CONFIGURABLE TASK MAPPING FOR MULTIPLE OBJECTIVES IN MACRO-PROGRAMMING OF WIRELESS SENSOR NETWORKS

Himanshu Singh[1] and R.S. Singh[2]

[1,2]Department of Computer Engineering, Institute of Technology, Banaras Hindu University, Varanasi, India.

[1]himanshu.singh.cse07@itbhu.ac.in, [2]ravi.cse@itbhu.ac.in

ABSTRACT

Macro-programming is the new generation advanced method of using Wireless Sensor Network (WSNs), where application developers can extract data from sensor nodes through a high level abstraction of the system. Instead of developing the entire application, task graph representation of the WSN model presents simplified approach of data collection. However, mapping of tasks onto sensor nodes highlights several problems in energy consumption and routing delay. In this paper, we present an efficient hybrid approach of task mapping for WSN – Hybrid Genetic Algorithm, considering multiple objectives of optimization – energy consumption, routing delay and soft real time requirement. We also present a method to configure the algorithm as per user's need by changing the heuristics used for optimization. The trade-off analysis between energy consumption and delivery delay was performed and simulation results are presented. The algorithm is applicable during macro-programming enabling developers to choose a better mapping according to their application requirements.

KEYWORDS

Wireless Sensor Networks, Macro-programming, Genetic Algorithm, Task Scheduling, Energy Conservation

1. INTRODUCTION

Wireless Sensor Network is becoming a popular research field in distributed systems due to its wide applications. Consider the network requirements of a particular farm scenario. A farmer, having different fields at different locations, may wish to collect normal data about the field conditions for example – temperature, water level in the soil and humidity continuously at regular intervals; and would also like to be informed about any event happening on the field like sudden flood and fire. A wired information system would be too costly for this case. The most feasible solution for such a system would be to use a wireless sensor network across all the fields where nodes have various sensors (also known as motes) attached to it for taking the measurements and a radio for wireless communication. To maximize the network lifetime, energy consumption by batteries should be minimum since it is not feasible to replace the batteries of nodes once deployed in farm. Also, the data delivery time should be very low, especially for fire and flood notifications.

Now, the farmer may not be educated enough to program the sensor nodes in WSN specific operating systems like TinyOS [1], Contiki [2], MantisOS [3] and to optimise the above mentioned parameters. Macro-programming [4] constructs allow the farmer to specify his requirements in the form of a task graph and a network graph. The tasks may be different types of sensing tasks, filtering tasks, routing tasks etc which may be specific to the requirements of a particular field. The network graph consists of network of sensor nodes on which the tasks have to be executed. Some of the macro-programming architectures like KairOS[5], Regiment [6] and COSMOS [7], which provide such high level abstractions to define the system model, face the problem of efficient mapping of one or more tasks on some sensor nodes. In this paper, we study this task mapping problem using the Genetic Algorithm [8]. The heuristic used is based on the optimization over multiple parameters –total energy consumption, efficient routing, and soft real time delivery requirements. In section 2, we present previous research in this area and their difference with our work. In section 3, we formulate the problem and then present the Hybrid Genetic Algorithm (HGA) in section 4. Simulation results of trade-off analysis have been shown in section 5 and we conclude in section 6 illustrating how different farmers can control optimization of one parameter (like energy) on the cost of others (like delivery time).

