INT J COMPUT COMMUN, ISSN 1841-9836
9(2):160-171, April, 2014.

A Multi-objective Optimization Algorithm of Task Scheduling in WSN

L. Dai, H.K. Xu, T. Chen, C. Qian, L.J. Xie

Liang Dai, Hongke Xu, Qian Chao
School of Electronic and Control Engineering
Chang'an University
Xi'an 710064, China
E-mail: ldai1981@gmail.com, xuhongke@chd.edu.cn

Ting Chen
School of Information Engineering
Chang'an University
Xi'an 710064, China
E-mail: tchen@chd.edu.cn

Lijing Xie
Information and Navigation College
Air Force Engineering University
Xi'an 710077, China

Abstract: Sensing tasks should be allocated and processed among sensor nodes in minimum times so that users can draw useful conclusions through analyzing sensed data. Furthermore, finishing sensing task faster will benefit energy saving. The above needs form a contrast to the lower efficiency of task-performing caused by the failure-prone sensor. To solve this problem, a multi-objective optimization algorithm of task scheduling is proposed for wireless sensor networks (MTWSN). This algorithm tries its best to make less makespan, but meanwhile, it also pay much more attention to the probability of task-performing and the lifetime of network. MTWSN avoids the task assigned to the failure-prone sensor, which effectively reducing the effect of failed nodes on task-performing. Simulation results show that the proposed algorithm can trade off these three objectives well. Compared with the traditional task scheduling algorithms, simulation experiments obtain better results.

Keywords: Wireless Sensor Networks (WSN); task scheduling; multi-objective optimization; improved NSGA-II .

1 Introduction

Sensing tasks should be allocated and processed among sensor nodes in minimum times so that users can draw useful conclusions through analyzing sensed data. Furthermore, finishing sensing task faster will benefit energy saving, which is critical in system design of wireless sensor networks. The primary objective of task scheduling in wireless sensor networks is to find an optimal strategy of splitting the original tasks received by SINK into a number of sub-tasks as well as distributing these sub-tasks to the sensors in the right order. The directed acyclic graph [1], independent task sets[2] and divisible load theory [3] are usually used as modeling tools for task scheduling in wireless sensor networks, but these models only take the makespan as the main objective, and assign the task to sensors. However, wireless sensor networks are widely applied to both abominable and military environments. Meanwhile, the complexity of networks, the limited energy of sensors, and potential physical or logical faults, bring challenge to task scheduling of wireless sensor networks.

Wireless sensor network is one kind of the widely used distributed real-time systems. How to assign tasks of system to sensors in unstable and unreliable network environment, and guarantee their deadlines is one of the key techniques in wireless sensor networks. In wireless sensor network environments, QoS guided task scheduling problem is complex and challenging, especially when the tasks have multiple needs.

In this paper, a multi-objective optimization algorithm of task scheduling is proposed for wireless sensor networks. It is the first time that the NSGA-II [4] algorithm is used to analyze the task scheduling for wireless sensor networks. Based on the characteristics of wireless sensor networks, makespan optimization, the energy-consuming balance optimization and task-performing probability optimization are included. A mathematical model used to optimize the task scheduling problem by NSGA-II was built and the solution was presented, and a detailed process to solute the multi-objective programming model is put forwards. The problem is solved with a multi-objective genetic algorithm (GA) optimization method combined with linear programming (LP) and a group of pareto solutions are provided.

2 Related work and motivation

Wireless sensor networks have restrictions due to energy, memory, and communication ability. We should realize the goal of the improvement and enhancement of sensor networks performance in real time, economy, power aware and harmony.

