Domain Monotonicity and the Performance of Local Solutions Strategies for CDPS-based Distributed Sensor Interpretation and Distributed Diagnosis

Norman Carver
Computer Science Department
Southern Illinois University
Carbondale, IL 62901
(carver@cs.siu.edu)

Victor Lesser
Computer Science Department
University of Massachusetts
Amherst, MA 01003
(lesser@cs.umass.edu)

Abstract

The growth in computer networks has created the potential to harness a great deal of computing power, but new models of distributed computing are often required. *Cooperative distributed problem solving* (CDPS) is the subfield of *multi-agent systems* (MAS) that is concerned with how large-scale problems can be solved using a network of intelligent agents working together. Building CDPS systems for real-world applications is still very difficult, however, in large part because the effects that domain and strategy characteristics have on the performance of CDPS systems are not well understood. This paper reports on the first results from a new simulation-based analysis system that has been created to study the performance of CDPS-based *distributed sensor interpretation* (DSI) and *distributed diagnosis* (DD). To demonstrate the kind of results that can be obtained, we have investigated how the *monotonicity* of a domain affects the performance of a potentially very efficient class of strategies for CDPS-based DSI/DD. *Local solutions strategies* attempt to limit communications among the agents by focusing on using the agents' local solutions to produce global solutions. While these strategies have been described as being important for effective CDPS-based DSI/DD, they need not perform well if a domain is nonmonotonic. We had previously suggested that the reason they have performed well in several research systems was that many DSI/DD domains are what we termed *nearly monotonic*. In this paper, we will provide quantitative results that relate the performance of local solutions strategies to the monotonicity of a domain. The experiments confirm that domain monotonicity can be important to consider, but they also show that it is possible for these strategies to be effective even when domains are relatively nonmonotonic. What is required is that the agents receive a significant fraction of the data that is relevant to their subproblems. This has important implications for the design of DSI/DD systems using local solutions strategies. In addition, while the work indicates that many DSI/DD domains are likely to be "nearly monotonic" according to our original definitions, it also shows that these measures are not as predictive of performance as other measures. This means that near monotonicity alone does not explain why local solutions strategies have performed well in previous systems.

1 Introduction

The explosive growth of computer networks and advances in networking capabilities are key factors behind the increasing interest in *intelligent agents* and *multi-agent systems* (MAS). Networks create the potential to access a great deal of computing power and information, but new models of distributed computing are often required to take full advantage of these resources. *Cooperative distributed problem solving* (CDPS) is the subfield of MAS that is concerned with how large-scale problems can be solved using systems of agents, distributed among a set of networked computers, working together cooperatively.[1] There are a wide variety of potential applications for CDPS techniques, including:

[1] CDPS is one of the earliest areas of research in MAS (e.g., [14]). It is also often simply called *distributed problem solving*. We prefer to use CDPS to emphasize that the agents are assumed to be cooperative.

sensor interpretation in distributed sensor networks; vehicle traffic monitoring and control; detection and diagnosis of faults in computer and telecommunications networks; information retrieval from distributed information sources and digital libraries; distributed perceptual processing for cooperating robots or autonomous vehicles; computer-supported cooperative work; and situation assessment for decision support systems. For appropriate applications, CDPS approaches can offer increased processing power, reduced computer and communication costs, more flexible processing, and increased reliability—as compared to centralized systems.

Despite the interest in MAS and the promise of CDPS, the current state of the field makes it difficult to build CDPS systems for real-world applications. There are not yet formal modeling techniques that are fully adequate for making quantitative predictions about how a CDPS design will perform in a particular domain, nor are there known classes of strategies that can guarantee a certain level of performance (in an appropriate domain). This means that it is difficult to consider a potential application and decide whether CDPS techniques are appropriate, make quantitative predictions about the performance that might be obtained from a CDPS system, or determine how to structure the system. In fact, relatively little research has been done to develop a broad understanding of how system and domain characteristics interact to affect the performance of CDPS systems, or to identify strategies that will perform well under particular conditions. Most CDPS research has limited its experimentation to small numbers of scenarios, to demonstrate that a particular technique can improve performance in *some* cases. One reason for this is that the systems and domains being studied have often been too complex for it to be practical to conduct the very large numbers of experiments that would be necessary to "map out" the interactions between CDPS strategies and domain characteristics.

We believe that it is critical that better CDPS modeling and analysis frameworks be developed if there is to be significant further application of CDPS techniques. This paper reports on the first results from an analysis system that we have created to improve our understanding of how the performance of CDPS systems is influenced by the characteristics of domains and CDPS strategies. Initially, we are focusing on *distributed sensor interpretation* (DSI) and *distributed diagnosis* (DD). The system is *simulation* based: results are obtained through empirical means, using abstract models of domains and CDPS systems. This makes it possible to do the large amount of experimentation that is required to develop quantitative models of how DSI/DD domain characteristics interact with CDPS strategies. An important consideration with simulation approaches is the need for domain and problem-solving models to be realistic enough that results can be related to real-world domains and strategies. We are using *belief nets* (BNs) and other probabilistic models as the basis for our simulations so that results can be stated in terms of standard probabilistic criteria.

The subject of the experiments reported on here was suggested by a paper of ours that appeared in AAAI-96 [5]. In that paper, we noted that much of the work on CDPS-based DSI had implicitly assumed that the "local solutions" produced by the agents would play a critical role in allowing global solutions to be produced without excessive communication among the agents. Because solutions are abstractions of the input data, they can be very compact relative to the raw sensor data. If CDPS systems can develop global solutions by communicating mainly the local solutions, this can drastically reduce the amount of information that must be communicated as compared with systems in which a great deal of the raw data must be shared. As we pointed out, though, the *nonmonotonicity*[2] of virtually all real-world domains means that there is no guarantee that local solutions (which are based on incomplete data) will be useful in identifying global solutions.

Nonetheless, "local solutions strategies" had been used with apparent success in several research systems. We suggested that the reason for this may be that many DSI (and DD) domains are what we termed *nearly monotonic*. The basic idea was that even though these domains are nonmonotonic in a strict sense, they nearly behave as if they

[2] For SI/diagnosis, nonmonotonicity refers to the fact that the best interpretation/diagnosis can change as additional data is considered. Only if a domain is (completely) *monotonic* would this not be the case. Nonmonotonicity is discussed further in Section 4.

are monotonic once certain conditions are achieved (like moderately high belief in the local solutions). Thus, while local solutions will not be *guaranteed* to be globally correct, in nearly monotonic domains they can be very likely to be correct, and so may be useful in developing global solutions.

This was our intuition, but many questions remained. Are DSI/DD domains indeed often "nearly monotonic?" Will local solutions strategies definitely perform well if a domain is nearly monotonic? Is it necessary for a domain to be nearly monotonic for local solutions strategies to be effective—or can such strategies be useful even in domains that are fairly nonmonotonic? Since the degree of monotonicity of domains will vary, what strategies are appropriate for domains with particular degrees of monotonicity and what level of performance can one expect? How is performance affected by design parameters like the number of agents and the problem decomposition? Furthermore, how should one assess the "degree of monotonicity" of a domain so the measure is meaningful for predicting the performance of local solutions strategies?

What these questions show is that it is not sufficient simply to note that local solutions will be useful if a DSI/DD domain is "almost monotonic." This observation provides virtually no guidance when designing a CDPS system, since it is extremely unlikely that any real-world domains are monotonic.[3] The development of modeling and analysis techniques that can answer the kinds of questions posed above is what is needed to support the design of CDPS systems. Only then will designers have the tools to determine whether a particular strategy should be considered for a potential application and to make predictions about the performance that might be obtained. The experiments reported on here are intended, in part, to demonstrate the ability to use our simulation approach to answer these kinds of questions. The results provide the first substantial *quantitative* information about the effect that domain monotonicity has on the performance of DSI/DD strategies that rely largely on local solutions. In fact, they represent one of a quite small number of quantitative studies of the relationship between a general domain characteristic and the performance of a class of CDPS strategies.

The experiments confirm that the monotonicity of a domain can be an important factor to consider when deciding whether to use "local solutions strategies" for CDPS-based DSI/DD. However, the situation is more complex than simply: these strategies work well if the domain is almost monotonic and work poorly otherwise. It turns out that their performance is strongly dependent on the fraction of the relevant data that each agent receives. When agents receive a significant fraction (e.g., 50%) of the data relevant to the events they are attempting to identify, there are local solutions strategies whose performance is excellent—and largely independent of the degree of monotonicity of the domain. As the fraction of the data decreases, however, performance rapidly degrades. For example, if agents receive only 20% of the relevant data, a domain must be quite monotonic to achieve a high probability of producing correct global solutions. These findings have important implications for the design of DSI/DD systems: local solutions strategies should be considered only if the data relevant to each event can generally be limited to a very small number of agents (or the domain is highly monotonic). Additional experiments do indicate that many DSI/DD domains are likely to be "nearly monotonic" according to the definitions we gave in [5]. However, they also show that these definitions do not correlate well with performance. Another contribution of the work to date is the development of two measures of domain monotonicity that are shown to correlate well with the performance of local solutions strategies.

The next three sections provide background on the key subjects of the papers: (1) distributed SI and diagnosis; (2) CDPS and the locals solutions strategy being evaluated; and (3) domain nonmonotonicity and near monotonicity. Section 5 summarizes the capabilities and basic design of the analysis system. Section 6 contains the results from the experiments. The paper concludes with a section on related research and a section that recaps our conclusions to this point, and lists our future plans for experimentation and extensions.

[3] Being nearly monotonic (as in [5]) is not the same thing as being almost monotonic—see Section 4.

2 Distributed Sensor Interpretation and Distributed Diagnosis

2.1 Sensor Interpretation and Diagnosis

Sensor interpretation involves the fusion of data from one or more sensors, over time, to determine the situation in the environment. An *interpretation* of a sensor data set is typically expressed as a set *events* that could have occurred in the environment and would explain the sensor data. Events may be things like aircraft or other vehicles moving through the sensed region, the detection of walls or other obstacles as a robot moves, or so forth. Like SI, *diagnosis* involves the determination of a set of events that could explain a data set. Here the events are *diseases* that patients have or *faults* in a system, and the data will be patient symptoms or information on improper system performance. Each set of explanatory diseases/faults is called a possible diagnosis.

Typically, there will be multiple possible interpretations/diagnoses for any reasonably-sized set of data/faults. The algorithm used by an SI/diagnosis system must not only identify possible interpretations/diagnoses, it must also select the "best" of the possibilities to report as the *solution*. Three commonly used probabilistic standards for selecting solutions are:[4]

1. The *Maximum A Posteriori* Interpretation (MAPI): MAPI(D)= argmax($P(I_i \mid D)$);
2. The *Maximum Likelihood* Interpretation (MLI): MLI(D)= argmax($P(D \mid I_i)$); and
3. The *Probable Hypotheses* Interpretation (PHI): PHI(D,t)= $\{\text{E} : \text{E is a toplevel event and } P(E \mid D) \geq t\}$, where t is a threshold for inclusion of a toplevel event.

The MAPI will be used as the solution standard for the experiments reported on in this paper, since it is considered optimal (when event/fault priors are known). If one has a *belief net* (BN) model of a domain, the MAPI can be determined by plugging the available data into evidence nodes and appropriately propagating its effects [20, 23]. For many real-world SI and diagnosis domains, though, it is impractical to compute the MAPI (computing the MAPI has been shown to be NP-hard in the worst case [26]). Instead, systems must use *approximate, satisficing* strategies. For example, the PHI would be considered an approximate strategy as would search-based BN inference methods [7, 18]. The need for approximate strategies is particularly an issue for real-world SI domains, because it is virtually never practical to develop the kind of complete domain model needed for a BN.[5] Nonetheless, it is appropriate to use the MAPI in the experiments because even if real-world systems cannot compute the true MAPI, they will often attempt to approximate it.

There are a number of reasons why we feel it is reasonable to focus on DSI/DD initially. For one thing, DSI and DD are increasingly important in many fields. *Sensor fusion* is a major area of research because advances in sensor technology are leading to the deployment of large networks of sophisticated sensors. Interest in getting robots and other computer systems to interact more autonomously with their environments is also driving the introduction of sensors into a wider array of environments (involving distributed characteristics). DD is increasingly important to automate fault detection as computer and telecommunication networks become larger and more complex. In addition, DSI and DD are closely related to the more general task of *situation assessment*, which is a critical aspect of most *decision support systems*—another rapidly growing application area. DSI and DD are also of interest for methodological reasons. They are particularly amenable to study with formal probabilistic methods. Because they

[4] In these definitions, I_i represents a possible interpretation/diagnosis (i.e., a set of event hypotheses) and D represents the data set. The MAPI is also known as the *most probable explanation* (MPE) [20]. Non-probabilistic standards have been explored as well—e.g., the *most parsimonious cover* [21].

[5] A detailed discussion of the differences between SI and diagnosis can be found in [6]. The key issue is that while diagnosis problems typically involve a fixed set of diseases/faults and possible data, SI problems involve multiple and often indeterminate numbers of *instances* of each event and data type. This means that SI systems must use *constructive* approaches, in which even the *possible* interpretation components must be determined as a part of the SI process. Furthermore, the possibility of multiple instances of events gives rise to the *data association problem* [1] and exponential growth in the number of possible interpretations with the amount of data that is evaluated.

have been a major focus of much of the previous research on CDPS, there are are both unanswered research questions and results that can help guide this research. Finally, previous experience has shown that complex DSI problems can be used to study most of the major issues that occur in CDPS.

2.2 Centralized vs. Distributed SI/Diagnosis

The types of SI problems for which CDPS techniques are of interest are generally large-scale problems involving multiple sensors (and perhaps multiple types of sensors). Typically, such large-scale sensor systems are "inherently distributed" because the sensors are geographically (spatially) distributed. As an example of the potential complexity of such problems, Brooks and Iyengar [2] state that a battlefield analysis system "may have to handle up to 100,000 reports from 156 separate sensor platforms covering over 800 km^2 each minute."

In *centralized* approaches to such problems, the data from all of the sensors is transmitted to a single "agent" that does all of the computation to interpret the data. In CDPS-based distributed approaches, both the data and the computation to interpret the data will be distributed among a group of SI agents. Typically, each sensor will be associated with a single agent and only that agent will have (direct) access to the sensor's data. Each agent may have multiple sensors under its control, either the same type with different coverage regions or different types with similar coverage regions. Generally, the agents will have to communicate with one another and share data or other information in order to determine an overall solution interpretation.[6] This is a key issue, because the choice of how to structure the interactions among the agents can have a big effect on the performance of these systems (we will consider this further in Section 3).

Distributed approaches to SI can have many advantages, particularly when the sensors (and their agents) are geographically distributed. For example, the use of multiple processors (operating in parallel on subsets of the data) has the potential to speed up processing and/or reduce the cost of the computing hardware. System costs can also be reduced if each agent is located in close physical proximity to its sensor(s), because this will lower the bandwidth requirements of the communications elements since much of the processing of each sensor's data will be done locally. Distributing the interpretation processing so that it is local to the sensors can have the added benefit of improving the real-time control over active, controllable sensors. Another important reason for considering CDPS approaches is that they can lead to more reliable systems, whose performance can degrade "gracefully" should processors and communication links fail.

As with SI, CDPS techniques are applicable principally to diagnosis involving large-scale, inherently distributed systems such as telecommunication networks and computer networks. An important application with these networks is the detection and diagnosis of faults: determining that certain components are functioning abnormally. Fault diagnosis is based on information like the level of packet traffic on a particular link, the failure of a "node" to respond to a message, and so forth. Some of this information may be routinely or periodically gathered to detect the possibility of faults, while other information may be the result of tests that agents actively decide to run once a fault is suspected.