2. RELATED WORKS

Task mapping is a widely studied problem in parallel and distributed systems. But in wireless sensor networks, it gained attention of researchers due to energy constraints and delivery time requirements. Most of the researches [9], focus on reducing the energy consumption by efficient MAC protocols which introduce sleep-awake scheduling of nodes. Unlike our work, they do not consider the possible optimization in routing and efficient task allocation. The work in [10] addresses the problem of task placement on sensor nodes, but they consider single hop networks only. The work in [11] also present the task mapping problem, but they focus only on the objective of minimizing the total energy consumption. With macro-programming becoming a popular field, work in [12] adopted a greedy method of task mapping but it maps the tasks only on the tree topology and not general network graphs. Some macro-programming languages like KairOS present the concept of dividing the flow of program into 'Control Flow Graphs' (CFG) and mapping CFGs onto sensor nodes. KairOS minimizes total number of edges that crosses CFG. It does not clearly offer any solution to energy constraints. [13] offers a solution based on greedy method particularly for macro-programming, but they do not offer any method to configure the optimization according to needs of application. COSMOS[7] and Srijan [14] which use mOS and mPL languages, program the WSNs using the concept of task graph. These applications do not provide the support of configuring the task mapping in application; however these act as perfect platforms wherein the HGA algorithm (presented in section 4) can be used to map the task graphs on motes with optimizations over multiple-objectives and allowing users to configure these optimizations by changing the heuristics used.

3. PROBLEM IDENTIFICATION

Tasks which need to be mapped onto sensor nodes, can be sensing tasks, routing tasks, activation tasks etc. Since these tasks need to execute in a particular order (Sensing, then routing, then activation) the tasks can be modelled as a directed acyclic graph. The sensor node which form a wireless network, are modelled as undirected network graph.

3.1. Objectives

Objective of this work is to obtain a mapping of every task into some sensor nodes, then ascertaining the routing path between the predecessor and successor and finally ascertaining start up time and scheduling order of every task.

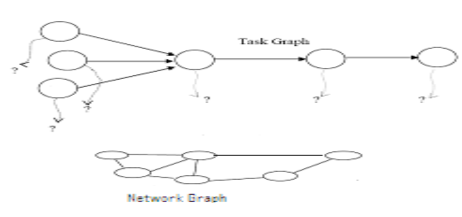

Figure 1: Task Graph needs to be mapped on Network Graph

Multi-objective optimization: In a real time environment, it becomes of a greater necessity to execute certain tasks with higher priority within given deadline and ensuring that minimum energy is consumed in completing all the tasks. Such requirements need simultaneous optimization of multiple objectives. Solutions to such problems are usually computed by combining multiple objectives into a single criterion to be optimized. The combining of objectives has the advantage of producing a single compromised solution, and normally does not require any human intervention. However, the problem is that if the optimal solution cannot be accepted because we chose an improper setting of the coefficients of the combining function, user intervention may be required. Such interaction should be easy and high-level. Therefore, we consider finding an optimal schedule as a problem of multi-objective optimization with non-measurable objectives. Genetic Algorithms are capable of multi-objective optimization. The algorithm explores a set of solutions that can only be improved in one way by being degraded in another instead of collapsing entirely. Therefore, we use a multi-objective GA which independently considers multiple objectives, and finds a set of solutions that satisfy all of the following objectives and simultaneously allows users to increase/decrease optimization of any parameter.

1. Minimize total schedule time cost
2. Minimize total schedule energy cost
3. Minimize tardiness value to achieve soft real time deadline

3.2. Directed Acyclic Task Graph

Let the task graph be *TG*= (*V*, *E*), where *V* be the set of vertices and E be the connecting edges. A vertex V_i corresponds to a task T_i. Each T_i is identified by ($Energy_i$, $Time_i$) where $Energy_i$ denote the energy and $Time_i$ is the time required for execution of T_i.. E_{ij} connecting $<V_i, V_j>$ denotes that the V_j could not start until and only until the task V_i has been executed. The weight on the directed edge $C_{i,j}$ between V_i and V_j denotes the amount of data transmitting from task T_i to T_j. Figure 1 illustrates a DAG task Graph T.