Some studies over the past decade have been conducted to reduce the overall energy consumption for task scheduling in wireless sensor networks by using diverse techniques [5-14]. Heemin presents an energy-efficient task assignment and migration framework for sensor networks. With the proposed framework, optimal task transformation and assignment is sought so as to minimize given cost function[6]. Younis M presented an optimization scheme for task allocation to gateways [7]. The task allocation problem is modeled as a zero-one nonlinear program. He study system partitioning of computation to improve the energy efficiency of a wireless sensor networking application. Wang explored system partitioning between the sensor cluster and the base station, employing computation-communication tradeoffs to reduce energy dissipation[8]. Also he showed that system partitioning of computation within the cluster can also improve the energy efficiency by using dynamic voltage scaling. Tian presented a task mapping and scheduling solution for real-time applications (RT-MapS) in WSNs[9]. RT-MapS incorporates wireless channel modeling, hyper-DAG extension, concurrent task mapping, communication and computation scheduling, and dynamic voltage scaling (DVS) methods. Tian also presented a task mapping and scheduling solution for energy-constrained applications in WSNs, energy-constrained task mapping and scheduling (EcoMapS)[10]. EcoMapS incorporates channel modeling, concurrent task mapping, communication and computation scheduling, and sensor failure handling algorithm. The performance of EcoMapS is evaluated through simulations with randomly generated directed acyclic graphs (DAG).Yu proposed an energy-balanced allocation of a real-time application onto a single-hop cluster of homogeneous sensor nodes connected with multiple wireless channels[11]. An epoch-based application consisting of a set of communicating tasks is considered. Each sensor node is equipped with discrete dynamic voltage scaling (DVS). The time and energy costs of both computation and communication activities are considered. He proposed both an Integer Linear Programming (ILP) formulation and a polynomial time 3-phase heuristic. Gu presented an application independent task mapping and scheduling solution in multi-hop Video sensor networks (VSNs) that provides real-time guarantees to process video feeds[12]. The processed data is smaller in volume which further releases the burden on the end-to-end communication. Using a novel multi-hop channel model and a communication scheduling algorithm, computation tasks and associated communication events are scheduled simultaneously with a dynamic critical-path scheduling algorithm. Dynamic voltage scaling (DVS) mechanism is implemented to further optimize energy consumption. Xie developed a novel task allocation strategy called BEATA (Balanced Energy-Aware Task Allocation) for collaborative applications running on heterogeneous networked embedded systems[13]. The BEATA algorithm aims at blending an energy-delay efficiency scheme with task allocations, thereby making the best tradeoffs between energy savings and schedule lengths. Besides, He introduced the concept of an energy-adaptive window, which is a critical parameter in the BEATA strategy. By fine-tuning the size of the energy-adaptive window, users can readily customize BEATA to meet their specific energy-delay trade-off needs imposed by applications. Further, he built a mathematical model to approximate energy consumption caused by both computation and communication activities. A task allocation framework for Underwater Acoustic Sensor Networks (UW-ASNs) that participate as a team to accomplish critical missions is developed by Kulkarni[14]. The team formed as a result of this task allocation framework is the subset of all deployed AUVs that is best suited to accomplish the mission while adhering to the mission constraints.

Most of the existing task scheduling algorithms is normal ones without consideration of harmony and reliability in our research. This problem will need to be further gone into for improving the wireless sensor networks characteristics on real time, economy, power aware and harmony.

The above discussed task scheduling algorithms are mainly for the problem of assignment tasks with the objective of minimizing the makespan and energy consumption, effective algorithms is discussed in these papers, and these algorithms will be able to distribute tasks to nodes with the expected value of the minimum earliest-finish-time and make them executed parallel and efficiently. However, tasks distributed to remote sensors may be unable to complete because of physical failures or attack, and the complexity, dynamics and open deployment of the wireless environment increases the possibility of that happening. Existing task scheduling policies are not consider this question, so tasks will be assigned to the sensors with lower reliability to perform, what make measurement tasks automatically cease to be in force due to sensor failure, and then lowered the efficiency and QoS of wireless sensor networks.

Compared with the previous research, the main contributions of this paper are presenting the concept of " task-performing probability ", and proposing a multi-objective optimization task scheduling algorithm for wireless sensor networks considering makespan optimization, the energy-consuming optimization and task-performing probability optimization simultaneously. Consideration shall also be given to ensuring a reasonable distribution of tasks to reliable sensors, taking into account the makespan and energy consumption. This algorithm schedules tasks avoiding allocation some tasks to unreliable sensors, thus effectively lowering the influence of sensor failure on task performing.

3 The Task Scheduling Model for WSNs

In this section, we describe mathematical models which were built to represent a task scheduling framework.

3.1 Task Model

Generally, a wireless sensor network consists of a set of heterogeneous sensors in abominable or military environments. Sensor nodes always break down due to hardware failure, software error, energy exhaustion and disturb from outer environment, so the precondition of the tasks accomplished successfully is the sensors to provide a stable hardware and software needed to perform the given tasks. Each sensor is in a "active" or "inactive" state at a given time. The task is impossibly accomplished on a inactive sensor, and the status information also will be missed or ineffective.

Let λ_i be the failure probability of sensor S_i , then the task-performing probability of the tasks accomplished successfully on sensor S_i is $1 - \lambda_i$. We assumed that the failure process of each sensor is independent and yields to poisson distribution.

We defined the task set as $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$. According to the previous assumption, the tasks in the task set is independent each other.