In centralized approaches to network fault diagnosis, a single "agent" collects all the information and diagnoses the state of the network. In distributed approaches, a set of distributed agents are responsible for diagnosing the system. Typically, the network is considered as a set of subnets, with a different agent "responsible" for diagnosing each subnet. Only this agent has direct access to information about the performance of the subnet, and only this agent can directly test components on the subnet (i.e., only that agent can communicate with subnet components without utilizing the network being diagnosed). Furthermore, it is common for detailed knowledge about a subnet (like components and topology) to be available only to the subnet's agent, with agents having only very abstract

[6] Obviously at a minimum the results from the individual agents must somehow be combined to arrive at a complete solution, but this is not the key issue in most CDPS-based DSI systems.

knowledge of the overall network. For instance, all an agent may know about the overall network is what subnets are directly connected to its own subnet.

Distributed diagnosis of network faults has many of the same potential advantages over centralized diagnosis as distributed SI has over centralized SI, and these advantages may in fact be easier to realize than in DSI systems. For example, in networks where component failures are rare, fault diagnosis can often be accomplished by a small number of agents working together—those responsible for the subnets in which components have failed. Of course, network-wide diagnosis will sometimes require that many diagnosis agents communicate with one another. An agent may need other agents to run tests because it cannot differentiate among multiple possible faults (diagnoses) in its subnet, or several components could have failed in different subnets.

3 CDPS-Based SI and Diagnosis

CDPS has traditionally been the branch of MAS that has been concerned with how large-scale problems can be solved by systems of cooperative and often homogeneous, distributed intelligent agents.[7] This section provides an overview of the key issues that arise in designing and assessing CDPS systems for SI and diagnosis. It also introduces the basic system strategy that will be considered in the remainder of the paper.

3.1 Basic Issues in CDPS

In CDPS, an overall problem is decomposed into a set of subproblems and each subproblem is distributed to a particular agent that will have primary responsibility for solving the subproblem. For some applications, the decomposition and distribution phases are critical. With inherently distributed problems such as we are considering, the decomposition and distribution phases are generally straightforward. For example, in DSI each agent receives data from its associated sensor(s), so a "natural decomposition" of the overall SI problem is for each agent to interpret the data from its sensor(s) and develop an interpretation for the region covered by these sensor(s). The agents' individual subproblem solutions (interpretations) must then be combined in some way to produce a solution to the overall problem: determining what is going on in the entire region covered by the sensor network.

The key issue in such situations is the existence of inter-agent *subproblem interactions* (constraints). These arise because the agents' subproblems (as determined by the data each agent gets from its sensors) are not *independent*. For instance, consider a situation assessment application involving the tracking and interpretation of vehicle movements. Each vehicle will typically generate data across multiple agents' sensors because: (1) it will move from a region covered by one agent's sensor(s) to a region covered by another agent's sensor(s), (2) sensor overlap will result in multiple simultaneous detections by different sensors, and (3) environmental phenomena can cause sensor data for regions other than the one the vehicle is physically in. If each agent's subproblem is to interpret its local sensor data, factors like these will lead to multiple agents developing interpretations about the same vehicle—interpretations that may not be consistent.

Subproblem interactions are not limited to these "natural decompositions" either. Consider the obvious alternative decomposition in which each agent is responsible for tracking and identifying the purpose of particular vehicle(s). Because of the factors noted above, agents will clearly need to get data from one another. Unfortunately, there is generally no "easy" way to determine what data is relevant to each agent's vehicle problem(s) (e.g., using position information). The first issue one faces is that even identifying the vehicles that should be tracked requires interpretation, so multiple agents may identify the same "new" vehicle to be tracked (and determining that this is the case can be difficult). More generally, it can be impossible to determine whether particular data is relevant to some event without at least partially interpreting the data (consider, for instance, situations where reflections can cause

[7] Other areas of MAS have dealt with systems in which the agents are heterogeneous, self-interested, or working on different problems.

vehicles to be detected at multiple positions or where vehicles in different regions may be acting in concert with one another). What this all means is that it is virtually never practical to decompose DSI/DD problems so that subproblem interactions are eliminated.

The existence of subproblem interactions means that CDPS agents cannot simply solve their subproblems in isolation and then combine the local solutions (e.g., by taking their union). Instead, they must communicate during the problem-solving process—at least to insure that their local solutions are *globally consistent* and perhaps also to obtain data or other information that is needed to solve their subproblems. A *coordination strategy* (or protocol) specifies how agents will interact: when and with whom they will communicate, and what information they will send or request.[8] In addition to coordination issues, a complete *CDPS strategy* must also specify how each agent will solve its local subproblem(s) (what algorithms each will use) and how the system will ultimately produce a global solution. The basic CDPS strategy that is considered in this paper will be described in Section 3.3.

3.2 CDPS Performance Metrics

In attempting to understand how the performance of CDPS systems is affected by the interactions between strategies and domain characteristics, there are a number of aspects of the performance of CDPS-based DSI/DD systems that one might evaluate. Among the most important are: (1) solution quality, (2) time to solution, and (3) amount of information that must be communicated. Solution quality has been a focus of our initial development of the analysis system. In SI/diagnosis problems, no matter how fast solutions are produced, if they are not "high quality" then they will probably not be useful. Of course, this raises the question of how to judge whether a CDPS system's performance is "high quality" or not. Given a metric like solution quality, one must have a standard of comparison. One possible standard is solution correctness: a system produces high quality solutions if there is a high probability that the solutions correctly identify the events/faults that have actually occurred. This standard is problematic when evaluating CDPS systems, however, because domains inherently vary in how accurately SI/diagnosis can be accomplished. A better approach is to compare the solution correctness performance of the CDPS system to the performance of a centralized system (that had access to all of the data available to the set of distributed agents). This is the standard that we will use in this paper.[9] Not only does it allow the effects of a particular distributed approach to be separated from the inherent limitations of the domain, it also compares the CDPS approach to the main alternative design paradigm.[10]

While solution quality has been our main focus, the other key measures of CDPS performance have not been ignored. The CDPS strategies that we are considering are of interest precisely because they can require limited communication among the agents. As part of our analyses, we can also produce statistical information about the amount of information that would be communicated among the agents. We are not yet explicitly considering time to solution, however time to solution and communication levels are typically interrelated: the more agents must communicate, the more time they must expend packaging information and processing received information, and the more time they may have to spend waiting for needed information. Thus, when conditions are such that communications are low, time to solution is likely to be better than it would be otherwise.

[8] The coordination strategy may be partially determined by the *agent organization*, which defines the roles and relationships particular agents have in the overall system.

[9] Note that we are implicitly saying that *approximate* solutions are acceptable in CDPS-based DSI/DD. For example, we will want a CDPS-based DSI/DD system to have a suitably high probability of identifying the MAPI of the data, but we will not require that it always identifies the MAPI. CDPS commonly involves approximate, satisficing problem solving.

[10] Whether the performance of a CDPS system is ultimately judged to be "acceptable" or not will depend on the goal one had in using a distributed approach. For example, if the reason was to take advantage of parallel processing to improve performance, then solution quality and time to solution should be at least as good as what could be obtained from a centralized system. On the other hand, if the reason was to save money by not having to use a supercomputer, then even if the CDPS system performs slightly worse than a centralized system in terms of time (or even solution quality), this may be a reasonable trade-off. If the reason was to develop a system that was less vulnerable to hardware failures, then a very different set of metrics must be used to judge success.

3.3 The Consistent Local Solutions Strategy (CLSS)

The main strategy that will be considered in this paper is one that has been used in several CDPS-based DSI research systems (e.g., [4, 10, 15]). We have dubbed it the *consistent local solutions strategy* (CLSS) [5]. Basically, agents first independently solve their local subproblems (using some interpretation algorithm), they then transmit their local solutions to all other agents whose solutions may have interdependencies, and if these agents' local solutions are "consistent," they are merged—e.g., by taking the union of the solutions—without doing a more complete assessment of what the globally best solution is.[11] The CLSS is of interest because it has the potential to be very efficient. If local solutions are often consistent then agents will be able to produce global solutions without transferring large amounts of data among themselves. This will result in relatively low communication rates because interpretation solutions are abstractions of the data, and can generally be represented using a small fraction of the storage that would be required for the data they are derived from. Time to solution would also be good because agents interpretation algorithms would be applied only to subsets of the data and the agents would not have to redundantly process data. Furthermore, there can be significant delays in problem solving when agents require substantial amounts of raw data from other agents, because this data will be transmitted only once an agent realizes it needs the data and makes a specific request for it or once the source agent realizes the other agent needs it.

Key questions that must be answered to determine whether the CLSS performs well are: (1) will the agents' local solutions frequently be consistent (so that they can be efficiently combined), and (2) does local solution consistency produce high quality global solutions? Much of the work on CDPS and CDPS-based DSI has assumed that these questions will be answered affirmatively. For example, Lesser and Corkill [15] referred to "consistency checking" of the tentative local solutions with results received from other nodes as "an important part" of their approach to CDPS. Unfortunately, the *nonmonotonicity* of virtually all real-world SI and diagnosis domains means that there is no guarantee that local solutions (based on incomplete data) are likely to be consistent, nor that consistency will be useful in identifying the best global solutions. Providing quantitative assessments of the importance of domain monotonicity to CDPS-based DSI/DD strategies that rely on local solutions is the primary contribution of the results presented in this paper. Before those results are presented, however, the issue of domain monotonicity will be considered in more detail.

4 Monotonicity, Nonmonotonicity, and Near Monotonicity

In a knowledge representation context, *nonmonotonicity* refers to the notion that the set of facts that are considered to be true does not grow monotonically as additional information is acquired. For SI/diagnosis, nonmonotonicity means that as additional data is processed (used to determine the solution), the interpretation/diagnosis that is considered best, may change. For example, consider a data set $\mathcal{D}$= $\{d_1, d_2, d_3, d_4, ...\}$ with two possible (not mutually exclusive) causes E_1 and E_2. Just because the interpretation $\{E_1, \overline{E_2}\}$[12] is considered best given the data subset $\{d_1, d_2\}$, this does not mean that it will remain so given $\{d_1, d_2, d_3\}$, let alone all of $\mathcal{D}$ (it does not even mean that E_1 will continue to be part of the best interpretation). Only if the domain is *monotonic* would this be the case. In the AI literature, a domain is classified as nonmonotonic unless it is monotonic (i.e., *completely* monotonic). Some knowledge

[11] This is obviously a very abstract description of the CLSS. We are deliberately being vague about many details of the strategy. In part this is to cover several variations that all fit this same general pattern, but it is also because only abstract models of systems and strategies are required in a simulation framework. One critical detail that has been ignored here is the question of what should be done when the local solutions are *not* consistent. We will explore several possibilities in the experiments section. Another issue that can be much more complex than it may appear is the question of when local solutions are "consistent." In the experiments reported on in this paper, agents are doing exact interpretation and every agent receives data about every event. This means that consistency reduces to *equality*. Consistency can be a complex issue, however, particularly when approximate interpretation strategies are being employed. An in-depth evaluation of consistency is beyond the scope of this paper.

[12] Event E_1 occurred and event E_2 did not.

representation languages assume domain monotonicity (notably, first order logic). However, nonmonotonicity is inherent in probabilistic representations via the notion of conditional probabilities [23]: the degree of belief in a statement is conditioned on the relevant evidence, and belief may go up or go down as the conditioning evidence is increased.

It is clear that the nonmonotonicity of a domain can affect the performance of strategies like the CLSS that rely on local solutions. Consider a situation in which there are two alternative interpretations I_a and I_b for a data set $\mathcal{D}$.[13] Now suppose that the data in $\mathcal{D}$ is distributed to two agents, A_1 and A_2, so that A_1 has data subset D_1 and A_2 has D_2, with each $D_i \subset \mathcal{D}$ ($D_i \neq \mathcal{D}$). Unless the domain is (completely) monotonic, it is entirely possible to have $P(I_a \mid D_1) > P(I_b \mid D_1)$ and $P(I_a \mid D_2) > P(I_b \mid D_2)$, but $P(I_a \mid D_1 \cup D_2) < P(I_b \mid D_1 \cup D_2)$. In other words, even though interpretation I_a is the most likely (best) interpretation given each agent's local data set separately, it may not be the globally most likely solution—even though the local solutions are consistent (here identical).

In general then, the consistency of local agent solutions provides *no guarantees at all* about the quality of the global solution. Does this mean that researchers intuitions about the importance of local solutions were wrong? Will local solutions be useful only if a domain is essentially monotonic? What about the fact that local solutions strategies had been used with apparent success in several DSI research systems? As noted earlier, in [5] we suggested that a possible explanation for why strategies like the CLSS might be effective in spite of nonmonotonicity is that many SI (and diagnosis) domains may be what we termed *nearly monotonic*. While nonmonotonic in a strict sense, under appropriate conditions these domains nearly behave as if they are monotonic. For example, once a fairly high degree-of-belief about an event is attained, it is very unlikely that further evidence would cause the belief to drop significantly, and it is unlikely that the event would not be part of the best global solution.

No single precise definition of near monotonicity will always be appropriate, since the value of a measure will depend on factors like the particular strategy being used to select solutions. In [5], we suggested several ways of statistically characterizing the monotonicity properties of SI/diagnosis domains that we believed could be useful for predicting how well local solutions strategies might perform. The basic approach was to relate the ultimate properties of event hypotheses (if all the data were processed) to their properties based on a subset of the data, using properties such as belief and solution membership. Given our focus in this paper on MAPI solutions, the two most relevant characterizations were:[14]

- the probability that an event hypothesis E would ultimately be in the MAPI, given that its current belief is p: $P(E \in MAPI(\mathcal{D}) \mid P(E \mid D) = p)$;
- the probability that an event hypothesis E would ultimately be in the MAPI, given that it is in the current MAPI and its current belief is p: $P(E \in MAPI(\mathcal{D}) \mid E \in MAPI(D),\ P(E \mid D) = p)$.

Being "nearly monotonic" would require that once an event hypothesis achieves a reasonable degree of belief (e.g., greater than 0.7), these probabilities would be "sufficiently high" to make it appropriate to assume that the event is in the global solution. This would allow a strategy like the CLSS to be effective, because if the appropriate conditions can be achieved, the consistency of local solutions would be highly predictive of global correctness.

The notion of near monotonicity developed from our observation that nonmonotonic domains could differ substantially in how they fail to be monotonic. For instance, in one domain the best interpretation may only rarely change as additional data is considered, while in another, changes may be frequent—even though both domains are "nonmonotonic." Furthermore, the ability to develop successful SI and diagnosis systems requires that there be some structure to the nonmonotonicity of a domain. At the very least, one must be be able to assemble a set of data sources such that if all of the resulting data is processed, it will be unlikely that access to additional data would change the answer. If this were not the case—if data from additional sensors/tests was likely to lead to different solutions—this

[13] For example, I_a could be $\{E_1, \overline{E_2}\}$ and I_b could be $\{\overline{E_1}, E_2\}$.

[14] As above, we are assuming that there is a complete data set $\mathcal{D}$ and that only a subset D of that data has been processed.

would imply that the system did not have a high probability of identifying the correct interpretation/diagnosis. A similar situation must exist if approximate interpretation is to be effective (there will have to be some way to ensure that an approximate solution is sufficiently unlikely to change as a result of doing further processing). Near monotonicity is an attempt to characterize useful structure in the nonmonotonicity of SI/diagnosis domains.

As we will see in Section 6.3, however, the particular characterizations of domain monotonicity given above do *not* turn out to be predictive of the performance of CLSS-like strategies. While it appears that many DSI/DD domains will indeed be nearly monotonic according to these definitions, it can be difficult to achieve appropriate belief levels in CDPS systems, so local solutions strategies may not perform well even if the domain is "nearly monotonic." As part of the work being presented here we will demonstrate two alternative measures of domain monotonicity that do correlate well with the performance of the CLSS. These measures will be defined in Section 5.2. Demonstrating that the previous characterizations of near monotonicity are not predictive of how the CLSS performs, but that other measures are is another of the useful contributions of this work. In fact, despite the obvious value in understanding the monotonicity properties of a domain, we are not aware of any previous research that has been done to measure or otherwise characterize the "degree of monotonicity" of domains.

