

RESEARCH

Open Access



Energy-aware task allocation for energy harvesting sensor networks

Neda Edalat* and Mehul Motani

Abstract

Ambient energy harvesting is a solution to mitigate the typical finite energy supply of sensor nodes in wireless sensor networks (WSNs). On the one hand, the uncertainty of energy availability in energy harvesting systems makes network protocol design challenging. On the other hand, the fact that energy is continuously replenished opens up avenues for protocol design based on prediction of future energy arrivals. One of the key application scenarios for sensor networks is task allocation, in which a central entity allocates tasks to a set of geographically distributed sensor nodes to accomplish an overall objective. In this work, we consider a scenario in which the sensor nodes are equipped with devices capable of harvesting ambient energy, e.g., solar panels to harvest the Sun's energy, and focus on energy-aware strategies for adaptive task allocation. We decompose the static task allocation problem into two phases: scheduling of the task graph and task mapping to the appropriate sensor nodes. The solution objectives are to minimize the makespan and maximize the fairness in energy-driven task mapping (i.e., energy-balancing), while satisfying the task precedence constraints and energy harvesting causality constraints. We employ a novel energy prediction model which incorporates seasonal changes in solar energy harvesting as well as sudden weather changes. In case of an error in available energy prediction, a dynamic adaptation phase runs to avoid violation of the task allocation objectives. The performance of our proposed algorithms, in terms of energy-balancing and scheduling length, is evaluated through simulation and compared with other approaches, including a genetic algorithm as a baseline. We achieve more balanced residual energy levels across the network while attaining a near optimum scheduling length. By utilizing the dynamic adaptation phase, for certain runs of simulation, the missing ratio, which is the percentage of times in which the task allocation fails due to a temporal shortage of energy availability, is dramatically decreased.

Keywords: Energy harvesting wireless sensor networks (EH-WSNs), Task allocation algorithms, Solar-powered sensor nodes, Energy harvesting prediction

1 Introduction

A sensor network, which is a network of collaborating embedded devices with capabilities of sensing, computing, and communicating, is used to run specific applications (such as target tracking and event detection). Energy is a most precious resource in running these applications [1]. Hence, nodes will operate for a finite duration which implies a finite lifetime of the applications of interest or additional cost to regularly change batteries. An alternative technique is the use of energy harvesting as a source of powering sensor nodes. This ensures sustainable operation of such systems. However, the harvested energy is

usually not sufficient to allow the sensor nodes to stay active all the time. Moreover, the time-varying availability of environmental energy results in dynamic changes of the system's available energy. Therefore, the dynamic resource and task allocation for energy harvesting wireless sensor networks (EH-WSNs) are required, presenting a new set of problems in the area of networking and communication [2].

In wireless sensor networks (WSNs), energy-aware task allocation algorithms deal with energy availability of batteries, which typically have a monotonically decreasing energy profile. However, in EH-WSNs, due to the fluctuating energy sources, the energy availability profile is uncertain, making task allocation a challenging problem.

*Correspondence: neda@u.nus.edu

¹Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore

Generally, two design considerations for energy harvesting systems are maximizing performance and ensuring energy-neutral operation. Energy-neutral operation means operating the network such that, at all times, the energy used is less than the harvested energy. Also, while ensuring energy-neutral operation, what is the *best performance* level that can be achieved in a given harvesting environment?

In this paper, we address the task allocation problem which allocates and schedules a set of tasks represented by a task graph to a set of geographically distributed sensor nodes to achieve an overall system objective. Such a scenario is depicted in Fig. 1. We consider a scenario in which the sensor nodes are equipped with solar panels.

1.1 Related works

Many applications for energy harvesting sensor networks, such as structural health monitoring [3], disaster recovery [4], and health monitoring [5], require real-time reliable network protocols and efficient task scheduling. In such networks, it is important to dynamically schedule node and network tasks based on the remaining energy and current energy intake, as well as predictions about future energy availability.

In the literature on task allocation for WSNs [9, 12, 13], given a directed acyclic graph (DAG) and initial available resources, the set of tasks is assigned to sensor nodes based on a known static energy model. However, the time-varying nature of EH-WSNs brings new considerations for energy model and designing task allocation schemes.

There is some existing work on resource management for real-time energy harvesting embedded systems in the literature. In some works, such as [14, 15], dynamic voltage scaling policies are used to reduce the energy consumption; however, they may violate the scheduling length constraint for the case of task allocation to

the multiple sensor nodes. In [16, 17], the authors constructed lazy scheduling algorithms (LSA), which are energy-clairvoyant, i.e., the energy generated in the future is known. Lazy scheduling algorithms can be categorized as non-work conserving scheduling disciplines where a lazy scheduler may be idle although waiting tasks are ready to be processed. Moreover, there is an assumption that tasks are independent from each other and preemptive. More precisely, the currently active task may be preempted at any time and have its execution resumed later, at no additional cost. Also, since LSA is only based on a "as late as possible" heuristic, it is more likely that the battery overflows from harvested energy which results in missed-recharging opportunities. Moreover, the above-mentioned works mainly focused on the task allocation to a single processor with the energy harvesting capability but not on the allocation of a task graph with precedence constraints to multiple nodes.

The most closely related works are [10, 11]. The work in [10] proposes a dynamic energy-oriented scheduling algorithm. By conducting decomposition, combination, concurrent execution, and admission control, their proposed method allocates tasks based on the dynamically changed available energy. The work in [11] presents a distributed mission assignment scheme for wireless sensor nodes with rechargeable batteries.

Our approach differs from these works in two ways. First, the problem we address is energy harvesting aware task allocation at the network level, i.e., how to assign nodes in the network to the tasks from the task graph with precedence constraints. Such problem is quite different from the task allocation problem tackled by those works, which instead focus on individual task scheduling at the node level. Second, they maximize the network profit within a given time horizon, rather than enabling the network to operate perennially.

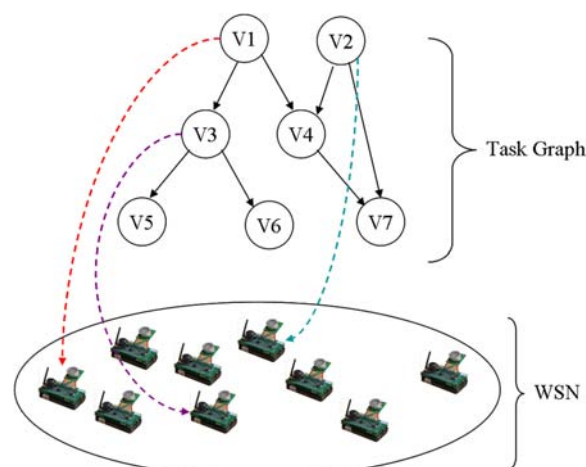


Fig. 1 An illustration of the task allocation problem. The task graph is shown, along with the mapping of tasks to nodes in the wireless sensor network

In this paper, we propose task allocation algorithms to deal with these issues. These algorithms account for the energy harvesting characteristics of sensor nodes (i.e., uncertainty in energy availability) and make the best use of available energy. The solution objectives are to minimize the scheduling length of the task graph and maximize the fairness in energy-driven task mapping, while satisfying energy harvesting causality constraints and the task precedence constraints. The fairness in task mapping means tasks with longer task lengths are assigned to the nodes with higher energy levels. The proposed solution results in the balanced energy levels among all the nodes in the network and minimum schedule length.

1.2 Contributions of this paper

In this paper, we propose the task allocation algorithms for EH-WSNs. The main contributions of this work are summarized as follows:

- We propose a novel task allocation algorithm for energy harvesting wireless sensor networks that operates in two phases: task scheduling of DAG and task mapping to the solar-powered sensor nodes. To the best of our knowledge, this is the first work on energy harvesting aware task allocation to *multiple solar-powered sensor nodes*.
- The problem is formulated in an optimization framework as an integer linear program. The proposed framework for our scheme is operated in two stages consisting of static and dynamic adaptation specialized for energy harvesting systems.
- An appropriate energy prediction model and algorithm are incorporated to increase the accuracy of our task allocation scheme.
- A genetic algorithm based multi-objective task allocation strategy has been implemented for the comparison purpose.
- The performance of our proposed algorithms in terms of the scheduling length and fairness in the

energy-driven task mapping objectives is evaluated through simulation.

