

Jinghua Zhu¹, Jianzhong Li, Hong Gao
School of Computer Science & Technology,
Harbin Institute of Technology, 318#
150001, Harbin, China.
zhujinghua@hit.edu.cn



Journal of Digital
Information Management

ABSTRACT: Collaborative processing among sensors to fulfill given tasks is a promising solution to save significant energy in resource-limited wireless sensor networks (WSN). Quality-of-Service (QoS) such as lifetime and latency is largely affected by how tasks are mapped to sensors in network. Tasks allocation is a well-defined problem in the area of high performance computing and has been extensively studied in the past. Due to the limitations of WSN, existing algorithms cannot be directly used. In this paper, a novel nested optimization technique based on genetic algorithm is proposed to assign tasks onto sensors with minimal cost while meeting application's QoS requirements. Optimal solution can be achieved by incorporating task mapping, routing path allocation, communication scheduling, dynamic voltage scaling. Performance is evaluated through experiments with randomly generated Directed Acyclic Graphs (DAG) and experiments results show better solution compared with existing methods.

Categories and Subject Descriptors

C.2.1[Network Architecture and Design]; Wireless Communication: D.4.4[Communications Management] Network Communication: H.3.1 [Group and Organization Interfaces] Collaborative Computing

General Terms

Wireless Sensor Networks, Genetic algorithms

Keywords: Wireless sensor network(WSN); Quality-of-Service(QoS); Tasks graph; Tasks scheduling; Genetic Algorithm(GA).

Received 11 Nov. 2006; Revised and accepted 11 Jan. 2007.

1. Introduction

Technology advances in embedded system and wireless communications in recent years are making complex and diverse applications of sensor network possible, such as target tracking, infrastructure monitoring, habitat sensing, and battlefield surveillance etc. WSNs usually consist of a large number of tiny sensor nodes with the capability of sensing, processing and communicating. Many emerging applications consist of various kinds of computation or communication tasks (e.g. sensing, filtering, image or speech processing, storing intermediate data, etc) and require various resources to collaboratively accomplish the given tasks. For example, in a target tracking application[18], sensor nodes are usually organized into clusters. Distributed signal detection and collaborative data processing such as LU factorization[13] or the Fast Fourier Transformation(FFT)[14] are performed within each cluster for detecting, identifying and tracking an object. However, due to the un-replaceable property of battery, applications for WSN will be imposed by energy consumption constraints to guarantee network lifetime. Hence, energy consumption is a key concern in WSN [1]. It is becoming a challenging

research issue of how to assign tasks onto sensor nodes in resource-restricted sensor network. Efficient scheduling of tasks onto the available resources in sensor networks is one of the key factors for achieving high performance.

Task allocation has been extensively studied in the area of parallel and distributed system[2], the same allocation problem onto sensor nodes can be transformed into a traditional task allocation problem in distributed system. However, in sensor networks, it is a challenging topic to accommodate the communication aspect at the same time. Traditional parallel processing systems generally assume point-to-point connections between all nodes, communications can occur simultaneously without contention. However, communication scheduling is needed to avoid contention in WSN due to the restricted resource. Thus, task allocation algorithms in high performance computing cannot be directly used in WSN.

Task allocation for WSN has been recently discussed in literatures [9] [10] which solve the similar problem with this paper. However, their communication model is only suitable for single-hop cluster, we focus on the communication scheduling in multi-hop cluster. It has been proven that short distance multi-hop transmission can save much more energy than long distance single-hop transmission. Single-hop cluster is perfect in a small scale network, however, in a large scale network, multi-hop cluster is much energy efficient for the significant reduced communication cost. In a single-hop cluster, there can be only one transmission on the wireless channel at a given time. Hence, the wireless channel is modeled as a virtual node C that executes one communication task at any time instance. Thus, a cluster can be modeled as a star-network where all sensors only have connections with the virtual node C [10]. However, in a multi-hop cluster, nodes transmit data to the other nodes by means of more than one node's relay. Routing path allocation plays an important role in energy consumption of tasks allocation. Communication model in single-hop cluster can not be used in a multi-hop cluster because it does not take inference avoidance into consideration. Communication scheduling in multi-hop sensor network is more complicated because of routing consideration.

In this paper, we consider tasks allocation onto a multi-hop cluster of heterogeneous sensor nodes connected by wireless channels. Heterogeneity in this paper embraces multiple meanings. First, multiple kinds of sensors may coexist, e.g., sound sensors, video sensors and temperature sensors etc. Different kind of sensors may have different processing capabilities; consequently execution time of a task on different nodes may vary. Second, different nodes may have distinct processor architectures and hence different nodes in a system are suitable for different kinds of tasks. Third, the bandwidth between different nodes pair may be distinct. To the best of the authors' knowledge, this is the first work for tasks allocation in a multi-hop cluster-based WSN that considers time and energy costs of both computation and communication.