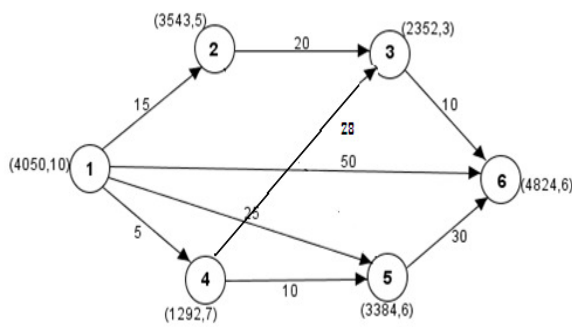

Figure 2: Task Graph

3.3. Network Graph

The network graph NG= (*P*, *L*) is used to denote the wireless sensor network. *P* is the set of sensor nodes which process the some tasks. *L* is the set of communicating undirected edges. The weight of edge $dist\_nodes_{i\ j}$, $comm\_time\_cost_{i\ j}$, $w_{i\ j}$ respectively denotes the distance, the communicating delay, the energy consumption for transmitting one unit data by vertex P_i and receiving one unit data by vertex P_j when data is transmitted between vertex P_i and P_j.

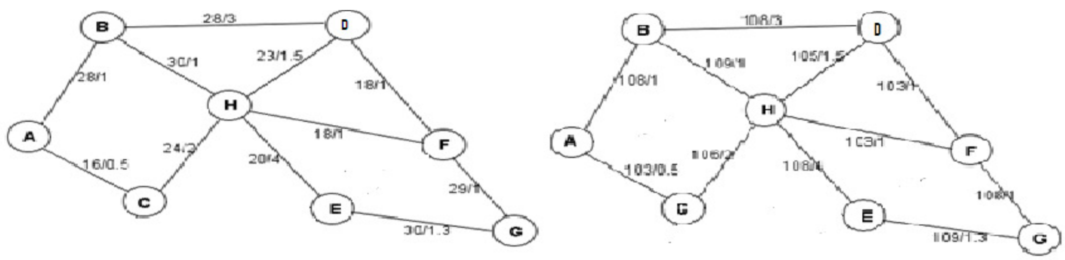

Figure 3 (a) Network Graph denoting distance (b) Network Graph denoting energy

In Figure 3(a), the data on the edge denote the physical distance between two nodes. In Figure 3(b), the data on the edge denote the energy consumptions for one unit transmitting of data.

We use the energy model presented in [15]. Energy is expended to serve the purpose of: (1) digital electronics, *E_ELEC,* (actuation, sensing, signal emission/reception), and (2) communication, *EAMP*. *EAMP* varies according to the distance *dist_nodes* between a sender nodes and a receiver. To transmit k bits for a distance *dist_nodes,* the radio expends $k \times$ (*E_ELEC* + *EAMP* $\times$ *dist_node*^{n}) J, where $n = 2$ for *dist_nodes* < D*o*, and $n = 4$ for *dist_nodes* >= D*o*, and D*o* is a constant threshold distance which depends on the environment. To receive k bits at the receiver, the radio expends $k \times$ *E_ELEC* J.

4. HYBRID GENETIC ALGORITHM

4.1. Background and Definitions:

Let initial energy of each node is e_0. As the system operates, the energy remaining at node k be e_k. The Execution Cost Matrix E is a single dimensional matrix that represents energy cost of execution of task i. Transmission Cost (E_{trans}) is calculated as the energy required/spent to transmit data from every single node to every other node in the network.

Transmission energy cost, when V_i transmits data to V_j, is the sum of the energy consumption for transmitting unit data ($W_{m\,n}$ = *E_ELEC* + *EAMP* $\times$ *dist_node*^{n}) on every edge in path of V_i and V_j taken product with data C_{kl}.

$$E_{trans(i,j)} = \lambda \sum_{(m,n)=(k,x)}^{(x,l)} C_{i,j} w_{m,n}$$

The whole energy consumption "Energy" of accomplishing task is the sum of the energy spent in completing task t_i on $node_j$ and transmitting data. That is:

$$\text{Energy} = \sum_{i=1}^{N} E(t_i, node_j) + E_{trans}$$

To formulize the objective of meeting the deadline of each task, let us assume that we have a task, say t_i, which has an execution time (e_i) and a deadline (d_i).

