Analyzing the Efficiency of Strategies for MAS-based Sensor Interpretation and Diagnosis*

Norman Carver
Computer Science Department
Southern Illinois University Carbondale
Carbondale, IL, USA
carver@cs.siu.edu

Ruj Akavipat
Computer Science Department
Indiana University
Bloomington, IN, USA
ruj_a@yahoo.com

ABSTRACT

One of the factors holding back the application of multi-agent, distributed approaches to large-scale sensor interpretation and diagnosis problems is the lack of good techniques for predicting the performance of potential systems. In this paper we use a consideration of Bayesian network inference algorithms to construct formulas that describe the computational and communication resources required by several strategies for MAS-based distributed SI/diagnosis.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence—*Coherence and coordination*

General Terms

Algorithms, Design, Performance

Keywords

computational complexity and agent systems; coordination of multiple agents/activities; distributed problem solving; distributed interpretation and diagnosis; Bayesian networks

1. INTRODUCTION

Distributed problem solving (DPS) is the subfield of *multi-agent systems* (MAS) that studies how large-scale problems can be solved using systems of cooperative, distributed agents. Among the problems that have been most widely studied by DPS researchers are *distributed sensor interpretation* (DSI) and *distributed diagnosis* (DD). Unfortunately, little of this research has been concerned with developing the types of formal techniques that would allow designers to predict the performance of MAS-based strategies for DSI/DD. We believe that this is a serious deficiency, and have been pursuing several approaches to help address it. One project has used simulation approaches to identify properties of SI/diagnosis domains that are predictive of the performance of various classes of MAS-based strategies [1]. In this paper we demonstrate how a consideration of algorithms for inference in Bayesian networks (BNs) can be used to develop formulas that describe the computational and communication resources required by MAS-based strategies for DSI/DD.

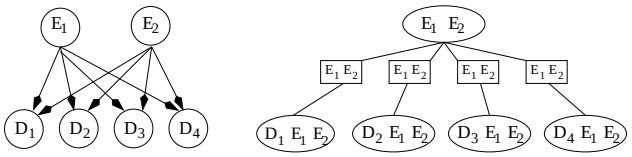

Figure 1: A simple SI BN and a possible join tree. The network has two event nodes, E_1 and E_2, and four data nodes, D_1 through D_4. In the join tree, *clusters* are ovals and *sepsets* are rectangles, labeled with their variables.

The goal in SI problems is to identify the best *interpretation* of sensor data, where an interpretation is a set of *events* in the environment that could have produced the observed data. The most commonly used standard for selecting the "best" interpretation is the *maximum a posteriori interpretation* (MAPI) of the (globally available) data. While it is typically impractical to compute the MAPI in real-world SI problems, SI systems often aim to approximate it.

The typical MAS-based approach to DSI/DD would partition the data set among a group of agents, each of which would be responsible for determining whether a certain subset of events had occurred (these are the agents' *subproblems*). Because each agent ends up with direct access to only a fraction of the data, agents must typically interact by exchanging data and/or results. The key question is: how should this be done so as to efficiently arrive at a high quality global solution (such as the MAPI)?

These days, the most popular approach for doing probabilistic reasoning is with Bayesian networks. The standard technique for doing *exact* inference in *multiply connected* BNs is to convert the BN to a *join tree*, operate on that structure, and compute probabilities of interest as needed (see [2] for details). Figure 1 shows a very simple BN for SI, along with one possible join tree version.

There are generally multiple join trees that can be constructed from any BN, and the join tree choice can have a significant impact on the cost of inferences. The selection of a join tree structure is one point where knowledge of the

*This material is based upon work supported in part by the Digital Society and Technologies program of the National Science Foundation under Grant No. IIS-0084135.

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

specific characteristics and inference goals for SI/diagnosis problems can be applied. Our analysis showed that for SI systems, the most efficient join tree structure is what we have termed the "central cluster" tree. These join trees have a cluster made up of the relevant events, to which all other (data containing) clusters are connected. This is the type of join tree shown in Figure 1. An additional benefit of this type of structure is that it suggests a straightforward method for distributing the BN among multiple agents: each agent's subproblem is represented as a central cluster join tree, with the subtrees linked at their central clusters, via sepsets. This allows agents to combine their local results through communications that effectively implement a join tree message pass (see [2]).

