RESOURCE ALLOCATION IN SENSOR NETWORKS USING DISTRIBUTED CONSTRAINT OPTIMIZATION

Sumit Chachra (student), Theodore Elhourani (student)
Michael Marefat (faculty advisor), and John Reagan (faculty advisor)
Department of Electrical and Computer Engineering
The University of Arizona, Tucson AZ 85721
Email contact:{marefat,reagan}@ece.arizona.edu

ABSTRACT

Several algorithms have been proposed for solving constraint satisfaction and the more general constraint optimization problem in a distributed manner. In this paper we apply two such algorithms to the task of dynamic resource allocation in the sensor network domain using appropriate abstractions. The aim is to effectively track multiple targets by making the sensors coordinate with each other in a distributed manner, given a probabilistic representation of tasks (targets). We present simulation results and compare the performance of the DBA and DSA algorithms under varying experimental settings.

KEYWORDS

Distributed Constraint Satisfaction/Optimization, Sensor Networks, Multiagent Systems, Resource Allocation

INTRODUCTION

The problem of constraint optimization is of assigning values to a given set of variables such that a global function evaluating the given value assignments is optimized. The constraint satisfaction version of the problem has been proven to be NP-Complete. Centralized solutions have been suggested to classic constraint satisfaction problems such as the n-queens problem (none of the n queens on a n × n board attack each other), graph coloring problem (no neighboring vertices in a graph can have the same color) etc, which assume that we have a global view of all the constraints and current values of all the variables. The problem becomes more complex when we treat each variable as an agent with only a local view of the problem. For example in the graph coloring problem an agent only knows its own color, the current color of its neighbors, and the constraints it has with them. Given only local information, the aim is to solve the problem through message passing-enabled negotiation. In this paper we explore resource allocation in sensor networks from various perspectives:

- Performance of Distributed Breakout (DBA) and Distributed Stochastic (DSA) Algorithms.

- Efficiency and optimality of each algorithm in a sensor network with multiple targets.

In particular we consider a set of Doppler Radar sensors. Each sensor has three radar heads, each with a 120 degree viewable arc. Only one head in a sensor can be activated at one time and it can track one moving object only. The sensor is able to detect the presence of a moving object and can give an approximate value of its velocity. The sensor is able to detect targets only within a certain range and a minimum number of sensors is required to accurately localize and hence track a target. Resource allocation/contention here consists of deciding which head of each sensor needs to be activated at a particular time, such that the targets are effectively tracked. The domain specific details are abstracted so that existing DCOP algorithms can be applied for coordination/negotiation between the sensors. A two layered architecture similar to the one in [7] has been used. The bottom layer abstracts the domain specific details and the top layer controls the coordination among the different sensors. The above system is modeled as a discrete event system and simulations are performed under various experimental conditions. The modeling takes into consideration uncertainties and errors in sensor readings as well as latencies in communication between the sensor nodes. An "off the shelf" discrete event simulator, DEVS-JAVA [1] is used to perform the simulations.

In the rest of this paper we discuss previous research and in particular the application of DCOP algorithms in the sensor network domain. We then formulate the resource allocation problem as a distributed constraint optimization problem. In the DCOP section we introduce DBA and DSA tailored to our domain of interest. Then we describe the simulations we conducted and show results. Finally we conclude and discuss future work.

PREVIOUS WORK

There has recently been a lot of interest in the area of (distributed) sensor networks [2, 3], since most of the problems faced are common to most distributed computing systems. Because each sensor performs its tasks independently it is natural to treat it as an agent and the entire network as a multiagent system (MAS). This enables us to apply theories which have been developed for MAS [4], directly to the sensor network domain.

Two of the two most popular algorithms for solving the distributed version of constraint satisfaction are DBA and DSA [5, 6]. Both the algorithms aim to minimize the constraint violations through message passing between the agents. In our problem formulation we model each agent to represent a variable. In DBA and DSA each agent tries to take on values which minimize constraint violations based on the values taken on by neighboring agents with whom they have constraints. DBA gets its name from the manner in which it comes out of local minima, when none of the agents which share constraints is able to find a new value for themselves. DSA gets its name from the manner in which each agent stochastically takes a decision whether to take on a new value or not.