This paper is structured as follows. In Section 2, we introduce the system overview considering the task and application model, prediction model, and proposed framework for our task allocation scheme in EH-WSNs. This is followed by the problem formulation in the form of integer linear programming in Section 4. In Sections 5 and 6, the energy harvesting aware task scheduling and mapping algorithms are described. The simulation results are discussed in Section 7. Section 8 provides a summary and conclusion.

2 System overview

In this section, the model for the task allocation problem in EH-WSNs is explained. Figure 2 shows the overall structure of this model. The centralized manager collects some initial data from the energy harvesting source. It then runs the energy prediction algorithms. Given the predicted energy and the task graph from WSNs application, the task allocation algorithm can be run. The output of this algorithm is the set of actions for sensor nodes. The main objective is to minimize the scheduling length of the task graph and maximize the lifetime of the sensor nodes. We assume the communication among the centralized manager and sensor nodes is done in most energy-efficient way such that this communication can be ignored.

We consider a discrete time model. The network consists of multiple solar-powered sensor nodes denoted as $n \in \mathcal{N}$ where \mathcal{N} is a set of all available sensor nodes. Tasks are units of execution that make up an application. The application for wireless sensor networks can be target tracking, event detection, etc. A single application may perform a variety of tasks such as sensing, computation, storage, and communication. Applications are represented by a DAG, composed of all tasks $m \in \mathcal{M}$ where \mathcal{M} is the set of tasks that must be completed for

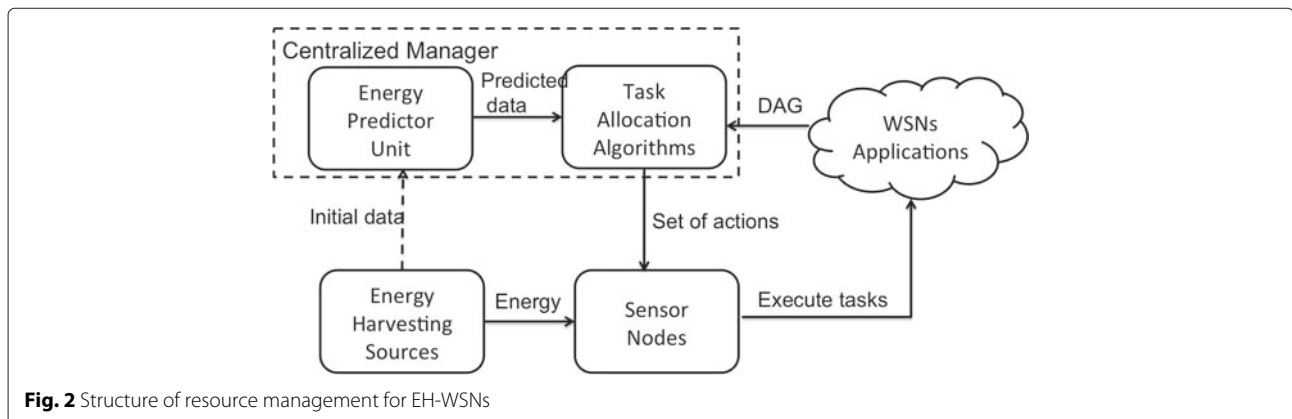


Fig. 2 Structure of resource management for EH-WSNs

the application. L_m denotes the size of the task m which refers to the number of time slots required to accomplish that task ($L_m \in \mathbb{Z}^+$). Tasks must be sequentially executed to satisfy the precedence constraints. A task has one or more inputs and once all inputs are available, the task is triggered to execute.

The harvesting period is divided into equal-length time slots which are indexed as $t \in \mathcal{T}$ where $\mathcal{T} \in \mathbb{Z}^+$. We consider an energy storage device with capacity B and discharge efficiency $\eta_d < 1$. \mathcal{H}_t^n is the amount of harvested energy in slot t in node n . The available energy at each node n and time slot t is denoted as \mathcal{A}_t^n and calculated as

$$\mathcal{A}_{t+1}^n = [\mathcal{A}_t^n + \mathcal{H}_t^n - \eta_d \cdot E_0 \cdot Y_t^{m,n}]_0^B,$$

where E_0 denotes the energy which is drained from the battery during each busy slot. E_0 is assumed to be constant for each node and task. $Y_t^{m,n}$ is a boolean function indicating whether node n is busy with task m in time slot t . The available energy is bounded by the capacity of battery B .

2.1 Framework of task allocation in EH-WSNs

In this part, the framework of our proposed task allocation scheme for EH-WSNs is described. It is a hybrid framework of static and dynamic stages.

The static stage is executed *offline* in the centralized manager which can be a node with the high computational capability such as Stargate (<http://platformx.sourceforge.net/home.html>). In this stage, the centralized manager runs the prediction algorithm to extrapolate information about the harvested energy in the next harvesting period. It then allocates an energy budget in terms of the harvesting rate for each time slot of the next period. The predicted information and the task graph are the inputs to our static allocation algorithm. The allocation outcomes and energy budget for each slots are then communicated to sensor nodes.

The dynamic stage is executed *online* in the sensor nodes. This stage may modify the real-time execution of the task allocations from the previous stage to adapt to unpredictable fluctuation in energy availability. At the beginning of each time slot, the corresponding energy budget is retrieved from the memory and is compared with the corresponding baseline error margin. If the energy budgets from the prediction do not match with actual harvested energy and the error margin is higher than the predefined threshold, then the dynamic adaptation method is executed. Adaptation allows the system to cope with variation in renewable energy source and maintain sustainable operation. Moreover, energy harvesting statistics are sent to the centralized manager to update the data for the energy harvesting prediction for the next harvesting period.

3 Prediction algorithm

In this section, the energy harvesting prediction method is described. An accurate prediction of the near-future harvested energy has a critical effect on decision-making for task allocation procedure in energy harvesting systems. A time series [22] is a sequence of observations of a random variable such as energy harvesting rate. A time series analysis provides a proper tool for forecasting future events. There are several methods for forecasting the near future of real data such as regression analysis, exponential smoothing, and moving average [22]. Weather-conditioned moving average (WCMA) is the common approach adopted for the weather prediction. It first introduced in [19] and then extended for cloud cover case and wind energy prediction in [20, 21]. Our prediction model called autoregressive weather-conditioned moving average (AR-WCMA) has its foundation on WCMA which is a low overhead solar energy prediction algorithm. The key idea of our prediction model is to apply an autoregressive (AR) time series model at the beginning of each day, since the sun irradiance is a periodic phenomena. It has been proven in [23] that a simple linear regression model is an appropriate model for real-time and random data. Then, the model is further improved by utilizing the moving average of the information from past days.

For this prediction algorithm, a day is divided into T equal duration of time slots. The power sampling and prediction are performed once per slot. To predict the harvesting energy in the next slot, the algorithm uses the values of power measurement $\tilde{e}(j) \in \tilde{E}_T$ of the current day. It also utilizes the values of power measurement $\tilde{e}(i, j) \in \tilde{E}_{D \times T}$ of the last D days, $D \in \mathbb{Z}^+$. The unit of power measurements and accordingly the prediction values are irradiance (W/m^2). The matrix $\tilde{E}_{D \times T}$ and the vector \tilde{E}_T are shown in Fig. 3. Assume that t slots have elapsed on the current day shown shaded in Fig. 3 and $\tilde{e}(t+1)$ is the estimation of the power value at the start of slot $t+1$.

The time series analysis begins with some response measurements of the structure at a particular sensor location. Assuming the response to be stationary, an AR process model is used to fit the discrete measurement data to a set of linear coefficients from past time history observations as follows:

$$\tilde{e}(t+1) = \sum_{i=1}^k b_i \tilde{e}(t+1-i) + r_k. \quad (1)$$

The response of the structure at sample index t , as denoted by $\tilde{e}(t)$, is a function of k previous observations of the system response, plus a residual constant term r_k . Weights of the previous observations $\tilde{e}(t+1-i)$ are denoted by the b_i coefficients which are calculated daily based on the available information of last D days.

