Task Allocation In Robot Mobile Wireless Sensor Networks

Amirhamzeh Razavinegad

Abstract: To solve the robot task allocation problem especially in unknown complex environment, a DPSO algorithm which decrease communication cost in wireless sensor network (WSN).Our simulation experiments show that at a small cost in assignment quality, DPSO algorithm sign ificantly extends the lifetime of the static sensor network , rathar than PSO ,centeralized algorithm, and an auction-based algorithm.

Index Terms: Minimum 7 keywords are mandatory, Keywords should closely reflect the topic and should optimally characterize the paper. Use about four key words or phrases in alphabetical order, separated by commas.

1 INTRODUCTION

Sensor networks have emerged and have been suc-cessfully deployed to solve problems as diverse as detecting , controlling,floods the temperature in office buildings, and monitoring hospital patients and an effective tool for monitoring large-scale environments[1]. For deploinng sensor networks in such environments, often with hundreds of nodes, wireless sensors must be low-cost and affordable.Hence, wireless sensors are typically highly limited in terms of battery life, sensing,computation, communication, and the actions they can perform. Robotic teams have several potential advantages over single-robot operations: some problems can be unsolvable for robots on their own, requiring cooperation of several agents; they may be able to complete a mission in less time; cooperative approaches that are more efficiently may exist; finally, they are more resistant to failure thanks to redundancy of components. We are motivated by a particular application of wirelessensor networks, namely, multi-robot task allocation (MRTA). The MRTA problem has been well-studied in the obotics community [2], and is simply stated as the problem of allocating tasks to robots. Higher-capability mobile robots may be dispatched to gather more accurate temperature or humidity readings, or to take soil samples which the static sensors are not equipped for. Although the static sensors are less capable than mobile robots, they are also much less expensive and can be deployed to cover a vast area at a low cost.In general, static sensors detect events which must be handled by mobile robots. These events are associated with a point in space where a mobile robot is needed to perform a task, such as gathering a more accurate temperature reading,spraying pesticides, or recharging a static sensor's battery. The static sensors must assign these events to mobile robots while simultaneously minimizing the distance travelled by the robots and the communication among the static sensors to prolong their battery life. Recently the multi-robot task allocation (MRTA) problem has risen to prominence and become a key research topic in the multi-robot domain[3,4] . The MRTA problem in its most general form is equivalent to the NP-Complete conjunctive planning problem.

- *Ali Bagherinia*
- *razavi.amihamzeh@gmail.com*
- *(Amirhamzeh Razavinejad, Dehdasht branch.Islamic Azad university Dehdasht branch,Dehdasht Iran)*

There are four different tasks:1)Uncorrelated tasks solved by a single robot; 2)Correlated tasks with separate robot asynchronously solving the task; 3)Synchronously correlated tasks, done by inseparably by multi-robots in a synchronous manner; 4)Sequentially correlated tasks, done inseparably by multi-robot doing the task sequentially. Any task can be represented as a combination of the above task types. This kind of task decomposition enables the defining tasks in the simulator even for very complex tasks. MRTA encourages the distributed problem solving.MRTA has two major subdivisions: Offline and Online. Offline MRTA is assigning resources to different tasks if the information of tasks such as arrival time distribution and priority of tasks is known a priori.