* This work is partly supported by the Key Program of the National Natural Science Foundation of China under Grant No.60533110 and Grant No.60473075, the National Grand Fundamental Research 973 Program of China under Grant No. 2006CB303000.

¹ To whom all correspondence should be addressed

Task allocation usually consists of two steps: mapping and scheduling. The first step aims to find a proper sensor node on which the task will be executed. The second step is used to order the execution of tasks on each sensor node. In the context of multi-hop clustered WSN, two additional steps: routing and Dynamic Voltage Scaling (DVS) are needed to further optimize the assignment. The cost function is used to determine the better solution when choosing task allocation from search space. The cost function should be defined according to the application that the users are interested in. Therefore, we present a general objective function that can accommodate various tradeoffs.

Task allocation is a well-known NP-complete problem; proper optimization technique is needed in order to achieve suboptimal solution in polynomial time. Heuristic-based techniques such as various list-scheduling heuristics, are widely used due to the low complexity, however, optimal solution can not always be found using these methods. Guided random search techniques such as genetic algorithm can generate good quality of output schedules. Task mapping can affect the overall communication workload and delay greatly because communications only occur among tasks mapped onto different sensor nodes. Further, different routing could lead to different energy consumption and communication delay. Thus, a novel nested optimization technique based on genetic algorithm is proposed in this paper with the aim to minimize the overall cost in network.

Paper Organization: We discuss the related work in Section 2. Task allocation problem and some preliminaries are defined in Section 3. The detail of the task allocation algorithm is described in section 4 including each step of the optimization technique. Experiments results are demonstrated in Section 5. We give conclusions and future work in Section 6.

2. Related Work

Task mapping and scheduling problem has been well studied in the area of distributed system, however it dose not take into account communications which have a big impact on energy and delay of network[2]. In [3], a novel energy-aware communication and task scheduling algorithm (EAS) statically schedule communication activities and computations onto heterogeneous Network-On-Chip (Noc) architectures under real time constraint. They focus on the architectures interconnected by 2D mesh networks with XY routing schemes. The communications in wireless sensor network are much more costly and can be done in a multi-hop fashion. In [5], an online task scheduling mechanism (CoRAI) is proposed to allocate the network resources between the tasks of periodic applications in WSNs. Upper bound frequencies of applications are evaluated according to bandwidth and communication requirements between sensors. The frequencies of the tasks on each sensor are optimized subject to the upper-bound execution frequencies. However, CoRAI does not address mapping tasks to sensor nodes, and energy consumption is not explicitly discussed. Task mapping mechanisms in wireless networks have been presented in [6],[7]. A TCP-oriented distributed task mapping approach is introduced in [6] for mobile ad hoc networks. A data fusion task mapping mechanism, DFuse, is presented for WSNs in [7]. Both solutions assume an existing underlying network communication mechanism. Communication scheduling between nodes is not addressed explicitly, which has a significant impact on communication efficiency and energy consumption in WSNs. Task mapping and task scheduling have been jointly considered for mobile

computing [8] and WSNs [9][10] recently. Task mapping and scheduling heuristics are presented in [8] for heterogeneous mobile ad hoc grid environments. However, the communication model adopted in [8] is not well suited for WSNs, which assumes individual channels for each node and concurrent data transmission and reception capacity of every node. In [9], communications over multiple single-hop wireless channels are first modeled as additional linear constraints of an Integer Linear Programming (ILP) problem. Then a heuristic algorithm is presented to provide a practical solution. Energy consumption optimization is addressed through energy-balanced task allocation. However, the energy-balanced solution in [9] does not take energy consumption constraints into account and cannot provide guarantee of application energy consumption constraints. EcoMapS algorithm in [10] present a generic task mapping and scheduling solution for single-hop clustered wireless sensor networks with the similar objective as this paper. Based on realistic energy models for computation and communication, EcoMapS aims to provide energy consumption guarantees with minimum schedule lengths. However, none of the above methods take into consideration allocating tasks onto sensor nodes in multi-hop cluster. Multi-hop is much preferred than single-hop for the sake of saving much energy. In order to get optimal solution, a novel nested optimization method is proposed to explore the whole search space, including genetic-based task mapping, genetic-based routing path allocation, task and communication scheduling and dynamic voltage scaling.