Distributed Resource Allocation, and in particular in the domain of sensor networks has been addressed several times over the last decade. In recent work [7] this problem has been formalized as a DCOP, in which the goal is to find assignments to each of the sensor variables, such that a global objective function F is minimized. In [8] the problem of scan scheduling in sensor networks has been

solved by using DBA and DSA, comparing both the algorithms on various metrics. In this work we have used DBA and DSA for the task of target tracking under various experimental setups. Both the algorithms have been adapted to the sensor network domain (see Algorithm 1 and 2). Modi [7] introduced the idea of a two layered architecture wherein the lower layer is responsible for abstracting the domain details and providing a discretisized version of the problem to the upper layer. The upper layer consists of a tailored core DCOP algorithm (ADOPT-SC). Modi tries to show that the two layered architecture enables the use of abstract DCOP algorithms in real world contexts. Our goal is to compare DBA and DSA in the sensor network domain and to pave the way for further evaluations of other existing algorithms. Scalability is claimed in [7], however it is not experimentally established, we do address scalability by experimenting with multiple targets and a relatively larger sensor network. Hence, our contributions can be summarized into three main points: first we developed a general simulation for the evaluation of DCOP algorithms for resource allocation in the sensor network domain, second we took into consideration scalability issues, and finally we evaluated both DBA and DSA in the simulation framework we developed.

PROBLEM FORMULATION

We reduce the given problem of resource allocation in sensor networks to that of constraint optimization using the formulation given in [9]. The variables in our domain are the set of n sensors. Hence $X = \{s_1, s_2, \ldots s_n\}$. The domain of each sensor is defined as $D = \{0, 1, 2, 3\}$, where 0 corresponds to the sensor performing scanning and 1, 2 and 3 represent the resource allocation of the respective heads. We define the scope Q_i to be the set of sensors consisting of the sensor s_i and all its neighbors (to which it can communicate with). Let f be the real-valued functional component defined over the scopes $Q_1, Q_2, \ldots Q_n$, $Q_j \subseteq X$. The global cost function F is defined by

$$F(V) = \sum_{j=1}^{n} f(Vj) \tag{1}$$

where $V = (V_1, V_2, \ldots V_n\}$, V_j is the set of value assignments to the sensors in Q_j. In this paper we view the constraint optimization problem as defined over a cost network [9], which is basically a 4-tuple $\mathcal{C} = \{X,D,C_h,C_s\}$, where C_h and C_s represent the hard and the soft constraints respectively. The hard constraint in our problem is that each sensor can allocate resources to only one head at any point of time (a head or sector as seen in figure 1). The soft constraint over which we optimize is represented by the function f. We calculate the function f in the following way: if a sensor X overlaps with sensor Y, and sensor Y has already allocated resources (activated) to the head(s) included under this overlap while X didn't allocate, we then increment f by O, where O is defined to be the area of a region covered by the sectors (or heads) of two different sensors. We would like to minimize the function f. By doing so, the unallocated areas covered by multiple sensors with a target roaming inside of them are minimized. The final goal is to reach a near optimal value of F under harsh time constraints resulting in an near optimal resource allocation and hence an effective tracking of the targets.

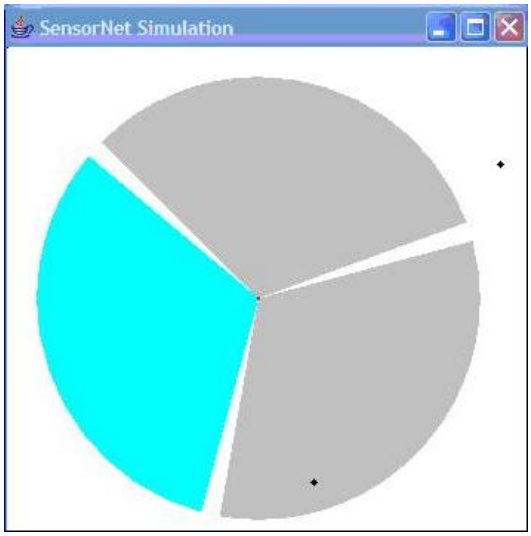

Fig.1. A representation of the Doppler Radar with three heads.

