



Parallel Robot Routing by Coarse-Grained Model

* Sherin.T

* PG Student, Department of Computer Science and Engineering, Bharath University, Selaiyur, Chennai-600073, India

ABSTRACT

Wireless sensor network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. This paper mainly focuses on robot routing problems with WSN. It proposes clustering-based parallel genetic algorithm with migration (CBPGA), in which mobile robot gathers all data from all sensors in turn reduces its travel cost. Simulation results confirm data gathering with near-optimal route

Keywords : Clustering-Based Parallel Genetic Algorithm, with migration ; Wireless Sensor

I. INTRODUCTION

A wireless network consisting of a large number of small sensors with low-power transceivers can be an effective tool for gathering data in a variety of environments. The data collected by each sensor is communicated through the network to a single processing center that uses all reported data to determine characteristics of the environment or detect an event. The communication or message passing process must be designed to conserve the limited energy resources of the sensors. Clustering sensors into groups, so that sensors communicate information only to cluster heads and then the cluster heads communicate the aggregated information to the processing center, may save energy[13].

Wireless Sensor Networks (WSNs) have and will play a vital role in our daily lives. Humans have relied on wired sensors for years, for simple tasks such as temperature monitoring, to complex tasks such as monitoring life-signs in hospital patients. Wireless Sensor Networks provide unforeseen applications in this new field of design [6]. From military applications such as battlefield mapping and target surveillance, to creating context aware homes [7] where sensors can monitor safety and provide automated services tailored to the individual user; the number of applications are endless.

Given a set of sparsely distributed sensors in the Euclidean plane, a mobile robot is required to visit all sensors to download the data and finally return to its base. The effective range of each sensor is specified by a disk and the robot must at least reach the boundary to start communication. The primary goal of optimization in this scenario is to minimize the traveling distance by the robot. This problem can be regarded as a special case of the Traveling Salesman Problem with Neighborhoods (TSPN), which is known to be NP-hard [2]. Data gathering can be obtained by employing one or more mobile robots that can autonomously gather data from all sensors.

In previous research, cluster heads consumes more energy during data delivery. These cluster heads deplete energy quickly, and many researches has been focused on this problem and also to decrease the load of the cluster head. However they fails successful data delivery to base stations with sparse sensor distribution environment. In response, Yuan et al. [2] presented methods which helps mobile robot to collect this data with minimal travel costs. However, their travel cost solution fails in situations involving sensors with overlapping communication ranges.

Based on this, we propose a clustering-based parallel genetic algorithm with migration (CBPGA) which helps to find shorter route for the data-gathering mobile robot. This algorithm allows the robot to gather all data from all sensors with reducing travel costs and finds near-optimal route within fewer generations. This is achieved by decreasing the number of visited nodes.

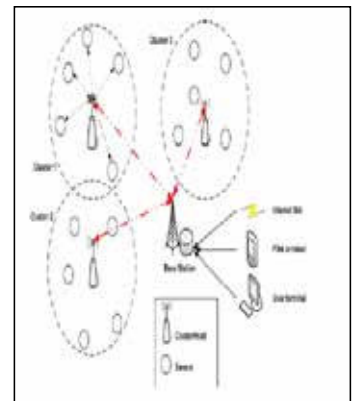


Fig 1 Sensor network

Our results and organization of the paper: In section II, the related research is described. In section III, reflection of the research work are described. In section IV, effective parallel genetic algorithm has been explained in detail. In Section V, we present simulations which provide further insights on the use of mobile robots for gathering data. The paper concludes with the summary of results in section VI.

II. RELATED WORKS

The idea of exploiting mobility for data collection and routing has received some recent attention. Yuan et al. formulate the problem of collecting sensor data using a single robot as a TSPN instance [1,2]. They do not address the time spent in downloading data. Ignoring transmission time can worsen the performance of the system drastically. Tirta et al. presented algorithms to schedule visits of a mobile agent to collect data from cluster heads [3]. The mobile robot need only to visit the specific location to download the data from sensor within the communication range. Hence it does not need to visit the

other sensor's location.