3. Preliminaries

3.1. Network Model

In this paper, we assume that sensors are grouped into *multi-hop* clusters, each cluster execute an application which is either assigned during system setup time or distributed by base stations during system run time. It is the cluster heads' responsibility to create schedules for appli-cation communication and computation. The topology of a communication network is modeled as a *topology graph* $TG = (P, L)$, where P is a finite set of vertexes and each represents a sensor node equipped with discrete *DVS*. L is a finite set of links. Each link $L_{ij} \in L$ represents a communication link from sensor node P_i to P_j . Figure1(a) shows an example of network topology graph. In the graph, some nodes are connected directly by a link and some others are connected indirectly by more than one links. We assume that the topology of sensor network is a prior knowledge and sensor network is static, will not change during sensor lifetime. In dynamic WSN, such topology knowledge will not be known in advance; hence we will defer the discussion of distributed tasks allocation in dynamic WSN to our future work.

3.2. Application Model

In this paper, we consider that a periodic real-time application consists of a set of computation and communication tasks, represented by a *Directed Acyclic Graph (DAG)* $G = (V, E, w, c)$. The nodes in V represent tasks to be executed without preemption. Further, each task might inherit a specific hard deadline θ . These deadlines must be met to fulfill the feasibility requirements of the specific application. In addition, the task graph inherits a period p that specifies the maximum allowed time between two successive executions of the initial task. Each edge $e \in E$ in task graph, denotes precedence or data dependencies among tasks. If two tasks, τ_i and τ_j , are connected by an edge then the execution of τ_i must be

finished before τ_i can be started. The positive weight $w(\tau_i)$ associated with node $\tau_i \in V$ represents its computation cost and the non-negative weight $c(e_{ij})$ associated with edge $e_{ij} \in E$ represents its communication workload.

The set $\{\tau_x \in V: e_{xi} \in E\}$ of all direct predecessors of τ_i is denoted by $\text{pred}(\tau_i)$ and the Set $\{\tau_x \in V: e_{xi} \in E\}$ of all direct successors of τ_i , is denoted by $\text{succ}(\tau_i)$. A task $t \in V$ without predecessors, $\text{pred}(\tau_i) = \emptyset$ is named *source* task and task without successors, $\text{succ}(\tau_i) = \emptyset$, is named *sink* task. If task τ_j is scheduled on sensor other than sensors on which its direct predecessor τ_i be scheduled, then a communication between these sensor nodes is required. However, if both of tasks are scheduled on the same sensor, the result delivery latency is considered to be zero and τ_j can start to execute just after τ_i is finished. A feasible implementation of an application must respect all timing constraints and precedence requirements when executed on an underlying architecture. A DAG may have multiple source tasks and one sink task. If there are more than one sink tasks, they will be connected to a pseudo sink task with computation cost equals zero. Fig. 1(b) shows an example of a DAG, where τ_1 and τ_2 are source tasks, τ_6 is a sink task, and τ_4 is the direct successor and direct predecessor of τ_1 and τ_6 , respectively. The weight c_{13} on edge e_{13} represent the communication cost from task τ_1 and τ_3 . The weight $w_1 = 5$ represent the computation cost of task.

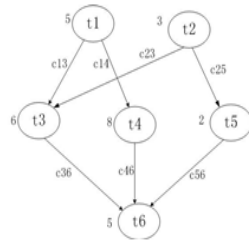
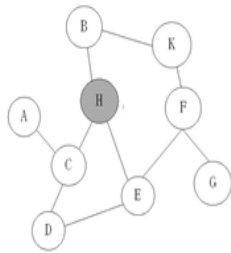


Figure 1 (a) Network Topology

Figure 1(b) Tasks Graph

3.3. Problem Formulation

For a given task graph, $G = (V, E, w, c)$, an initial step maps each task in G into one of the available sensors in $TG = (P, L)$. We use function $\phi: V \rightarrow P$ to represent this task mapping step. Task mapping affects the total communication load because only tasks assigned onto different sensors can generate communication load. The routing path between two sensors is then allocated and the function $\Omega: E \rightarrow L$ is used to denote this routing path allocation step, where L is the set of link sequences. The task scheduling and communication step determine execution order of tasks assigned onto the same sensor and communication time between sensors. We need to know communication delay when task scheduling, hence routing path allocation step is executed first. Tasks voltage assignment step determines the execution clock speed of each task to reduce energy consumption by utilizing what would otherwise be slack time. We use function $\Psi: V \rightarrow C$ to represent the task voltage assignment step, where C is the set of possible clock speeds of tasks.

We can define power-efficient tasks allocation problem for multi-hop WSN as follows:

Given task graph $G = (V, E, w, c)$ and network topology graph $TG = (P, L)$

Find functions of ϕ, Ω, Ψ such that

$$\min F_3 = \sum_{\tau \in V} c(\tau) + \sum_{e_{ij} \in E} w(e_{ij}) + w1 * \max(L(n)) + w2 * \max(E(n))$$

Subject to $\forall \tau \in V, \theta(\tau) \leq d$

- $c(t)$: computation cost of task t in DAG
- $w(e)$: communication cost of edge e in DAG
- n : sensor node n
- $w1$: weight for latency
- $w2$: weight for maximum energy consumption
- $L(n)$: the latency of sensor node n
- $E(n)$: the energy consumption of node n .

