20, 20, 20, 20 seconds respectively.

The user measurement requests are given in Table 5.3.

Table 5.3 Configuration of tasks

Area Index	Location, Radius	Expected No. Samplings
1	(30,30), 50	2,2,1,0,1,0,0,0
2	(90,70), 40	2,2,1,1,1,0,0,0
3	(150,40), 50	0,0,0,1,1,1,0,1

where to simplify the question we choose the areas to monitored as circular areas.

The combined information of sensors as well as user requests are as shown in Figure 5.2, in which the black (resp., white) nodes represent the sensors of type 2 (resp., 1). The type of each sensor node is also denoted by the number beside each node. And the number inside of each node denotes the index of the sensor. The circles with dotted lines illustrate the measurement areas requested by the users. The number at the center of each circle denotes the index of the area.

From Figure 5.2, we have sensors 1, 2, 4, 5 are located inside area 1, sensors 5, 6, 8, 9 inside area 2, and sensors 7, 8, 10, 11 inside area 3. In area 2, sensors 5, 8, 9 are of type 2, whereas sensor 6 is of type 1. Note task 3 can only be performed by a sensor of type 1. Therefore in order to satisfy the user requirement on area 2, which requires at least one measurement of task 3, sensor 6 must be selected for measurement of task 3. Similarly in area 3, sensors 7, 8, 11 are of type 2 and sensor 10 is of type 1. Therefore in order to measure task 4 (which can only be sampled by a sensor of type 1) and task 6 (which can only be sampled by a sensor type 2) at area 3, at least one sensor among sensors 7, 8, 11 should be assigned task 4, and sensor 10 must be selected to perform task 6. Also since sensor 5, which can perform the measurements

of tasks 1, 2, 5, are within area 1 as well as area 2, the selection of sensor 5 can more efficiently contribute to the accomplishment of user requests of tasks 1, 2, 5 in both area 1 and area 2. Therefore as so to save measurement energy, sensor 5 should be at least assigned the tasks 1, 2, 5. Note at least two samplings of tasks 1, 2 are expected at area 2. Then either sensors 6, 8, 9 should be chosen to perform tasks 1,2, or any two from the three sensors 6, 8, 9 should be chosen to perform task 1, 2 respectively.

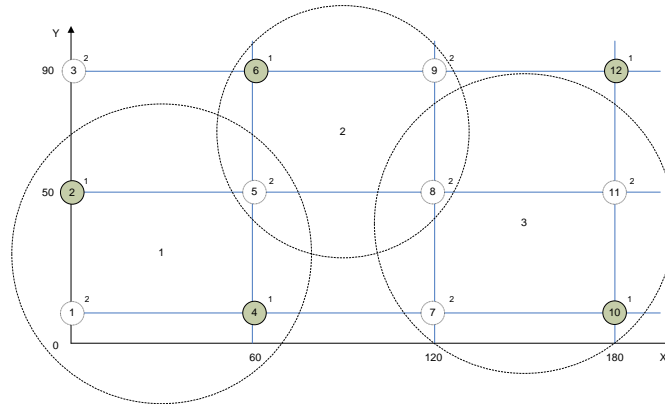


Figure 5.2 Configurations of Sensors and User Requests

We develop a Java simulation program to create the ILP formulation for the given task assignment problem and use SAT4J solver to resolve it. By running the simulation program, we have the proposed ILP formulation for the given small-sized sensor network consists of 3444 binary variables and 16620 constraints (at the full form). Based on the analysis in the previous chapter, the time interval U_T is chosen as 42000 seconds, which equals 10 times of the sampling period of the overall sensor network. The simulation result shows the optimal energy consumption during U_T is 218752.460 mJ. And it takes the solver 175.763 seconds to find such optimal solution. The optimal task assignment obtained by resolving the ILP formulation is as shown in Table 5.4, in which 0 denotes no task is assigned.

From our prior analysis, we have sensor 5 should be assigned tasks 1, 2, 5, sensor 10 should be assigned task 6, and sensor 6 should be assigned task 3, to fulfill the user measurement requirements. The optimal assignment that we obtain by solving the proposed ILP formulation using SAT4J gives us the same solution.