DISTRIBUTED CONSTRAINT OPTIMIZATION IN SENSOR NETWORKS

The nucleus of both algorithms (DBA and DSA) remains the same with minor modifications on the lower layer interface level. Please refer to DBA and DSA below in Algorithm 1 and Algorithm 2 respectively. In DBA agents try to assign values to their assigned variables while minimizing the constraints violations among them. DBA is a synchronous distributed algorithm that solves/minimizes DCOP problems. In the first step agents assign random values to their variables and send these values to all their respective neighbors. A neighbor of agent A_1 is defined to be any agent sharing constraints with A_1. After an agent receives all the value assignments of his neighbors he evaluates the current value assignments and finds the best improvement he can make if he possibly re-instantiate his variables. He then sends the improvement to his neighbors and starts receiving the possible improvements of his neighbors. After receiving all the improvement messages the agent decides to change the value of his variables, to the values which yield his computed maximum improvement, only if his improvement is the largest among all of his neighbors. In the cases where all the agents in a neighborhood have a zero improvement (local-minima) the constraint violations are increased, consequently the local-minimum is escaped. DSA (Distributed Stochastic Algorithm) is similar to DBA in that it uses a message passing protocol to achieve coherence. However, DSA is fully asynchronous. Agents send their current values to their neighbors and then randomly decide whether they should change their values or not. The choice of a new value depends on the current state of the agent and the states of the neighbors.

As mentioned in the problem statement section the agents represent sensors and the domain of each sensor consists of four values: {scan, allocate head 1, allocate head 2, and allocate head 3}. However, the domain of a sensor is not static as opposed to the domain of a node in a graph coloring problem. When no target is detected the sensor's domain is reduced to the singleton {scan}, since in this case, scan is the most logical solution to the problem. The dynamics of the domain are controlled by the lower layer which updates the probabilities of the presence of a target for each sector.

```
Set the weight for each head's constraint violation
while conflict in resource allocation do
  send current s to all the neighbors
  Form the V vector based on received messages from neighboring sensors {V is nothing but the
  agent view}
  cost ← f(V)
  CR ← the best possible cost reduction if s' is chosen; ∀s' in D, s' ≠ s: send CR to all the
  neighboring sensors and collect their respective improvements
  if CR > CR_i; ∀i ∈ Neighboring_Sensors and i ≠ Current_Sensor then
    s ← s'
  else if ∃i ∄j, j ≠ i, j ≠ Current_Sensor : CR_i = CR and CR_j > CR then
    k ← breakTie() {break the tie appropriately such that only one of the sensors changes its
    value}
    if k = Current_Sensor then
      s ← s'
    end if
  else if none of the sensors in V can offer a cost improvement then
    Increase the weights associated with the heads constraint violation by a certain amount
  end if
end while
```

Algorithm 1: Distributed Breakout Algorithm (DBA)

```
set probability p
while conflict in resource allocation do
  change ← 0
  cost ← f(V) {Calculate the cost/violation based on the current allocation}
  if ∀s' in D, s' ≠ s: and f(V') ≤ cost and Random_Number ≤ p then
    change ← 1
    s ← s'
  end if
  if change ≠ 0 then
    send current s to the neighboring sensors
  end if
end while
```

Algorithm 2: Distributed Stochastic Algorithm (DSA)

SYSTEM DESIGN AND EXPERIMENTS

Experiments were conducted using DEVS-JAVA [1]. Each sensor is designed as a coupled DEVS model containing the upper and lower layer atomic DEVS elements. In figure 1, the window to the right shows the lower-layer "LocalReasoning1" and the DCOP upper layer "DBA1" of sensor "sensorDBA1". The left window shows the corresponding GUI. "LocalReasoning1" receives the targets' locations and updates the probability that a target is present in each of the three sectors covered by the sensor. In our experimentations we take this probability to be directly proportional to the distance of a target from the sensor. This computation is done in the external transition function o the model. A DEVS atomic model receives messages from other models through "in-ports" and sends messages to other models through "out-ports". The mathematical behavior of a DEVS atomic model is given as:

$< S, ta, \delta_{int}, X, \delta_{ext}, Y, \lambda >$, where

S is the set of admissible states

$ta: S \rightarrow R$

$\delta_{int}: S \rightarrow S$ (internal transition function)

X set of external inputs.