The total cost is the summation of the computation and communication cost of all tasks, and the weighted summation of maximum latency and maximum energy consumption among all nodes. Usually, the lifetime of network is the duration of alive time of the node dies first, so the energy consumption of the node consumes the most needs to be minimized. By adjusting $w1$ and $w2$, users can change the cost function to satisfy the requirement of application. Since the genetic selection operation usually chooses the maximum result in the search space, we transform the above minimum fitness function into maximum function: $F_3' = 1 / F_3$.

4. Tasks Allocation Algorithm

4.1. Algorithm Framework

Tasks mapping and tasks scheduling problems could be handled as separate independent problems or these could be handled as one integrated problem. Optimal solution of the integrated problem can provide the best results. However since both mapping and scheduling are computationally hard problems, solving them together is more difficult than solving them one by one. On the other hand, task allocation problem defined in the above involves several interacting steps; it is unlikely that a good solution will be found by optimizing each step independently. We use two nested genetic algorithms to explore the solution space efficiently. There are a GA-based task mapping algorithm (GA-Mapping) and a GA-based routing path allocation algorithm (GA-Routing). Figure 2 shows the design steps of energy efficient task allocation in this paper.

Genetic algorithms imitate the principles of natural evolution to solve search and optimization problems, and are a promising technique for system-level design which has a large solution space. GA is especially suitable for multiple-objective optimization. Starting with an initial population, a genetic algorithm evolves a population using the crossover and mutation operations. Since the performance of genetic algorithm depends on encoding, crossover and mutation schemes, we need to select these schemes carefully. The detail of the algorithm steps will be introduced in the following sections.

4.2. Genetic-based Task Mapping

The more tasks are assigned to a sensor, the more energy it will consume for computations. Energy constraint can be expressed as $E(\phi, P_i) \leq E_{residual}^P$ where $E(\phi, P_i)$ means the energy consumption of a sensor P_i under the task mapping function ϕ , $E_{residual}^P$ is the residual energy of sensor. If there is an edge e between two tasks assigned to the same sensor, its value $w(e)$ is changed to zero. This task mapping step affects the total communication load. We will represent a task mapping solution as an array of integers. For example, in Figure3, task $t1$ is mapped to sensor $p6$ in the individual $p1$. Our crossover operation is two-point crossover which is widely used in GAs. Both parent individual $p1$ and $p2$ are divided at the same two points and the child individual $c1$ is generated from the first part of $p1$, the second part of $p2$ and the third part of $p1$. Our mutation operation changes the values of randomly selected genes into new values, which make up

a new individual. Our selection operation chooses the population with the highest fitness computed through the fitness function. The objective of mapping step is the same with task allocation problem. For each individual, we perform routing path allocation, task scheduling, communication scheduling and task voltage scaling to evaluate the fitness value.

4.3. Genetic-based Routing

Topology of WSN can be depicted by graph like figure 4, an equivalent expression is tree model. Any node in graph can be root of tree and each path from root to leaf node corresponds to a valid routing path in network graph. In order to simplify the genetic operations, in this paper, sensor network is expressed by a tree junctions and the genes are expressed by the tree junctions. By this coding method, the length of each chromosome is the same and the genetic operations are carried out in the tree junctions. To explain this procedure, see the topology graph in Fig.4. Let node S be the source and node D be destination. All routes are expressed by the network tree model shown in Fig.5. In the network tree model, each tree junction is considered as a gene and the path is represented by the chromosome. By using this gene coding method, the routing loops can be avoided. Fig.6 shows the chromosome coding. The genes in a chromosome have two states "active" and "inactive". A gene is called active if the junction is in the route; otherwise the gene is in "inactive" state. Active state is represented by

child path: $S \rightarrow B \rightarrow H \rightarrow G \rightarrow E \rightarrow D$ and $S \rightarrow H \rightarrow C \rightarrow D$. Mutation operation aims to keep the diversity of populations and avoid getting into local optimal solution. Similarly, mutation requires guarantee legal path. First randomly select the population to be mutated and randomly delete some edges in this path. Find out the disjoint branches and make them connected using the other edges in the network tree. A new legal path can be formed by this way.