1. Related Work

The problem of assigning mobile robots to static sensors is an instance of the multirobot task allocation problem,in which a group of robots is given a list of tasks to complete. The goal is to construct a schedule of tasks for each robot which minimizes the total cost to complete all the tasks. This problem is strongly NP-hard, and so a common approach is to greedily assign tasks to robots as the robots become available [5]. Another popular approach is a freemarket based auction system, where the mobile robots place bids on tasks based on the cost to accomplish them [6]. The task allocation problem we consider has the additional requirement of minimizing the communication between static sensors to prolong the network lifetime, and the additional networking structure provided by the static nodes. Heuristic-based algorithms have been developed to deploy the mobile robots to an initial configuration with good coverage of the environment, where after deployment they act as static sensors [7] [8]. Wang,et al., have studied redeploying these mobile sensors in response to sensor failures and new events using a grid-based ommunication framework [9], and have also developed an auction algorithm for the coverage problem in hybrid networks of both static and mobile sensors, where the mobile agents bid on coverage holes [10]. Again, these algorithms are designed to achieve coverage of the environment with the mobile sensors, rather than handle specific events which is our goal. A few real-world systems have utilized hybrid networks of both static and mobile sensors. In [11], Vasilescu, et al., deploy a team of ten static nodes and two mobile nodes to monitor underwater environments. The mobile nodes act as data ferries between the static sensors to address the difficulties of underwater radio communication. In [12], Sukhatme, et al., deploy ten static buoys and a single mobile boat to monitor microorganism levels in a lake. The mobile boat shares its sensor readings with the network of static buoys, which instruct the boat on what areas to observe next.

1.1. Centralized Algorithm

In a centralized algorithm, a lead sensor is selected to perform all of the decision making. Since the sensors each have a map with the positions of all the other sensors, we select the sensor closest to the centroid of all the sensors as the leader, sl. Whenever a robot is not assigned to a static sensor, either because it has just been initialized or was released from its previous assignment by the local MtE assignment algorithm,it sends a request for assignment to the nearest static sensor. This request contains the mobile robots's ID, the ID of the nearest static sensor (so that the leader can send the mobile robots its assignment through a path in the connectivity graph C), and the current (x; y) position of the mobile robot. The static sensors forward the request along the shortest path in C to sl. Similarly, when the number of robots needed by a static sensor si changes, either because robots successfully handled events or because more events occurred, the static sensor sends the new need num events(si) to sl through the shortest path in C. The leader, sl, accumulates a list of the number of mobile robots num events(si) needed by each static sensor. sl also gathers, for each unassigned mobile robot m, its position pos(m), and the ID closest(m) of the closest static sensor to that robot. Every time-step, sl makes assignments using the greedy Algorithm 1. Algorithm 1 iteratively makes the assignment of lowest cost, in the same manner as the greedy algorithm used for the MtE problem. Again, we use a greedy algorithm since we are solving an online problem announcing the winner of the auction, which percolates to all the static sensors. The static sensors remove m from their list a of ongoing auctions, and the winner adds m to the list of robots working for it. m then proceeds to the winner's location, and executes the MtE assignment algorithm[13].

2.2 . Auction of Mobile Robots

In our second approach, the mobile robots hold auctions for their services among the static sensors, in a manner similar to [10]. When a mobile robot becomes available, either initially or because it was released from its previous assignment, it sends an announcement of an auction to the neighboring static sensors. This announcement contains the mobile robot's ID, position, and the ID of the nearest static sensor. The static sensors percolate the announcement through the entire network. Each of the static sensors forms a list a of the mobile robots holding auctions. When a static sensor decides it needs more mobile robots to aid it, it sends a bid to the nearest mobile robot in a with the position of the static sensor. A static sensor can place several bids at a time, up to the number of mobile robots it needs. After a mobile robot m has called an auction, it forms a list b of the static sensors which have placed bids. The mobile robot then waits a time dauction before it selects a winner.dauction should be chosen based on the communication delay and the number of static sensors, so that all of the static sensors have time to place bids. If after dauction time passes no bids have arrived, m continues waiting and accepts the first bid it receives (since calling a new auction incurs a high overhead). Otherwise, the nearest static sensor in b to m is greedily selected as the winner. m then sends a message announcing the winner of the auction, which percolates to all the static sensors. The static sensors remove m from their list a of ongoing auctions, and the winner adds m to the list of robots working for it. m then proceeds to the winner's location, and executes the MtE assignment algorithm [14 p:10iros].