III. REFLECTIONS OF THE RESEARCH WORK:

The mobile robot (R) moves from the start position

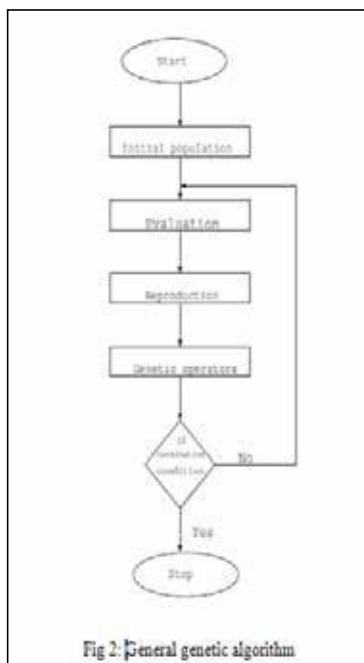
(S) to download data from each sensor within its range (rs) and comes back to its original position after performing data gathering task[1].The objective [8] is to compute a tour for each robot in such a way that minimizes the time to collect data from all devices. In order to download the data from a device, a robot must visit a point within the communication range of the device. Then, it spends a fixed amount of time to download the data. Thus, the time to complete a tour depends on not only the travel time but also the time to download the data, and the number of devices visited along the tour.

Mainly our research is to reduce the travel cost. This can be achieved by means of two things. One is to reduce the number of visited nodes. Secondly, by providing the shortest path between the nodes. For this purpose, we propose the algorithm known as CBPGA (clustering based parallel genetic algorithm)[1].

IV. EFFECTIVE PARALLEL GENETIC ALGORITHM

A. Parallel Genetic Algorithms (PGAs):

To solve some problems, genetic algorithms (GAs) may require hundreds or thousands of functions evaluations. Depending on the cost of each evaluation , the GA may take days, months or even years to find an acceptable solution[10].



Fortunately, GA works with the population of independent solutions, which makes it easy to distribute the computational load among several processors. For a long time, the parallel nature of genetic algorithms has been recognized, and many have successfully used parallel GAs to reduce the time required to reach adequate solutions to complex problems. [4]

The PGA consists of multiple computing nodes, those depends on type of PGA used. There are 3 major types of PGA's, they are master-slave, coarse-grained, fine-grained[5]

• Master-Slave GA:

In Master-Slave GA, one computing node will be the master and the others are slaves. The master node is responsible to hold the population and performs most of the genetic algorithm operations. The master will assign one or more computing intensive tasks to slaves by sending one or more chromo-

somes to them and it would then wait for the slaves to return their results.

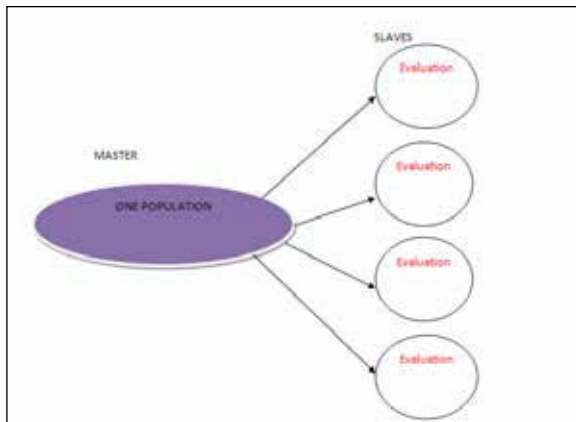


Fig 3: master-slave GA

• Coarse-Grained GA:

In Coarse-Grained, the population is divided into computing nodes which have a sub-population and executes genetic algorithm on its own. The nodes will exchange chromosomes with each other ensuring that good solutions can be spread to other nodes. This exchange can be called as migration where a node sends its best chromosome to other nodes. The other nodes which are having the worst chromosomes will be replaced by the received ones[5].