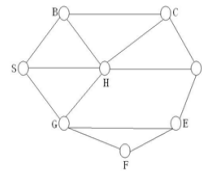


Figure 4. Network topology of routing

1	2	3	4	5	6	7	8	9
BHG	CH	DCBG	EFH	HD	CDG	EF	DC	DG

Figure 6. Gene coding for routing path allocation

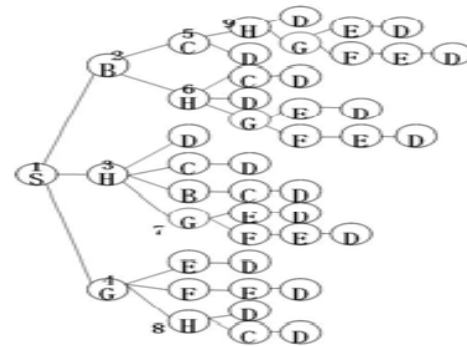


Figure 5. Network tree model

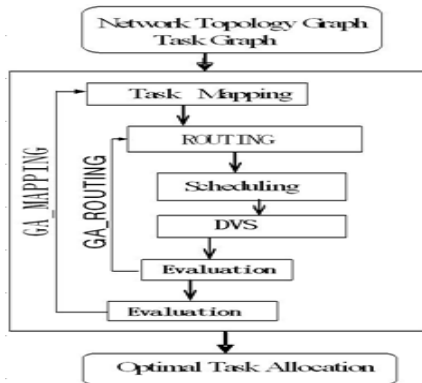


Figure 2. Algorithm Framework

	t1	t2	t3	t4	t5	t6
P1	6	9	5	3	7	1
P2	2	8	1	7	9	4
C1	6	9	1	7	7	1
C2	2	8	5	3	9	4

Figure 3. Two-point crossover in task mapping

Crossover operation must guarantee obtain legal routing path. We use single point crossover because simple operations are needed to get fast responses. We must be careful to choose the crossover site because crossover operation can only be executed among those paths which have meeting point. Meeting point is the common node in paths. For instance, two parent paths $S \rightarrow B \rightarrow H \rightarrow C \rightarrow D$ and $S \rightarrow H \rightarrow G \rightarrow E \rightarrow D$ can be crossed at node H to generate two

Fitness function is used to evaluate the populations. In routing path allocation step, each population represents a path from source node to destination node. There are two metrics to evaluate a path's performance. The first one is energy cost. The less energy consumption, the good the path is. The second one is communication delay. In order to complete the real-time application before deadline, path with small delay is preferable. Hence, we define the fitness function as the follows: $f(P)=1/(\sum_{l \in P} c(l) \times DT)$, here $c(l)$ is the cost of link l . DT is the actual delay time from source node to destination node. The exact delay can be computed only after task scheduling and communication scheduling, so this algorithm will be evaluated after the scheduling.

4.4. Scheduling

For task scheduling, we adopted a list-scheduling algorithm which uses the mobility of each task as its priority. The mobility of a task is defined as the difference between the ASAP start time and the ALAP end time. To get these times, we need to know the communication delay of an edge. The communication delay can be computed easily by communication scheduling procedure.

List-scheduling^[12] has been widely adopted in task scheduling, in this paper, we assign tasks priority according to the bottom level of each task. Bottom level is defined recursively as the following:

$$bl(\tau_i) = w(\tau_i) + \max_{\tau_j \in \text{succ}(\tau_i)} \{c(e_{ij}) + bl(\tau_j)\}$$

can be computed by traveling the task graph from bottom to top. To meet the application deadline, it is necessary to compute the start time and the finish time of each task. If a task is mapped onto a sensor node different to its preceding task,

then a communication is needed between these two sensors. A routing path can be attained through algorithm proposed in section VI.C; however the communication links in this path might be occupied by other communication at the same moment, hence, communication scheduling is necessary to avoid such conflict. The start time and the finish time of a communication edge can be determined through the following algorithm.

In pseudocode in Fig. 7, $t_s(e_{ij}, L_k)$ and $t_f(e_{ij}, L_k)$ denote the start time and finish time of communication edge e_{ij} on link L_k respectively. $EAT L_k$ is the earliest available time of link, $\sigma(e_{ij}, L_k)$ is the actual communication time. Line 8 update the earliest available time of each link according to the finish time of the communication so that other communication task can reuse the link later. Line 10 shows that the finish time of a communication on a path is the finish time of the communication on the last link in this path. The last line records the total energy consumption of this path; $D(R)$ is the distance between source node and destination node. By far, we can evaluate the routing path using the two metrics: energy cost and communication delay.