2. Proposed Approac

3. 1. Cost Function

The cost function includes the distance traveled by the mobile robots, energy consumed for data transmission between static sensors, the energy of the primary processing in static sensors and time of running events in the mobile robots.

$$\alpha\left(\sum_{m\in M} d_m + \sum_{(u,v)} EComm\right) + b\sum_t EProc$$

d: distance traveld by each robot m

(u,v): Enerhy consumed by Communication of each packet between two sensor nodes

E proc): Enerhy consumed for processing a task t at each sensor node

a,b: coefficient that define importance of distance or others factors

Due to the importance of coefficients that reduce the distance traveled by a mobile robot or a reduction in energy consumption for data transfer between the application of static sensors α, β are defined. The proposed method, unlike other mentioned methods that assignment of robots to static sensors and assining events to robots in the form of two independent phases (two sub-problems) performed by both the cost function together in a single phase (an atomoic problem) considered in trying to minimize the cost function. This problem is a NP problem and we present the details of the DPSO algorithm is used for solving it as below..

3.2. Proposed Algorithm

Allocation algorithm centrally runs and should minimum above cost function. Propose an algorithm that assigns the mobile robots to static sensor nodes in a sensor network (represented as a graph) and scheduling of events in each mobile robot while satisfy above cost function has very large search space; on the other hands the allocation problem is a NP-Complete problem. We have used a particle swarm optimization for allocation in this network. The goal of allocation is to minimize the cost function as result increasing network lifetime. allocation model is an integer program with an equation that includes cost function with integer coefficients (Equation 1).

Min Cost function =

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} ec_{ij}a_{ij} + \sum_{i=1}^{n-1}\sum_{k=i+1}^{m} cc_{ik}a_{ij}\left(1-\sum_{j=1}^{m} a_{ij}a_{kj}\right) \quad \text{(Equation 1)}$$

$$\sum_{j=1}^{t} a_{ij} = 1,2,3,\ldots.,n \quad \text{(Equation 2)}$$

$$a_{ij}\ (0,1) \quad \text{(Equation 3)}$$

Equation 2 states that each mobile robot should be allocated to exactly one sensor node. Equation 3 states that a_{ij} is binary variable. Now, we want to design particle to present a sequence of events in sensor networks that is represented by a weighted graph. In my approach solutions are encoded as n × m matrixes, called position matrix, in which m is the number of available sensor nodes at the time of allocation and n is the number of tasks. The position matrix of each particle has the two following properties:

1) All the elements of the matrices have either the value of 0 or 1. In other words, if X_{id} is the position matrix of i-th particles in a d-dimensional space, then:

$$X_{id}\ (0,1)$$

2) In each row of these matrices only one element is 1 and others are 0.

In position matrix each row represents a task allocation and each column represents allocated events in mobile robot. Velocity V_{id} of each particle is considered as a n×m matrix whose elements are in range$[-V_{max}, V_{max}]$. Also Pbest and nbest are n×m matrices and their elements are 0 or 1 as position matrices. P_{id} represents the best position that i-th particle has visited since the first time step andP_{gd} represents the best position that i-th particle and its neighbors have visited from the beginning of the algorithm. In this paper we used star neighborhood topology for P_{gd} . In each time step P_{id} and P_{gd} should be updated:

$$V_{id}^{new} = weight \times V_{id}^{old} + C_1 \times rand_1 \times (P_{id} - X_{id}) + C_2 \times rand_2 \times (P_{gd} - X_{id})$$

(Equation 4)

$$X_{id}^{new}(n,m) = \begin{cases} 1 & if\ V_{id}^{new}(n,m) = max\{V_{id}^{new}(n,m)\} \\ 0 & otherwise \end{cases}$$

(Equation 5)