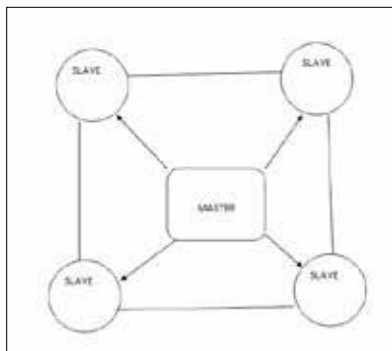


Fig 4: Coarse grained model

• Fine-Grained GA:

In Fine-Grained, each computing node only has a single chromosome and are arranged in a spatial structure. Here each node communicates only with other neighbouring nodes and the population is the collection of all the chromosomes in each node. To execute a genetic operation, a computing node must interact with its neighbours. The good traits of a superior individual can be spread to the entire population due to the overlapping of neighbourhoods.

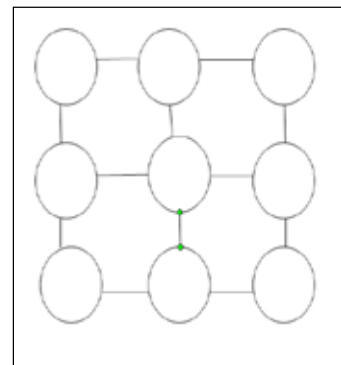


Fig 5: Fine grained model

In this paper, we adopt the structure of coarse grained model.

B. Migration Strategy

The migration model divides the population in multiple sub-populations. These subpopulations evolve independently from each other for a certain number of generations called isolation time. After the isolation time a number of individuals is distributed between the subpopulations called migration. The number of exchanged individuals is called migration rate. The selection of the individuals for migration immigrants takes place in the following manner: At each migration interval a node will send the best individual to the next node using the communication link. The sent individual will then replace the worst individual in the new subpopulation [9,14].

C. Overview of CBPGA

In order to generate the near-optimal shortest route, we combine the structure of the coarse-grained PGA (parallel genetic algorithm) [5] with the clustering algorithm, travel cost reduction scheme (CA and RS) and migration mechanism [10] that form CBPGA to obtain the near-optimal solution of TSPN.

The coarse-grained genetic algorithm[5] has been chosen for proposed algorithm due to the use of MPI(Message Passing Interface) cluster. Since the MPI environment will have a large overhead of communication between the computers[15]. So we go for coarse-grained genetic algorithm which has lowest communication overhead when compared to other PGA types. In this PGA implementation, all the computing nodes randomly creates their own subpopulation and each of them will execute genetic algorithm on its own. One of the computing nodes will be assigned special task to gather results from all the other nodes and then choose the best result as the output of parallel genetic algorithm. This node is called as the collector node.

In proposed algorithm, each chromosome is encoded as a series of node id's that are in the path from source to destination. The first gene in the chromosome is always the source and the last gene in the chromosome is always the destination. Since different paths may have different number of intermediate nodes, the chromosomes will be of variable length. However, the maximum length of a chromosome cannot exceed the total number of nodes in the network. Any repeated nodes in the chromosome signify that the path represented by the chromosome contains a loop and in network routing, any loop should be eliminated.

D. Clustering Algorithm (CA):

In order to decrease the costs, the number of visited nodes should be decreased and this can be done through the clustering algorithm (CA). Here, the CA classifies several sensors that share a common intersection area as the same cluster[1]. The robot only needs to visit the intersection areas of these clusters to download data from all sensors.

E. Parallel genetic Algorithm

For large population sizes, GA is computationally infeasible. Hence the use of PGA.

• Genetic Representation:

Each and every chromosome represents a feasible route, and it can be encoded in an ordered sequence of visited nodes. Since the mobile robot moves from the start position and comes back to the original position after gathering required data its start and ending positions are the same. Also can be known as the first and last gene on chromosome are the same.

Every visited node should appear only once except the first and last gene. If not the chromosome is considered as invalid one.