3.2 Makespan Model

Suppose that a wireless sensor network consists of m sensors, $S = \{S_1, S_2, \cdots, S_m\}$. There are n independent tasks competing the m sensors. We aim to scheduling the n tasks to the m sensors reasonably, to make the minimum makespan of the tasks. And we characterize a $n \times m$ matrix X satisfying the scheduling results. When $x_{i,j} = 1$ $(x_{i,j} \in X)$, it meant that we schedule the task τ_i on sensor S_j to process, otherwise, $x_{i,j} = 0$.

The task-processing time of each sensor can be estimated by forecasting techniques and history status based on task type. The task-processing time can be represented by a $n \times m$ matrix Y, in which the matrix element $y_{i,j}$ represents the estimated task-processing time that task τ_i runs on sensor S_j.The task-processing time of sensor S_j is the sum of all the tasks processing time that the tasks run on S_j.

It can be expressed as follow:

$$T_j = \sum_{i=1}^{n} y_{i,j} \tag{1}$$

Then, the makespan that the n tasks scheduled to the m sensors according to the scheduling result X is expressed as:

$$T(\Gamma, S, X) = Max_{S_i \in S}[R(S_j)] \tag{2}$$

3.3 Energy Consumption Model

Because of the limited energy in wireless sensor networks, the research on low power technology and long lifetime is pivotal in the architecture of wireless sensor networks. The energy consumption of wireless sensor networks mainly composed of communication and task-performing energy consumption.

Communication energy consumption is relevant to the minimum energy-communication for a given standard distance P_0 and the distance $d_{i,j}$ between sensor S_i and sensor S_j.

$$P_{i,j} = (4\pi)^2 d_{i,j}^2 \beta P_0 / d_0^2 G_t G_r \lambda^2 \tag{3}$$

where G_t and G_r are the antenna gains (with respect to an isotropic radiator) of the transmitting and receiving antennas respectively, λ is the wavelength, and β is the energy consumption factor. Since $(4\pi)^2 \beta P_0 / G_t G_r \lambda^2$ is constant, one unit of communication energy consumption is indicated as $d_{i,j}^2 / d_0^2$.

Task-performing energy consumption of sensor S_j for finishing all its tasks is:

$$E_j = C_j T_j \tag{4}$$

where C_j is the task-performing energy consumption of one certain task in one unit time.

The relative task-performing energy consumption of sensor S_j can be represented as follow:

$$R_j = E_j / Max_{S_i \in S}(E_j) \tag{5}$$

Excluding energy consumption, the lifetime of wireless sensor networks is the critical estimated part in task scheduling too. To extend network lifespan, we should balance the energy consumption of every sensor during the period of ordinary operation. We can measure the uniformity of sensors' residual energy based on entropy theory.

The uniformity of sensors' residual energy after data transmission in t can be represented as follow:

$$H_{it} = -\sum p(E_{it}) \log p(E_{it}) \tag{6}$$

where E_{it} is the residual energy of sensor S_j in t, and H_{it} is the value of residual energy entropy in the network. The higher H_{it}, the more residual energy in the network, then the longer lifetime of the network.

Thus, we define the evaluation function of energy consumption for sensor S_j is that:

$$C(S_i) = -d_{i,j}^2 R_j H / d_0^2 \tag{7}$$

The less $C(S_i)$ means that the energy consumption of data transmission and executing tasks has less effect on the residual energy in the network.

The evaluation function of energy consumption for the network is defined as:

$$C(L, S, X) = \sum_{i=1}^{m} C(S_i) \tag{8}$$

The smaller the value of $C(L, S, X)$, the longer network lifetime.

3.4 Task-performing Probability

The concept of task-performing probability comes from the survivability of distributed system. Survivability represents that tasks are capable of being performed steadily and regularly in the networks.

Definition. The task-performing probability means the probability that task could be accomplished successfully on specified sensor.

Let $P(\tau_i, S_j)$ be the task-performing probability that task τ_i accomplished on sensor S_j. In accordance with the illustrations in the previous section, the task-performing probability is $\exp(-\lambda_j t)$ that sensor be in the normal (ACTIVE) state within t. Because the tasks can be performed normally only when the given sensor is in a "normal" state, so we can get:

$$P(\tau_i, S_j) = \exp(-\lambda_j R(S_j)) \tag{9}$$

Let $P(\Gamma, S, X)$ is the task-performing probability the n tasks to the m sensors according to the scheduling result X, then we can get:

$$P(\Gamma, S, X) = \exp[-\sum_{i=1}^{i=n} \sum_{j=1}^{j=m} \lambda_j x_{i,j} R(S_j)] \tag{10}$$

Taking the task-performing probability as goal is to avoid the tasks scheduled to the sensors with low reliability.