Table 5.4 Optimal task assignment

Mote Index	Task Assignment
4	1,1,1,0,0,0,0,0
5	1,1,0,0,1,0,0,0
6	0,0,1,0,0,0,0,0
8	1,1,0,1,0,0,0,0
10	0,0,0,0,1,1,0,1

Next we analyze the other assignments. We first show the obtained optimal assignment is a feasible solution to the given user requests and configurations. In order to prove this, it suffices to check if each of the user requests has been satisfied. From the assignment result, we have at area 1, sensor 4 is responsible for the measurements of tasks 1, 2, 3, whereas sensor 5 is responsible for tasks 1, 2, 5. That is, we have two measurements for tasks 1, 2, and one measurement for tasks 3, 5. This satisfies the user requirement on area 1 while respecting the restrictions on location and sensing capability of both sensors. Similarly, at area 2, tasks 1, 2, 5 are taken care of by sensor 5, task 3 by sensor 6, tasks 1, 2, 4 by sensor 8; and at area 3, task 4 by sensor 8, and tasks 5, 6, 8 by sensor 10. Such assignment satisfies the user requirements of two samplings for tasks 1, 2, one for tasks 3, 4, 5 at area 2, and one for tasks 4, 5, 6, 8 at area 3. It should be noted that under such assignment, the measurements of tasks 1, 2 by sensor 5 and task 4 by sensor 8 are simultaneously used for the measurements at areas 1, 2, and areas 2, 3 respectively, which help to save the measurement energy consumption. Then we show that the obtained optimal assignment is also indeed an optimal solution. From the above analysis we know the computed assignment takes advantage of sensors 5, 8 (which are shared by areas 1, 2 and areas 2, 3 respectively) to work for the two areas simultaneously. Therefore the resulting energy consumption is less than any other assignment which requires sensor 1 or 2 to measure tasks 1, 2 at area 1, sensor 6, 9 at area 2. Meanwhile the computed assignment has not assigned any measurement other than what is requested. And so we can conclude that this is an optimal solution to the given assignment problem (since it is a feasible solution with a consumption cost smaller than the other possible solutions). This demonstrates the correctness of our ILP formulation and the corresponding software developed. It further demonstrates the feasibility

of applying the proposed ILP formulation to solve the energy-efficient task assignment problem on small-sized sensor networks.

In the following we compare the proposed ILP formulation based approach to a baseline task assignment scheme, namely random task assignment, which assigns the given set of tasks among sensor networks without considering energy efficiency. A random assignment among the above simple sensor network is shown in Table 5.5.

Table 5.5 Random task assignment

Mote Index	Task Assignment
2	1,1,1,0,1,0,0,0
5	1,1,0,1,0,0,0,0
6	1,1,1,0,1,0,0,0
7	0,0,0,1,0,0,0,0
10	0,0,0,0,1,1,0,1

It can be checked the random assignment given in Table 5.5 satisfies all the user requests and restrictions on the given sensor network: In area 1, sensors 2, is assigned tasks 1, 2, 3, 5 and sensor 5 is assigned tasks 1, 2; In area 2, sensors 5 is assigned tasks 1, 2, 4, and sensor 6 is assigned tasks 1, 2, 3, 5; In area 3, sensor 7 is assigned task 4, and sensor 10 is assigned tasks 5, 6, 8. Totally 15 tasks are assigned and the energy consumption corresponding to such random assignment is 2474638.80mJ. Here the random task assignment is achieved by setting the objective function of the proposed ILP formulation to be zero and then using SAT4J to solve the resulting ILP problem (with no objective function). However if the optimal task assignment as shown in Table 5.4 is applied, in total 13 tasks are assigned in the network and the resulting energy consumption is 218752.46mJ, which is more energy-efficient than the random assignment scheme. And so it is critical to apply an energy-efficient approach for task assignment problem of wireless sensor networks.

5.2.2 Simulation On Multiple Wireless Sensor Networks

In the previous subsection we report our simulation result on a simple small-sized wireless sensor network which consists of twelve sensors of different types. In the following

we apply the proposed ILP formulation on multiple wireless sensor networks to evaluate the feasibility and performance of our proposed approach.