We consider two types of times: earliest and latest start times of each task. The earliest start time of task t_i (estStT[t_i]) is the length of the longest path from the entry task to t_i.

$$estStT[t_j] = \begin{cases} 0 & \text{if } \neg\exists t_i : (t_i, t_j) \in E \\ \max\limits_{t_i \in P(t_j)} \{estStT[t_i] + e_i\} & \text{otherwise} \end{cases}$$

The latest start time of task t_i (lstStT[t_i]) is defined as the latest time at which t_i can start, such that t_i and all successors of t_i have a chance of meeting their deadlines.

$$lstStT[t_j] = \begin{cases} d_j - e_j & \text{if } \neg\exists t_i : (t_j, t_i) \in E \\ \min\left\{ \min\limits_{t_i \in S(t_j)} \{lstStT[t_i] - e_j\}, d_j - e_j \right\} & \text{otherwise} \end{cases}$$

For achieving soft real time deadlines, our approach considers the communication time between two tasks and time taken to execute those tasks in calculating the latest start times of tasks. Tardiness of a task is defined as the difference between actual start time in schedule and latest start time of that task. The cost of a schedule is defined in terms of sum of maximum tardiness among all tasks. The objective is to find a schedule with the maximum tardiness minimized.

4.2. Algorithm

4.2.1. Algorithm 1 - Hybrid Genetic Algorithm

Input**:** Network topology, DAG task graph, initial population size: init_population, evolutional generation: T.

Output**:** Task mapping scheme, scheduling scheme and routing scheme.

Step1: i=1; // evolutional generation count
Step2: generate initial GA_ population (Pop) using create_initial_Chromosome
Step3: Calculate schedule_time(), schedule_energy(), tardiness() of Population
Step4: Repeat
Step5: calculate the fitness value f(Pop[i]) of the individuals in population using calculat e_fitness. // computing individual fitness
Step6: [child1, child2] = Roulette_Selection(Pop[i]);
// Select the individuals having optimal fitness value using Roulette criteria[16]
Step7: Pop[i+1] ← OptimalSelect(Pop[i]); // selection operation
Step8: GTi ← Hybrid_operation(child1, child2); //hybrid operation
Step9: GSi+1 ← Evolove(GTi) //mutation evolution operation
Step10: i=i+1; Calculate schedule_time(), schedule_energy(), tardiness() of newly generated population
Step11: until (i>Threshold) or (termination criterion is met).
Step12: Ouput Best individual found so far;

4.2.2. Algorithm 2 – create_Initial_Chromosome

Supposing the number of DAG tasks which are scheduled to m sensor is n. The allowable energy deviation is ΔE.

Step1: *n* tasks are mapped to sensors.
Step2: mapping a routing path for every edge of DAG
Step3: Get(i , v1 , v2) ; //the start point and endpoint of the edge i is respectively mapped to nodes v1 and v2 in sensor Network.
If v1=v2 then there is no use for routing
Else
Find (v1,v2,k) // Use Floyd-warshall algorithm to find shortest path
Step4: Select (v1,v2,k); // select the least energy cost path from the k paths.
End

4.2.3. Algorithm 3 – Calculate_fitness(Pop[i])

Step1: TMax = Maximum time of execution of tasks in Pop;

EMax = Maximum energy of execution of tasks in Pop;
TDMax = Maximum tardiness of all tasks.

Step2: Fitness of i^{th} Chromosome, pop[i].fitness = W1*fitness_time/Tmax + W2*fitness_energy/Emax + W3*fitness_tardiness/TDmax; where fitness_time = Tmax - pop[i].schedule_time and W1, W2, W3 are user defined weights of different objectives.