Lowering the influence of sensor failure on task performing, then to maximize the task-performing probability of task set Γ, that is, we will maximize $P(\Gamma, S, X)$ by our task scheduling algorithm to obtain the most suitable X. Thus, the quality of service for wireless sensor networks will be improved.

In eq. 9, let $L(\tau_i, S_j) = -\lambda_j R(S_j)$, we can see that if we want to increase the value of $P(\tau_i, S_j)$, then we must lower the $L(\tau_i, S_j)$.

Let $L(\Gamma, S, X) = \sum\limits_{i=1}^{i=n} \sum\limits_{j=1}^{j=m} \lambda_j x_{i,j} R(S_j)$, then we can see that maximizing $P(\Gamma, S, X)$ means to Minimize $L(\Gamma, S, X)$.

The task scheduling algorithm taking the task-performing probability as goal makes the value of $L(\Gamma, S, X)$ as minimal as possible.

4 Optimal Task Scheduling Based on Improved NSGA-II

4.1 Multi-Objective Optimization Problem

In the case of a Multi-Objective Optimization (MOO) problem, there is usually no single solution that is optimum with respect to all objectives. There are a set of optimal solutions known as pareto optimal solutions. Without additional information, all these solutions are equally satisfactory. The goal of MOO is to find as many of these solutions as possible.

In the research of pareto front many methods have been proposed. David Schaffer first implemented a multi-objective evolutionary algorithm called the vector-evaluated genetic algorithm or VEGA in 1984. His algorithm started off well but tended to converge to a single solution. To prevent the convergence to a single solution, Goldberg and Richardson suggested using a non-dominated sorting procedure coupled with a niching strategy called sharing. Sharing takes into account that individuals in the same niche must share the available resources. This concept is integrated into the pareto genetic algorithm by increasing the cost of chromosomes as a function of their distance from each other. Closely grouped chromosomes will find their costs increased more than chromosomes that are spaced far apart.

The multi-objective genetic algorithm (MOGA)[15] starts by finding all non-dominated chromosomes of a population and gives them a rank of one. These chromosomes are removed from the population. Next, all the non-dominated chromosomes of this smaller population are found and assigned a rank of two. This process continues until all the chromosomes are assigned a rank. The largest rank will be less than or equal to the size of the population. Usually, there are many solutions that have the same rank. The selection procedure uses the chromosome ranking to determine the mating pool. MOGA also uses niching on the cost to distribute the population over the pareto optimal region [16].

Non-dominated Sorting Genetic Algorithm (NSGA)[17] ranks chromosomes in the same manner as MOGA. The NSGA algorithm then calculates a unique value. This unique value is related to the distance between each solution and its two closest neighbors. Distance may be calculated from the variable values or the associated costs. The resulting values are scaled between 0 and 1 and subtracted from the cost. Further information about the evolution of this method can be found in [4,16,17].

As discussed above, the goal of MTWSN is to find the solutions giving the best trade-off between the three conflict objectives, known as Pareto optimal. MOGAs[4,17] are recognized to be well qualified to tackle multi-objective optimization problems. NSGA-II[4] is one of the most popular MOGAs. Some concepts of multi-objective optimization problem are defined as follows.

Definition 1 (Multi-objective optimization problem). Given an n-dimensional decision vector $x = \{x_1, x_2, \cdots, x_n\}$ in the solution space X, find a vector x^* that maximizes a given set of k objective functions $f(x^*) = \{f_1(x^*), f_2(x^*), \cdots, f_k(x^*)\}$. The solution space X is generally restricted by a series of constraints, such as $g_j(x^*) = b_j$ for $j = 1, 2, \cdots, m$.

Definition 2 (Dominance). A vector $u = \{u_1, u_2, \cdots, u_n\}$ is said to dominate a vector $v = \{v_1, v_2, \cdots, v_n\}$ if and only if u is partially less than v, i.e., $\forall\ i = 1, 2, \cdots, n,\ u_i \leq v_i \land \exists i =$

$1, 2, \cdots, n,\ u_i \leq v_i$.

Definition 3 (Pareto optimal solution). A solution $x_u \in X$ is said to be pareto optimal if and only if there is no $x_v \in X$ for which $v = f(x_v) = \{v_1, v_2, \cdots, v_n\}$ dominates $u = f(x_u) = \{u_1, u_2, \cdots, u_n\}$.