In this simulation we select 15 wireless sensor networks that consist of 9, 12, 16, 20, 25, 30, 36, 42, 49, 56, 64, 72, 81, 90, 100 sensors respectively. Among these sensor networks, sensors are evenly distributed (in form of sensors array) in an area with a length of 80 meters, a width of 180 meters and a left-bottom corner $(10, 10)$ in the field. The locations of sensors are decided as follows. Let n be the number of sensors at each row, m be the number of sensors at each column, and (x_i, y_j) be the location of a sensor s_{ij} , at row i and column j . Then $x_i = 10 + (i - 1) * 180/n$, $y_j = 10 + (j - 1) * 80/m$. In contrast to the simulation done in the prior subsection, we assume that all sensors have the same sensing capability, i.e., of the same type. Such assumption is not over restricted since in some real applications of sensor networks, a sensor network may consist of identical sensor nodes. In addition we assume that each sensor is capable to perform all the measurement tasks. Such an assumption is made since we expect to guarantee the existence of a solution to the proposed ILP formulation for each sensor network so that we can focus on studying the performance of the proposed ILP-based approach. Moreover in our simulation we select 5 sets of user requests. The user requested consist of 1, 2, 3, 4, 5 measurement requirements respectively: each set $G_n = \{r_i\}$ for $i = 1, \dots, n$, and $n \in \{1, \dots, 5\}$, consists of n measurement requirements as shown in Table 5.6.

Table 5.6 User requests

Request Index	Area, Location	Expected No. Samplings
1	(30,30), 50	2,2,1,0,1,0,0,0
2	(90,70), 40	2,2,1,1,1,0,0,0
3	(150,40), 50	0,0,0,1,1,1,0,1
4	(20,70), 30	2 2 1 1 1 0 1 1
5	(175,70), 35	0 1 0 1 0 1 0 0

In total we have performed $15 \times 5 = 75$ simulations on each sensor network for each group of user request. The simulation results for the number of variables and constraints, the running time and energy consumption of each case are demonstrated in Figure 5.3, 5.4, 5.5,

5.6 respectively.

From Figure 5.3, 5.4, we have the number of variables and constraints of the proposed ILP formulation increase with the increasing of the size of a sensor network and the number of measurement requirements. For a small-sized network (of less than 50 sensors), such increase (with the number of measurement requirements) is not significant. Whereas with the increase of the size of a sensor network, the number of variables and constraints increase faster with the increase of the number of measurement requirements. Also the simulation result shows us the scale of the proposed ILP formulation for small-sized as well as middle-sized networks with respect to a small amount of (at most 5) measurement requirements, which helps us to estimate the size and difficulty of an ILP problem formalized for a given sensor network and a set of user requests.