4.2.4. Algorithm 4 – Hybrid_operation(child1, child2)

```
Step1: //Crosses over two individuals pop[child1] and pop[child2]
Step2:  if (random(0,20) > 15) then num_bits = random(M/2,M) ; //M is no. of tasks
        else      num_bits = random(0,M/2) ;
Step3: for i = 0 to num_bits            //Mutation between child1 and child2
                pos = random(0,M) ;
                temp = pop[child1].schedule[pos] ;
                pop[child1].schedule[pos]=pop[child2].schedule[pos] ;
                pop[child2].schedule[pos] = temp ;
        End
```

4.2.5. Algorithm 5 – Evolve((offspring)

```
M is the number of tasks
Step 1:  Initialize num_bits = 0.
Step 2:  if (random(0,20) > 15)
                num_bits = random(M/2,M)
         else
                num_bits = random(0,M/2)
Step 3:  for ( i=1 to num_bits)
         Assign a random node to a random node;
```

5. PERFORMANCE AND ANALYSIS

5.1. Simulation scene and parameters

The HGA algorithm was simulated by varying the number of tasks and number of nodes. The energy cost of performing tasks is taken randomly between 1000J and 5000J, the data to be transferred is a random number between 5 to 50 bits, and the time delay for communicating between two nodes is a random number between 1 and 5. Distance between two nodes is a random number from 10 to 100 units. The transmission energy is calculated as described in section 3.2. When the energy of a node becomes less than 0 J, the node is believed as dead and is excluded for further scheduling in particular population. The weights of optimization W1 (weight of energy optimization), W2 (weight of total time of execution of tasks in the schedule), W3 (weight of total tardiness of all tasks) are varied and based on simulation using these parameters; we now discuss the results in next subsection.

5.2. Results

5.2.1. Comparison of Total Schedule Time

As observed in Figure 4 and 5, when energy minimization is priority in figure 5, the schedule time using Hybrid objective is larger than that in figure 4 where time minimization is priority. To illustrate, Figure 4 shows the total time of execution of all tasks when W2 (weight of time requirement) is larger than W1 (weight of energy optimization), whereas Figure 5 shows total time of execution when W1 is larger i.e. energy optimization is priority.

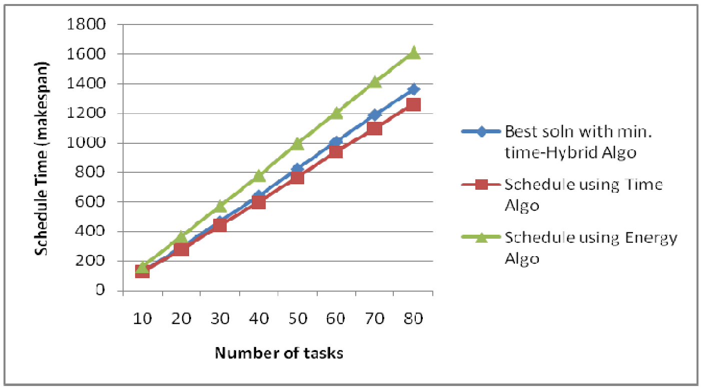

Figure 4. Best solution with minimum time only Vs Best solution with minimum energy only Vs Best Solution using Hybrid objectives with time as priority

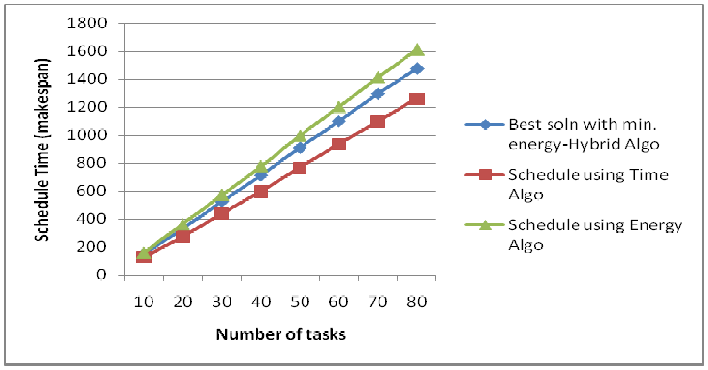

Figure 5. Best solution with minimum time only Vs Best solution with minimum energy only Vs Best Solution with Hybrid objectives with energy as priority

5.2.2. Comparison of Total Schedule Energy

Figure 6 and 7 show total energy consumption of solution. The energy consumption is larger when time minimization is priority for the user, shown in Figure 6, than when energy minimization is priority, shown in Figure 7.