Definition 4 (Pareto optimal set and front). Let $A \subseteq X$. The nondominated set regarding A, represented by X_p, is defined as $X_p = \{z \in A | z\ is\ nondominated\ regardingX\}$. The corresponding objective function values in the objective space are defined as $Y_p = F(X_p) = \{f(z) | z \in X_p\}$, where X_p is called the pareto optimal set and Y_p is called the cohere pareto optimal front.

However, the solutions found by original NSGA-II are likely to be inferior or only comparable to that by classical heuristic search algorithms because of premature convergence. To find perfect solutions, a delete operator for NSGA-II is proposed to enhance the search capability. When selecting the elitist, if neither of the two individuals in a population wins out and their genes are the same, then delete one of them. Furthermore, a circulation selection is presented to preserve excellent genes of the parent population. Suppose there are K individuals in a population $(ind_1, ind_2, \cdots, ind_K)$ when the crossover operations are carried out. The first time the operation is carried out with (ind_1, ind_2) as parents, the second time (ind_2, ind_3) are taken as parents, and so on. Similarly, the last child is done by (ind_K, ind_1). By this way, K offspring individuals are generated. The genes of each parent are inherited by two offspring individuals, thus avoiding the loss of excellent solutions.

4.2 An Efficient Task Scheduling Algorithm

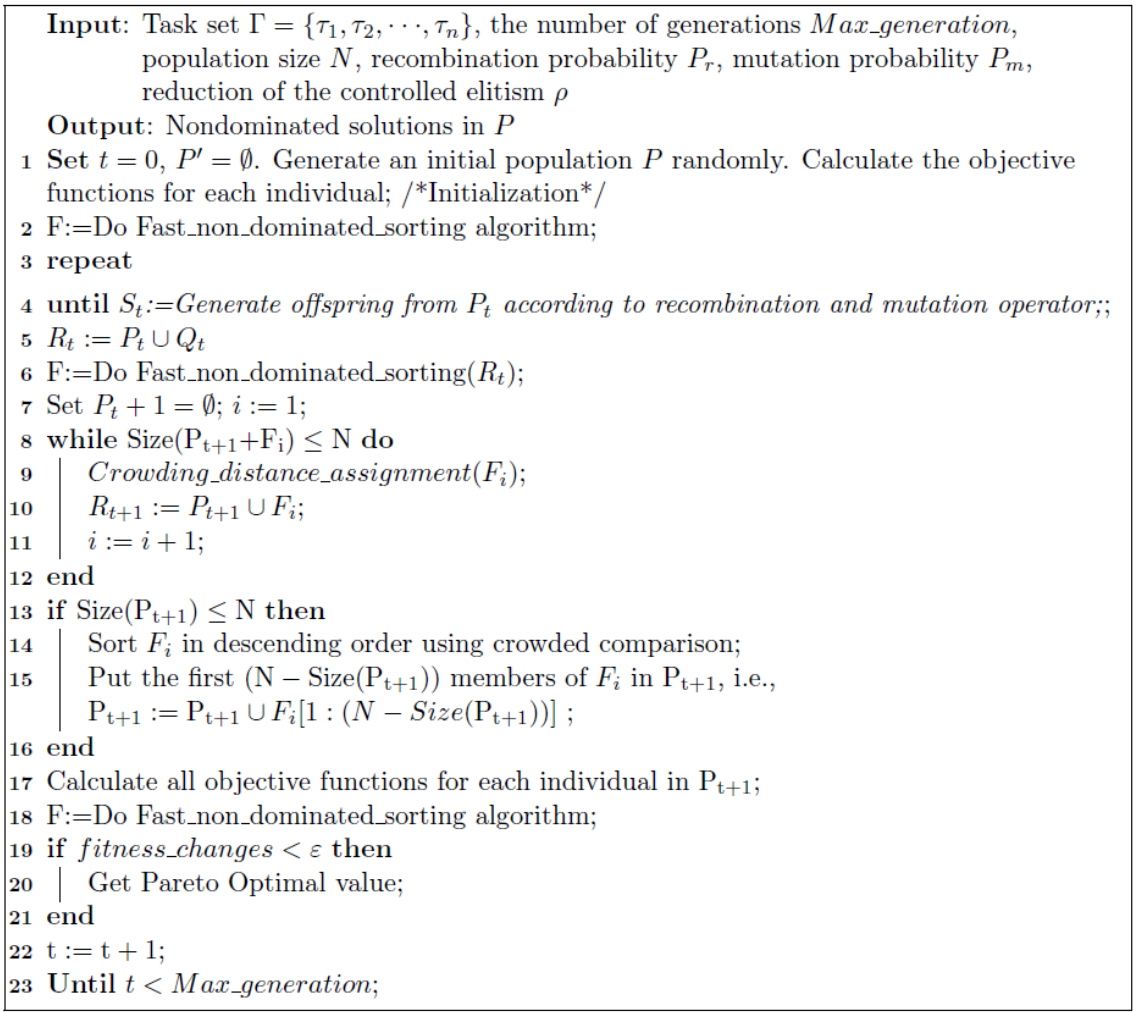
The motivation here is to provide the user with a set of pareto optimal solutions by the NSGA-II algorithm and give it the flexibility to choose the best possible solution from this set, depending on the specific application requirements.