In (4) $V_{id}^{new}(n,m)$ is the element in n-th row and m-th column of the i-th velocity matrix in the updated time step of the algorithm and $X_{id}^{new}(n,m)$ denotes the element in n-th row and m-th column of the i-th position matrix in the updated time step. C_1 and C_2 are positive acceleration constants which control the influence of P_{id} and P_{gd} on the search process. Also $rand_1$ and $rand_2$ are random values in range [0, 1] sampled from a uniform distribution. weight which is called inertia weight was introduced by Shi and Eberhart [15] as a mechanism to control the exploration and exploitation abilities of the swarm. Usually w starts with large values (e.g. 0.9) which decreases over time to smaller values so that in the last iteration it ends to a small value (e.g. 0.1). Equation (5) means that in each row of position matrix value 1 is assigned to the element whose corresponding element in velocity matrix has the max value in its corresponding row. If in a row of velocity matrix there is more than one element with max value, then one of these elements is selected randomly and 1 assigned to its corresponding element in the position matrix.

The pseudo code of the proposed DPSO algorithm is stated as follows: Create and initialize a n×m -dimensional swarm with P particles

```
do{
    for (each particle i=1 to P) do
    {
      if  f(X_id) > f(P_id)
          P_id = X_id
      if  f(P_id) > f(P_gd)
          P_gd = P_id
    }
    for (each particle i=1 to P) do
   {
      update  the velocity matrix using Equation (4)
      update  the position matrix using Equation (5)
   }
} While (stopping condition is false)
```

In commutative and associative tasks such as Min(), Max(), Sum() and Avg() at each task allocation to a node states of its neighbor nodes is given to it. Thus tis nod after finishing executing its task wait to receive data from its neighbors. Then send total result as a message. For example in This action cause less mobile robot moving,decreasing communication in sensor networks that implies less energy consumption.

4. EXPERIMENTAL RESULTS

To evaluate the three static sensor assignment algorithms,we ran experiments in simulation to compare the assignment quality and communication costs. The use of simulation allows us to run large scale experiments involving hundreds of mobile robots and static sensors. The task allocation problem is equally relevant to real-world deployments of hybrid networks. We compare the three algorithms to a modified version of SWANS [16], in which the grid heads use our local MtE assignment algorithm.We evenly distribute one hundred static sensors on a square environment to achieve full coverage. In Fig.1 the maximum number of messages sent by a single sensor is shown. The centralized and auxtion based algorithms have the highest values, since their total communication is higher then energy consumption is the highest. But even though the proposed algorithm has higher total communication than the centralized algorithm,the maximum number of messages sent for a single robot is approximately half as much in the online scenario for two hundred events. This is because in the centralized algorithm, all the messages are routed through the leader. In Fig.2. The proposed algorithm, on the other hand mobile robots moving is the least algorithm runs events the most quickly than others algorithms.