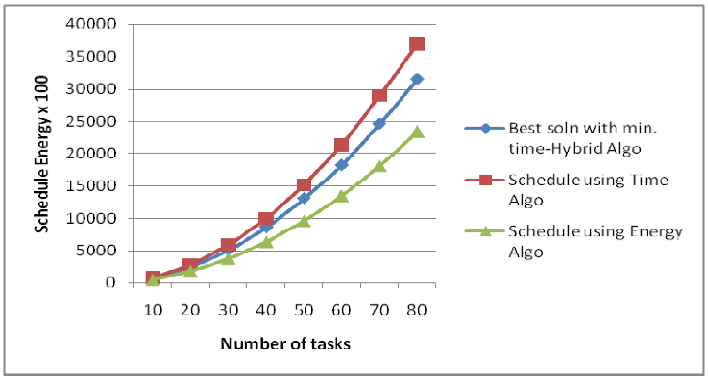

Figure 6. Best solution with minimum time only Vs Best solution with minimum energy only Vs Best Solution using Hybrid objectives with time as priority

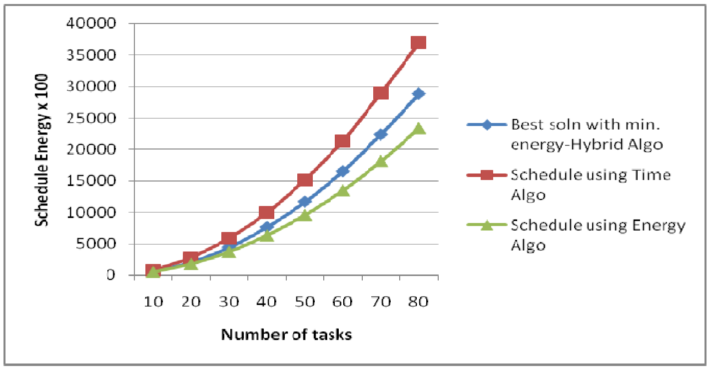

Figure 7. Best solution with minimum time only Vs Best solution with minimum energy only Vs Best Solution with Hybrid objectives with energy as priority

5.2.1. Comparison of Total Tardiness

When objective of achieving soft real time deadlines are included, the tardiness value of best schedule decreases as shown in Figure 8.

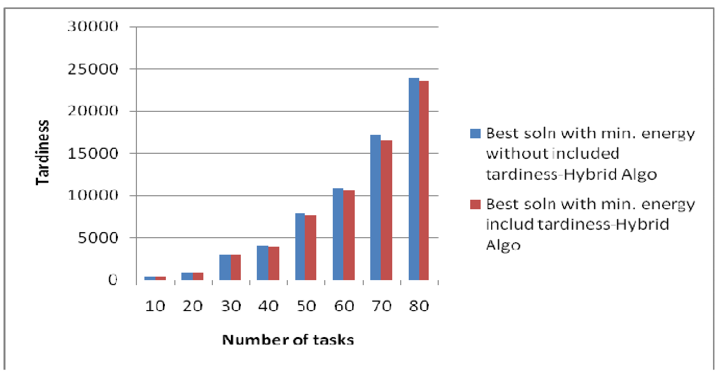

Figure 8. Best Solution using Hybrid Algorithm without considering tardiness value Vs Best Solution w.r.t energy using Hybrid Algorithm considering tardiness value

6. CONCLUSIONS

For scheduling tasks with limited energy in WSN, we implemented Hybrid Genetic Algorithm that will find a mapping considering many objectives. Based on heuristic used – fitness value, users can configure the mapping for a particular objective, just by changing the weights W1, W2, W3 etc [Section 4.2.1], thus changing the underlying routing and mapping scheme. Although Genetic Algorithm requires high computing, the algorithm will be useful particularly in macro-programming with on-the-fly reprogrammable sensor nodes where only one time mapping is required during macro-programming.