5 The Analysis System

The analysis system provides three main types of capabilities: (1) the ability to model a wide range of SI/diagnosis domains; (2) software tools for analyzing properties of these domains; and (3) software for simulating the application of CDPS strategies in specified domains and collecting performance information. In this section, we will describe how domains are represented and the various tools that have been implemented—including the main measures of domain monotonicity that are used in the the experiments described in Section 6.

5.1 Domain Representations: Grammars and Belief Nets

The analysis system supports two different formalisms for modeling SI and diagnosis domains: *stochastic context free grammars* (SCFGs) [11, 19] and *belief nets* (BNs) [20, 23]. SCFGs have been used as the primary representation to this point. However, many of the analysis tools use BN inference techniques, so we have developed software to convert between SCFG and BN models (and to produce explicit joint models when practical). SCFGs may not be familiar to many people, so they will be introduced and contrasted with BNs in this section.

In our discussion of grammars and domain models we will assume that both the events and data are *binary*. The terms *positive* and *negative* will be used for the two possible values. Positive means that the event occurred or the data was its non-default value. Negative means that the event did not occur or the data had its default value (the value that the sensor/test would report if no events had occurred). When we say that "an event E causes datum d," we mean that the event causes the sensor/test to report the positive value for the datum type d. If specific data is being shown, then notation like d will be used to denote the positive value and $\overline{d}$ the negative. For example, if d corresponds to a particular acoustic frequency band in the environment, d would denote that energy was detected in the band while $\overline{d}$ would mean it was not.

An SCFG is a context free grammar in which each production has an associated probability that denotes the likelihood that the production would actually be used in a derivation when it is applicable. For example, a production $0.3 : E \rightarrow d$ means that 30% of the time, an E would produce only d. The probabilities are subject to the constraint that they must sum to one for productions with the same LHS (single nonterminal). An SCFG constitutes a complete probabilistic model of a domain, but while an SCFG can directly be used to generate simulated data sets, it is not straightforward to use one to make most probabilistic inferences. Instead, an SCFG must be converted into a joint or BN model of the domain and these models then used in computing desired probabilistic inferences (such as the

conditional probabilities of events given data or the MAPI of data sets). Figure 1 shows two example SCFGs and corresponding BNs.

There are several reasons why SCFGs have been used as the primary domain representation. One reason is that we felt they were a very intuitive way to conceptualize SI/diagnosis domains, given our goal of understanding how domain characteristics relate to CDPS system performance. SCFGs provide a very direct model of the causal relations in a domain. A production of the form $p : E \rightarrow d$ not only denotes that event E is a direct cause of d, but that $P(E \; causes \; d \mid E)$=$p$. Contrast this with the probabilistic information in a BN. There, the causal connection would be denoted by making the E node the (immediate) parent of the d node. If E was the only parent of d then $P(E \mid d)$ (and $P(E \mid \overline{d})$) would need to be provided. $P(E \mid d)$ does not represent the probability that E *causes* d, though. Rather it gives the probability that d *co-occurs* with E (when E has occurred). This will not be the same as the SCFG probability p when there are multiple causes for d. While the conditional probabilities in a BN may be more reasonable quantities to gather experimentally,[15] for our purpose in developing models of domains and understanding their characteristics, we believe SCFGs are useful.

Another potential advantage of SCFGs is that they can easily represent "correlated" events/data. For instance, suppose that some event E will cause the sensor to report *either* the data set $\{d1, d2, d3, \overline{d4}, \overline{d5}\}$ or the set $\{\overline{d1}, d2, \overline{d3}, d4, d5\}$. This could occur if the data depends on the state of the target being sensed or which side of the target is facing the sensor, for instance. An SCFG can easily model this situation by using two E productions with appropriate RHSs. It is less straightforward to represent in a BN. We cannot simply introduce E_1 and E_2 nodes as the immediate children of E, with these particular data sets as their immediate children. BNs assume conditional independence of the "children" of a node, so there is no way to specify E_1 and E_2 as mutually exclusive results of E. Instead, a more complex network representation will be needed. While BNs are capable of representing any probabilistic model (joint), a BN model will directly correspond to the causal structure of a domain only when the immediate results of any event are conditionally independent.[16]

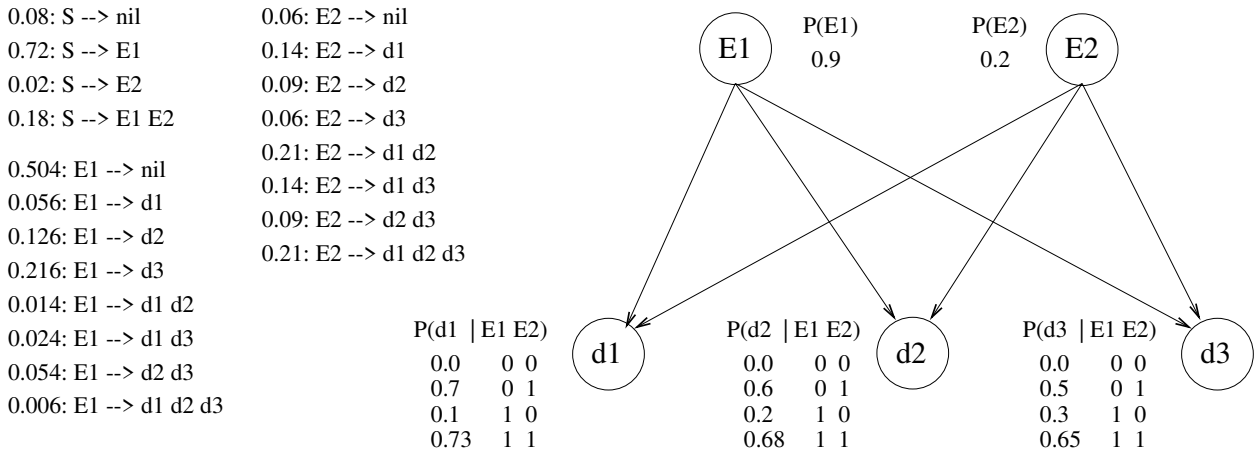
Actually, both SCFGs and BNs have their idiosyncrasies. If more than one event can cause some sub-event or datum, a traditional BN requires that the probability of the sub-event/datum must be specified for *every combination* of the possible causes—even if the causes are independent. Since this results in an exponentially increasing number of probabilities with the number of causes, this has led to the development of BN extensions like *noisy OR* nodes that can more compactly represent independent causes. SCFGs assume that causes are independent, so they do not suffer from this problem. Of course, if the causes do interact then SCFGs have a problem. For example, suppose that events E_1 and E_2 each cause d with high probability, but that they "interfere" with each other so that d is extremely unlikely when both E_1 and E_2 occur. This is straightforward to deal with in a BN (assuming the number of interacting causes is not too large), but an SCFG would require the introduction of a (non-terminal) symbol like $E_{1\&2}$ to explicitly represent the joint occurrence of the interacting events (though this may be reasonable if one is trying to understand how domain characteristics affect performance).

Another reason for our interest in SCFGs is that an extension of SCFGs has been used successfully by Whitehair and Lesser [27, 28] to model both complex SI domains and SI problem solvers that use sophisticated heuristic search techniques. The extension involves the addition of sets of *attributes* to the symbols of the grammar and "*semantic functions*" to the productions—i.e., an *attribute* SCFG.[17] This makes it much easier to model complex, real-world SI domains, where sensor data and events often involve multiple continuous/semi-continuous-valued attributes (e.g., location). We are currently working to extend our analysis tools to support an attribute SCFG domain representation.

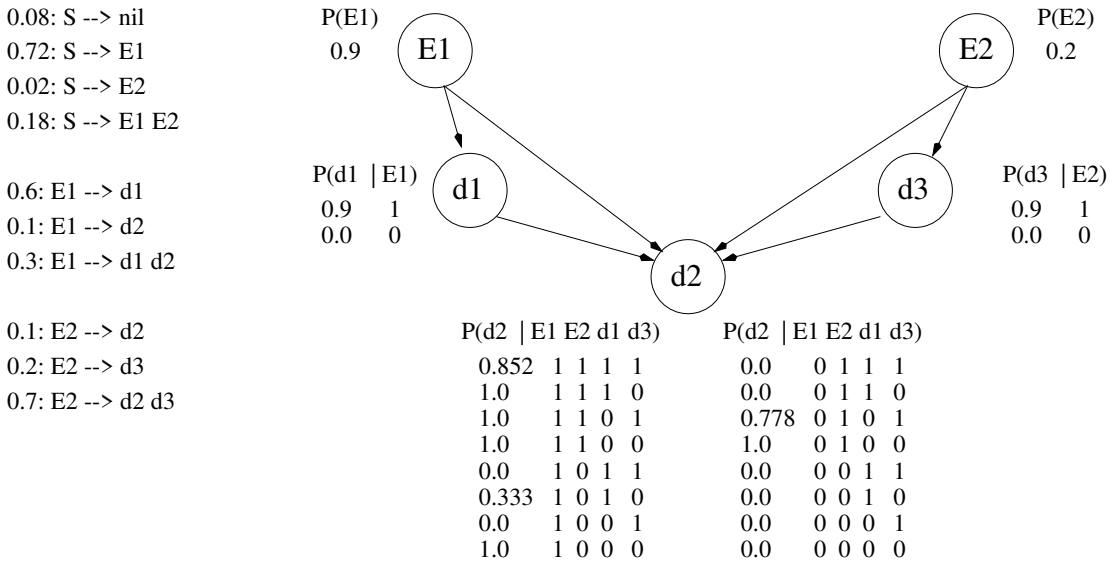
[15] For example in a medical diagnosis domain it is rarely possible to gather direct causal probabilities—e.g., that disease D causes *causes* symptom S with probability p.

[16] See [23] p. 441–442 for a discussion of BN representational efficiency.

[17] Attribute (or attributed) grammars have been explored in research on syntactic pattern recognition. See [12, 19] for additional information.



(A) An "independent" SCFG and corresponding belief net.



(B) A "correlated" SCFG and corresponding belief net.

Figure 1: Example SI/diagnosis stochastic context free grammars and corresponding belief networks.
In the grammars shown, S is the *start symbol*, E1 and E2 are *nonterminals* that represent the events to be identified, and d1, d2, etc. are *terminals* that represent the sensor data. All events and data are binary (positive/negative), as described in the text. Only the positive event/data symbols are shown explicitly in the grammar—e.g., a sequence of productions that does not produce d2 among its terminals implicitly means that d2 was negative. In the (A) grammar, the probability of only event E1 occurring is 0.72, the probability of both E1 and E2 occurring is 0.18, and so forth. Similarly, the probability that an E1 event will produce only data d1 is 0.056, the probability an E1 will produce d1 and d2 is 0.014, and so forth. In both grammars, the probabilities of E1 and E2 occurring are *independent*: all possible combinations of E1 and E2 can be produced (including the case where neither event occurred) and the production probabilities can be computed from the priors p(E1)=0.9 and p(E2)=0.2 by appropriate multiplication. In the (A) grammar, the probabilities of d1, d2, and d3 are also (conditionally) independent. By contrast, in the (B) grammar, the probabilities of d1, d2, and d3 are *not conditionally independent* given the events: each event can produce only a subset of the possible data combinations and the probabilities of combinations cannot be obtained by multiplying the individual data probabilities. The (A) grammar is what we refer to as an *independent* grammar. The (B) grammar is what we refer to as a *correlated* grammar. Note that in the BN probability tables, 1's stand for positive values of the events/data and 0's stand for negative values.

The final reason for our interest in SCFGs has to do with our eventual goal of developing a more general CDPS analysis framework. Grammars like CFGs have been used as the basis for modeling and analyzing a wide range of search-based approaches to problem solving.[18] A key example is the work of Kumar and Kanal—e.g., [13]. Using CFGs as the basis for modeling *discrete optimization problems*, they have been able to analyze the relationships among various AI search techniques as well as dynamic programming and branch and bound algorithms. It is our intention to use similar CFG-based approaches to model search-based CDPS domains other than SI and diagnosis. We believe it will be possible to adapt many of our analysis tools because of the similarities between interpretation and parsing, and the role that parsing and parse trees play in formalisms like Kanal and Kumar's (with a grammar-based search representation, parsing effectively equals problem solving—a parse tree encodes a possible solution).

5.2 Analysis Tools

A number of software tools have been developed for analyzing properties of SI domains defined by SCFGs and/or BNs. Currently, the majority of these deal with the monotonicity of a domain and the ability to reliably perform interpretation/diagnosis in the domain.

The main assessment of monotonicity that will be used to present the results of the experiments is what we call the *all subsets monotonicity measure* (ASMM). Basically, it is the probability that the MAPI of a randomly chosen (non-empty) subset of a data set, will be equal to the MAPI of the entire data set, assuming all subsets are equally likely to be chosen and each data set occurs with the probability determined by the grammar.[19] The ASMM of a (completely) monotonic domain is 1.0, otherwise it is between 0 and 1. As we will show later, the ASMM of a domain correlates well with the performance of CLSS strategies. Formally, ASMM(domain) is:

$$\frac{\sum_{\mathcal{D}_i} \left(P(\mathcal{D}_i) \sum_{D_j \subseteq \mathcal{D}_i} \frac{E(MAPI(D_j), MAPI(\mathcal{D}_i))}{2^{|T|}-1} \right)}{\sum_{\mathcal{D}_i} P(\mathcal{D}_i)}$$

where:

- T is the set of "terminals" in the domain—i.e., the possible data types;
- $\mathcal{D}_i$ ranges over all the possible combinations of the domain data—except the case that represents all "default" data (when no event has occurred). With binary data, this would be all cases except the one in which all the data is "negative." In grammar terms this means that $\mathcal{D}_i$ ranges over all the non-empty subsets of the terminals T, with these subsets then being "completed" by adding the terminals not in the subset with their negative values—i.e., $\mathcal{D}_i = \mathcal{D}_i^{'C}$, where $\mathcal{D}_i^{'} \in \ (\mathcal{P}(T) - \{\emptyset\})$ ($\mathcal{P}(T)$ denotes the power set of T), and $\mathcal{D}_i^{'C}$ is the "completed version" of $\mathcal{D}_i^{'}$ (e.g., if $\mathcal{D}_i^{'}$=$\{d1, d2, d5\}$ then $\mathcal{D}_i^{'C} = \{d1, d2, \overline{d3}, \overline{d4}, d5\}$, when $T = \{d1, d2, d3, d4, d5\}$);
- D_j ranges over the non-empty subsets of $\mathcal{D}_i$; and
- $E(x, y) = 1$ if x=y else 0.

Another monotonicity measure that we have implemented is the probability that the MAPI of an arbitrary subset of the data that is a particular fractional amount of all the data, is the same as the MAPI of the complete data set. For example, the probability that if an agent gets 50% of the available data, its local MAPI solution will be

[18] To give a simple example of using grammars to model general search problems, consider that any *state-space search* domain can be modeled using a *regular grammar*. Recall that in state-space search, actions are encoded as *operators*, and an operator transforms a state (that it is applicable in) into another state. A regular grammar model of a state-space domain has a production $S_i \rightarrow o_j \ S_k$ for each instance where operator o_j is applicable in state S_i and results in state S_k (with the S_i's being nonterminals and the o_j's terminals).

[19] Using the notation from Section 4, it is $P(MAPI(D) = MAPI(\mathcal{D}))$.

the true MAPI. We have termed this the *partial subsets monotonicity measure* (PSMM). The formal definition of PSMM(domain,fraction) is very similar to the definition of the ASMM(domain). The only difference is that instead of D_j ranging over *all* non-empty sets of $\mathcal{D}_i$, it now ranges over all subsets such that $|D_j|/|T| = \text{fraction}$.[20] The PSMM also correlates well with CLSS performance.