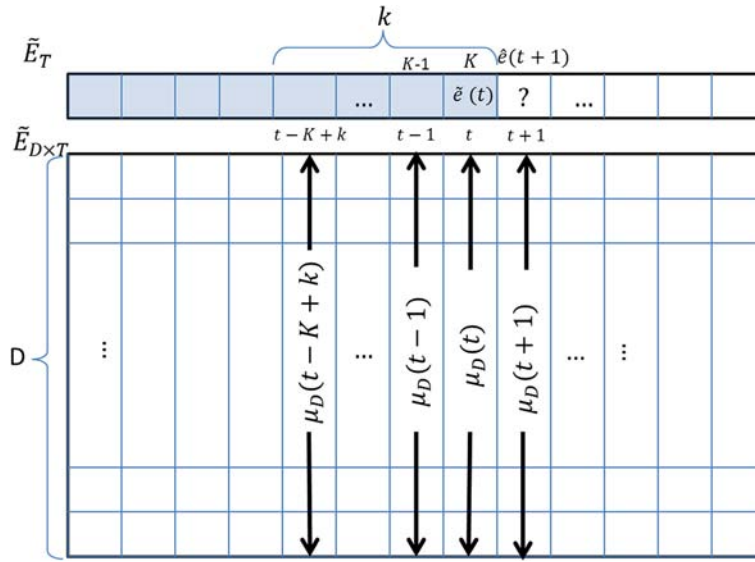


Fig. 3 Prediction framework for solar-powered sensor nodes

The goal is to predict the power $\hat{e}(t+1)$ at the beginning of slot $t+1$ (marked with a \hat{e} in Fig. 3). In Fig. 3, $\mu(t+1)$ represents average of power measured at beginning of all $t+1$ slots for last D days. The predicted power for the next slot considers the power measurement from the present information of last k slots $\tilde{e}(t+1)$ and the average power $\mu(t+1)$ of the same slot ($t+1$) for past D days:

$$\hat{e}(t+1) = \alpha \cdot \tilde{e}(t+1) + (1 - \alpha) \cdot \mu_D(t+1) \Phi_K. \quad (2)$$

The prediction algorithm consists of two terms as shown in (2). The first term is the effect of the AR, and the latter one is the conditioned average term. The AR term determines the effect of the current day to the predicted value, while the conditioned average term is the average of past D days on time slot ($t+1$) which scaled by the conditioning factor Φ_K . The parameter α shows the trade-off between these two terms ($0 \leq \alpha \leq 1$).

In (2), $\mu_D(j)$ is the average of power measured at beginning of slots j in the past D days:

$$\mu_D(j) = \frac{\sum_{i=1}^D e(i, j)}{D} \quad (3)$$

Φ_K is a conditioning factor for $\mu(t+1)$, and it is a function of previous K ($K \in \mathbb{Z}^+$) time slots before time slot ($t+1$) of current day (can be seen in Fig. 3). Φ_K shows how much brighter or cloudy the current day is compared to previous days. It is calculated using (4), which is a weighted average of ratio $\eta(K) \in H_k$, (shown in (5)), where each ratio $\eta(k)$ compares the current day measured power of one particular slot to the average of the past days. Since the slots earlier than t are assumed to be less correlated to the future slot $t+1$, the weights $\theta(k) \in \Theta$ calculated in (6) decrease from 1 to $1/K$ starting at the slot t .

$$\Phi_K = \frac{(\Theta_K)^T \cdot H_K}{\sum_{k=1}^K \theta(k)} \quad (4)$$

$$\eta(k) = \frac{\tilde{e}(t-K+k)}{\mu_D(t-K+k)} \quad (5)$$

$$\theta(k) = k/K \quad (6)$$

The estimated value $\hat{e}(t)$ in following sections are denoted as \mathcal{H}_t . All measurements and estimations are in irradiance unit (W/m^2) which can be converted to the energy unit Ah . The details are explained in the simulation section.

4 Problem formulation

In this section, we explain the problem formulation for the task allocation in sensor networks equipped with energy harvesting devices. Task allocation problem which consists of task scheduling and mapping plays an essential role in parallel processing. Solving this problem results in execution sequence of tasks and assignment of tasks to the sensor nodes. The main objectives are to minimize the makespan and maximize the fairness (energy-balancing), while satisfying the task precedence constraints and energy harvesting causality constraints.

4.1 Decision variables

The decision variable for this problem is defined as follows:

X : valid mapping and scheduling of tasks $m \in \mathcal{M}$ to the sensor nodes $n \in \mathcal{N}$ at time slot $t \in \mathcal{T}$:

$$X_t^{m,n} = \begin{cases} 1, & \text{if task } m \in \mathcal{M} \text{ is assigned to sensor} \\ & \text{node } n \in \mathcal{N} \text{ at time slot } t \in \mathcal{T} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Although, we have one decision variable, it is useful to define the following quantities, which depend upon X . S^m : start time slot of scheduled task $m \in \mathcal{M}$. It can be calculated as follows:

$$S_m = \operatorname{argmax}_t \left(\max_n X_t^{m,n} \right) \quad (8)$$

Y : an indicator function to show the busy time slot.

$$Y_t^{m,n} = \begin{cases} 1, & \text{if node } n \text{ is busy with task } m \\ & \text{at time slot } t \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

This value can be calculated as a function of S_m and $X_t^{m,n}$ as follows:

$$Y_t^{m,n} = \begin{cases} 1, & t = S_m, S_m + 1, \dots, (S_m + L_m); \\ & \forall n = \operatorname{argmax}_m \max_t X_t^{m,n} \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

4.2 Objective function and constraints

The following objective function and linear-integer constraints describe the optimization problem to be solved in order to compute an optimal task allocation.

$$\min_X w_1 \left(\max_{m \in \mathcal{M}} (S_m + L_m) \right) + w_2 \sum_{\substack{t \in \mathcal{T} \\ n \in \mathcal{N} \\ m \in \mathcal{M}}} g_t^{m,n} X_t^{m,n} \quad (11)$$

Subject to:

$$S_k \geq S_j + L_j + T_c, \quad \forall (j, k) \in \mathcal{M}, P_{j,k} > 0. \quad (12)$$

$$X_t^{m,n} \in \{0, 1\} \quad \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, \forall t \in \mathcal{T}. \quad (13)$$

$$\sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} X_t^{m,n} = 1, \quad \forall m \in \mathcal{M}. \quad (14)$$

$$\sum_{m \in \mathcal{M}} Y_t^{m,n} \leq 1, \quad \forall n \in \mathcal{N} \quad \forall t \in \mathcal{T}. \quad (15)$$

$$\mathcal{A}_{t+1}^n = [\mathcal{A}_t^n + \mathcal{H}_t^n - \eta_d \cdot E_0 \cdot Y_t^{m,n}]_0^B, \quad \forall n \in \mathcal{N}, \forall t \in \mathcal{T}. \quad (16)$$

where T_c is the communication cost between immediate tasks precedence. $g_t^{m,n}$ is the cost associated with assigning task $m \in \mathcal{M}$ to sensor node $n \in \mathcal{N}$, at time slot $t \in \mathcal{T}$. This cost value is a linear function of \mathcal{A}_t^n and task size (L_m). The cost function is defined as follows:

$$g_t^{m,n} = L_m \times (\mathcal{A}_t^n)^{-1}. \quad (17)$$

The objective function shows the trade-off between the two task allocation objectives with weights w_1 and w_2 . The first part corresponds to minimizing the scheduling

length by minimizing the maximum of the finish times of all the tasks. The second part relates to minimizing the task mapping cost in order to achieve a balanced energy availability through the network.

The constraint in (12) states the precedence constraint where a task cannot start until its predecessors are completed, and data has been communicated to it if the preceding job were executed on different node, illustrated as a constant value T_c . As stated before, L_m is the size of task m , which refers to the number of time slots required to accomplish that task.

The constraint in (13) refers to boolean task mapping constraint. (14) and (15) show task allocation constraints. More precisely, (14) specifies that each task must be assigned once to exactly one node and (15) specifies that at most one task can be assigned to each sensor node at each time slot. The constraint shown in (16) indicates the energy harvesting causality constraint. This constraint keeps track of energy availability at each time. Moreover, this constraint avoids the energy harvesting overflow at each node by considering the capacity of the battery (B) as an upper bound for energy availability.

Altogether, the task allocation problem for wireless sensor networks equipped with energy harvesting systems is described in the abovementioned objective function and constraints. This problem is NP-complete in general [24], and heuristic algorithms are applied to obtain a practical solution. Our proposed heuristic approach is presented in the next section.