Now, we can construct the cover set using the optimal pareto solutions generated by the improved NSGA-II algorithm. The chromosomes of a genetic algorithm contain all the building blocks to a solution for the genetic operators and the fitness function. In our implementation, each individual node is represented by a one-bit binary number called gene. This one-bit gene defines the status of the sensors as follows:

$$x_{i,j} = \begin{cases} 1,\ if\ task\ \tau_i\ runs\ on\ S_j \\ 0,\ otherwise \end{cases} \tag{11}$$

We assign each individual with three fitness functions, makespan, energy consumption and task-performing probability. By introducing the non-dominated sorting approach and the crowded distance operator the replacement scheme is executed. First, a combined population $R_t = P_t \cup Q_t$ is formed with the parent population P_t and the offspring population Q_t, where t is the number of generation. Therefore the population R_t will be of size 2N. And it is sorted according to the non-domination and crowded comparison. By adding solutions from the first front till the size exceeds N, the new parent population P_{t+1} is formed. After that, the solutions of the last accepted front are sorted according to the crowed comparison and the first $(N - Size(P_{t+1}))$ points are picked. In this way, the population P_{t+1} of size N is constructed. Subsequently, it is used for the circulated selection, crossover, and mutation to create a new population Q_{t+1} of size N. The recombination operator used in this paper is K-point crossover. After recombination, the mutation operator is applied to complement some genes in the chromosomes of the child randomly.

This entire process is repeated until the difference of fitness values among the current pareto optimal set and the previous one is less than a chosen precision (ε). The main procedure of



```
Input: Task set $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$, the number of generations $Max\_generation$,
       population size $N$, recombination probability $P_r$, mutation probability $P_m$,
       reduction of the controlled elitism $\rho$
Output: Nondominated solutions in $P$
Set $t = 0$, $P' = \emptyset$. Generate an initial population $P$ randomly. Calculate the objective
functions for each individual; /*Initialization*/
F:=Do Fast_non_dominated_sorting algorithm;
repeat
until $S_t$:=Generate offspring from $P_t$ according to recombination and mutation operator;;
$R_t := P_t \cup Q_t$
F:=Do Fast_non_dominated_sorting($R_t$);
Set $P_t + 1 = \emptyset$; $i := 1$;
while Size($\mathrm{P_{t+1}+F_i}$) $\leq$ N do
    $Crowding\_distance\_assignment(F_i)$;
    $R_{t+1} := P_{t+1} \cup F_i$;
    $i := i + 1$;
end
if Size($\mathrm{P_{t+1}}$) $\leq$ N then
    Sort $F_i$ in descending order using crowded comparison;
    Put the first ($\mathrm{N} - \mathrm{Size}(\mathrm{P_{t+1}})$) members of $F_i$ in $\mathrm{P_{t+1}}$, i.e.,
    $\mathrm{P_{t+1}} := \mathrm{P_{t+1}} \cup F_i[1 : (N - Size(\mathrm{P_{t+1}}))]$ ;
end
Calculate all objective functions for each individual in $\mathrm{P_{t+1}}$;
F:=Do Fast_non_dominated_sorting algorithm;
if $fitness\_changes < \varepsilon$ then
    Get Pareto Optimal value;
end
t := t + 1;
Until $t < Max\_generation$;
```

Algorithm 1: A Genetic Approach to Task Scheduling Algorithm

algorithm is described as algorithm 1:

5 Performance Evaluation

In this section, we will present the simulation results as the performance evaluation of our proposed task scheduling algorithm. The performance of our proposed algorithm is compared with RT-Maps [9], EcoMaps [10], and EBTA [11] task scheduling algorithms in wireless sensor networks in terms of the makespan, and energy consumption of task scheduling. For our experiments, we use 50 sensor nodes and a sink, which is not power constrained. Nodes are distributed in a 100X100 meter square area yielded to random poisson distribution. To evaluate the performances of the task scheduling algorithms properly, experimental parameters of the four algorithms were set as the same ones.