Other domain assessment tools evaluate how well particular interpretation/diagnosis strategies performance in a domain, in terms of the probability that they would produce the correct solution (i.e., the actual causes of the data). There are currently tools to evaluate the three probabilistic solution strategies mentioned in Section 2.1 (the MAPI, MLI, and PHI). In Section 6.1 we will refer to the results from these tools as, for example the "probability of MAPI correctness." The tools are constructed in a manner similar to the monotonicity tools: each possible data case is considered, the particular strategy is applied and its result compared to the (distribution of) actual causes for data, with the final result being the weighted average of the fraction of correct interpretations (over the probabilities of the data cases). Note also that whether one considers these to be tools to evaluate interpretation strategies or tools to evaluate domain characteristics depends on the current point of view. One may have a grammar and be trying to find the best interpretation strategy to use with that grammar, or one may be considering domains and be trying to find those that can be reliably interpreted.

The system also include tools to generate simulated data sets that adhere to the statistical properties of the domain models and to simulate the distribution of these data sets among groups of agents in various configurations. These data sets are then used as input to tools that can simulate CDPS-based DSI/DD and centralized SI/diagnosis, using specific abstract CDPS and interpretation strategies, and collect statistics about how strategies perform in the domains. Currently, the performance statistics deal with solution quality, the probability of communication, and the amount of information that must be communicated. In designing the abstract CDPS strategies, several strategies can be chosen for simulating local and centralized SI/diagnosis problem solving. These include the MAPI (which is used in the experiments reported here) and the other probabilistic standards given in Section 2.1. While we spoke originally of these as standards for selecting solutions, here we speak of them in terms of simulations because real-world DSI/DD systems may use different algorithms to achieve these standards or may use algorithms that only *approximate* a standard like the MAPI. A key aspect of our approach to simulation, though, is that SI/diagnosis problem solving is always being simulated with probabilistic approaches so results can be compared to strategies in real systems. CDPS strategies are currently specified via code (Lisp/C++), but we are working to adapt attribute SCFGs to be able to define strategies grammatically.

The analysis system software is written in Common Lisp and C++. The Lisp software has been tested under Lispworks 3.2.2 on Sun workstations and Allegro CL 5.0/5.0.1 under Win9x and Linux. For BN software, we have been using *SMILE*, a library of C++ routines developed by the Decision Systems Laboratory at the University of Pittsburgh, which is the basis for their GeNIe BN system.[21]

6 Experimental Results

This section contains the results from our initial experimentation with the analysis system. Before we present the results, the characteristics of the domain models that were used for the experiments will be described. The main subject of the experiments is the effect that the monotonicity of a domain has on the effectiveness of "local solutions strategies" like the CLSS. These results are presented in Section 6.2. The question of the near monotonicity of DSI/DD domains and its relation to the performance of CLSS strategies is examined next. Later sections will show results from experiments that compared the PSMM to the ASMM, and that considered the effect that common (overlapping) data has on the CLSS.

[20] Using the notation from Section 4, it is basically $P(MAPI(D) = MAPI(\mathcal{D}) \mid |D|/|\mathcal{D}| = fraction)$.

[21] See www2.sis.pitt.edu/~genie.

6.1 The Domains

The experimental results shown in this paper were based on a set of 330 grammars that were selected from over 15,000 randomly generated grammars. The grammars in the set all share certain structural features, but vary in "degree of monotonicity" (as measured by the ASMM). The grammars in the test set correspond to *two level* BNs: they have a set of *top-level* symbols that are produced from the *start symbol* and a set of *terminals* that are directly produced by these top-level symbols. The top-level symbols represent the events/faults that would be reported as answers and the terminals represent the data/findings. Two top-level/event symbols (e.g., $E1$ and $E2$) and ten terminal/data symbols (e.g., $d1$, $d2$,...) were used. The probabilities associated with the *Start* to top-level productions were always such that the top-levels were *independent*. The probabilities associated with the top-level to data productions come in two varieties: *independent* and *correlated*. In the independent grammars, data elements are conditionally independent given each top-level: each top-level can produce every combination of data elements, with probabilities determined through appropriate products of $P(E\ causes\ d_i \mid E)$ type information. In the correlated grammars, the data is not conditionally independent: each top-level may produce only a subset of the possible combinations of data (with non-zero probability). Examples of both classes of grammars appear in Figure 1.

In selecting from the randomly generated grammars, each grammar was assessed to determine its ASMM and the *probability of MAPI correctness* (see Section 5.2).[22] Grammars with a probability of MAPI correctness less than 0.85 were discarded. MAPI correctness effectively represents the best performance that could be obtained from a centralized system. One way of thinking about the 0.85 lowerbound is that domains in which even a centralized system cannot do well are not of interest. SI systems would not be built for such domains without first improving the sensors or adding additional sensors so that the system could perform well. In fact, because of the small number of top-level events being used, we selected grammars to have a probability of MAPI correctness of at least 0.9 when possible. For ASMMs between 0.5 and 0.7, grammars with acceptable MAPI correctness probabilities were very hard to generate, so all of the acceptable grammars in this range were used. With ASMMs greater than 0.7, grammars were randomly chosen from among those generated. In addition, higher MAPI correctness bounds were imposed: for ASMMs between 0.7 and 0.8 the MAPI correctness lowerbound was 0.88, while with ASMMs greater than 0.8 the lowerbound was 0.9. Because behavior generally becomes increasingly similar as monotonicity increases, the density of the chosen grammars was decreased as ASMMs increased from 0.7 to 1.0. No grammars with ASMMs less than 0.5 were generated with two top-levels and ten data elements, and no acceptable "independent" grammars were generated with an ASMM less than 0.65.

The use of two-level grammars may appear to be a major limitation, but this is not the case given the strategies being evaluated. Since the MAPI strategy is used for both the centralized system and for the local agent interpretations, only the top and data levels of the grammar ultimately play a role in the analysis computations. The probability information in a multi-level grammar would effectively be collapsed into a two-level grammar. While it is true that multiple grammar/model levels may be useful both for building models of complex domains as well as for practical (e.g., approximate) DSI/DD systems, their existence does not affect the ultimate ambiguity or monotonicity of the domain.[23]

The choice of two top-levels and ten data elements was made largely to keep experiment runtimes reasonably fast. Two top-level symbols is also a useful abstraction of many SI problems, where data may result from: (1) "noise" (environmental events that are not of interest, sensor noise, etc.); (2) one or more events/targets of interest;

[22] These measures were assessed *exactly* for the selected domains.

[23] Having intermediate-level events/faults is useful because these events inherently represent uncertainty/ambiguity about top-level events. So, instead of always driving interpretation inferences up to the top-level, it may be more efficient to stop at the intermediate levels until additional data is acquired. The availability of intermediate-level event models also can enhance distributed problem solving because agents may now communicate information at different levels of abstraction. Carver and Lesser [4] show how this can be useful for CDPS-based DSI.

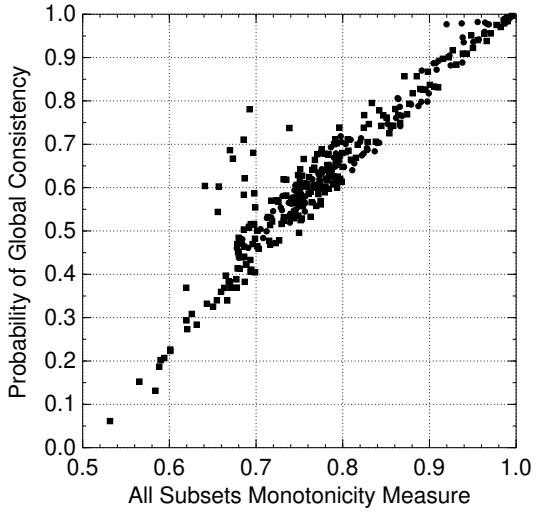
or (3) both noise and target events. The addition of more top-levels would then effectively represent the need to not only identify if there are targets of interest, but to identify what the targets are. This generally makes interpretation more difficult, since the number of possible interpretations (combinations of top-level events) increases exponentially. In limited experiments with four and eight top-levels, the randomly generated grammars tended to have lower ASMM values and lower probabilities of MAPI correctness. Thus the grammars used here represent a fairly easy interpretation environment. However, when grammars were matched by ASMM and probability of MAPI correctness, the results of experiments with larger numbers of top-levels were completely consistent with the results using two top-levels.

Ten terminals (data elements) were found to provide a reasonable trade-off between speed and complexity for the experiments. Ten terminals allows each event to produce up to 1024 different data combinations with various probabilities. Also, because the ASMM is an average over every possible subset of data, one would expect there to be little difference in the way two domains with the same ASMM would perform when considering agents with, say, 20% of the global data, whether that 20% is just two pieces of data or whether it is ten or twenty pieces of data. To confirm the adequacy of ten terminals, a limited set of experiments with twenty terminals were run. Again, the results were consistent with what is presented in this paper (when using the same data percentages for the agents and matching domains by ASMM and probability of MAPI correctness). This is precisely the approach that must be taken to generate a broad model of CDPS performance: using simplified domains so that experiments with large numbers of variations can be run, while conducting limited tests to ensure the results are meaningful for larger, more realistic domains.

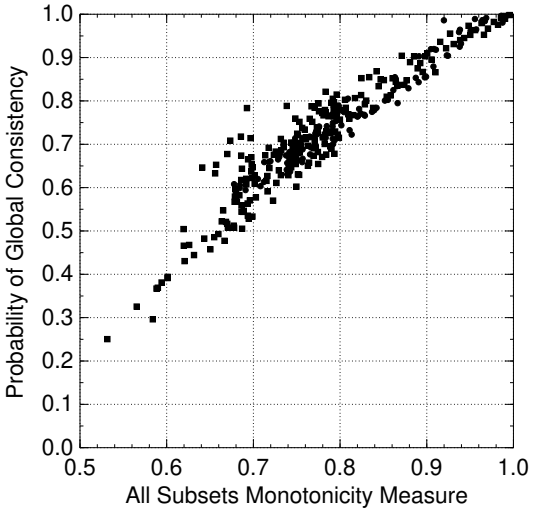
6.2 Domain Monotonicity and the CLSS

This section presents the results of experiments concerning the effect that domain monotonicity has on the performance of several variations of the consistent local solutions strategy or CLSS (described in Section 3.3). In all of the CLSS simulation experiments reported on here, each of the agents was considered to have used the MAPI strategy to produce its local interpretations (based on locally available data). The MAPI of the complete, globally available data set served as the solution comparison standard, since this is the optimal solution a centralized system could produce with access to the same data. The global data was *randomly distributed* among the agents, given the particular distribution characteristics noted for the experiment. This means that each agent potentially received data relevant to *all* of the top-level events, and each agent was expected to produce complete interpretations (i.e., interpretations that identify the occurrence/non-occurrence of every top-level event). Because of this, consistency of local solutions reduces to equality. All the simulations were based on 10,000 data sets for each domain. 95% confidence intervals can be determined from the formula $p = \hat{p} \pm 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{10^4}}$ (Bernoulli trials, with p the true probability and $\hat{p}$ the measured value). Thus, individual data points in the graphs are good to within about 1% (the maximum range is ± 0.0098 when $\hat{p} = 0.5$).

The first set of experiments effectively assume a *two-agent* system. Figure 2 shows the likelihood that the two agents' local solutions will be *consistent*, as a function of domain monotonicity (as determined by the ASMM). Two cases are shown: (A) each data gets 50% of the overall data with no data in common between the agents, and (B) each agents gets 60% of the data such that each agent gets a unique 40% of the overall data and 20% of the data is common to both agents. As one would expect, the more monotonic the domain is (the higher the ASMM), the more likely the local solutions are to be consistent. In addition, having common data among the agents also appears to increase the probability of consistency. What is perhaps unexpected, is the degree to which nonmonotonicity affects global consistency. A reasonable first pass estimate of the probability of global consistency would be $p^2 + (1-p)^2/3$, where p is the probability that a local solution will be globally correct, and we assume these probabilities are independent



(A) Each agent gets 50% of the the global data, no data in common.



(B) Each agent gets 60% of the the global data, 20% of the global data common to the agents.

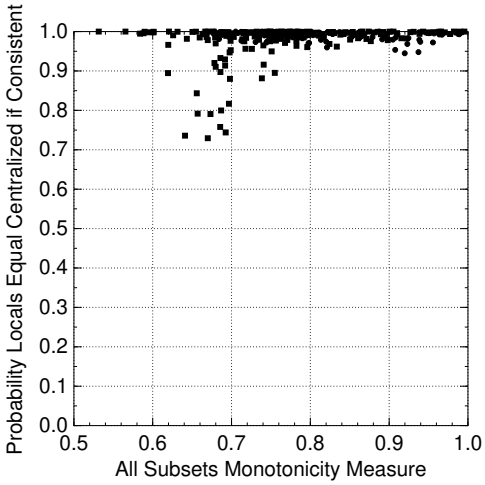
Figure 2: Global consistency of pairs of local solutions.
This graph shows the probability that two agents local MAPI solutions are globally consistent (here identical), as a function of domain monotonicity (ASMM). Results are shown for two cases with the agents having different fractions of the globally available data and different amounts of common data.

for the two agents.[24] Since the probability of correctness with half the data is nearly identical to the ASMM (see Figure 15), we might expect the probability of global consistency for an ASMM of 0.5 to be 0.33 and for an ASMM of 0.6 to be 0.41. In the case where the data is evenly divided between the agents, the probability of global consistency is well below these figures when the domain is relatively nonmonotonic (ASMM < 0.7). This suggests that the probability of both agents correctly identifying the solution is not independent when domains are nonmonotonic.

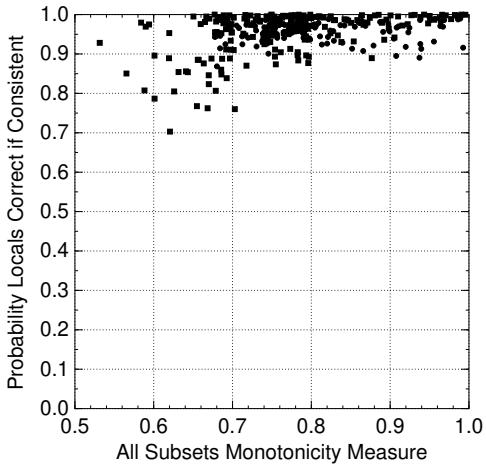
The global consistency results in Figure 2 suggest that CLSS-like strategies may not perform well unless a domain has a high ASMM value. Recall that the effectiveness of the CLSS was predicated on: (1) local agent solutions frequently being consistent and (2) consistency being highly predictive of global correctness. It does turn out that that global consistency of local solutions is almost uniformly highly predictive of global correctness here. Figure 3 (A) shows that globally consistent local solutions are virtually always over 90% likely to be the same as the global MAPI (i.e., what a centralized system would produce). However, Figure 2 makes it clear that in a relatively nonmonotonic domain, the frequency with which local solutions will be consistent can be quite low. This means that the system will often have to apply a different procedure to determine its solution than simply merging the (consistent) local solutions. We were deliberately vague earlier about what should take place if the local solutions were not consistent. One possible approach is for the agents to effectively resort to centralized interpretation in these situations: once global *in*consistency is detected, one agent transmits all of its data to the other, and that agent interprets the complete data set. We will refer to this variation as CLSS-1. CLSS-1 would obviously produce solutions that are highly likely to be globally correct, since local solution consistency is very predictive of global correctness and lack of consistency causes the system to compute the global solution. Unless global consistency is likely, though, CLSS-1 would drastically reduce the runtime performance advantage that would come from having multiple agents that can work in parallel.[25]

[24] There are four possible interpretations with two top-level events. Thus p^2 is the probability that both agents get the (same) correct solution. The probability that both get incorrect solutions is $(1-p)^2$, but since there are three possible incorrect solutions, the probability of consistency is then $(1-p)^2/3$.