REFERENCES

[1] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An operating system for wireless sensor networks,” in Ambient Intelligence. New York, NY: Springer-Verlag

[2] Dunkels, A., Gronvall, B., and Voigt, T. 2004. “Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors.” In Proceedings of the 29th Annual IEEE international Conference on Local Computer Networks (November 16 - 18, 2004). LCN. IEEE Computer Society, Washington, DC.

[3] Bhatti, S., Carlson, J., Shucker, B., Gruenwald, C., Torgerson, A., and Han, R. 2005. “MANTIS OS: an embedded multithreaded operating system for wireless micro sensor platforms.” Mob. Netw. Appl. 10, 4 (Aug. 2005).

[4] A. Pathak and V. K. Prasanna, “Energy-efficient task mapping for data-driven sensor network macro-programming,” in International Conference on Distributed Computing in Sensor Systems (DCOSS), June 2008.

[5] R. Gummadi, O. Gnawali, and R. Govindan, “Macro-programming wireless sensor networks using kairos,” in International Conference on Distributed Computing in Sensor Systems (DCOSS). Springer, 2005, pp. 126–140.

[6] R. Newton, G. Morrisett, and M. Welsh, “The regiment macroprogramming system,” in IPSN ’07: Proceedings of the 6th international conference on Information processing in sensor networks. New York, NY, USA: ACM, 2007, pp. 489–498.

[7] A. Awan, S. Jagannathan, and A. Grama,“Macroprogramming hetero-geneous sensor networks using cosmos,” in EuroSys ’07: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007. New York, NY, USA: ACM, 2007, pp. 159–172.

[8] Haupt, R.L., Haupt, S.E., “Practical genetic algorithms”, John willy & Sons, 2004.

[9] Li, Y., Ye, W. and Heidemann, J. “Energy and latency control in low duty cycle MAC protocols.” In Proceedings of the IEEE Wireless Communications and Networking Conference, New Orleans, LA, USA, March 2005. SMAC and TMAC

[10] Y. Yu and V. K. Prasanna, “Energy-balanced task allocation for collaborative processing in wireless sensor networks,” MONET, vol. 10, no. 1-2, pp. 115–131, 2005

[11] Y. Tian, E. Ekici, and F. Ozguner, “Energy-constrained task mapping and scheduling in wireless sensor networks,” in IEEE International Conference on Mobile Adhoc and Sensor Systems Conference RPMSN Workshop, 2005.

[12] Z. Abrams and J. Liu, “Greedy is good: On service tree placement for in-network stream processing,” in ICDCS ’06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems. Washington, DC, USA: IEEE Computer Society, 2006, p. 72.

[13] A. Pathak and V. K. Prasanna, “Energy-efficient task mapping for datadriven sensor network macroprogramming,” in International Conference on Distributed Computing in Sensor Systems (DCOSS), June 2008.

[14] A. Pathak and Mahanth K. Gowda, “Srijan: a graphical toolkit for sensor network macroprogramming”, In proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, ACM , New York, USA, 2009.

[15] T. Rappaport, “Wireless Communications: Principles & Practice”. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[16] Fitness proportionate selection, Wikipedia, Retrieved on 2nd February, 2011. http://en.wikipedia.org/wiki/Fitness_proportionate_selection

Authors

Himanshu Singh is an undergraduate student at Institute of Technology, B.H.U, India. His research interests include Computer Networks and Wireless Sensor Networks.

R. S. Singh is Assistant Professor at Institute of Technology, B.H.U, India His research interests include Graph Algorithms and High Performance Computing.