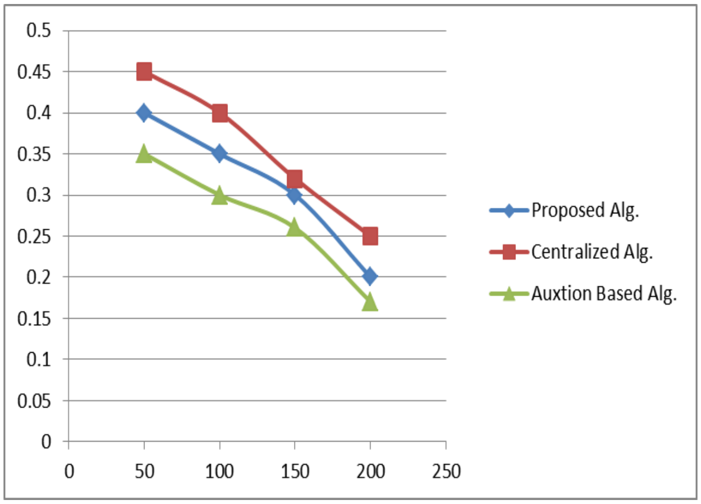

Fig.1. Remaning average of energy in three algorithms

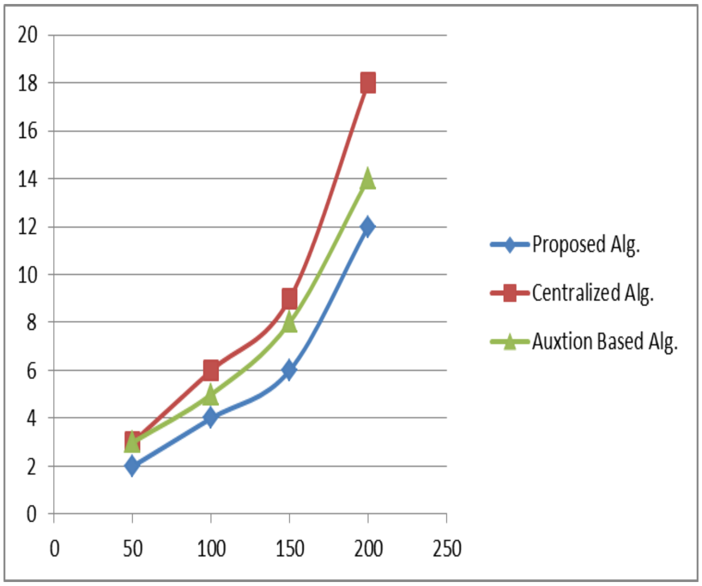

Fig.2. Response time in three algorithms

5. CONCLUSION

In this paper, we presented the event assignment problem for allocating mobile robots to events in hybrid wireless sensor networks. We discussed two algorithms for solving this problem: a centralized algorithm, an auction-based approach. Then we propose an approach that assignment of robots to static sensors and assining events to robots in the form of two independent phases performed by both the cost function together in a single phase considered in trying to minimize the cost function.We evaluated the advantages and disadvantages of each in terms of assignment qualityresulted from less moving of robots,total communication cost, and expected network lifetime. Our results showed a lower assignment quality for the other algorithm, but with a low communication cost, a high expected network lifetime, in contrasted proposed approach runs events processingthe most quickly than others algorithms and energy consumption isn'y heigh.

REFERENCES

[1]. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci,"Wireless sensor networks: a survey," Computer Networks, vol. 38,no. 4, 2002.

[2]. Alejandro R. Mosteo, Luis Montano ,"A survey of multi-robot task allocation", AMI-009-10-TEC.

[3]. ARAI T, PAGELLO E, PARKER L E. Advances in multirobot systems. IEEE Transacitons on Robotics and Automation,2002,18(5):655-661.

[4]. CAO Y U, FUKUNAGA A S, KAHNG A B. Cooperative mobile robotics: antecedents and directions. Autonomous Robots,1997,4(1):7-27.

[5]. B. P. Gerkey and M. J. Matari´ c, "A formal analysis and taxonomy of task allocation in multi-robot systems," The Intl. J. of Robotics Research, vol. 23, no. 9, 2004.

[6]. R. M. Zlot, A. T. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in IEEE International Conference on Robotics and Automation, 2002.

[7]. A. Howard, M. J. Matari´ c, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," Autonomous Robots, vol. 13, 2007.

[8]. B. Zhang and G. S. Sukhatme, "Controlling sensor density using mobility," in EmNets '05: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors, 2005.

[9]. G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation inmobile sensor networks," in In Proc. of IEEE INFOCOM, 2005.

[10]. G. Wang, G. Cao, and T. La Porta, "A bidding protocol for deploying mobile sensors," in ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols, 2003.

[11]. I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems, 2005.

[12]. G. S. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B. Stauffer, and D. A. Caron, "The design and development of a wireless robotic networked aquatic microbial observing system," Environmental Engineering Science, vol. 24, no. 2, 2007.

[13]. www.cn.ece.cornel.edu/publications/papers/allerton2002/slides.pdf.

[14]. SWANS Simulator: http:\\jist.ece.cornell.edu.