5 Proposed algorithm for task allocation in EH-WSNs

This section describes our solution approach for task allocation. The proposed scheme consists of a hybrid framework of static and dynamic stages. The static task allocation problem is divided into two phases. The first one is the task scheduling phase to determine the proper sequence of tasks. The main objective for the first phase is to minimize the scheduling length (shown in the first part of the objective function) and also satisfy the precedence constraint (12). This phase, which is explained in Section 5.1, results in the lower and upper bounds for the starting time of each task.

Given the bounds from the first phase, the second phase, explained in Section 5.2, runs the task mapping to the appropriate sensor nodes. The objective of the second phase is to maximize the energy-balancing among the nodes, considering the energy harvesting characteristics. It addresses the second part of the objective function as well as the constraints (13) to (16).

If the error margin, which is the difference between the energy predicted value and the actual harvested energy, is higher than some predefined thresholds, the dynamic

adaptation stage (Section 5.3) is executed. Adaptation allows the system to cope with variation in renewable energy supply and maintain the system's sustainability.

5.1 List scheduling and critical nodes path tree (CNPT) Heuristic

This part presents the list scheduling algorithm. It uses to satisfy the first constraint (12) while optimizing the first part of the objective function. It computes a task sequence provided by a DAG to obtain the earliest start time (EST) and the latest start time (LST) of each task considering a critical path (CP). A CP of a task graph is defined as a path with the maximum sum of node and edge weights from an entry node to an exit node. Given these values, the tasks are queued into a list. The EST and LST for a task i can be computed recursively by traversing the DAG downward from the entry node and upward from the exit node respectively as follows:

$$EST_m = \max_{i \in \text{pred}(m)} EST_i + L_i \quad (18)$$

$$LST_m = \min_{i \in \text{succ}(m)} LST_i - L_i \quad (19)$$

where $\text{pred}(m)$ and $\text{succ}(m)$ are the set of immediate predecessors and successors of m respectively and L_i is the length of the task i . After the listing phase, the task graph is sequentialized into a queue and ready for the task assignment phase. The main objective of this part of the scheduling process is to determine the minimum scheduling length (makespan) while satisfying all precedence constraints. The tasks are queued for assignment to sensor nodes based on LST. This is because it may give a sensor node the chance to harvest more energy before the task is assigned to it.

5.2 Energy harvesting aware task assignment heuristic

In this phase, the proposed heuristic algorithm assigns the tasks to maximize the energy-balancing considering the energy harvesting characteristics. It satisfies the energy-neutral constraint and avoids the energy overflows.

The following steps describe the basic idea of our proposed energy harvesting aware task allocation heuristic:

- Step 1. Sort all the tasks based on LST from List Scheduling and CNPT in queue Q
- Step 2. Update the energy availability \mathcal{A}_t^n of all nodes based on the predicted energy for the current slot
- Step 3. For the task dequeued from queue Q , calculate the cost associated with assigning the correspond task m to all sensor nodes $n \in \mathcal{N}$ at the current time slot t as

$$C_t^{m,n} = g_t^{m,n} a_t^n$$

where $g_t^{m,n}$ can be calculated from (3.10). a_t^n is an indicator defined as

$$a_t^n = \begin{cases} 0, & \mathcal{A}_t^n \geq B \forall \text{ node } n \in \mathcal{N} \text{ at time slot } t \in \mathcal{T} \\ 1, & \text{otherwise.} \end{cases} \quad (20)$$

This indicator shows that the cost is zero if the overflow occurs at the node. Hence, we immediately allocate the first task from the queue to that node.

As a result, since the listing in the queue is based on LST , it increases the chance that the node can harvest enough energy and gains the failure tolerance. In the other word, the task scheduling is *as late as possible* but within a boundary that results in the minimum schedule length. Since the task mapping is based on the available energy, in \square STEP 2 \square the available energy for each slot is updated. \square STEP 3 \square assigns the task based on the cost associated with each node for accomplishing the corresponding task and handles the overflow situation when the harvested energy may become greater than the maximum capacity. The proposed algorithm is shown in *Algorithm 1*.

Algorithm 1: Energy Harvesting Aware Task Allocation

Given: EST and LST of all the tasks $m \in \mathcal{M}$ from CNPT, $\mathcal{A}_t^n \forall n \in \mathcal{N}$ and $t \in \mathcal{T}, L_m \forall m \in \mathcal{M}$.

Sort all the tasks in respect to LST in queue Q .

For all tasks m dequeued from Q ,

For all $n \in \mathcal{N}$,

For $EST_m < t < LST_m$,

If $\mathcal{A}_t^n \geq B$,

Assign task m to sensor node n ,

$S_m = t$,

Else

$g_t^{n,m} = L_m \times (\mathcal{A}_t^n)^{-1}$

EndIf

EndFor

EndFor

$n^* = \arg \min_{t,n} (g_t^{n,m})$ (finding the best node)

$S_m = \arg \min_{n,t} (g_t^{n,m})$ (finding the best schedule)

EndFor

We briefly explain a potential extension of the current model to multi-hop scenario. In this case, a task i and its immediate predecessor task j might be allocated to nodes that are in one or more hops away of each other. Assume that j needs the data from i for execution. Hence, one needs to find the way to communicate this data through multi-hop. In WSNs, the communication is broadcast in nature. To represent the broadcast feature of wireless communication, the DAG representation of application is extended to Hyper-DAG. The weight of edges between two computation tasks is represented as a separate communication task. Unlike in single-hop

networks, there might be multiple simultaneous communications in multi-hop networks. In order to schedule the communication task, one models multi-hop channel as a virtual node on which only the communication task can be scheduled. To avoid the interference between simultaneous communication tasks, one may consider the penalty of infinity cost for the case of interference and zero otherwise. In case that the source and destination of a communication task are one or more hop away from each other, one can schedule based on the path generated by a low complexity routing algorithm [25].

5.3 Online dynamic adaptation stage

This stage is based on the real-time execution of these tasks allocated to the sensor nodes based on the actual energy harvesting data. At the beginning of each time slot, the corresponding energy budget is retrieved from memory and a corresponding baseline error margin is looked up. If the energy budget from the prediction does not match with actual harvested energy and the error margin is higher than predefined threshold, the dynamic adaptation phase is executed. Adaptation allows the system to cope with the variation in the renewable energy and maintain system sustainability. In addition, energy harvesting statistics with new updates are sent to the centralized manager to update energy harvesting prediction of the next harvesting period.

The basic idea is that if a sensor node due to an environmental uncertainty does not have enough energy to execute the task, then it slows down the processor rate and stretches the execution time of that task. As a result, one may gain more time for the sensor node to harvest energy from environment at the price of operating at the lower processor power. This technique is called dynamic voltage and frequency scaling (DVFS)[6]. Based on the initial schedule, all tasks are executed at the full speed of the processor (P_{\max}), which is not an energy-efficient scheme. We need to make use of the task slacks for the energy saving. DVFS is applied to stretch the execution time of each task and slow down the power of processor. Then, we shift all other potentially affected tasks accordingly. Assume the DVFS-enabled processor has K discrete operating frequencies $f_k : \{f_k | 1 \leq k \leq K, f_{\min} = f_1 < f_2 < \dots < f_K = f_{\max}\}$ and the power consumption with regards to f_k is denoted as P_k . We assume a linear relationship between the power P_k used for executing tasks and their execution time L . We can say the higher the power P_k , the shorter the execution time L .

We define a decelerate factor v_k as the normalized frequency of f_k with respect to the maximum frequency f_{\max} , that is:

$$v_k = f_k / f_{\max} \quad (21)$$

If task m is stretched by a slowdown factor v_k , then its actual execution time at frequency f_k is L_m / v_k .

Assume that initially the task m is assigned to the sensor node n considering the full speed of the processor. Let st_m and ft_m denote the starting time and finish time of the task m accordingly. The task m has enough energy to finish its execution if the following inequality holds:

$$A(st_m) + \mathcal{H}(st_m, ft_m) \geq \mathcal{D}(st_m, ft_m) \quad (22)$$

where $\mathcal{C}(st_m)$ is the energy available at time instant st_m and $\mathcal{H}(st_m, ft_m)$ is the harvested energy during task execution and $\mathcal{D}(st_m, ft_m)$ is the energy demand of executing task. If this inequality does not hold, the decelerate factor is adjusted until the following equality holds,

$$A(st_m) + \mathcal{H}(st_m, st_m + \frac{ft_m - st_m}{v_n}) = \mathcal{D}(st_m, st_m + \frac{ft_m - st_m}{v_n}). \quad (23)$$

Subsequently, all of the tasks $\tau \in \text{succ}(m)$ need to be delayed dl_m which is the delay caused by stretching the task via DVFS. The result of this dynamic adaptation phase is to reserve time for the sensor node with lack of energy to harvest from the environment and execute with lower CPU frequency. Note that the schedule length in our formulation is not a hard constraint (or deadline). In the dynamic adaptation phase, the task is missed only if extending the task length to meet the energy-neutral condition causes the frequency f_k to be below f_{\min} . The sequence of the algorithm is shown in *Algorithm 2*.