[25] The approximate level of global consistency that is needed to make a CLSS-1 system worthwhile can be obtained by using a simple model in which the expected time to solution is considered to be $pT(\frac{D}{n}) + (1-p)(T(\frac{D}{n}) + T(D) + T(C))$, where p is the probability

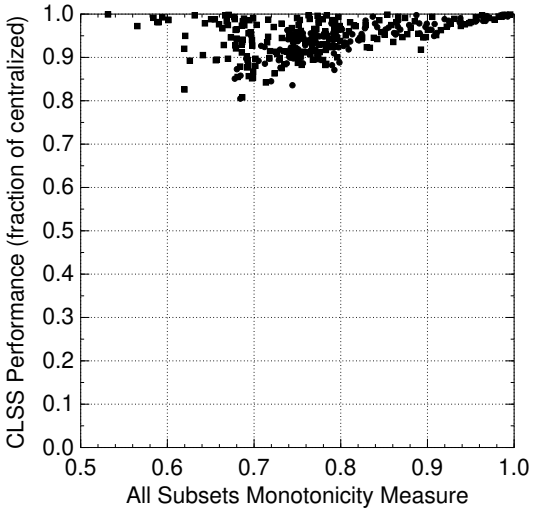


(A) Probability that the local agent solutions are equal to the centralized solution when consistent.

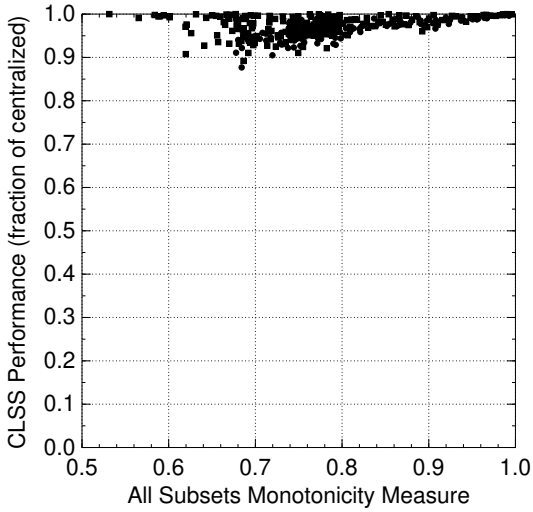


(B) Probability that the local agent solutions identify the actual causes when consistent.

Figure 3: Correctness of local solutions when consistent, two agents.
This graph shows the probability that two agents local MAPI solutions are "correct" when they are globally consistent (here identical), as a function of domain monotonicity (ASMM). Each agent gets 50% of the the global data with no data in common. Two aspects of correctness are shown: (A) whether the locals solutions are equal to the MAPI of all the data—the solution that would be produced by a centralized system, and (B) whether the local solutions are the true cause of the data.



(A) Each agent gets 50% of the the global data, no data in common.



(B) Each agent gets 60% of the the global data, 20% of the global data common to the agents.

Figure 4: Performance of CLSS-2, two agents.
This graph shows the solution quality performance of CLSS-2, as a function of domain monotonicity (ASMM). The result is given as a fraction of the performance of a centralized system (i.e., the performance is the ratio of the probability that a CDPS system using the CLSS-2 produces the correct answer to the probability that a centralized system using the MAPI will produce the correct answer). Results are shown for two cases with the agents having different fractions of the globally available data and different amounts of common data.

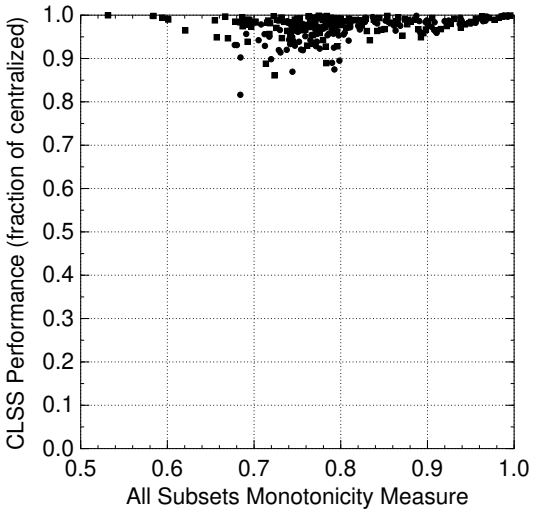
Resorting to centralized interpretation is one possible response when faced with inconsistency in the local agent solutions, but other approaches are possible. The next CLSS variation that we considered was when the local solutions were not globally consistent, the local solution (interpretation) with the *higher conditional probability* (given the local data) would be selected as the global solution. This variation will be referred to as CLSS-2. CLSS-2 is potentially very efficient, since it merely requires passing an additional real number with each local solution (it completely eliminates the need to communicate sensor data among the agents). The idea behind it is that some data is more critical to correctly identifying events—particularly in the more nonmonotonic domains—so one of the agents may have better data, and we might be able to identify which has the better data by considering the degree of belief in the local solutions. Figure 4 shows the results of experiments with CLSS-2. It turns out that the strategy performs quite well. Solution quality performance is typically around 90% or better of the performance of a centralized system (using the MAPI). Still, performance generally drops off as domains become less monotonic, and there are some domains that do not perform as well. For example, CLSS-2 can guarantee that performance would be at least 95% as good as a centralized system only in domains with an ASMM above about 0.9 (without data overlap).

A third possible CLSS variation is to look at the *absolute* probabilities of the local solutions rather than just their relative probabilities, and do additional communication and interpretation until a local solution with a belief (conditional probability) greater than a specified threshold is reached. This will be referred to as CLSS-3. The reason why this might be better is that it is possible in SI/diagnosis domains with 2 top-level events for the MAPI to have a probability just over 0.25.[26] Thus the most likely of the two local solutions could potentially have a probability of only, say, 0.26. Hardly a very well believed interpretation to automatically select as the global solution. Another advantage of this type of strategy is that because it is *incremental*, it can adapt to a poor data set or data distribution without requiring further communication and processing in situations where this is not necessary. Figures 5 and 6 show the results from experiments with CLSS-3. For these experiments, the acceptance threshold for solutions was set to 0.8 (a local solution has to have a conditional probability given the data it is based on of at least 0.8, to be accepted as the global solution). If neither initial local solution has a high enough probability, additional data will be communicated between the agents and used to develop new local solutions, until an acceptable level is reached in at least one agent. At this point the solution with the higher probability is returned. Performance is quite good (particularly with some common data). For example, CLSS-3 can guarantee that performance would be at least 94% as good as a centralized system in domains with an ASMM above 0.8 (without common data). In addition, Figure 6 shows that this level of performance does not require extremely frequent communication. When the ASMM is greater than 0.7, communication of data and further interpretation will typically be required less than 20% of the time.

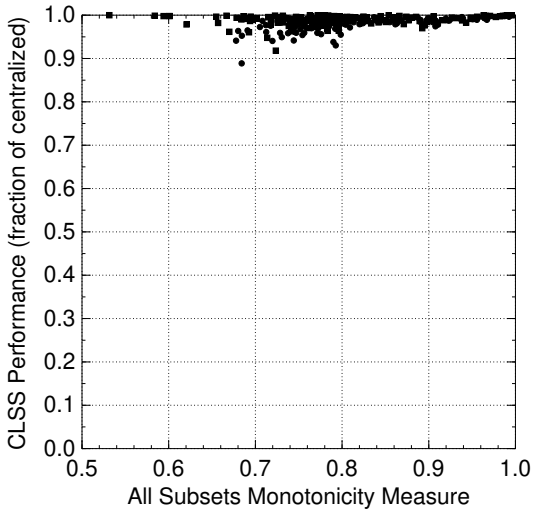
The results presented so far have shown that it is possible to use CDPS strategies for DSI/DD that rely largely on local solutions and achieve effective performance in two-agent systems. While consistency of local solutions can be quite low unless a domain is "highly monotonic," simple extensions to the basic strategy can produce solution quality

of global consistency, $T(\frac{D}{n})$ represents the time to compute the interpretation (e.g., MAPI) of one nth of the data set, $T(D)$ represents the time to compute the interpretation of the entire data set, and $T(C)$ is the time to communicate full data set to one agent. This can then be compared to $T(D)$, which would be the time for a centralized system to interpret the data set. For example, if we consider a two-agent system and ignore communication time, the CDPS to centralized ratio becomes $T(\frac{D}{2})/T(D) + (1-p)$. Now, if the runtime for the interpretation computation grows linearly with the amount of data then *speedup* from a two-agent system would be $1/(\frac{3}{2}-p)$. If $p = 0.7$ as would occur for an ASMM of around 0.8, the speedup would be only 1.25 (i.e., two agents would be only 25% faster than one). If the interpretation computation were quadratic with the amount of data then the speedup would be $1/(\frac{5}{4}-p)$. Now if $p = 0.7$, the speedup would be 1.8, and if $p = 0.8$, the speedup would be 2.2. While these figures are well below the maximum potential speedup of 4, they are very reasonable for doubling the number of agents. Of course, the addition of delays for communication could greatly reduce these speedups. Note also that the runtime for many BN algorithms depends on the topology of the network, but is independent of the amount of data that is actually known (i.e., independent of the number of evidence nodes whose values are set).

[26] Since there are four possible interpretations with two binary top-level events, an interpretation can potentially be the "most likely" and yet not be much over 0.25—if all the interpretations have approximately the same degree of belief.

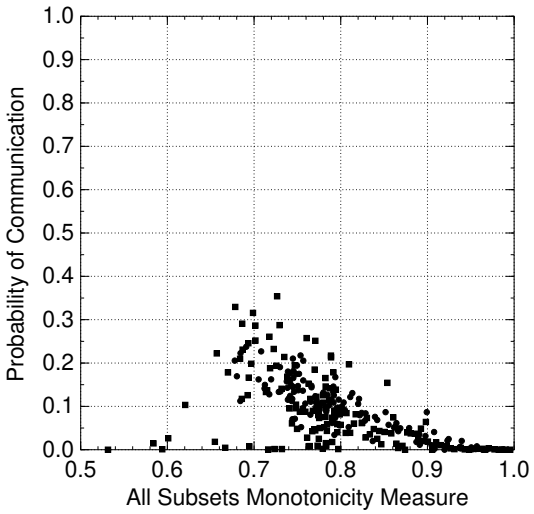


(A) Each agent gets 50% of the the global data, no data in common.

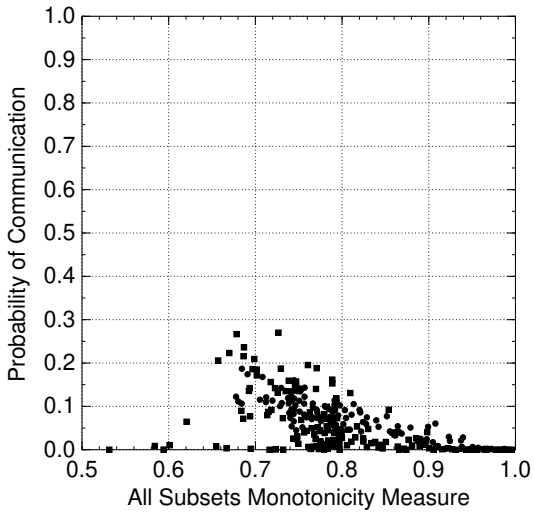


(B) Each agent gets 60% of the the global data, 20% of the global data common to the agents.

Figure 5: Performance of CLSS-3, 0.8 acceptance threshold, two agents.
This graph shows the solution quality performance of CLSS-3, as a function of domain monotonicity (ASMM). When the local solutions are not consistent, data will be communicated to get a solution probability over 0.8. The result is given as a fraction of the performance of a centralized system (i.e., the performance is the ratio of the probability of the CDPS system using the modified CLSS producing the correct answer to the probability that a centralized system using the MAPI will produce the correct answer). Results are shown for two cases with the agents having different fractions of the globally available data and different amounts of common data.



(A) Each agent gets 50% of the the global data, no data in common.



(B) Each agent gets 60% of the the global data, 20% of the global data common to the agents.

Figure 6: Probability of communication with CLSS-3, 0.8 acceptance threshold, two agents.
This graph shows the probability that information will need to be communicated with CLSS-3, as a function of domain monotonicity (ASMM). When the local solutions are not consistent, data will be communicated to get a solution probability over 0.8. Results are shown for two cases with the agents having different fractions of the globally available data and different amounts of common data.

(and runtime) performance that are very good (relative to a centralized system). Furthermore, the performance can be made fairly independent of the degree of monotonicity of the domain. A key question, though, is whether similar results can be obtained as the number of agents is scaled up?

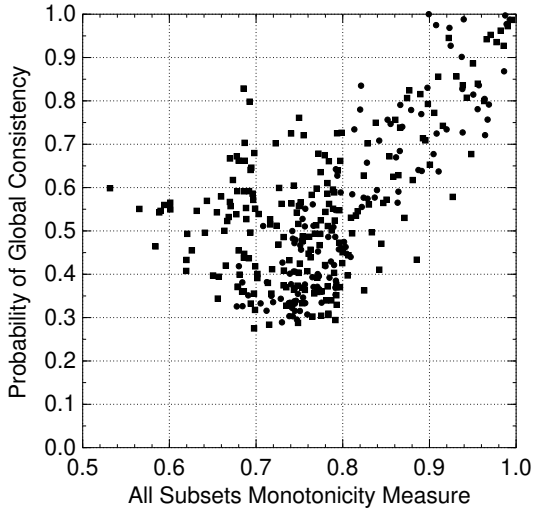
To answer this question, we repeated the previous experiments effectively assuming systems of five agents or ten agents. One of the main differences this makes is that each agent will receive a smaller fraction of the global data than in a two-agent system. Presumably, this will have a negative effect on local solution consistency and probably on the ultimate performance of the system. How significant will the effect be? Figure 7 shows the probability of consistency of *pairs* of agent local solutions, receiving data fractions that would be reasonable in five and ten-agent systems. As compared with Figure 2, the reduced data fractions result in the probability of consistency generally dropping off more rapidly as the ASMM of the domain decreases, and there is more variability (among domains with similar ASMMs).[27] Furthermore, Figure 8 shows that the probability of *all* five or ten local solutions being consistent drops off extremely rapidly. This clearly indicates that some method other than merging consistent local solutions will very frequently have to be used unless the domain is highly monotonic. Another question is whether local solution consistency is still predictive of correctness. Figure 9 shows that consistency of *pairs* of local solutions is no longer terribly predictive of global correctness. In Figure 10 we see that consistency of all five local agent solutions often is predictive (though there is much variation among domains with similar ASMMs), but once we get to ten agents, even having all ten local solutions consistent (identical) does not give a high probability that these solutions are correct (the local solutions are consistent, but wrong).

These results strongly suggest that CLSS-like strategies will not perform well in systems involving five and ten agents, unless the domain is very highly monotonic. To get a sense of CLSS performance with five and ten agents, we first tried CLSS-2, modified slightly to handle more than two agents: when the local agents solutions are all consistent (here identical), they are automatically taken as the global solution, and when they are not all consistent, the one with the highest conditional probability is chosen as the global solution. Figure 11 shows that this strategy did not perform well for most domains. For example, to guarantee performance within 90% of a centralized system, a domain would have to have an ASMM of approximately 0.96 or higher. From Figure 8, we know that global solutions are predominantly being chosen here based not on consistency, but rather on the relative conditional probabilities of the local solutions. Unlike the two-agent case, these results show that the relative conditional probability of the local solutions is a not reliable indicator of global correctness given the reduced data fractions. In other words, even the "best" of the interpretations based on only 20% (or 10%) of the data is not a good predictor of the best global interpretation unless the domain is highly monotonic. This is not too surprising, since we will see in Section 6.4 that it is very unlikely that the MAPI of a random 20% or 10% fraction of the data is the same as the global MAPI.