Input: communication edge e_{ij} in DAG ,
routing path R attained by GA-RPA

Output: the start and finish time of each link in path R

Comm_scheduling ($e_{ij}, R[L_1, L_2, \dots, L_m]$)

1. FOR k FROM 1 TO m DO
 - Find the earliest available time of each link L_k ,
2. IF $k=1$
3. $t_s(e_{ij}, L_k) = \max\{EAT L_k, t_f(\tau_i)\}$
4. ELSE^o
5. $t_s(e_{ij}, L_k) = \max\{EAT L_k, t_f(e_{ij}, L_{k-1}) - \sigma(e_{ij}, L_k), t_s(e_{ij}, L_1)\}$
7. $t_f(e_{ij}, L_k) = t_s(e_{ij}, L_k) + \sigma(e_{ij}, L_k)$
8. $EAT L_k = t_f(e_{ij}, L_k)$
9. END FOR
10. $t_f(e_{ij}, R) = t_f(e_{ij}, L_m)$
11. $Energy(R) = \alpha + \beta D(R) \gamma$

Figure 7. Pseudocode of communication schedule

4.5. Dynamic Voltage Scaling

One promising low-power techniques for energy-limited embedded system is to scale the voltage of tasks. For the task voltage assignment, we take advantage of the voltage and clock speed selection algorithm proposed by Schmitz and Al-Hashimi^[4]. This algorithm determines the operating speed of each task assigned on the DVS-enabled PE. It first estimates the slack time of each task considering the deadline and precedence constraint. It then calculates $\Delta E(\tau_i)$ for a task τ_i which has slack time. $\Delta E(\tau_i)$ is the energy gain when the time slot for τ_i is increased by Δ_t (with a lower clock speed). After increasing the time slot for the task τ_i with the largest $\Delta E(\tau_i)$, by a time increment Δ_t it repeats the same sequence of steps until there is no task with slack time.

5. Experiments Evaluations

The goals of our experiments are (1) to measure and compare the energy consumption of our nested optimized

technique against the random-based approach; (2) to evaluate the optimized schedule length of our method against the list-scheduling heuristic approach.

The cost function defined in this paper is a general form; therefore we can present general metric which can accommodate various tradeoffs by adjusting parameter w_1 and w_2 . Performance metrics include energy consumption and schedule length.

Genetic algorithm used the following parameters throughout the simulation: Population size = 20; Crossover probability=1; Mutation probability = 0.01; Max number of iterations = 1000. The nested genetic algorithm proposed in this paper can guarantee to coverage to the global optimal solution for the following three reasons: 1) the crossover possibility is 1; 2) the mutation possibility is a number in (0,1); 3) selecting population according to proportion and always keep the best population. However, the high search complexity of genetic algorithm usually baffle its application, therefore the performance of our algorithm can be improved by limiting the total iteration number.

We started by estimating the efficiency of each optimization technique at the task mapping, routing path allocation and task voltage scaling steps. For our experiments, we generated random task graphs g_1 to g_{16} .

Figure 8 shows the energy consumptions of application tasks under various optimization configurations. In this group of experiments, parameter w_1 and w_2 are set to be zero. Therefore, the objective is to minimize the total energy consumption under the real time constraint. Experiments results were normalized against the energy consumption obtained by a task allocation technique which uses random task mapping, shortest path routing and no task voltage scaling.

The first bar for each tasks graph represents the result when we applied only task DVS. The second bar represents the result when we used the GA-Routing algorithm for routing path allocation, as well as task voltage scaling technique (DVS +R). The third bar shows the result when the GA-Mapping algorithm for task mapping is also applied (DVS+R+TM). The energy consumption is reduced by 16%, 26.5%, and 39.6% on average by the DVS, DVS+R, and DVS+R+TM techniques respectively. These reduction ratios are dependent on the characteristics of the task graph (e.g., slack time and communication load) and the performance of the random configurations. For example, the dynamic voltage scaling (DVS) showed small energy reductions because the random task mapping and shortest path routing generate little slack time. From these results, we realize that all steps have large effects on energy saving, and it is necessary to optimize the energy consumption at all steps. We did not compare the task scheduling step with other techniques because the list scheduling is universally popular. Table 1 shows the execution time of our nest genetic algorithm.

We also compare the quality of the solution produced by this paper with those produced by other heuristic algorithm. In this set of experiments, parameter w_1 is set to be one and w_2 is set to be zero. The objective is to obtain the minimal schedule length while meeting application's QoS requirements such as time and energy constraint. Table 1 compared the schedule length of the nested genetic algorithm and the listing algorithm along with the optimal schedule for the random task graph. The solution obtained by the genetic algorithm is better than the list scheduling algorithm and is with 10% of the optimal solution.