Algorithm 2: Dynamic Adaptation Phase

Require: initial schedule, actual harvesting rate, actual energy available;

For task $m \in M'$ which M' is a set of tasks mapped to sensor node $n \in N$,

If $A(st_m) + \mathcal{H}(st_m, ft_m) < \mathcal{D}(st_m, ft_m)$,

Calculate v_n from (23);

For all the task $\tau \in \text{succ}(m)$,

$st_m \rightarrow st_m + dl_m$;

EndFor

EndIf

EndFor

5.4 Computational complexity

In this part, the complexity of the proposed heuristics is calculated. Considering an application with m tasks, a network with n sensors and ε which is defined as $\varepsilon = \max_{m \in \mathcal{M}} (LST_m - EST_m)$ the complexity of the listing stage is $O(m)$ and the energy harvesting aware task assignment heuristic has the complexity of $O(n m \varepsilon)$ for the worst case. Hence, worst-case total complexity is $O(m) + O(n m \varepsilon) \simeq O(n m \varepsilon)$. The dynamic adaptation phase has a low complexity of $O(k l)$ where k is the number of tasks mapped to the sensor node which has the energy

prediction error higher than predefined threshold and l is the number of those tasks's successors in the DAG.

6 Multi-objective genetic algorithm for task allocation

The task scheduling and mapping into a parallel and distributed computing system is a well-defined NP-complete problem. It is considered as one of the most challenging problems in parallel computing [29]. The task allocation problem is included in this class of combinatorial optimization problems.

Genetic algorithms (GAs) have been widely used as beneficial meta-heuristics for obtaining high-quality solutions for a broad range of combinatorial optimization problems including the task allocation problem [30]. Another distinct feature of the genetic search is that its inherent parallelism can be exploited to further reduce its running time. Hence, as a baseline, with which to compare our task allocation scheme, we implement a multi-objective genetic algorithm. This algorithm starts with an initial population of feasible solutions. Then, by applying some operators, the best solution can be found after some generations. The selection of the best solution is determined according to the value of the fitness function. In this section, the detailed implementation is presented.

Representation: The first step in designing a genetic algorithm for a particular problem is to develop a suitable representation scheme, i.e., a way to represent individuals in the GA population. A chromosome represents a mapping of tasks to sensor nodes. Figure 4 shows an example of such a representation of the chromosome. Tasks 2, 3, 6, and 9 will be scheduled on sensor node 1, tasks 5 and 8 on sensor node 2, and tasks 1, 4, and 7 on sensor node 3.

Fitness function: A fitness function attaches a value to each chromosome in the population, which indicates the quality of the schedule. The main objectives of the task allocation, which are minimizing scheduling length and maximizing energy-balancing considering the energy harvesting, are represented in the fitness function. In this case, the fitness function (\mathfrak{F}) needs to express two different objectives as follows:

$$\mathfrak{F} = w_1 \left(\max_{m \in M} (S_m + L_m) \right) + w_2 \left(\frac{1}{n} \sum_{i=1}^n (\mathcal{A}_i - \bar{\mathcal{A}})^2 \right) \tag{24}$$

where $S_m + L_m$ is the finish time of task m and $\max_{m \in M} (S_m + L_m)$ is the schedule length of a task graph. \mathcal{A}_i is the available energy level of sensor node i considering the recharging with the energy harvester. $\bar{\mathcal{A}}$ is the mean value which is calculated as $\bar{\mathcal{A}} = \frac{1}{n} \sum_{i=1}^n \mathcal{A}_i$ where n is the number of sensor nodes. In order to show how balanced the energy levels are, the deviation from the mean of the remaining energy level of all the sensor nodes is evaluated. The lower this value is, the more balanced available energy level among all the sensor nodes. Also, the lower scheduling length is better. So, our objective is to minimize the defined fitness function which is proportional to the schedule length and the energy variance. w_1 and w_2 are the weights to set the priority for the objectives.

Selection operator: There are two selection phases used for the genetic algorithm. The first one is *parent selection*. This selection phase is used to select the parents for mutation and crossover based on the fitness. We have applied the fitness-proportional roulette wheel selection and tournament selection. The second selection phase is survival selection among the reproductive chromosome

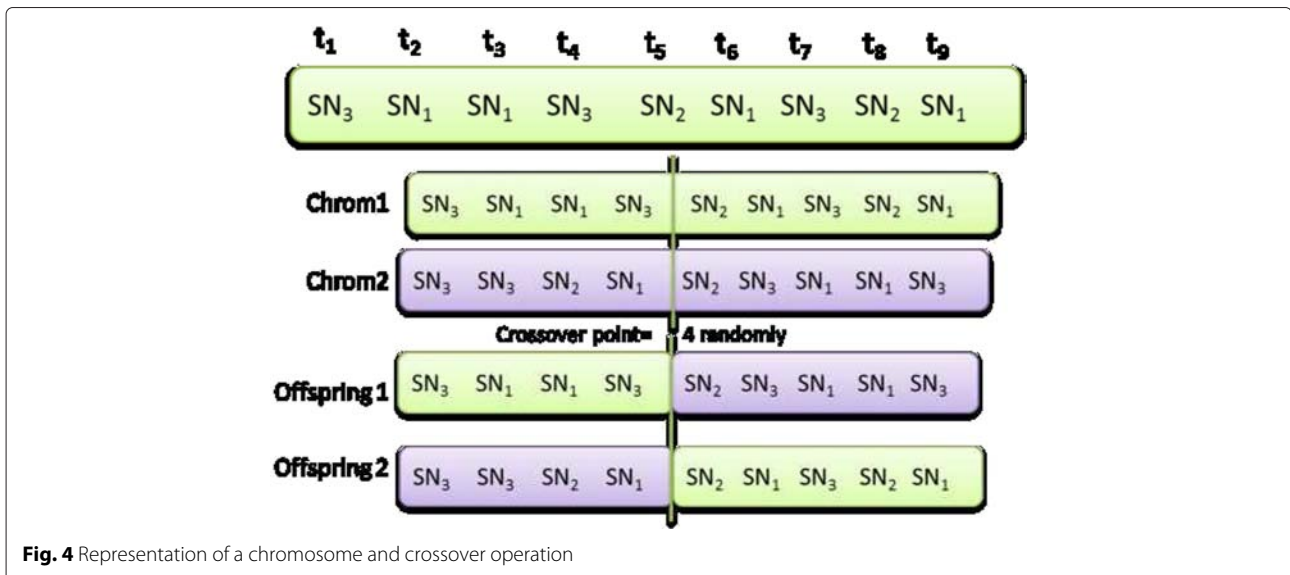


Fig. 4 Representation of a chromosome and crossover operation

after crossover. For this phase, we use the Genitor selection, a.k.a. delete worst, which means among parents and offsprings deleting the worst.

Crossover and mutation operator: Each chromosome in the population is subjected to crossover with probability P_c . Two chromosomes are selected from the population, and a random number $r \in [0, 1]$ is generated for each chromosome. If $r < P_c$, these chromosomes are subjected to the crossover operation using single point crossover as shown in Fig. 5. For this work, the *non-uniform self-adaptation mutation*, which is a fitness-dependent mutation rate, is utilized. The mutation rate initially set to be some high values then the lower value once it reaches to near to optimal fitness.