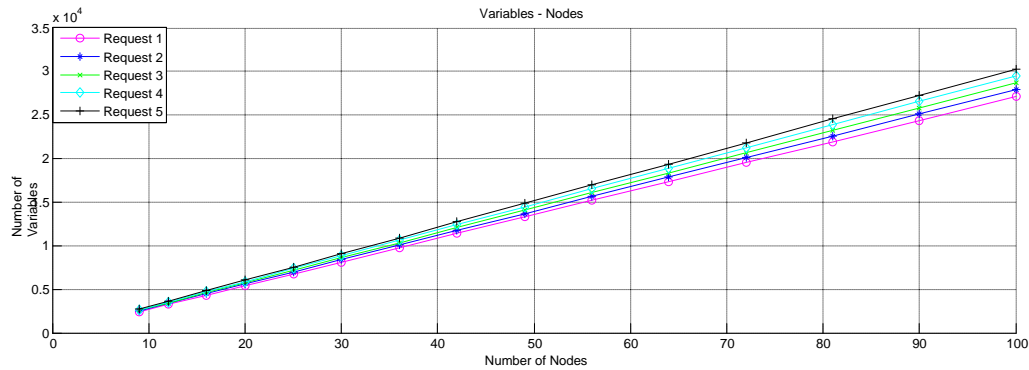


Figure 5.3 Number of variables vs nodes

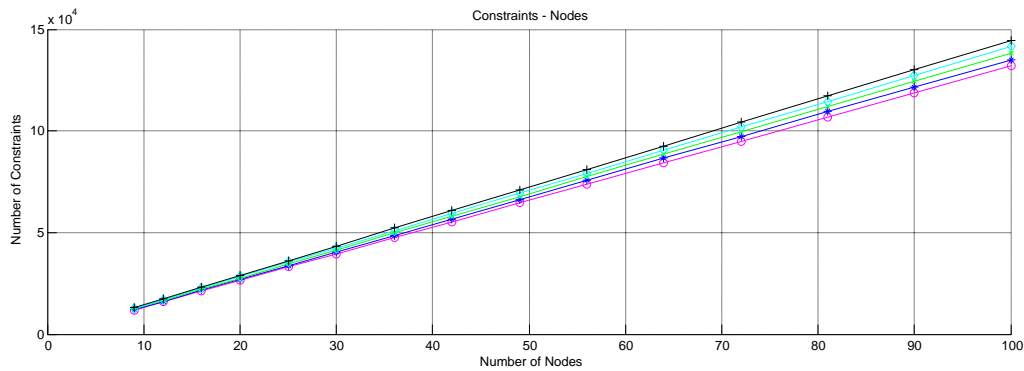


Figure 5.4 Number of constraints vs nodes

As mentioned before, ILP is a NP-hard problem. Therefore we may not be able to solve it in polynomial time. Due to this feature, an ILP solver may not always provide an optimal solution (efficiently). In such situation, SAT4J, the solver adopted in our simulation to resolve an ILP problem, can only return a satisfiable solution (a solver is not sure if the returned solution is an optimal one) or even no solution (a solver is unable to judge if there exists a solution to the given ILP problem). To avoid a solver from keeping on searching, we choose 18000 seconds (5 hours) as an upper bound of our simulation time. I.e., after running the solver for five hours, we will terminate the solver and retrieve its outputs no matter whether an optimal solution has been found. The running time of SAT4J for each case is shown in Figure 5.5.

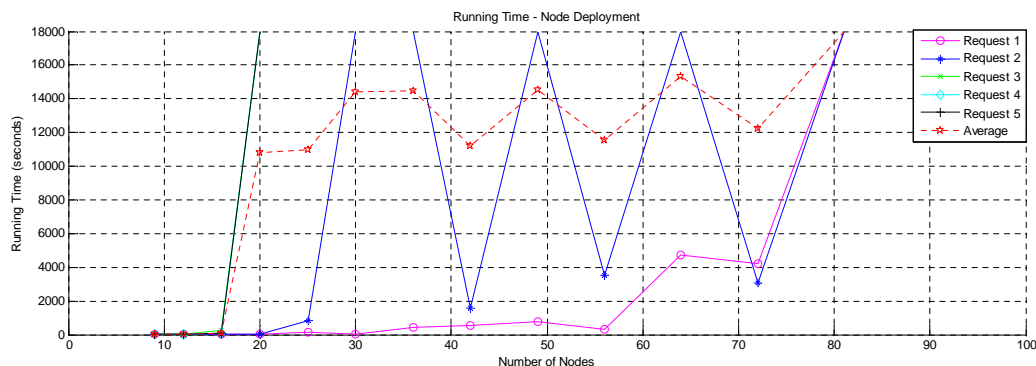


Figure 5.5 Running time vs nodes

Each line in Figure 5.5 corresponds to the simulation result of the selected 15 sensor networks with respect to a certain group of user request, as illustrated in the legend of Figure 5.5. The solid line with stars represents the average of the five simulation results of each selected sensor network. Such plot is drawn for statistic analysis since SAT4J cannot work efficiently for any ILP problem. For instance, for request 2, namely G_2 , SAT4J solver could find the optimal solutions for the sensor networks consisting of 42, 56, 72 sensors, whereas it spent much more time on returning only satisfiable solutions for the sensor networks consisting of 36, 49, 64, 81 sensors. For the other cases (G_1, G_3, G_4, G_5), as shown in Figure 5.5, SAT4J solver needs more time on calculation with the increase of the network size and the number of user requests. The simulation result also shows that at least 5 hours are needed to obtain a

solution to a small-sized or middle-sized sensor network generally.