$\delta_{ext}: Q \times X \rightarrow S$, external transition function, $Q = \{(s,e) | s \in S, 0 \leq e \leq ta(s)\}$

Y set of outputs

$\lambda: S \rightarrow Y \cup \{\varnothing\}$

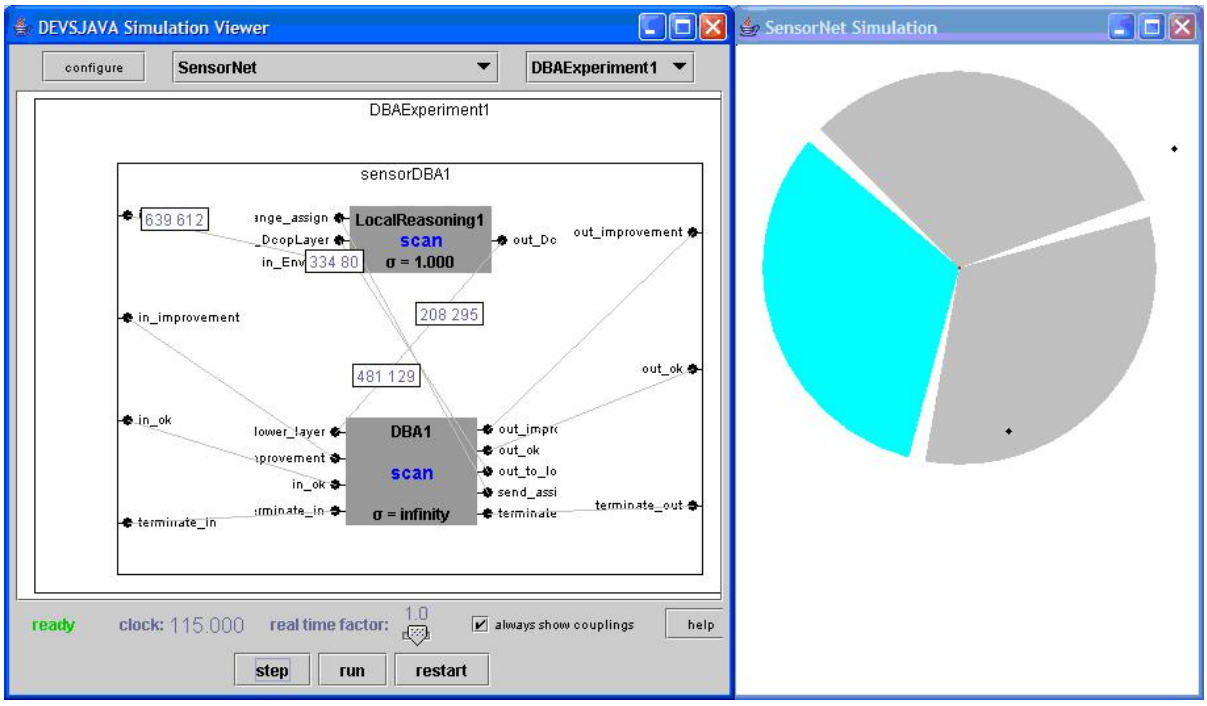

Fig. 2. An example DEVS-JAVA sensor model utilizing DBA and the corresponding display.

A sensor is initially given a static range within which it can detect targets (gray areas in figure 1 represent the range of a sensor). After updating one of the three probability values, the lower layer decides whether to notify the upper layer depending on the extent of the change. We consider a similar discretisization of the probabilities as in [7]. If the probability of a target goes above 0.6 we assign a value P (present) to the associated head, otherwise if the value falls between 0.2 and 0.6 we consider the head to be in the state U (unknown), and finally if the value is below 0.2 then we assign the value NP (not present). A discrete value from {P, U and NP} is passed to the upper layer in the event of a transition between any two probability levels (only such a transition is considered to be significant from the upper layer's perspective). The upper layer is where the DCOP algorithm sits and all the negotiation takes place. A display of the sensor network (shown in figure 1) was developed to help visualize the dynamics of the system.