7 Simulation results

The performance of our proposed algorithms are evaluated through simulation. We have run several sets of simulations to investigate the following aspects:

- Performance of the energy harvesting prediction algorithm
- Performance of the task allocation scheme in terms of energy balancing and scheduling length objectives
- Performance of the dynamic adaptation phase

To the best of our knowledge, this is the first work that addresses an energy harvesting aware task allocation at the network level. For small scale problems, we are able to solve mixed-integer linear programming (MILP) by the optimization toolbox in Matlab using syntax `intlinprog`. The results can be used as a baseline for our proposed heuristic. A multi-objective genetic algorithm based task allocation scheme (explained in details in Section 6) is also implemented. For large scale

problems, the proposed scheme is validated by comparing the results of the genetic algorithm approach and some modified heuristics such as the critical node path tree (CNPT) algorithm [18] and extended CNPT (E-CNPT) algorithm [13]. The strategy for extended CNPT (E-CNPT) algorithm in [13] is to assign the tasks along the most critical path first to the nodes with earliest execution start times. This algorithm operates by adjusting the number of sensors in each scheduling iteration and then choosing the schedule with the minimum energy consumption.

The energy harvesting profile is retrieved from the National Renewable Energy Lab website [26]. The data for the power consumption of a sensor node is extracted from the Mica-2 data sheet [27]. Some of the implementation specifications are shown in Table 1.

The example of a task graph and its corresponding task allocation to five sensor nodes are shown in Fig. 5. The results from the convergence of a genetic algorithm based multi-objective fitness function for the minimum and average value after several generations are shown in the Fig. 6.

7.1 Simulation results for the energy harvesting prediction algorithm

To evaluate the performance of the proposed algorithm, simulations are carried out based on the available solar irradiance data from the solar energy received at different times of day and night. We consider the measurement trace of a solar panel in four consecutive days, with both the sunny and cloudy conditions from National Renewable Energy Lab [26]. The reported data is from California Solar Initiative (CSI) from 26 October 2011 for 4 days. The unit of this data is an average solar irradiance (mW/m^2), and it is for every 1 h. An irradiance unit is converted to an energy unit by linear conversion considering a solar panel

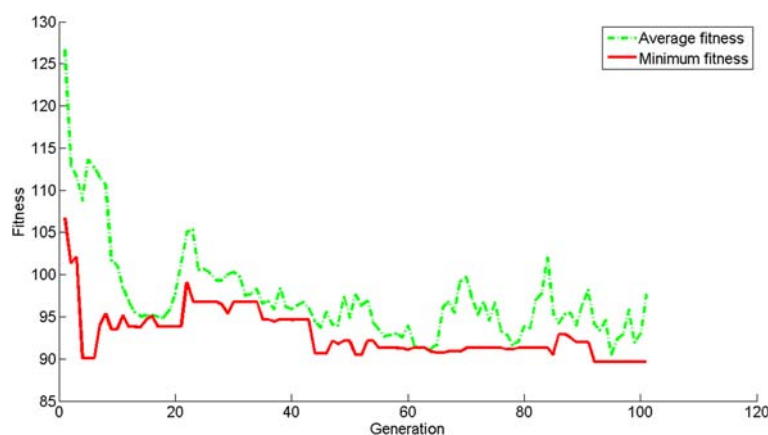


Fig. 5 Convergence of multi-objective fitness function of genetic algorithm-based task allocation

Table 1 Simulation configuration

Recombination	Single point crossover with probability 0.6 – 0.9
Parent selection	a) roulette wheel b) binary tournament
Survivor selection	GENITOR, a.k.a. \times delete worst \times
Population size	10 chromosomes
Number of task in the task graph	7 – 30
Number of sensor nodes	2 – 50
Maximum number of predecessors in task graph	2 or 3
Stopping criterion for genetic algorithm	10 unchanging generations

size 9.6×6.4 cm, a solar cell efficiency 10% and a harvesting efficiency 80%. We use mAh as the unit of the energy quantity because the voltage is fixed at 1.2 V.

To present the algorithm evaluation results, the following values used as algorithm parameters are as follows: $T = 24$, $\alpha = 0.7$, $D = 4$, and $K = 4$. Recall that α , K , and D denote weighing factor, number of previous slots,

and number of previous days for each solar power data set. Figure 7 shows how closely our prediction algorithm can track the actual measured data. The evaluation metric used for this part is the average prediction error. It is computed as

$$\epsilon = \frac{100}{n} \sum_{i=1}^n \frac{|e(i) - \tilde{e}(i)|}{e(i)}$$

where n is the time horizon of the predication which is $D \times T$.

We compare the energy prediction accuracy of WCMA, ARMA, and AR-WCMA. WCMA and AR-WCMA are addressed in the prediction algorithm part. ARMA refers to autoregressive moving average. This comparison can be seen in Fig. 8 for the different weather conditions (i.e., sunny day and cloudy day).

ARMA only uses values from the previous day; hence, if the weather conditions change from one slot to another, this method has a large error in prediction (i.e., close to 30%). On the other hand, WCMA only takes into account the last few slots but not the useful information about the weather condition of the past few days. AR-WCMA produces much better results because it uses the values of the same slot over previous days and also the previous slots of the same day, which helps to calibrate against the actual weather condition. To conclude, WCMA, ARMA, and AR-WCMA results in the average error of 23.6, 28.6 and 8.7%, respectively.

7.2 Discussion on results for energy-balancing and scheduling length objectives

In this section, the performance evaluation of our scheme in terms of our objectives which are minimizing the scheduling length and maximizing the energy-balancing considering energy harvesting characteristics are presented. The quantity that used for energy-balancing