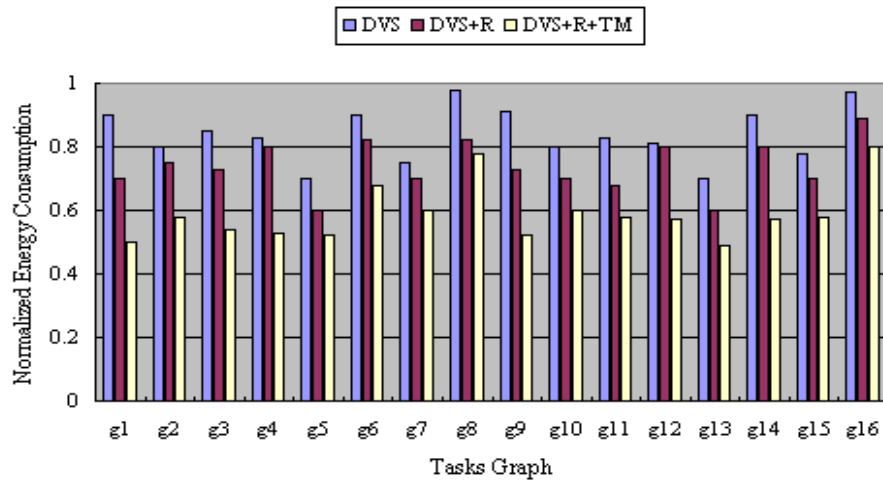


Figure 8. Effect of the nested genetic-based algorithm for task allocation($w_1=0, w_2=0$)

Table 1: Execution time of DVS+R+TM Algorithm

TG	Nnode/Nedge	Time(sec)	TG	Nnode/Nedge	Time(sec)
g1	20/32	48.2	g9	30/40	43.9
g2	15/24	16.8	g10	22/35	37.9
g3	18/32	34	g11	14/27	26.3
g4	40/77	70.2	g12	16/33	32
g5	20/27	36	g13	21/40	50
g6	15/30	19.5	g14	14/19	70
g7	12/19	16	g15	40/60	65
g8	11/24	15.4	g16	28/19	40

Table 2: Comparison of Optimal Schedule (OPT), Nested Genetic algorithm (GA) and List-Scheduling (LA) ($w_1=1, w_2=0$)

Task Graph	OPT	GA	LA	(GA-OPT)/OPT%
g1	147	156	198	6.1
g3	240	256	290	6.7
g5	280	305	339	8.9
g7	125	136	157	8.8
g9	347	358	370	3.2
g11	132	145	160	9.8
g15	438	455	475	3.9

We also compare the quality of the solution produced by this paper with those produced by other heuristic algorithm. In this set of experiments, parameter w_1 is set to be one and w_2 is set to be zero. The objective is to obtain the minimal schedule length while meeting application's QoS requirements such as time and energy constraint. Table 1 compared the schedule length of the nested genetic algorithm and the listing algorithm along with the optimal schedule for the random task graph. The solution obtained by the genetic algorithm is better than the list scheduling algorithm and is with 10% of the optimal solution.

Prolonging network lifetime as much as possible is the common goal of resource-limited wireless sensor networks. Lifetime has been defined in various ways. In the simplest way, a network may be considered alive when any of the sensors is alive. Making energy consumption balanced can avoid the network dying too soon. We achieve such object by setting parameter w_1 to zero, w_2 to 1. In figure9, we compare the lifetime improvement of nested GA-based approach in this paper with energy-balanced heuristic approach proposed in [9]. We tested tasks graph with tasks number 15,20,25,30,35 and 40, in a real sensor network with 20 nodes deployed in a square field of 200X200 meters. The communication radius is 20 meters. Each node can communicate with the cluster head in a multi-hop way. From figure9 we can see that our approach can get a lifetime improvement of 3.5x, while the heuristic approach in [9] can get a lifetime of 1.5x in average.

Performance of the nested genetic algorithm is shown in Fig.10. This figure shows the generation number versus the fitness values which is decided based on the fitness function f . The high fitness means that the task allocation solution has low delay and energy consumption. From the figure we

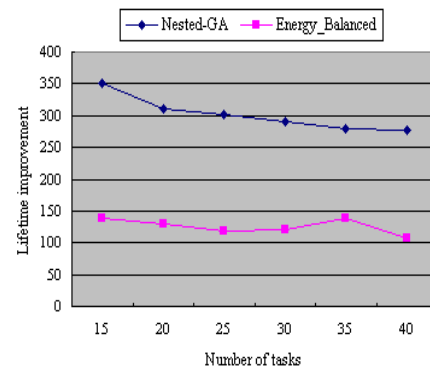


Figure 9. Performance of Nested Genetic Algorithm ($w_1=0, w_2=1$)

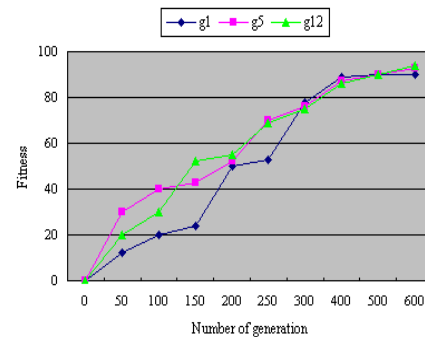


Figure 10. Performance of Nested Genetic Algorithm

can see that our nested genetic algorithm always coverage within a small number of generations.