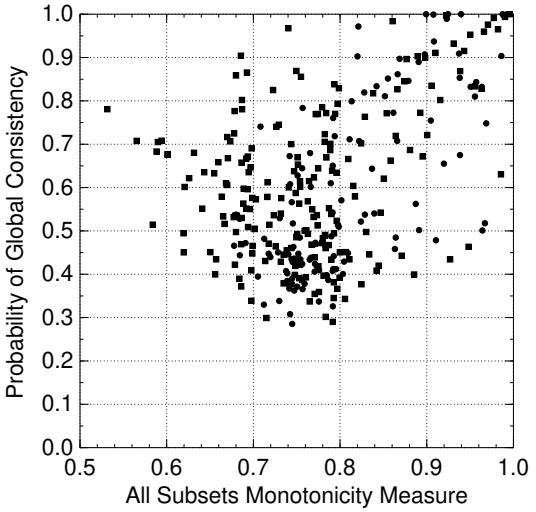
The results from the CLSS-2 experiments clearly indicate that in larger systems of agents, high quality global solutions cannot be developed simply by selecting from the agents' local solutions. Instead, agents will have to communicate data and do additional interpretation—perhaps constructing alternative solutions. This is what CLSS-3 does. In simulating systems of five and ten agents, we modified CLSS-3 to attempt to take advantage of the local solutions by using the conditional probabilities of the local solutions to direct the incremental transfer of data: data was incrementally transferred to the agent with the highest probability local solution, first from the agent with the next highest probability solution, then (if necessary) from the next highest probability solution, and so forth. Figure 12 seems to indicate that this strategy performed quite well in most domains when using five agents.[28]

[27] The apparent increase in consistency as the ASMM drops below 0.7 is due to the fact that nonmonotonicity often results from a small subset of the data being critical to identify each event, so as agents get less data there is more chance they will fail to identify the events and so be consistent—but wrong.

[28] Note though that performance was worse than when CLSS-3 was used with two agents (Figure 5). This may seem odd at first, since incremental interpretation is being done to obtain solutions above the same 0.8 probability threshold. The reason that performance is worse with five agents is that when all of the agents local solutions are consistent (identical), the consistent solution is used as the global solution, but consistency of the agents' local solutions is not terribly predictive of global correctness.

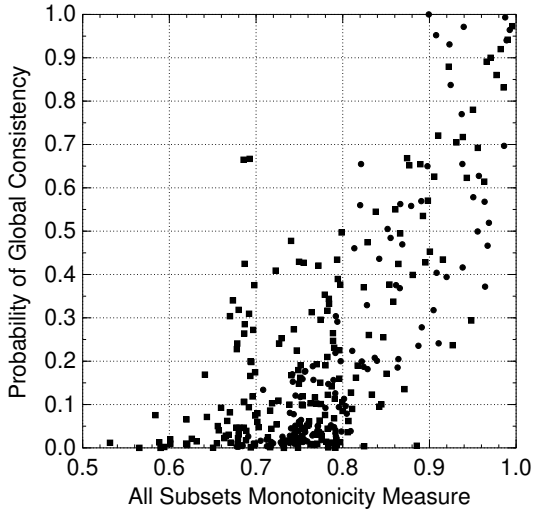


(A) Each agent gets 20% of the the global data, no data in common.

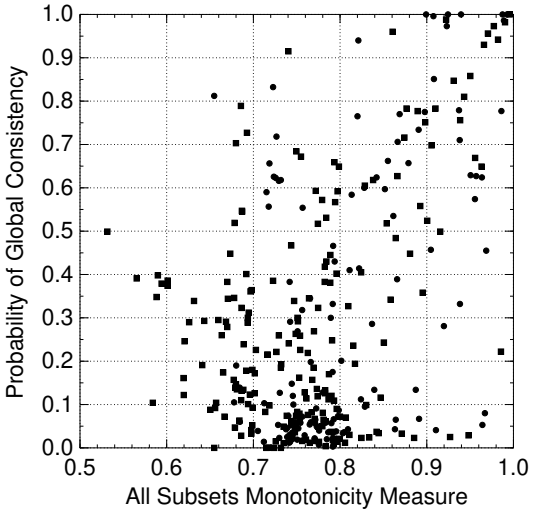


(B) Each agent gets 10% of the the global data, no data in common.

Figure 7: Global consistency of pairs of local solutions.
This graph shows the probability that two agents local MAPI solutions are globally consistent (here identical), as a function of domain monotonicity (ASMM). Results are shown for two cases with the agents having different fractions of the globally available data.

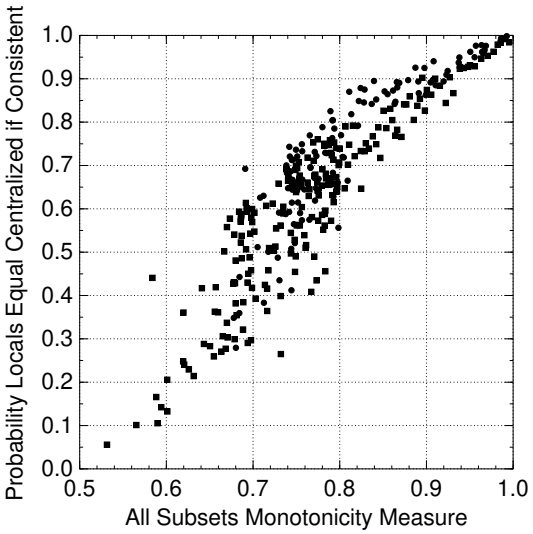


(A) Five agents, each gets 20% of the the global data, no data in common.

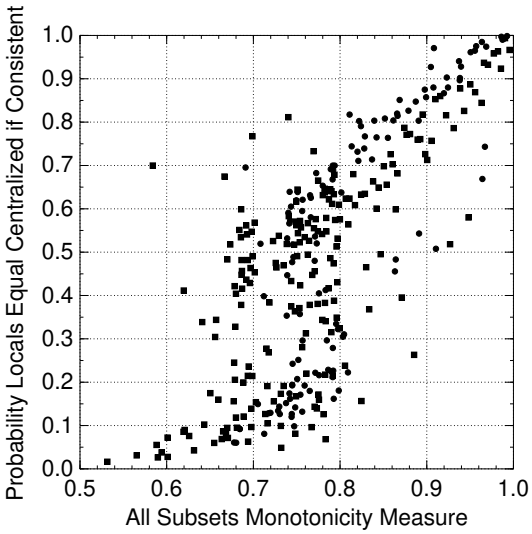


(B) Ten agents, each gets 10% of the the global data, no data in common.

Figure 8: Global consistency of all local solutions, five/ten agents.
This graph shows the probability that all five/ten of the agents' local solutions are globally consistent (here identical), as a function of domain monotonicity (ASMM).

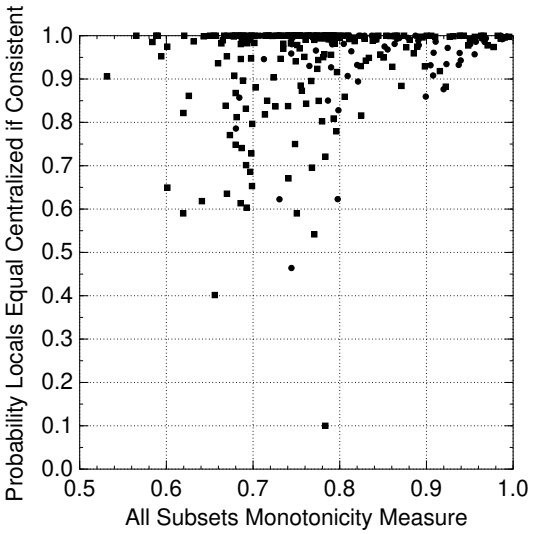


(A) Each agent gets 20% of the the global data, no data in common.

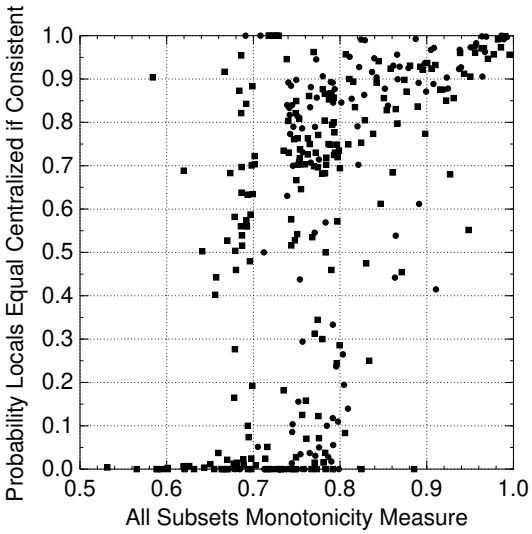


(A) Each agent gets 10% of the the global data, no data in common.

Figure 9: Probability that two local (agent) solutions are equal to the centralized solution when the two are consistent. This graph shows the probability that two agents' local solutions are the MAPI of the complete, global data set when they are consistent (here identical), as a function of domain monotonicity (ASMM).

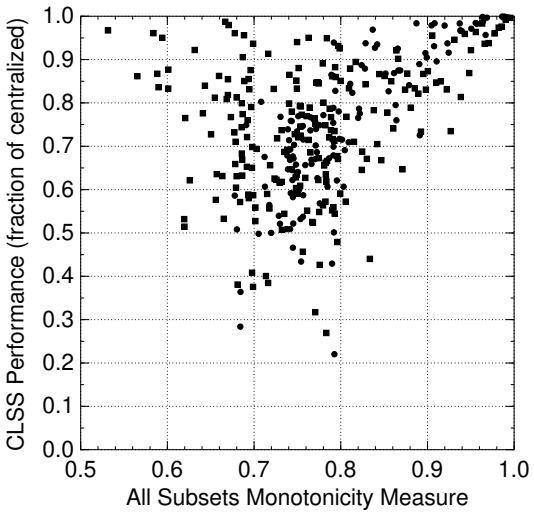


(A) Five agents: each gets 20% of the the global data, no data in common.

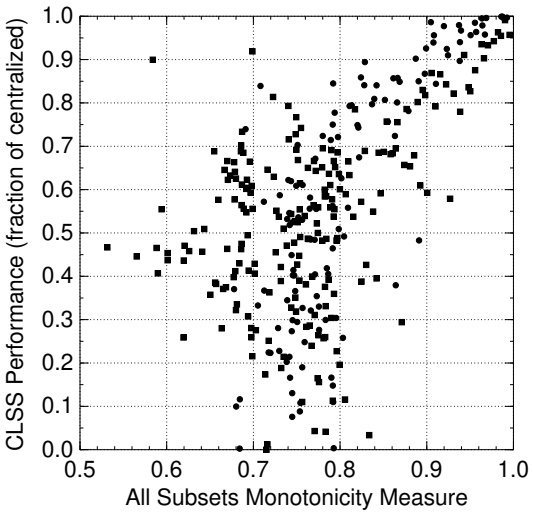


(B) Ten agents: each gets 10% of the the global data, no data in common.

Figure 10: Probability that the local (agent) solutions are equal to the centralized solution when all are consistent. This graph shows the probability that agents' local solutions are the MAPI of the complete, global data set when they are *all* globally consistent (here identical), as a function of domain monotonicity (ASMM).



(A) Five agents: each gets 20% of the the global data, no data in common.



(B) Ten agents: each gets 10% of the the global data, no data in common.

Figure 11: Performance of CLSS-2, five and ten agents.
This graph shows the solution quality performance of CLSS-2, as a function of domain monotonicity (ASMM). The result is given as a fraction of the performance of a centralized system (i.e., the performance is the ratio of the probability that a CDPS system using CLSS-2 produces the correct answer to the probability that a centralized system using the MAPI will produce the correct answer).

However, Figure 13 shows that this performance required that a great deal of data be communicated. For example, when domains had an ASMM of around 0.8, on average approximately 50% of the global data was communicated to the agent that produced the solution, so 70% of the global data was processed to generate the solution. The situation is even worse for systems of ten agents (there was obviously no reason to experiment with larger numbers of agents).

What can we conclude from the experiments presented above? It appears that they tell us that CLSS-like strategies are unlikely to be useful for CDPS-based DSI/DD, unless: (1) the domain is highly monotonic (as judged by a measure like the ASMM); or (2) a very small number of agents is being used. When based on only 10 or 20% of the global data, the agents' local solutions were usually not very useful for identifying global solutions—unless the domain was highly monotonic. The local solutions were only infrequently consistent, and even when they were consistent they were not extremely likely to be globally correct. While it is true that incremental strategies like CLSS-3 can produce high quality solutions in these domains, extensive communication (and redundant processing) is required—with the local solutions contributing little. So overall, performance appears likely to be unsatisfactory in systems of more than two or three agents.

When interpreting the results, however, we must keep the experimental setup in mind. One of the key aspects of the simulations was that data was randomly distributed among the agents, so agents potentially got data relevant to all top-level events, and agents were expected to produce complete interpretations. In other words, each agent had the exact same subproblem: identify the occurrence of all the possible events. This is an important element of the experimental setup to consider because it need not be the case in large-scale DSI/DD systems. While it is nearly always impractical to develop DSI/DD systems in which each agent's local subproblems are independent of those of all other agents, this does not mean that all the agents must have identical or even interacting subproblems. Typically, each agent's subproblem(s) will interact with some agents and will be independent of other agents' subproblems. For example, in a distributed vehicle monitoring system, the movement of a vehicle through regions monitored by different agents will cause subproblem interactions among these agents, but agents responsible for regions the vehicle does not enter may have completely independent subproblems.

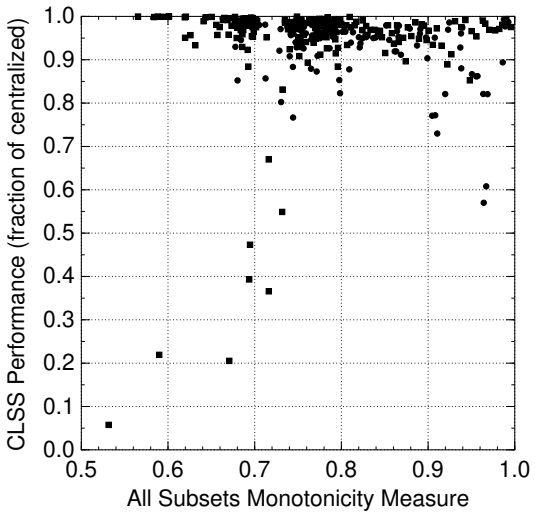
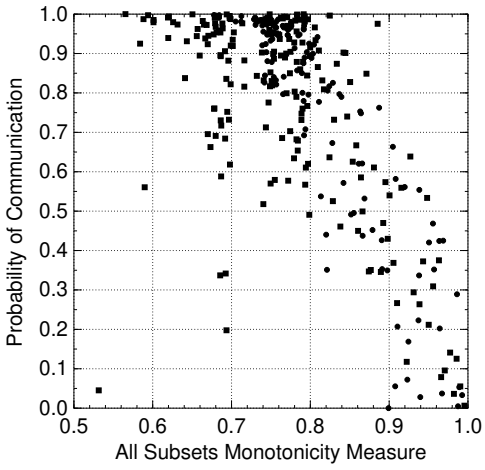
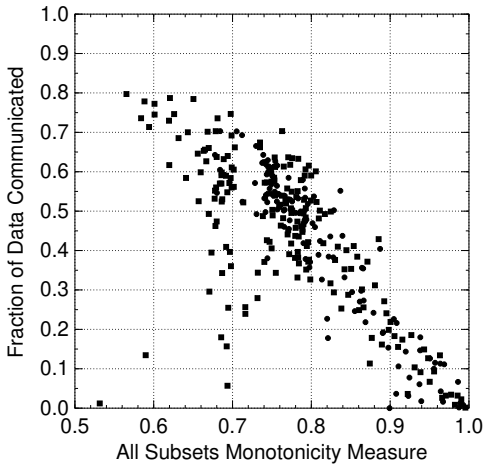


Figure 12: Performance of CLSS-3, 0.8 acceptance threshold, five agents.
This graph shows the solution quality performance of CLSS-3, as a function of domain monotonicity (ASMM). The result is given as a fraction of the performance of a centralized system.