The performance of the genetic algorithm is greatly affected by a number of factors, such as the population size, the probability of mutation and crossover, and the method of scheduling. We have run a number of experiments with different values of these parameters to determine the optimal set for our network size. Finally, the GA parameters used in our simulations are listed in table 1. Although we allow the GA to run for a maximum of 100 generations, we have observed

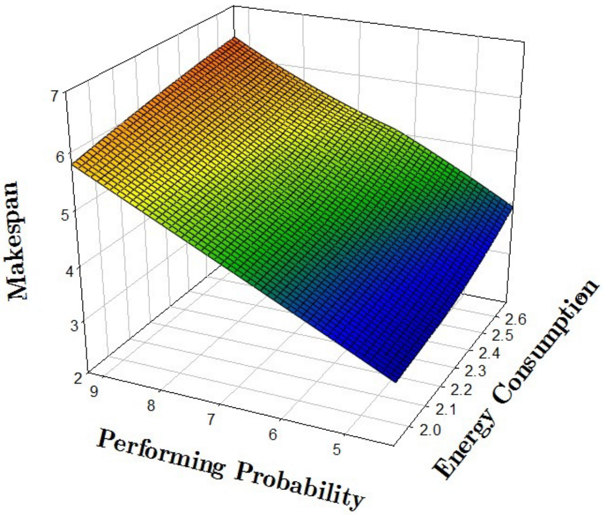


Figure 1: Schematic diagram of statistic information for scheduling

that the best solution is typically found within 40 generations.

Table 1: GA parameters used in Simulation

Parameter	Value
Population size	200
Recombination rate	0.9
Mutation rate	0.005
Reduction rate	0.5

The simulation results are shown in Fig. 1 to Fig. 3.

Figure 1 delineates the statistical information of the proposed task scheduling algorithm. We investigates the space of optimum configuration parameters (makespan, energy consumption and task-performing probability), using unconstrained optimization, to highlight the trade-offs faced by the WSNs considered. In order to solve the optimization problem, we chose a pareto-compliant ranking method based on evolutionary techniques, namely the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Fig. 1 shows the result of the joint optimization of the three objective functions. These results show that the higher the balanced energy consumption or the higher the probability of task performing, the shorter the makespan. The important outcome of the results in Fig. 1 is that it provides details about the optimal network configuration, since tuning the network with the parameters derived from the execution of the NSGA-II algorithm guarantees that the network performance is not biased toward any one of the performance indicators.

The makespan metric is compared between proposed task scheduling algorithms for wireless sensor networks in Figure 2. It explains that the makespan for wireless sensor networks is lower for proposed algorithms than RT-Maps [9], EcoMaps [10], and EBTA [11]. This is because that in proposed algorithm the tasks have been finished on minimum makespan as the target function.

Figure 3 shows the comparative performance in terms of energy consumption of sensor nodes.With the increase of the number of tasks in the network, the energy consumption of task scheduling and data transmission in the algorithms increases. However, the energy consumption in the proposed algorithm is lower than RT-Maps [9], EcoMaps [10], at fixed number of tasks.

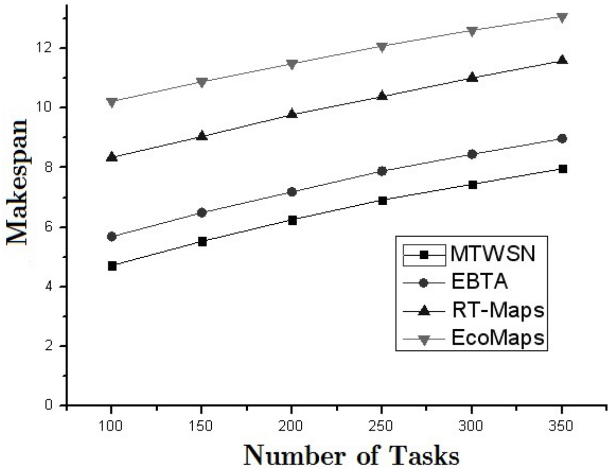


Figure 2: Makespan of different scheduling algorithms

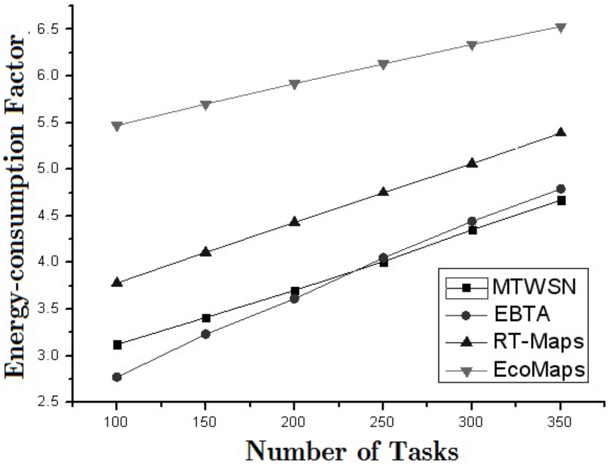


Figure 3: Energy-cosumption factor of different scheduling algorithms

Our proposed task scheduling algorithm results in lowest energy consumption for higher number of tasks. This shows the efficiency of the proposed algorithm.