2. RESOURCE FORMULAS

Using the above developments, it is possible to produce formulas describing the computation and communication resources required by particular DSI/DD strategies. For this paper, we will assume the most constrained situation, which is that the overall SI/diagnosis problem cannot be decomposed in any way. To simplify the presentation, we will make several assumptions, such as that all data and event variables are binary, all agents receive the same amount of data, and so forth. The network topology will also obviously affect the strategies that can be used by the MAS as well as the performance. For this paper, we will assume a situation in which every agent has a dedicated connection directly to every other agent. The following notation will be used:

- m is the number of agents;
- d is the number of data elements per agent;
- n is the number of events;
- C_* is the units of time required to multiply/divide two entries in agents' potentials;
- C_v is the variable portion of the communication cost—i.e., the portion of cost that depends on the amount of information being sent—in time units per number of potential entries;
- C_f is the fixed portion of the communication cost—e.g., the time to establish a connection (handshake) and the propagation delay—in units of time.

We will initially focus on "exact" interpretation—i.e., on computing the true MAPI of the complete *global* dataset. One possible strategy is for every agent to first process its local data to develop a local potential for the possible interpretations (events combinations), and to then have every agent communicate its potential to a single agent that will merge the local potentials to produce global probabilities that can be used to identify the true global MAPI. The initial processing of the local data can be done in parallel. This will require $2d2^n$ multiplications/divisions, for a time of $2d2^nC_*$. Each agent must then send the 2^n values in its potential to the merging agent, requiring a communication time of: $C_f + 2^nC_v$. Again, this can be done in parallel by the agents. The multiplication/division effort required to integrate each local potential at the receiving agent is: $2 \cdot 2^nC_* = 2^{n+1}C_*$. Putting the pieces together, the cost/time for this strategy to compute the global MAPI is: $(2d2^n + (m-1)2^{n+1})C_* + C_f + 2^nC_v$.

These formulas put us in a position to answer a question that has been of interest in previous DPS research on DSI/DD: what is the relative cost of merging the effects of one agent's local data with that of another agent, by merging local results as opposed to sending the raw data and having the receiving agent (re)process it? We have already seen that the cost to merge results from another agent is $C_f + 2^nC_v + 2^{n+1}C_*$. If the raw data is instead to be sent, the communication costs will be $C_f + d2^nC_v$, and the multiplication/division costs will be $d2^nC_*$. So the total time to merge another agents raw data is: $C_f + d2^nC_v + d2^nC_*$. Assuming that $d >> 2$, it is clearly more efficient to merge the effects of two agents' data by communicating results (a potential) rather than raw data.

Obviously, using a single agent to merge a set of agents' local results is not the only strategy that could be used in MAS-based DSI/DD. If we distribute the merge process, we can take further advantage of the parallel computing power of the agents. One scheme that we have analyzed involves successive exchanges of fractions of the potentials among "partner" agents, with multiplication/division steps being done in parallel. $lg(m)$ steps are required to perform the merging (we will assume that m is a power of 2) so the total cost of communication is $\sum_{i=1}^{lg(m)} C_f + 2^{n-i}C_v = lg(m)C_f + (2^n - 2^n/m)C_v < lg(m)C_f + 2^nC_v$, and the total cost of computations is $(2^n - 2^n/m + 2^n)C_* < 2^{n+1}C_*$. Overall then, the time to merge the potentials is less than $lg(m)C_f + 2^nC_v + 2^{n+1}C_*$ (compared to $C_f + 2^nC_v + (m-1)2^{n+1}C_*$ for the strategy that merges at a single agent). Since the distributed merging strategy trades reduced computation time for more communication, the more effective strategy will depend on the relative values of C_f and C_*.

Large-scale SI systems are typically faced with massive amounts of data, making MAPI computation intractable. Since DPS focuses on large-scale problems, most work on MAS-based DSI has inherently assumed that *approximate, satisficing* problem solving is not only acceptable but is necessary. One possible approximation is to compute the MAPI of only a subset of the globally available data. The goal in designing such a system would be to develop strategies for selecting data subsets such that the resulting approximate solutions have a high probability of being the same as the MAPI of the entire data set. We have studied such strategies using our simulation system [1]. An obvious approximate strategy is to merge results from only some of the agents, continuing until either the available time is exhausted or some termination criterion is satisfied. We have run simulation experiments with just such a strategy. The simulations can be used to develop statistical information that can be combined with the direct analyses given above to develop formulas for the average cost of these approximate strategies (and their solution quality).

3. REFERENCES

[1] N. Carver and V. Lesser. Domain Monotonicity and the Performance of Local Solutions Strategies for CDPS-based Distributed Sensor Interpretation and Distributed Diagnosis. *International Journal of Autonomous Agents and Multi-Agent Systems*, 6:35–76, 2003.

[2] C. Huang and A. Darwiche. Inference in Belief Networks: A Procedural Guide. *International Journal of Approximate Reasoning*, 11:1–158, 1994.