(A) Probability that communication is required.



(B) Fraction of data that must be communicated.

Figure 13: Communication requirements of CLSS-3, 0.8 acceptance, five agents.
This graph shows the probability that communication of data among the agents will be required, when using CLSS-3 with a 0.8 acceptance threshold and five agents, as a functions of the ASMM. Each agent gets 20% of the the global data with no data in common.

This suggests a more general way to interpret the experiments. Instead of viewing the test systems of agents as representing complete DSI/DD systems, the test systems can be viewed as *independent subsystems* of complete systems. In other words, they can represent subsets of two/five/ten agents whose subproblems interact with each other, but are independent of the subproblems of the other agents in the system. Viewed in this way, the results are not as negative as it first appears. If a large-scale DSI/DD system can be designed in such a way that subproblem interactions are limited to groups of two to three agents, then a strategy like CLSS-3 should perform well even in domains that are relatively nonmonotonic. Of course, it is not always going to be possible to insure that small subsets of agents are independent of all other agents. Often what may happen is a situation like this: agent A_1 is responsible for events E_1 and E_2, agent A_2 is responsible E_2 and E_3, agent A_3 is responsible E_1 and E_4, and so forth. Here, there are not independent subsets of agents that are all responsible for the same set of events. It is the case, though, that only a small subset of the agents are responsible for each event. While a local solution selection strategy like the MAPI may not be effective in this situation, the results of experiments presented in the next section suggest that approximate strategies that focus on individual events can be effective under these conditions. The key is limiting the distribution of data relevant to each event to a small number of agents, so each agent has a substantial fraction of the data relevant to the events it is responsible for. An important aspect of our future research will be to develop measures of subproblem interaction that are meaningful for predicting performance in these kinds of situations.

Another conclusion that can be drawn from these experiments is that a DSI/DD domain's ASMM value correlates well with the performance of local solutions strategies. This means that the ASMM is an appropriate way to assess the "degree of monotonicity" of a domain when considering how well a CLSS-like strategy will perform. We also want to stress that an important capability of a framework like ours is that it provides *quantitative* assessments of various aspects of system performance as a function of domain characteristics like the ASMM. Armed with this kind of information and knowledge of the target domain's degree of monotonicity, the designer of a DSI/DD system can make quite specific predictions about how particular strategies would perform if they were implemented. By contrast, mere intuitions about general trends (e.g., local solutions strategies will perform better the more monotonic a domain is) are virtually useless when it comes time to make design decisions. How monotonic does a domain have to be to achieve a desired level of performance and how should one measure the degree of monotonicity? These are the questions we must have answers to in order to be effective in designing CDPS systems.

6.3 Near Monotonicity and the CLSS

In Section 4 we discussed the notion of DSI/DD domains being "nearly monotonic" and that we had proposed that this might explain why local solutions strategies can be effective even though DSI/DD domains are nonmonotonic. We also presented two of the monotonicity measures that we had suggested could be useful in determining whether a DSI/DD domain was nearly monotonic or not. The experiments described in Section 6.2 show that CLSS-like strategies can definitely be effective if a domain has a relatively high ASMM value. However, having a high ASMM value is not the same as being nearly monotonic according to our original conception. So we have not yet answered the questions that remained from the previous work: (1) do local solutions strategies perform well if a domain is nearly monotonic, and (2) are many DSI/DD domains indeed nearly monotonic? This section presents the results from experiments that were designed to help answer these questions.

The key difference between the originally proposed measures for near monotonicity and the ASMM is that the near monotonicity measures are event-based rather than interpretation-based (they consider events individually). To evaluate this type of measure, we used a minor variation of the first of the two near monotonicity measures defined in Section 4. We will refer to this as the *hypothesis threshold monotonicity measure* (HTMM), with HTMM(t) defined as: $P(E \in MAPI(\mathcal{D}) \mid P(E \mid D) \geq t)$, where t is a probability threshold. Being nearly monotonic would require that this value is sufficiently high (e.g., ≥ 0.9) for a reasonable t ($\ll 0.9$).

Figure 14 shows the results of assessing the HTMM against the ASMM, for several belief thresholds. The graphs show that the HTMM of a domain is only weakly correlated with the ASMM. Since the ASMM generally correlates well with CLSS performance, this means that the HTMM would not be a good predictor of how well CLSS-like strategies would perform. Somewhat surprisingly, the graphs also show that the vast majority of the domains could reasonably be classified as nearly monotonic. For example, in 92% of the domains, once an event hypothesis has a degree-of-belief of at least 0.7, there is a 90% or higher probability that the event is part of the global MAPI solution.

These findings cast doubt on our proposed explanation for why local solutions strategies have performed well in previous CDPS-based DSI/DD systems. Even though most of the domains used in the experiments would be considered nearly monotonic, local solutions strategies did not necessarily perform well. The problem with our characterizations of near monotonicity—relative to predicting how CDPS systems will perform—is that they ignore the fraction of the data that is available to the agents. This will often be critical in determining whether a domain being nearly monotonic is useful or not. If agents receive a relatively small fraction of the relevant data, it is likely that none of them will achieve a degree-of-belief in their event hypotheses as high as, say, 0.7. However, the ability to achieve this level of belief from mainly local data would be required if near monotonicity is to lead to local solutions strategies being effective.

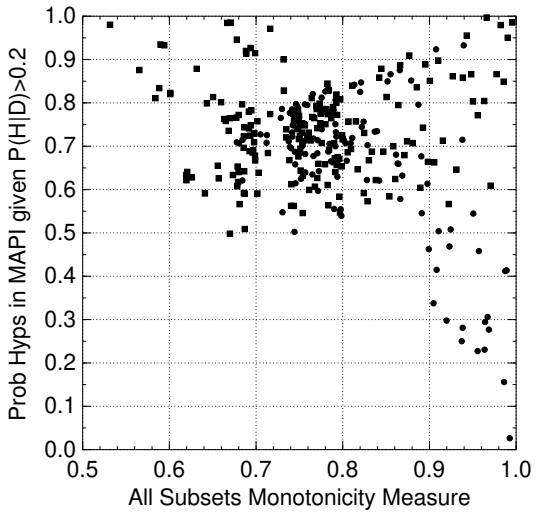
It does appear that we were correct about DSI/DD domains being nearly monotonic. The fact that the vast majority of the (randomly generated) experiment domains are nearly monotonic suggests that many DSI/DD domains will turn out to be nearly monotonic. This adds further support to the notion that local solutions strategies can be effective even in large-scale DSI/DD systems—if these systems are appropriately structured. Instead of agents that use the MAPI to determine their local solutions, consider agents that use approximate strategies like the PHI (Section 2.1), in which events are considered individually. Now suppose that the domain is nearly monotonic and each agent gets a significant fraction of the data that is relevant to the events it is responsible for.[29] This is likely to allow the agents to achieve a sufficient level of belief in the events so that their local solutions will have a high probability of being globally correct. It also again indicates that the key to using local solutions strategies in large-scale systems of agents is to limit the number of agents that are responsible for each event. In fact, we might surmise that this is what explains the successful use of local solutions strategies in previous DSI systems. Not only were the domains nearly monotonic, but because these systems were generally small scale, they were very likely to have been structured appropriately to take advantage of the near monotonicity.

6.4 The Partial Subsets Monotonicity Measure (PSMM)

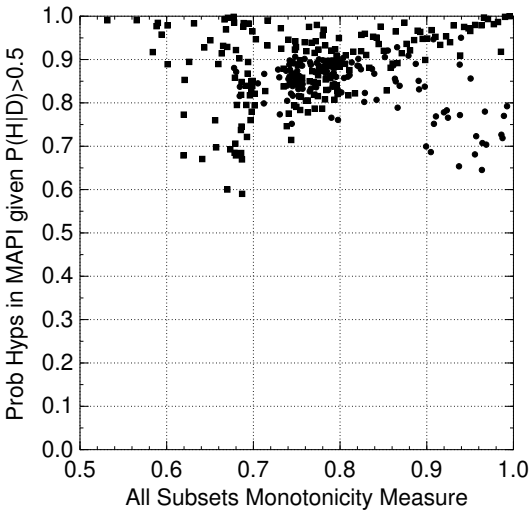
In this section we will consider the alternative partial subsets monotonicity measure (PSMM) introduced in Section 5.2. Again, the PSMM is the basically probability that the MAPI of some fixed fractional size data subsets will be the same as the MAPI of the complete data sets. The relationship between the ASMM and the PSMM using three different fractional subset sizes is shown in Figure 15 (these fractions were chosen because they are relevant to the experiments in the previous two sections). As can be seen, the ASMM is essentially identical to the 50% PSMM for most of the domains. As expected, having smaller fractions of the data leads to performance that drops off increasingly rapidly as the ASMM of a domain decreases. Clearly, the PSMM of a domain also correlates well with the performance of CDPS-based DSI/DD systems using CLSS variants and could be used a meaningful measure of domain monotonicity.

It is not surprising that the 50% PSMM is identical to the ASMM. In fact, this is good news for the simulation experiments because it is evidence that the number of data elements being used is sufficient for the data fractions being investigated. Since the ASMM is based on an average of all data subsets, the similarity of these measures suggests that the effect of decreasing or increasing the data fraction (relative to 50%-fraction data sets) has a comparable

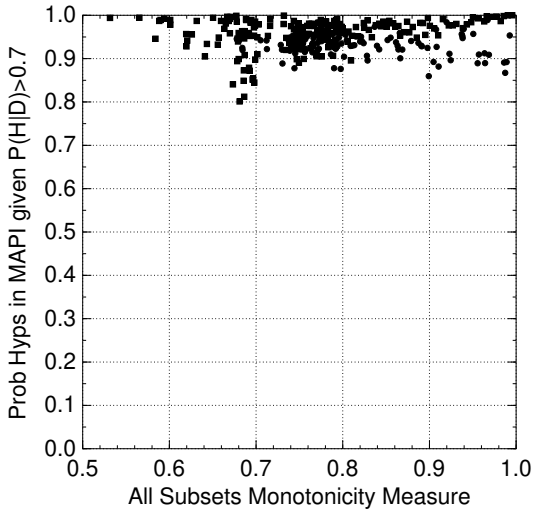
[29] Data is "relevant" to an event if it can affect the degree-of-belief (conditional probability) of the event.



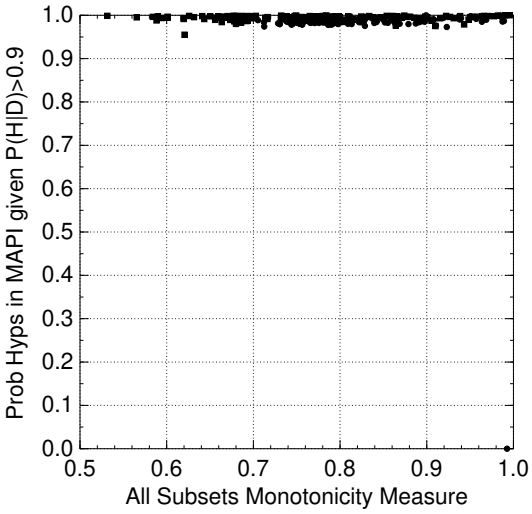
(A) Threshold of 0.2.



(B) Threshold of 0.5.

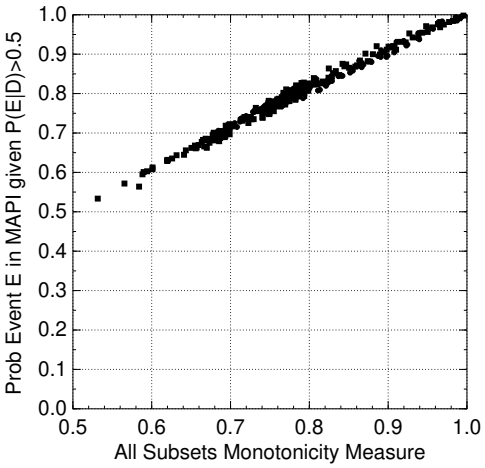


(C) Threshold of 0.7.

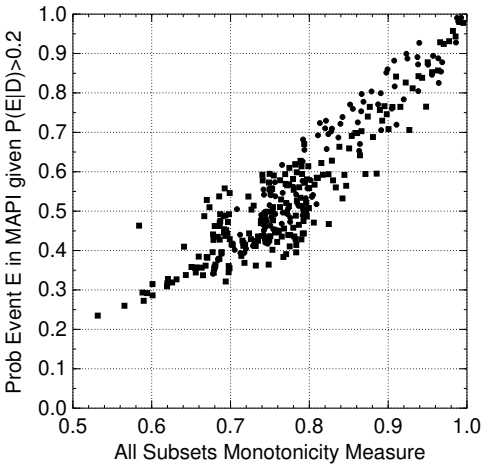


(D) Threshold of 0.9.

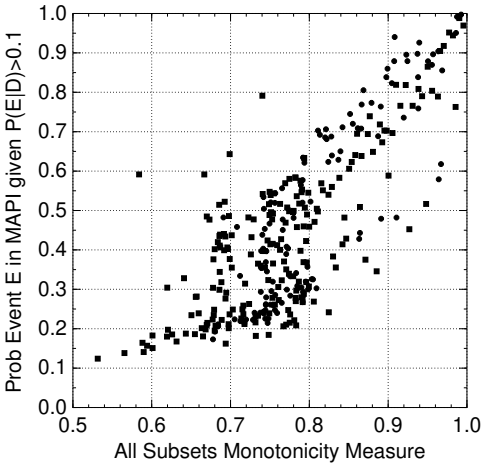
Figure 14: Hypothesis threshold monotonicity measure (HTMM) vs. ASMM, for different thresholds.
This graph shows the probability that event hypotheses are part of the global MAPI solution, once their conditional probability is above the specified threshold, as a function of each domain's ASMM value.



(A) 50% of the the global data.



(B) 20% of the the global data.



(C) 10% of the the global data.

Figure 15: Partial Subsets Monotonicity Measure (PSMM) vs. ASMM.
This graph shows the partial subsets monotonicity measure (PSMM) based on data subsets that are 50%, 20%, and 10% of the complete data sets, as a function of the ASMM monotonicity.

(though opposite) effect. If too few data elements were being used, then we would expect behavior to be skewed in some way. For example, small data fractions might produce much worse behavior than they should. This would cause the 50% PSMM and the ASMM to differ—but they do not.

6.5 The Effect of Common Data

Another issue that is of interest when considering the use of local solutions strategies for DSI/DD systems is the appropriate amount (if any) of common or overlapping data that there should be among the agents.[30] As the amount of overlap increases, the amount of redundant work being done by the agents increases, negatively affecting time-to-solution performance. On the other hand, when there are inter-agent subproblem interactions, common data would be expected to increase the likelihood that local solutions will be globally consistent. For strategies based on the consistency of local solutions, this will tend to reduce the need to do further communication and data processing, improving time-to-solution performance. Thus it could be advantageous to trade-off some redundant processing for overall performance improvements.