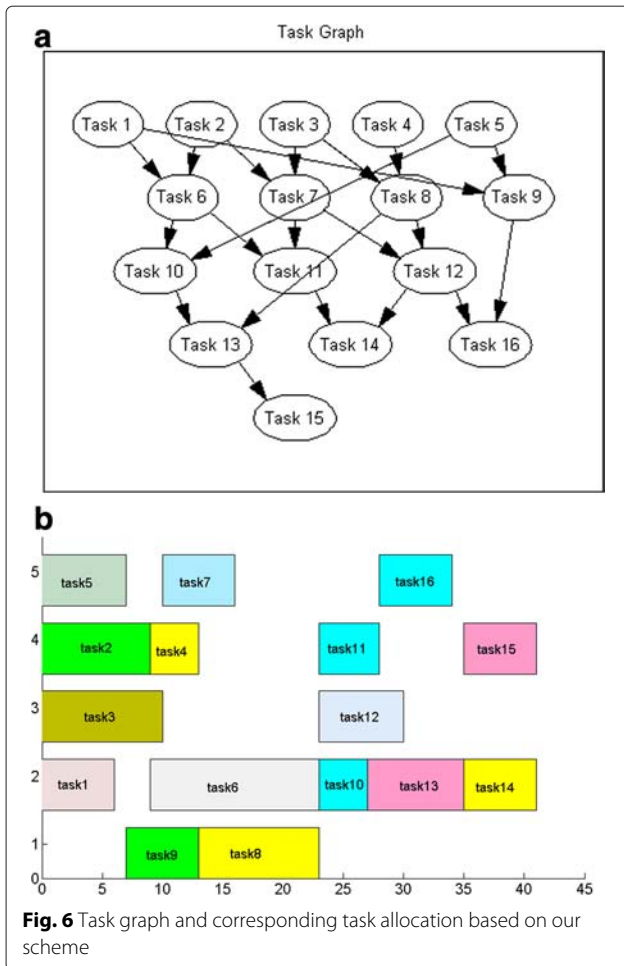


Fig. 6 Task graph and corresponding task allocation based on our scheme

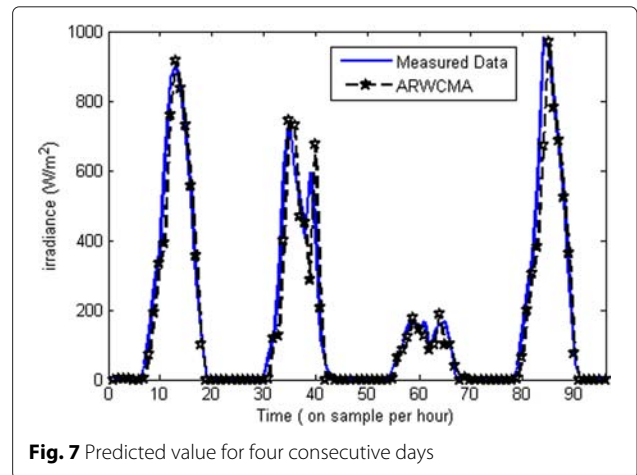
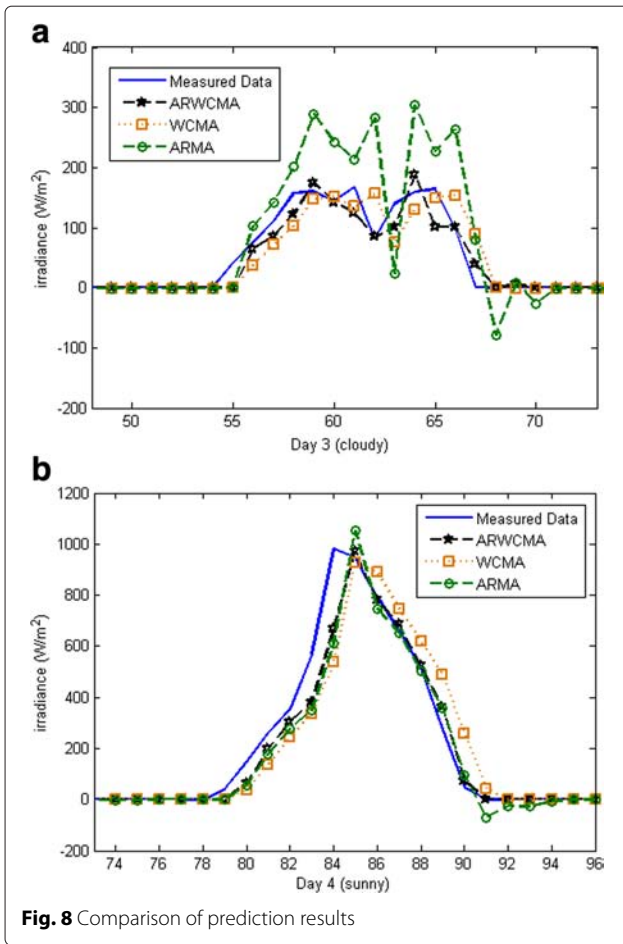


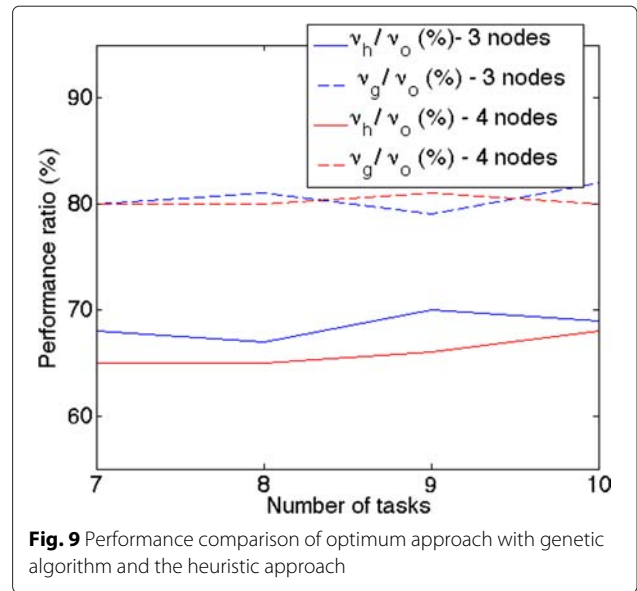
Fig. 7 Predicted value for four consecutive days



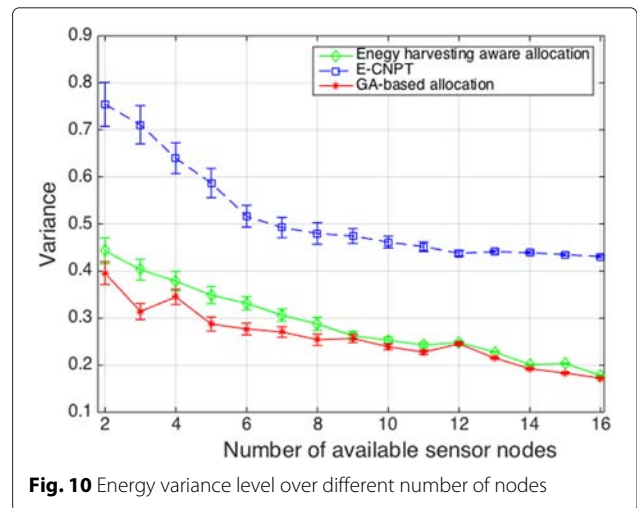
evaluation is the energy variance over all nodes which calculated as follows:

$$v = \frac{1}{n} \sum_{i=1}^n (\mathcal{A}_i - \bar{\mathcal{A}})^2$$

where \mathcal{A}_i is the energy availability level of sensor node i after the allocation and $\bar{\mathcal{A}}$ is the mean value. We first conducted simulations for small scale problems, with 3×4 sensor nodes and 7×10 tasks with maximum 2 predecessors. The results are averaged over more than 100 instances where each point has a 95 % confidence interval with a 10 % precision. The optimization toolbox in Matlab is used that calls the solver by syntax `intlinprog` to solve MILP for this small scale problem. The performance ratios of our proposed heuristic (v_h) and genetic algorithm (v_g) over the optimum solution (v_o) are shown in Fig. 9. We can see that the heuristic achieved up to 70 % of the solution obtained by the MILP-based approach for the conducted simulations. Moreover, the results from the genetic algorithm are good enough (up to 82 %) that can be later used as a baseline for the larger scale problem.



We then consider the simulations for larger scale problems. Task graph with 30 tasks and the maximum 3 predecessors are generated randomly. Similarly, the results shown in Fig. 10 are averaged over more than 100 instances so that each point has a 95 % confidence interval with a 10 % precision. This result shows the variance of energy level over 2 to 16 nodes. The lower energy variance means achieving the more balanced energy level among the nodes. This result shows that the energy level of nodes using our scheme is more balanced than the E-CNPT algorithm in [13]. This is because although the E-CNPT is an energy-efficient algorithm which schedules based on the the energy consumption of tasks, it considers constant energy profiles for nodes and it does not consider the energy variation of the harvested energy. The energy-balancing results from the GA-based scheme with



optimum result and our heuristic are fairly close to each other.

In Fig. 11, the scheduling length resulted from our scheme is compared with the CNPT algorithm proposed in [18], whose only objective is to minimize the scheduling length for each energy budget. Hence, in the best case, we can have the scheduling length near to the optimum scheduling length over different energy budgets as can be seen in Fig. 11. It is observed that for the different number of available sensor nodes, the results are fairly near to the CNPT and GA-based schemes.

In order to validate the performance of our approach, more complex system parameters such as the number of tasks and precedences in DAG and number of sensor nodes are considered. This result is shown in Table 2. Our method attains more balanced residual energy level while achieving a near optimum scheduling lengths, for different system parameter settings.

7.3 Discussion on results for dynamic adaptation stage

In this section, the performance of our proposed dynamic adaptation algorithm in the presence of the variable solar-based energy harvesting is evaluated. For this evaluation, we consider eight different prediction error ratios between 0 and 40 percentage points. For each error ratio point, 100 runs of the task allocation simulation with the random initial available energy of all the nodes are executed. Then, the percentage of the allocation failure, called *missing ratio*, is measured for the static allocation and static allocation with the dynamic phase. An allocation failure occurs once tasks and their precedences are missed due to unavailability of resources. In the dynamic adaptation phase, the task is missed only if stretching the task length to meet the energy-neutral condition causes the frequency f_k to be below f_{min} .