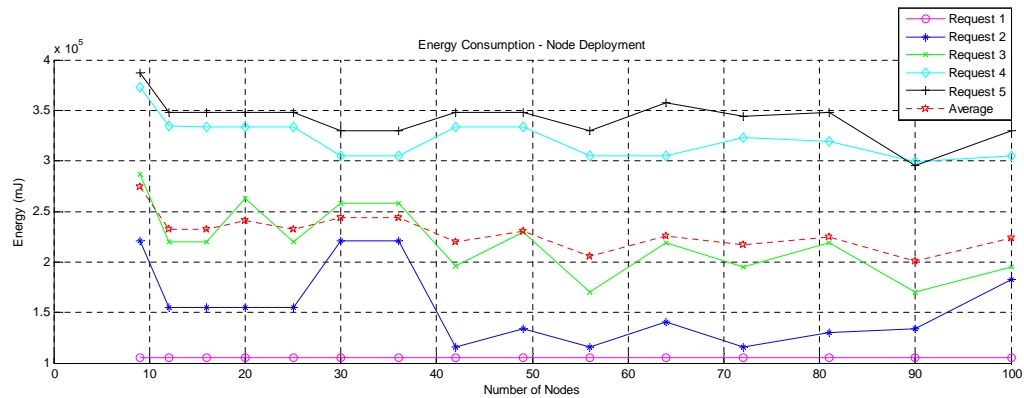


Figure 5.6 Energy consumption vs nodes

Figure 5.6 illustrates the energy consumption for each case. We can see the more requests are satisfied, the more energies are needed. This is as estimated. For a specific request, if a sparser network can satisfy a given request, then a denser network deployed in the same area could also satisfy the request with at least the same energy consumption. This is because if a denser network can also satisfy the user request, then for each requested area, it will contain at least the same number of sensors for performing the required measurement tasks. Note the measurement and communication energies consumed at each sensor in our simulation are the same. We have at least the same amount of energy is needed in the denser network. Also since the requested areas may overlap, then in a denser sensor networks, it is possible for more sensors to be shared by multiple requested areas and therefore could fulfill multiple requests for multiple areas simultaneously. Then from this analysis we estimate that the energy consumption plot should be non-increasing, or probably decreasing with the increase of the network size. However the simulation result is not exactly as what we have analyzed above. This is because SAT4J solver does not always return an optimal solution especially when an ILP problem becomes more complicated (in form of object function, variables, constraints, etc.) Therefore it may only report a suboptimal solution. This may result in a network that could have a smaller energy consumption is eventually assigned the tasks in a costly way. However we can still tell from the figure that the energy consumption has the tendency to decrease with

the increase of the size of a network. Such tendency is more evident in the plot (the solid line with stars) consisting of the average of the five simulation results of each selected sensor network.

In our work we focus on small as well as middle-sized sensor networks applied to agriculture. In such applications a sensor network of a size of 100 sensors could usually satisfy the users' needs. Our simulation results demonstrate that it is feasible to apply the proposed ILP formulation to obtain at least a suboptimal task assignment for small as well as middle-sized wireless sensor networks (consisting of at most 100 sensors) with respect to a small amount of user requests (consisting of at most 5 measurement requirements).

CHAPTER 6. Conclusion And Future Work

6.1 Summary Of Dissertation

In this dissertation we studied the problem of energy-efficient task assignment of wireless sensor network with the application to agriculture.

The main contributions of this dissertation are summarized as follows.

1. We formulated the tasks using 3-tuple. The proposed modeling of tasks captures the user requirements on the areas and types of the physical quantities to be monitored as well as the expected number of samplings of certain physical quantities at a field. In the prior works tasks are modeled using a directed acyclic graph without considering the aforementioned user requirement factors. A directed acyclic graph representation focuses on modeling the dependency/precedence relations among the tasks. However in the applications of agriculture consist of simple, independent measurement tasks as studied in this dissertation, such directed acyclic graph based representation is not necessary since the tasks we consider are not cooperative tasks.
2. We formulated the energy-efficient task assignment problem as an instance of Integer Linear Programming problem. The objective function of the proposed ILP formulation models the overall energy consumption including measurement as well as communication energies consumed in the sensor network under a task assignment. The constraints of the proposed ILP problem model the restrictions on a task assignment resulting from the user requirements, sensor locations, and sensor measurement capabilities. For communications in the network, at each sensor node we combine as many measurement results as possible into one message so as to save the communication energy (i.e., the energy

for delivering the messages). Such combination of communications is not considered and formulated using binary variables in the prior works.

3. We designed and implemented a task assignment system that solves the task assignment problem following the proposed ILP formulation and assigns the tasks among a sensor network according to the obtained energy-efficient assignment. The proposed ILP based approach is formal and automatic. We also applied the proposed ILP formulation and task assignment system to small as well as middle-sized sensor networks. The simulation results show that the proposed ILP formulation is correct and it is feasible to apply the proposed formal and automatic approach on energy-efficient task assignment of small as well as middle-sized sensor networks with a small amount of user requests.
4. The proposed ILP formulation and task assignment approach are general enough to be applied to solve the task assignment problem of wireless as well as wired networks.