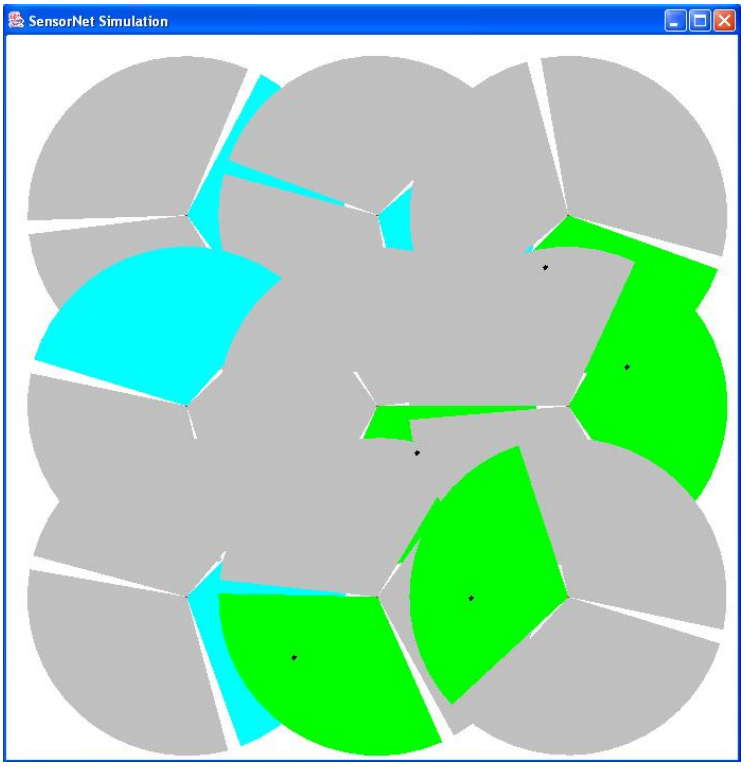

Fig. 3. The graphical interface of the simulator, black dots represent targets.

We performed one experiment for each of DBA and DSA. In a setting similar to the one shown in figure 3, (nine sensors and five targets) we ran the simulation for 2000 target steps. Two plots were generated showing the global evaluation function F versus time for each of the algorithms.

RESULTS

Figure 4 shows the global evaluation function F versus time steps. The spikes in the plot represent the instances where an increased violation is occurring in the sensor field. DBA reacts to these violations and tries to minimize the global function F. The

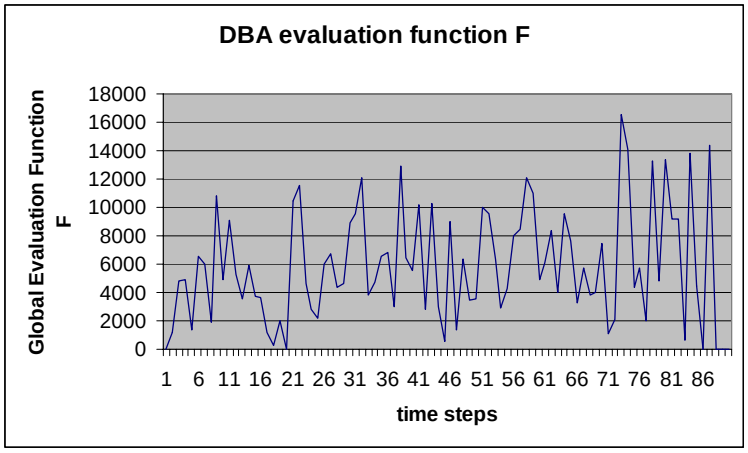

Fig. 4. The DBA global evaluation function

In figure 5 the individual evaluation functions are shown. Again the increase in the violation of constraints appears as sequences of spikes in the plots. Resource contention takes place when two or more sensors detect constraint violations and try to resolve them. In the plot this represented by the sequences of spikes occurring at the same time.