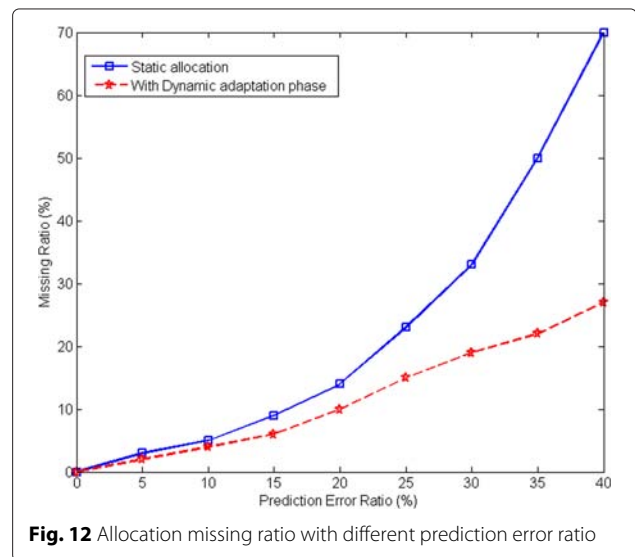
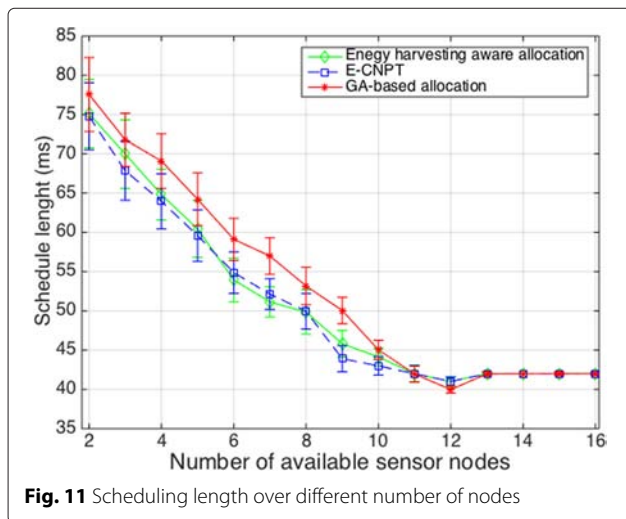
Table 2 Results from different system parameters

	Number of sensor nodes	Number of tasks	Number of precedence	Schedule length (ms)	Variance
Our approach	30	15	2	29	0.23
GA	30	15	2	29	0.19
E-CNPT	30	15	2	29	0.37
Our approach	40	22	3	36	0.27
GA	40	22	3	35	0.24
E-CNPT	40	22	3	35	0.48
Our approach	50	30	3	48	0.32
GA	50	30	3	47	0.28
E-CNPT	50	30	3	46	0.55

Figure 12 shows the significant improvement in terms of the missing ratio by implementing the dynamic adaptation phase.

8 Conclusions

In this paper, the task allocation problem for a sensor network powered by harvesting energy from natural environmental phenomenon is considered. The goals are to minimize the makespan and maximize the fairness in energy-driven task mapping (i.e., energy-balancing), while satisfying the task precedence constraints and energy harvesting causality constraints. The problem is precisely formulated as a mixed integer linear problem. The proposed framework for this problem has a hybrid framework of static and dynamic adaptation stages. Heuristic algorithms are proposed to solve the static problem in two phases: scheduling of the task graph and mapping to



the appropriate solar-powered sensor nodes. The performance of our proposed algorithms, in terms of energy-balancing and scheduling length, is evaluated through simulation and compared with other approaches, including a genetic algorithm as a baseline. We achieve more balanced residual energy levels across the network while attaining a near optimum scheduling length. The results also show the significant improvement in terms of allocation failure ratio due to implementing the dynamic adaptation stage.

Competing interests

The authors declare that they have no competing interests.

Received: 23 September 2014 Accepted: 1 December 2015

References

1. B Stefano, M Yousof Naderi, Ch Petrioli, D Spenza, *Wireless sensor networks with energy harvesting*, *Mobile Ad Hoc Networking: The Cutting Edge Directions*, (2013), pp. 701–736
2. A Kansal, J Hsu, S Zahedi, MB Srivastava, Power management in energy harvesting sensor networks. *ACM Trans. Embedded Comput. Syst.(TECS)*. **6**(4), 136–142 (2007)
3. A Cammarano, D Spenza, C Petrioli, Energy-harvesting WSNs for structural health monitoring of underground train tunnels. *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, 75–77 (2007)
4. M Gorlatova, P Kinget, I Kymissis, D Rubenstein, X Wang, in *Proc. of annual international conference on Mobile computing and networking*. Challenge: ultra low power energy-harvesting active networked tags (EnHANTs) (ACM, 2009), pp. 253–260
5. G Virone, A Wood, L Selavo, Q Cao, L Fang, T Doan, in *Proc. of Conference on Distributed Diagnosis and Home Healthcare (D2H2)*. An advanced wireless sensor network for health monitoring, (2006), pp. 23–27
6. J Zhuo, C Chakrabarti, Energy-Efficient Dynamic Task Scheduling Algorithms for DVS Systems. *ACM Trans. Embedded Comput. Syst.(TECS)*. **7**(2), 234–250 (2008)
7. W Li, FC Delicato, AY Zomaya, Adaptive energy-efficient scheduling for hierarchical wireless sensor networks. *ACM Trans. Sensor Netw. (TOSN)*. **9**(3), 23–30 (2013)
8. HS AbdelSalam, S Olariu, Toward efficient task management in wireless sensor networks. *IEEE Trans. Comput.* **60**(11), 78–96 (2011)
9. N Edalat, T Chen-Khong, X Wendong, An auction-based strategy for distributed task allocation in wireless sensor networks. *Comput. Commun.* **35**(8), 916–928 (2012)
10. T Zhu, A Mohaisen, Y Ping, D Towsley, DEOS: Dynamic energy-oriented scheduling for sustainable wireless sensor networks. *INFOCOM*, 2363–2371 (2012)
11. T La Porta, C Petrioli, D Spenza, Sensor-mission assignment in wireless sensor networks with energy harvesting. *IEEE SECON*, 413–421 (2011)
12. A Pathak, VK Prasanna, Energy-efficient task mapping for data-driven sensor network macroprogramming. *IEEE Trans. Comput.* **59**(7), 955–968 (2010)
13. Y Tian, J Boangoat, E Ekici, F Ozguner, in *Proc. of 20th International Parallel and Distributed Processing Symposium*. Real-time task mapping and scheduling for collaborative in-network processing in DVS-enabled wireless sensor networks, (2006), pp. 15–23
14. S Liu, Q Qiu, Q WU, in *Proc. of Design, Automation, and Test in Europe*. Energy Aware Dynamic Voltage and Frequency Selection for Real-Time Systems with Energy Harvesting, (2008)
15. S Liu, Q WU, Q Qiu, in *Proc. of Design Automation Conference*. An Adaptive Scheduling and Voltage/Frequency Selection Algorithm for Real-time Energy Harvesting Systems, (2009)
16. C Moser, D Brunelli, L Thiele, L Benini, in *Proc. of Conference on Distributed and Parallel Embedded Systems*. Lazy Scheduling for Energy Harvesting Sensor Nodes, (2006)
17. C Moser, D Brunelli, L Thiele, L Benini, Real-time scheduling for energy harvesting sensor nodes. *Real-Time Syst. J.* **37**(3), 233–260 (2007)
18. T Hagres, J Janecek, in *Proc. of International Conference on Parallel Processing Workshops*. A high performance, low complexity algorithm for compile-time job scheduling in homogeneous computing environments, (2003), pp. 56–62
19. J Recas, C Bergonzini, D Atienza, TS Rosing, in *Proc. of VITAE'09*. Prediction and Management in Energy Harvested Wireless Sensor Nodes, (2009)
20. C Renner, Solar harvest prediction supported by cloud cover forecasts. *IEEE ENSSys*, 45–50 (2013)
21. A Cammarano, C Petrioli, D Spenza, in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*. Pro-Energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks, (2012), pp. 234–240
22. RH Shumway, DS Stoffer, *Time Series Analysis and Its Applications*. (Springer-Verlag, 2000), pp. 250–278
23. DC Montgomery, LA Johnson, JS Gardiner, *Forecasting and Time Series Analysis*. (McGraw-Hill, 1990)
24. MR Garey, DS Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. (San Francisco, WH Freeman & Co, 1979)
25. JN Al-Karaki, AE Kamal, Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Commun.* **11**(6), 6–28 (2004)
26. National Renewable Energy Lab website. <http://www.nrel.gov/midc/>. Access date July 2014
27. Crossbow Technology INC. Mica-2 Data Sheet. [Online] Crossbow. <http://www.xbow.com>. Access date July 2014
28. DK Noh, L Wang, Y Yang, H Khac, L Tarek, in *Proc. of 5th IEEE International Conference on Distributed Computing in Sensor Systems*. Minimum variance energy allocation for a solar-powered sensor system (DCOSS, 2009), pp. 67–88
29. H El-Rewini, TG Lewis, HH Ali, *Task Scheduling in Parallel and Distributed Systems*. (Prentice-Hall International Editions, 1994)
30. AS Wu, H Yu, S Jin, K-C Lin, G Schiavone, An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Trans. Parallel Distributed Syst.* **15**(9), 824–834 (2004)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com