• Initial population:

Some factors to take into account when the initial population

is generated randomly. An initial population with high diversity has the higher chance of evolving great child chromosomes and it provides feasible solution[11]. To generate population with high diversity, we propose a Chromosome Generation Algorithm.

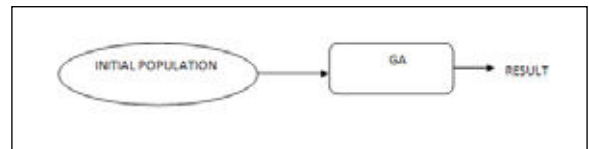


Fig 8: Initial Population

• Chromosome Generation Algorithm:

In genetic algorithms, a chromosome (also sometimes called a genome) is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. The chromosome is often represented as a simple string, although a wide variety of other data structures are also used[12].

• Fitness Function:

After creating the initial population, each individual is evaluated and assigned a fitness value according to the fitness function [11]. Fitness evaluation are done based on two different criteria. There are

- Classification Accuracy
- Size of the subset

$$\text{fitness}(x) = w_1 * \text{accuracy}(x) + w_2 * (1 - \text{dimensionality}(x))$$
 Here,

- $\text{accuracy}(x)$ = test accuracy of the classifier built with the gene subset represented by x
- $\text{dimensionality}(x) \in [0,1]$ = the dimension of the subset
- w_1 = weight assigned to accuracy
- w_2 = weight assigned to dimensionality
- High classification accuracy and low dimension has high fitness.

Fitness score:

The measurement of how good that chromosome is at solving the problem. The higher the fitness score, the smaller will be the error rate for the solution.

Selection Operation:

Selection is used to choose the parent chromosomes for the crossover operation. The selection scheme used in the algorithm is the pair wise tournament selection. In this selection scheme, a parent for the crossover operation is selected by randomly choosing two chromosomes from the population. The one with the higher chromosome between the two will be selected as a parent. To select two parents, this operation is performed twice.

Crossover:

Crossover is performed on the two parent chromosomes selected using the selection scheme described above. To ensure that the paths generated by the crossover operation are still valid paths, the two chromosomes selected must have atleast one common node other than the source and destination nodes. If more than one common node exists, one of them will be randomly chosen with equal probability. The chosen node is called the crossover point. For example, assume that we have the following parent chromosomes:

Parent chromosome 1=[A B C G H I X Y Z]
 Parent chromosome 2=[A K L M I T U Z]

Where A and Z are the source and destination nodes respectively. In this example the common node is node I. Therefore, crossover operation will exchange the first portion of chromo-

some 1 with the second portion of chromosome 2 and vice-versa. As a result, the following child chromosomes will be generated:

Child chromosome 1:[A B C G H I T U Z]
Child chromosome 2:[A K L M I X Y Z]

These two chromosomes would then become new members of the population.

Mutation:

Each chromosome produced by the crossover operation has a small chance to be mutated based on the mutation probability. For each chromosome that is chosen randomly, with equal probability, among the intermediate nodes in the path from sender to receiver. Once the mutation point is chosen, the chromosome will be changed starting from the node after the mutation point and onwards. For example assume that the following:

chromosome has been chosen to be mutated.
Original chromosome:[A C E F G H I Y Z]

Where A and Z are the sending and receiving nodes respectively. Assume also that the node G has been chosen as the mutation point. The mutated chromosome would become like this:

Mutated chromosome:[A C E F G x1,x2,x3... Z]

The mutated chromosome now contains a new path from G to Z where xi is the ith new node in the path. The new path is generated randomly; the same way as the paths in the initial population is generated

Migration operation:

We evaluated the following migration styles:

Tree based migration or child to parent migration: In this migration model a child node will send the best individual to the parent node using the communication link at each migration interval. The sent individual will replace the worst individual in

the parent sub-population

Neighborhood Migration : In this migration model migration will take place between two adjacent nodes using links corresponding to the edges of the nodes. The sent individual will then replace the worst individual in the neighbors population.