6. Conclusion

In this paper, we proposed an energy- and QoS-aware task allocation algorithm to optimize the energy consumption of applications submitted to a multi-hop cluster-based wireless sensor network. The objective is to minimize the total energy consumption of the system under real-time constraint. The proposed algorithm saves energy at various steps, i.e. task mapping, routing, and task voltage scaling. We use nested genetic algorithm to explore the solution space efficiently. Simulations on random generated tasks graphs show that our algorithm reduced the energy consumption of the wireless sensor network compared with the existing algorithm which uses random task mapping and minimal path routing. Our future work is to develop energy-efficient and QoS-garanteed distributed task allocation algorithm for dynamic sensor networks.

Acknowledgments

The authors wish to thank our mentor, Professor Jianzhong Li, as well as the reviewers for their comments on this paper.

References

- [1] Akyidiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Elsevier Computer Networks Journal*, 38 (4) 393–422.
- [2] Dogan, A., O'zgu'ner, F (2002). Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogenous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13 (3) 308 - 323.
- [3] Hu, R., Marculescu.(2004).Energy-Aware Communication and Task Scheduling for Network-on-Chip Architectures under Real-Time Constraints. *In: Proc. Design, Automation and Test in Europe Conf., Paris, France.*
- [4] Schmitz, M. T., Al-Hashimi, B. M(2001). Considering Power Variations of DVS Processing Elements for Energy Minimisation in Distributed Systems. *In: Proc. International Symposium on System Synthesis*, 250-255.
- [5] Giannecchini, S., Caccamo, M., Shih, C.-S(2004). Collaborative resource allocation in wireless sensor networks. *In: Proc. of Euro micro Conference on Real-Time Systems*, 35-44.
- [6] Basu, P., Ke, W., Little, T. D. C (2003). Dynamic task-based anycasting in mobile ad hoc networks. *Mobile Networks and Applications*, 8 (5) 593-612.
- [7] Kumar, R., Wolenetz, M., Agarwalla, B., Shin, J., Hutto, P., Paul, A., Ramachandran., U (2003). DFuse: A framework for distributed data fusion. *In: Proc. of The ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 114-125.
- [8] Shible, S., Castain, R., Siegel, H. J. Maciejewski, A. AT. Banka, T., Chindam, K., Dussinger, S., Pichumani, P. Satyasekaan, P., Saylor, W., Sendek, D., Sousa, J., Sridharan, J., Sugavanam, P., Velazco, J (2004). Static mapping of subtasks in a heterogeneous ad hoc grid environment. *In: Proc. of Parallel and Distributed Processing Symposium.*
- [9] Yu, Y., Prasanna, V.K (2005). Energy-balanced task allocation for collaborative processing in wireless sensor networks. *Mobile Networks and Applications*, 10 (1-2) 115–131.
- [10] Tian, Yuan., Ekici, Eylem., F'usun O'zgu'ner.(2005). Energy-Constrained Task Mapping and Scheduling in Wireless Sensor Networks. MASS 2005 Workshop RPSMN05.

- [11] Goldberg, D.E(1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.
- [12] Adam, T. L. Chandy, K. M . Dickson, J. R (1974). A comparison of list schedules for parallel processing systems. *Communications of the ACM*, 17, 685–689.
- [13] Conard, M. Marrakchi, M., Robert, Y., Trystram.D(1988). Parallel Gaussian elimination on an MIMD computer. *Parallel Computing*, 6:275–295.
- [14] Cormen, T. H. Leiserson, C. E R. L.(1990). *Rivest. Introduction to Algorithms*. MIT Press.
- [15] Zhang, Y., Hu, X., Chen, D. Z(2002). Task scheduling and voltage selection for energy minimization. *In: Design Automation Conference (DAC)*.
- [16] Zhu, D., Melhem, R., Childers, B (2001). Scheduling with dynamic voltage/speed adjustment using slack reclamation in multi-processor real-time systems. *In: IEEE Real-Time Systems Symposium (RTSS)*.
- [17] Poornachandran, R., Ahmad. H., Cam, H (2005). Energy-Efficient Task Scheduling for Wireless Sensor Nodes with Multiple Sensing Units. *Proc. of IWSEASN'05*.
- [18] Vercauteren, T., Guo, D., Wang, X. (2005). Joint multiple target tracking and classification in collaborative sensor networks. *IEEE Journal on Selected Areas in Communication*, 23 (4) 714–723.



JingHua Zhu is a Ph.D. candidate in Computer Science. Her research interests include Qos of sensor network.



JianZhong Li is a professor in Computer Science and supervise Ph.D candidates. His research interests include data mining, data warehouse, sensor network, grid and bioinformatics.



Hong Gao, is a professor, in Computer Science and supervise Ph.D candidates. Her research interests include data mining, data warehouse and sensor network.