6.2 Future Work

We formulated the problem of energy-efficient task assignment of wireless sensor network as an instance of an Integer Linear Programming problem. In our objective function, we focus on the overall energy consumption among the sensor network due to a task assignment without considering the issues of energy-balance. Energy-balance is another importance issue that should be taken care of in the applications of wireless sensor networks since a sensor may be assigned too many tasks, which may cause the sensor node dead due to no power quickly. Therefore we need formulate an objective function to consider the requirements of energy-balance as well as energy efficiency. Several possible ways can be followed for such purpose: (1) Formulate an ILP with multiple objective functions, (2) Introduce weights for energy-efficiency and energy-balance concerns so as to consider the trade-off between them, (3) Set an upper bound of the workload (in terms of the number of tasks assigned or the energy to be consumed) at each sensor, etc.

The task model adopted in our work is simple, which only captures the user requirements of

location, measurement ability and the expected amount of measurements. It does not take the requirements of measurement precision, measurement time, and priority into account. Also in our work we did not consider the scheduling of the tasks at a sensor node, for example, whether a task can be finished before its deadline. New binary variable should be introduced and corresponding constraints should be formulated to capture the restrictions of these issues for task assignment of wireless sensor networks.

Moreover the designed task assignment system could be enhanced by introducing task assignment feedback. I.e., after task assignment is sent out, the system should monitor the execution of the task assignment among the sensor network and resend task assignment command/reassign tasks in case of unsuccessful assignment arising due to loss of communication or presence of faulty sensors.

In the above we discuss how to improve the proposed ILP formulation and designed task assignment system. Next we discuss the directions to extend/deepen this research. In this work we consider the sensor network deployed for the applications to agriculture, which in general does not require mobility of sensors. However in other real applications, a sensor network may consist of mobile sensors, e.g., an application of wireless sensor networks to keep track of animals. Therefore an interesting research direction would be to explore this problem in the mobile sensor network setting and explore (the possibility of) an ILP formulation for respecting sensor mobility.

Another interesting research direction would be to consider the task assignment problem among a sensor network with redundant (backup) sensor nodes. The redundant sensor nodes are introduced so that in case of sensor failure, redundant sensors could join in the network so as to maintain the quality of service of the network. A task assignment approach that could adapt the changes of wireless sensor works could be challenging.

The proposed ILP problem can be resolved using provided general ILP solvers. However since ILP is a NP-hard problem, we may not get good solutions to our specific ILP problems by using the existing general ILP solvers. Further with the increasing of the network size and the number of user requests, the proposed ILP problem gets more and more complex.

Computation efficiency becomes a big issue that can not be ignored. Therefore it is critical to design a heuristic algorithm particularly for the task assignment problem to decrease the computation complexity and running time of the optimization procedure. Also the proposed ILP formulation could be further simplified, e.g., the modeling of communication energy consumption (in our work in order to correctly compute the communication energy we enumerate all the possible combined communications, which exponentially increases with the number of physical quantities to be monitored).

ACKNOWLEDGEMENTS

I would like to thank Dr. Wensheng Zhang for his consistent guidance, patience and help throughout this research and the writing of this dissertation. His deep insights and valuable directions inspired me to keep on exploring this research and pursuing a higher standard. Without his continuous encouragement, guidance and help, I would not have completed this work.

I would also like to thank my committee members Dr. Ying Cai and Dr. Ratnesh Kumar for their efforts, valuable advices and contributions to this work. Especially, I would like to thank Dr. Kumar, who also serves as my major professor for my PhD degree at Electrical Engineering at Department of Electrical and Computer Engineering, for his understanding and support, without which I would not be able to pursue and accomplish this second degree during my Phd studies.

Deep thanks to my friends Yanfei Wang, Yang Peng and Xuejia Lu at Department of Computer Science, and Ka Yang, Xi Chen and Wei Zhou at Department of Electrical and Computer Engineering for their warm help and guidance so that I could resolve several installation and implementation issues of TinyOS. Last but not least, I express my hearty appreciation to my beloved parents and sister for their ceaseless love, concern and understanding.