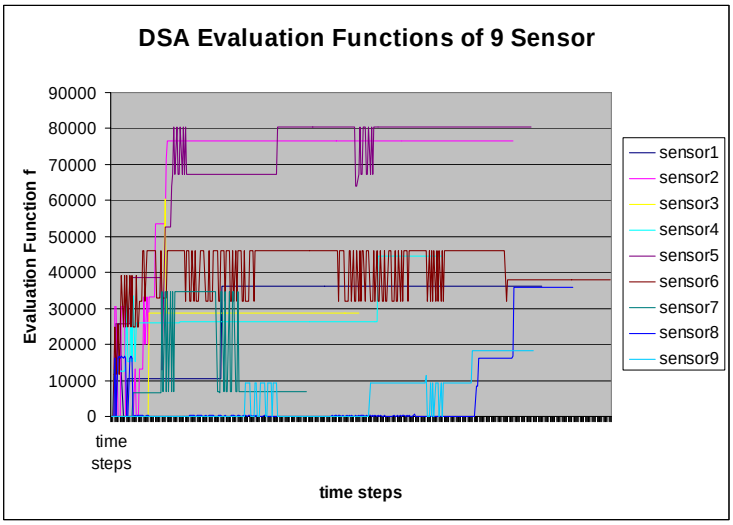

Fig. 5. The DSA individual evaluation functions

CONCLUSIONS AND FUTURE WORK

In this paper we investigated two popular DCOP algorithms, DBA and DSA, for distributed resource allocation in the sensor network domain. We believe that this work prepares for additional large scale experimentation in distributed resource allocation in the sensor network domain and other domains too. In the near future we will be evaluating other DCOP algorithms with improved models of the lower layer. In a later stage we will be investigating the resource allocation problem using the tools and techniques we developed in this research in different real world domains such as autonomic computing and multirobot systems.

REFERENCES

[1] B. P. Zeigler and H. S. Sarjoughian, "Introduction to devs modeling and simulation with java," 2001. [Online]. Available: http://www.acims.arizona.edu.

[2] V. Lesser, C. L. Ortiz, and M. Tambe, Eds., Distributed Sensor Networks: A Multiagent Perspective. Kluwer Academic Publishers, 2003.

[3] W. Zhang, Z. Deng, G. Wang, L. Wittenburg, and Z. Xing, "Distributed problem solving in sensor networks," Proceedings of AAMAS-2002, July 15-19, Bologna, Italy, poster paper. 2002.

[4] G. Weiss, Ed., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, 1999.

[5] M. Fabiunke, "Parallel distributed constraint satisfaction," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99), 1999: 1585–1591.

[6] S. Fitzpatrick and L. Meertens, "An experimental assessment of a stochastic, anytime, decentralised, soft colourer for sparse graphs," Proceedings of the first symposium on Stochastic Algorithms: Foundations and Applications, 2001: 49–64.

[7] P. Scerri, J. Modi, W. Shen, and M. Tambe, "Are multiagent algorithms relevant for real hardware? A case study of distributed constraint algorithms," Proceedings of the Eighteenth Annual ACM Symposium on Applied Computing, March 2003.

[8] W. Zhang, Z. Xing, G. Wang, and L. Wittenburg, "An analysis and application of distributed constraint satisfaction and optimization algorithms in sensor networks," Proceedings of the 2nd international Joint Conference on Autonomous Agent and Multiagent Systems (AAMAS-03), Melbourne, Australia, July 14-18, 2003: 185–192.

[9] R. Dechter, Constraint Processing. Morgan Kaufmann Publishers, 2003.

[10] M. Yokoo, Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-agent Systems, ser. Springer Series on Agent Technology. Springer-Verlag, 2001.

[11] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara, “The distributed constraint satisfaction problem: Formalization and Algorithms,” IEEE Transactions on Knowledge and Data Engineering, Vol. 10, no. 5, September/October 1998.