6 Conclusions

The optimization of task scheduling is studied to reduce the energy consumption and ensure the effective information acquisition in wireless sensor network. A multi-objective optimization algorithm of task scheduling is proposed for wireless sensor networks in this paper. It is the first time that the NSGA-II algorithm is used to analyze the task scheduling for wireless sensor networks. Based on the characteristics of wireless sensor networks, makespan optimization, the energy-consuming balance optimization and task-performing probability optimization were included. A mathematical model used to optimize the task scheduling problem by NSGA-II was built and the solution was presented, and a detailed process to solute the multi-objective programming model was put forwards.

Bibliography

[1] Z. Zeng, A. Liu, D. Li (2008), A Highly Efficient DAG Task Scheduling Algorithm for Wireless Sensor Networks, *In Proc. of ICYCS 2008*,Zhang Jia Jie , Hunan , China, 570-575.

[2] J. Lin, W. Xiao, F. L. Lewis (2009), Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks. *IEEE Transactions on Instrumentation and Measurement*, 58(6):1886 - 1896.

[3] L. Dai, Y. Chang, Z. Shen (2011), An Optimal Task Scheduling Algorithm in Wireless Sensor Networks, *Int J Comput Commun*, ISSN 1841-9836, 6(1):101-112.

[4] K. Deb, A. Pratap, S. Agarwal (2002), A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182-197.

[5] J. Mao, C. G. Cassandras, Q. Zhao (2007), Optimal Dynamic Voltage Scaling in Energy-Limited Nonpreemptive Systems with Real-Time Constraints. *IEEE Transactions on Mobile Computing*, 6(6):678 - 688.

[6] P. Heemin, B. S. Mani (2003), *Energy-efficient Task Assignment Framework for Wireless Sensor Netwoks*, UC Los Angeles: The Berkeley Electronic Press.

[7] M. Younis, K. Akkaya, A. Kunjithapatham (2003), Optimization of Task Allocation in a Cluster-based Sensor Network. *In Proc. of the Eighth IEEE International Symposium on Computers and Communication*, Netherlands: IEEE Computer Press, 329-334.

[8] A. Wang, A. Chandrakasan (2002), Energy-efficient DSPs for Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 19(4):68-78.

[9] Y. Tian, J. Boangoat, E. Ekici (2006), Real-time Task Mapping and Scheduling for Collaborative In-network. *In Proc. of the 20th International Parallel and Distributed Processing Symposium.* Rhodes Island: IEEE Computer Press, 1-10.

[10] Y. Tian, E. Ekici (2005), Energy-constrained Task Mapping and Scheduling in Wireless Sensor Networks. *In Proc. of the Workshop on Resource Provisioning and Management in Sensor Networks.* Washington, D C: IEEE Computer Society, 211-218, 2005.

[11] Y. Yu, V. K. Prasanna (2005), Energy-balanced Task Allocation for Collaborative Processing in Wireless Sensor Networks. *ACM/Kluwer Mobile Networks and Applications Journal*, 10(1):115-131.

[12] Y. Gu (2007), Real-time Multimedia Processing in Video Sensor Networks, *Signal Processing : Image Communication*, 22(3):237-251.

[13] T. Xie, X. Qin (2008), An Energy-Delay Tunable Task Allocation Strategy for Collaborative Applications in Networked Embedded Systems. *IEEE Transactions on Computers*, 57(3):329 - 343.

[14] I. S. Kulkarni, D. Pompili (2010), Task Allocation for Networked Autonomous Underwater Vehicles in Critical Missions. *IEEE Journal on Selected Areas in Communications*, 28(5):716-727.

[15] C. M. Fonseca, P. J. Fleming (1993), Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, *In Genetic Algorithms: Proceedings of the Fifth International Conference*, San Mateo, CA: Morgan Kaufmann, 416-423.

[16] R. L. Haupt, S. E. Haupt (2004), Practical Genetic Algorithms. John Wiley & Sons.

[17] N. Srinivas, K. Deb (1995), Multiobjective optimization using nondominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, 2(3):221-248.