BIBLIOGRAPHY

- [1] <http://www.IBM.com/CPLEX>.
- [2] <http://lpsolve.sourceforge.net/5.5>.
- [3] <http://www.mathworks.com/help/toolbox/optim/ug/bintprog.html>.
- [4] <http://www.solver.com/xlspresolv4.htm>.
- [5] <https://projects.coin-or.org/Clp>.
- [6] <http://www.ampl.com/index.html>.
- [7] <http://minisat.se>.
- [8] <http://reasoning.cs.ucla.edu/rsat>.
- [9] <http://www.sat4j.org>.
- [10] P. Aghera, D. Krishnaswamy, D. Fang, A. Coskun, and T. Rosing. Dynaheal: Dynamic energy efficient task assignment for wireless healthcare systems. In *2010 Design, Automation, Test in Europe Conference and Exhibition (DATE)*, Dresden, Germany, March 2010.
- [11] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38, 2002.
- [12] D. Anurag, S. Roy, and S. Bandyopadhyay. Agro-sense: Precision agriculture using sensor-based wireless mesh networks. In *Innovations in NGN: Future Network and Services 2008*, Geneva, May 2008.

- [13] A. Baggio. Wireless sensor networks in precision agriculture. In *First Workshop on Real-World Wireless Sensor Networks*, Stockholm, Sweden, June 2005.
- [14] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1), 2004.
- [15] A. Danak, A. Rahimi-kian, and B. Moshiri. Management of wireless sensor networks using the theory of auctions. In *ACS/IEEE International Conference On Pervasive Services*, Lyon, France, June 2006.
- [16] J. Giacomini and F. Ulaby. Wireless sensor network as a measurement tool in precision agriculture. In *XVIII IMEKO World Congress-Metrology For A Sustainable Development*, Rio de Janeiro, Brazil, September 2006.
- [17] V. Giordano, P. Ballal, F. Lewis, B. Turchiano, and J. Zhang. Supervisory control for mobile wireless sensor network: Math formulation, simulation and implementation. *IEEE Transactions on Systems, Man, Cybernetics*, 36(4), 2006.
- [18] C. Hernandez, P. Gonzalez, J. Hernandez, and L. Diaz. Wireless sensor networks and applications: A survey. *International Journal of Computer Science and Network Security*, 7(3), 2007.
- [19] H. Park and M. Srivastava. Energy-efficient task assignment framework for wireless sensor networks. In *CENS Technical Reports*, 2003.
- [20] A. Pathak and V. Prasanna. Energy-efficient task mapping for data-driven sensor network macroprogramming. *IEEE Transactions On Computers*, 59(7):995–968, 2010.
- [21] H. Qin, Z. Li, Y. Wang, X. Lu, G. Wang, and W. Zhang. An integrated network of roadside sensors and vehicles for driving safety: Concept, design and experiments. In *IEEE Conference on Pervasive Computing and Communication*, Mannheim, Germany, March 2010.

- [22] H. Sahota, R. Kumar, A. Kamal, and J. Huang. An energy-efficient wireless sensor network for precision agriculture. In *Symposium On Computers And Communications*, Riccione, Italy, August 2010.
- [23] Y. Tian, J. Boangoat, E. Ekici, and F. Ozguner. Real-time task mapping and scheduling for collaborative in-network processing in dvs-enabled wireless sensor networks. In *Proceedings 20th IEEE International Parallel and Distributed Processing Symposium*, Rhodes Island, Greece, April 2006.
- [24] Y. Tian, E. Ekici, and F. Ozguner. Energy-constrained task mapping and scheduling in wireless sensor network. In *Workshop on Resource Provisioning and Management in Sensor Networks (RPMSN05)*, November 2005.
- [25] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Hurley. Transforming agriculture through pervasive wireless sensor networks. *IEEE Computer Society*, 6(2), 2007.
- [26] N. Xu. A survey of sensor networks applications. *IEEE Communications Magazines*, 40, 2002.
- [27] Y. Yu and V. Prasanna. Energy-balanced task allocation for collaborative processing in wireless sensor networks. *Mobile Networks and Applications*, 10(1):115–131, 2005.
- [28] B. Zhao, M. Wang, Z. Shao, J. Cao, K. Chan, and J. Su. Topology-aware energy efficient task assignment for collaborative in-network processing in distributed sensor systems. In *IELP Working Conference on Distributed and Parallel Embedded System*, Milano, Italy, 2008.