Isolated model : In this model there is no communication between nodes. This model is used as a control test to show the importance of communication between nodes. In our experiment this model is referred to as the no communication model.

V. SIMULATION RESULTS

In this paper, we propose a clustering-based parallel genetic algorithm with migration (CBPGA) for solving the robot routing problem of WSN using coarse-grained model. Simulation experiment provides the routing protocol with WSN in which mobile robot gathers all data from all sensors which confirm data gathering with near-optimal route. Hence it reduces the number of visited nodes and reduces the travel costs.

VI. CONCLUSION

This paper proposed a coarse grained parallel genetic algorithm for solving the shortest path routing with the primary goal of achieving faster computation speed and decreasing the number of visited nodes. The accuracy and computation time of the algorithm is dependent on population size and the number of computing nodes. The accuracy decreases linearly with larger number of computing nodes whereas the computation time decreases exponentially. The accuracy and execution time are higher when population size is more. Due to increase in population the computation time increases and this can be reduced by using larger number of computing nodes.

ACKNOWLEDGEMENTS

This work was supported by the management of Bharath University, I would like to thank for the encouragement of this research work

REFERENCES

- [1] Ko-Ming Chiu, Jing-Sin Liu, "Robot Routing Using Clustering-Based Parallel Genetic Algorithm with Migration" | [2]B. Yuan, M. Orlowska, and S. Sadiq. "On the optimal robot routing problem in wireless sensor networks", IEEE Trans. on Knowl. and Data Eng., Vol 19, Issue 9, 2007, pp. 1252–1261. | [3] Y. Tirta, Zhiyuan Li, Yung-Hsiang Lu, and S. Bagchi. Efficient collection of sensor data in remote fields using mobile collectors. International Conference on Computer Communications and Networks (ICCCN), pages 515–519, Oct 2004. | [4] E. Cant' u-Paz. "Efficient and accurate parallel genetic algorithms," Kluwer Academic Publishers, 2001. | [5] S. Yussof, R. A. Razali, O. H. See, A. A. Ghapar and M. M. Din, "A Coarse-grained Parallel Genetic Algorithm with Migration for Shortest Path Routing Problem," IEEE International Conference on High Performance Computing and Communications (HPCC) 2009, 2009, pp. 615 – 621. | [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, Aug 2002. | [7] . Meyer and A. Rakotonirainy, "A Survey of Research on Context-Aware Homes," Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing, Adelaide Australia, 2003 | [8] D. Bhaduria and V. Isler, "Data gathering tours for mobile robots", IEEE/RSJ International Conference on Intelligent Robots and Systems 2009, 2009, pp. 3868-3873. | [9] Rebaudengo M. and Sonza Reorda, M., "An experimental analysis of the effects of migration in parallel genetic algorithms", Euromicro Conference on Parallel, Distributed and Network-Based Processing (EMPDP) 1993, 1993, pp. 232 – 238. | [10] Golub M. and Jakobovic D., "A New Model of Global Parallel Genetic Algorithm," Information Technology Interfaces (ITI) 2000, 2000, pp. 363 – 368. | [11] Pedro A. Diaz-Gomez and Dean F. Hougen, "Initial Population for Genetic Algorithms: A Metric Approach," University of Oklahoma | [12] Matsumura T., Nakamura M., Miyazato D., Onaga K. and Okech J., "Effects of Chromosome Migration on a Parallel and Distributed Genetic Algorithm", Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN) 1997, 1997, pp. 357 – 361. | [13] Wendy Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Proceedings of the Hawaii International Conference on System Sciences, 2000, January. | [14] Endi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions On Wireless Communications, Vol. 1, No. 4, 2002, October. | [15] Plamenka B. and Milena L., "Migration Policies for Island Genetic Models on Multicomputer Platform", International Symposium on Algorithms and Computation (IDAACS) 2007, 2007, pp. 143 – 148. | [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100]