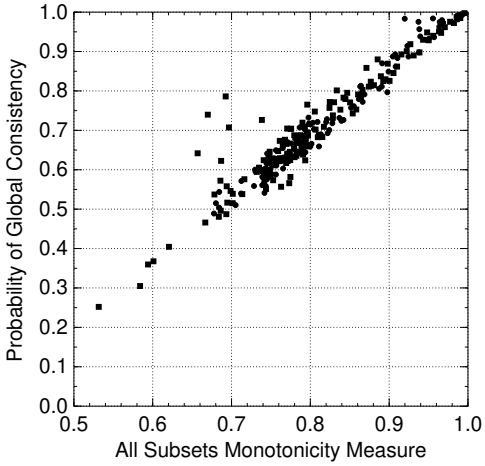
Figures 2, 4, and 5 in Section 6.2 do include cases with overlap, but the inclusion of common data also increases the fraction of the global data that is available to each agent. We see definite improvements in the performance measures in these figures, but the effect of common data is confounded with the effect of having access to a higher fraction of the global data. Figures 16 and 17 show the results of experiments where the fraction of the global data available to each agent is held constant at 50%, while the amount of overlap is increased. Comparing Figure 16 to Figure 2 (A), we see that as overlap increases, the probability of the local solutions being globally consistent does increase as expected, and that the effect is more pronounced as a domain becomes more nonmonotonic. What is most interesting is the effect that we see when comparing Figure 17 to Figure 4 (A). Overall solution quality performance (using CLSS-2) is generally not improved by increasing the overlap when the overall fraction of the global data being used is decreased. This once again demonstrates that the fraction of the global data that each agent receives is critical for the success of local solutions strategies.

7 Related Research

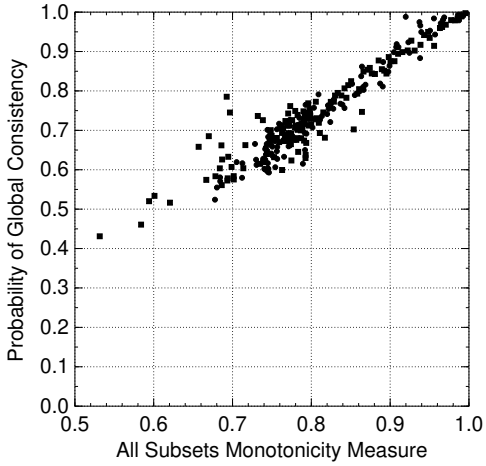
As we said in the introduction, little research has been devoted to developing a broad, general understanding of how the performance of *CDPS* systems relates to domain and strategy characteristics. There has, though, been some work that has evaluated how various strategies perform in particular applications (as we have done here for DSI/DD). A good example is the work by Sen and Durfee that considered the performance of several possible strategies for distributed meeting scheduling—e.g., [24, 25]. There have also been a number of proposed approaches for formalizing CDPS systems—e.g., [29, 30, 32]. Unfortunately, these frameworks do not model CDPS systems at a sufficient level of detail to be able to predict performance attributes like solution quality, or they apply only to restricted classes of problems. It is the case that a great deal of work has been done in conjunction with heterogeneous and self-interested agents, both to understand the performance characteristics of systems of these agents and to design coordination protocols that guarantee performance attributes (such as Pareto-optimality). However, this research has not generally addressed the issues that are of primary concern in CDPS systems.

To date, arguably the most substantial research involving quantitative analysis of CDPS systems over broad classes of problems has been the work carried out by Decker and Lesser using the TAEMS framework—e.g., [8, 9]. TAEMS is based on the use of explicit models of the structure of agents' tasks/subproblems, along with quantitative representations of the interrelationships among the tasks. Relationships among tasks worked on by separate agents

[30] Common data could result from sensor/test data being reported to multiple agents. Data from different sensors monitoring the same region has much the same effect.

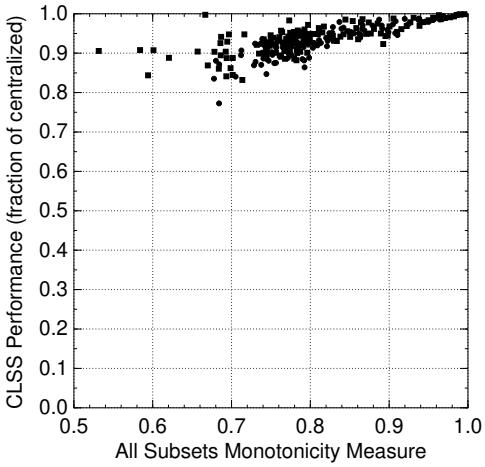


(A) Each agent gets 50% of the the global data, 10% of the global data common to the two agents.

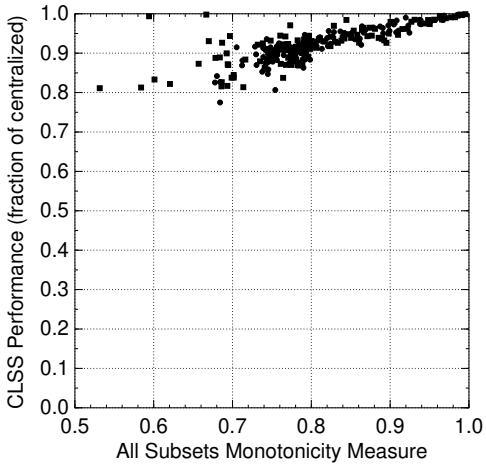


(B) Each agent gets 50% of the the global data, 20% of the global data common to the agents.

Figure 16: Global consistency of pairs of agents' local solutions, for different levels of data overlap.
This graph shows the probability that two agents local MAPI solutions are globally consistent (here identical), as a function of domain monotonicity (ASMM). Results are shown for two cases with the agents having a constant 50% fraction of the globally available data, but with different fractions of common data. For example, in (A) 40% of the global data has gone to one agent only, a disjoint 40% to the other agent only, and half of the remaining 20% to both agents. Thus, 20% of each agent's data is common to both agents. These graphs should be compared with Figure 2 (A), which has the same 50% fraction of the global data, but no common data.



(A) Each agent gets 50% of the the global data, 10% of the global data common to the agents.



(B) Each agent gets 50% of the the global data, 20% of the global data common to the agents.

Figure 17: Performance of CLSS-2, two agents, for different levels of data overlap.
This graph shows the solution quality performance of CLSS-2, as a function of domain monotonicity (ASMM). The result is given as a fraction of the performance of a centralized system (i.e., the performance is the ratio of the probability that a CDPS system using CLSS-2 produces the correct answer to the probability that a centralized system using the MAPI will produce the correct answer). Results are shown for two cases with the agents having a constant 50% fraction of the globally available data, but with different fractions of common data. For example, in (A) 40% of the global data has gone to one agent only, a disjoint 40% to the other agent only, and half of the remaining 20% to both agents. Thus, 20% of each agent's data is common to both agents. These graphs should be compared with Figure 4 (A), which has the same 50% fraction of the global data, but no common data.

are termed *coordination relationships*. The framework includes a simulation component that generates task structures with specified statistical characteristics, which it uses to simulate distributed problem solving and produce quantitative, statistical information about the performance of particular coordination strategies.[31] Work with TAEMS has produced some interesting insights into various aspects of coordination strategies. For example, it shows why no single coordination algorithm is optimal and how meta-level communications can be used for efficient dynamic re-organization. The main problem with the TAEMS work is that the simulations (and analytic work) have not been based on detailed, realistic models of domains and problem solvers. This significantly limits the usefulness of the results. For example, DSI was one of the CDPS applications investigated. However, the simulated task structures were based on a conceptual model of the SI process that ignored some of the more difficult aspects of SI: determining how many events are potentially represented in noisy data and selecting the data that should be processed to gather evidence for each event. Furthermore, because the tasks that agents carry out will depend on both the strategies being used and the characteristics of the domain, the use of task structures as the basis for the simulations means that it is impossible to separate out these two components. As a result, it is very difficult to relate the TAEMS results to particular SI strategies or domains, and difficult to model aspects of performance like solution quality. So while the TAEMS work examined how particular coordination strategies perform given various task characteristics, what actual domains and systems the task characteristics are representative of is, frankly, difficult to determine.

Diagnosis problems have been much studied with the belief network community and some distributed approaches have been proposed—e.g., [31]. Here an overall BN is decomposed appropriately (using the concept of a *multiply sectioned Bayesian network*) and the subnets are distributed among a set of machines. The key difference between the distributed BN work and our own work with BNs is that we are using BN algorithms simply to *simulate* SI/diagnosis problem solving, for abstract, simplified system models. The distributed BN research by contrast aims to develop BN inference algorithms that can be effective in real-world domains. While this may be a reasonable approach for some problems, it is a very different model from the CDPS model of loosely coupled, intelligent agents, and approximate, satisficing problem solving. So far, the distributed BN approaches that have been developed would not be practical for problems anywhere near the scale of the battlefield analysis system described by Brooks and Iyengar [2] (noted at the start of Section 2.2). In fact, because it is impossible or impractical to develop complete models of most DSI domains [6], current BN methods are in general not appropriate for real-world DSI.

There are other "distributed" approaches to SI and diagnosis that also do not fit the model we have presented. We have used the term "*cdps-based* distributed SI/diagnosis" specifically to distinguish the approaches to DSI/DD that we are concerned with from these alternatives. For example, in the field of "sensor fusion," DSI typically means that the data from multiple sensors is fused (combined) in a distributed manner, but results are left in the form of "distributions," and all of the interpretation is ultimately done by a *single, central unit* [2]. Furthermore, the distributed processing is usually fixed; it is not done by "intelligent agents" that can adjust their actions to respond to changing characteristics of the data. Similarly, in the field of distributed computing, distributed fault detection/diagnosis research often focuses on variations on Byzantine agreement algorithms. The main issues are the number of nodes that can fail and still have faults detected, and the number of test cycles required to guarantee detection. The diagnosis process itself (i.e., determining what is wrong) is not the critical element, and nodes are again not "intelligent agents." Even with MAS-based approaches, there are alternatives. For instance, DD may be done using a set of agents where each has completely different diagnosis knowledge or capabilities. While alternative types of "distributed" approaches may be appropriate for many applications, consideration of their merits relative to CDPS approaches is beyond the scope of this paper.

[31] Note that TAEMS work by Prasad et. al. [22] also used grammars to study CDPS. However, these grammars were not SCFGs, they were not used to model the actual (causal) structure of domains (merely to generate TAEMS tasks), and they had nothing to do with SI/diagnosis.

8 Conclusions and Future Directions

The current inability to predict how potential CDPS system designs will perform in an application domain is a serious hindrance to the widespread use of CDPS techniques. We believe that simulation-based approaches are currently the most practical way to address this deficiency. This led us to implement a simulation-based analysis system to study how domain and strategy characteristics interact to affect the performance of CDPS systems. We have focused on DSI/DD for a variety of reasons, including the range of important applications for these techniques, the ability to use standard probabilistic techniques in the simulations, and the amount of previous CDPS research concerned with these problems. The simulation system will be used both to derive performance information that can help in designing real-world CDPS-based DSI/DD systems and to explore the appropriateness of simulation-based approaches for studying CDPS performance.

The experiments reported on in this paper were intended to demonstrate how quantitative models that relate the performance of CDPS systems to particular domain characteristics can be obtained from a simulation-based framework like ours, and how this information can be useful in answering the kinds of questions that designers of CDPS systems are interested in. The experiments concentrated on the relationship between domain monotonicity and DSI/DD strategies that rely largely on the consistency of local solutions. This was an issue that we had raised in an earlier paper, but for which we had little to go on other than intuition. The results of the experiments confirmed some of our intuitions, but also produced unexpected findings that are directly relevant to the design of CDPS-based DSI/DD systems. Furthermore, even if the experiments had shown that all of our intuitions were correct, there is a huge difference between intuition and experimental confirmation. Not only may intuitions be incorrect, but they do not provide the quantitative information about the power of an effect that designers need to predict how a system will perform or to make detailed design decisions.

The experiments confirmed that DSI/DD strategies based on the consistency of local solutions can be effective in producing high quality solutions—but only under certain conditions. As expected, domain monotonicity can be a key factor to consider, with these strategies tending to perform better the "more monotonic" the domain is. Prior to this work, however, it was not clear just how monotonic a domain would need to be for local solutions strategies to achieve an acceptable level of performance, whether these kinds of strategies could perform well if a domain was quite nonmonotonic, or even what an appropriate measure of domain monotonicity was. The experiments went a long way toward answering these questions. They showed that unless CDPS systems are limited to two or three agents, domains must be very highly monotonic for local solutions to be useful in identifying global solutions, when every agent is responsible for all events. One of the surprising findings from the experiments, though, was that a group of two or three agents can perform well even in relatively nonmonotonic domains, by using simple and efficient extensions of the basic strategy. So it is not simply the case that local solutions strategies perform well only in domains that are just about completely monotonic. Furthermore, while the results at first appear to rule out the use of CLSS-like strategies in large-scale systems of agents, we observed that by re-interpreting our agent systems as *subsystems* of an overall system, the experiments suggest methods for designing effective large-scale DSI/DD systems. The critical requirement is that the number of agents among which the data relevant to each event is distributed must be small (so agents receive a significant fraction of the data needed to identify the events they are responsible for). With an interpretation strategy like the MAPI that considers complete interpretations, this is best accomplished by creating a large system from small independent subsystems of agents (where the agents are all responsible for the same subset of events). With approximate strategies that consider events individually, as long as the domain is nearly monotonic, it must simply be the case that a small number of agents are responsible for each event.

Another important contribution of these experiments is confirmation that the ASMM and PSMM values for a DSI/DD domain correlate well with the performance of CLSS-like strategies. While the exact ASMM and PSMM

measures will be impractical to apply to real-world domains, they can be approximated quite easily.[32] An unexpected finding was that the hypothesis-based monotonicity measures that we had originally proposed for judging whether a domain was nearly monotonic, are not well-correlated with the performance of CLSS-like strategies. This was the case despite the fact that it appears likely that many DSI/DD domains would be considered nearly monotonic according to these definitions. While our original definitions would be appropriate for evaluating centralized SI/diagnosis systems, the experiments make it clear that the failure to consider the fraction of data available to each agent is what makes these definitions unsuitable for CDPS systems.

While we believe that simulation approaches are one of the best ways to develop a better understanding of CDPS performance, simulation approaches are not without their risks. The use of abstract strategies, agents, and domain models is critical to being able to run the large numbers of experiments needed to map out the effects that interactions among these aspects of a system have on performance. At the same time, the results must be meaningful for real-world CDPS systems and domains, or they will not have much value. In particular, it must be possible to map between real domains and strategies, and the abstract domains and strategies. These are key concerns in our work. One way that we have addressed these issues is by maintaining separate models of domains and CDPS strategies, so that the characteristics of each can be clearly determined. Another part of our approach is the use of BNs and probabilistic models like the MAPI to simulate SI/diagnosis problem solving. Real-world systems may use different algorithms when computing the MAPI or they may only approximate it, but our results should be useful because they will be expressed in terms of formal standards.

An important aspect of our future research will be to verify that the results from the simulation are applicable to real-world domains and systems. For example, to make sure that we are not exploring abstract domains that have no real-world analogues, we will be using the domain measures that we develop to assess real-world domains. As a first step, we have applied the ASMM to several large BN models that are available in the *Bayesian Networks Repository*.[33] We have a found a range of monotonicity levels in the models we have looked at, from 0.95+ down to around 0.55. This helps to confirm the usefulness of experimenting with domain models over a wide range of monotonicity levels. There are some domains that are highly monotonic and some that very nonmonotonic, so no single CDPS strategy will always be best.

The analysis system is still at an early stage of development, with many extensions under development and planned. One of the extensions that is currently being pursued is the implementation of additional probabilistic inference algorithms. The main objective is to support simulations with increasingly complex models of DSI/DD domains (domains with larger event and data sets, and multiple levels of events). We are particularly interested in bigger domain grammars to support experiments with systems of very large numbers of agents, since relatively little previous research has been done with large-scale CDPS systems. The SMILE package that we are currently using for BN inference implements only sampling-based approximation algorithms and does not include native MAPI (belief revision) algorithms. Research has shown that the effectiveness of the various approximate inference algorithms for BNs is strongly dependent on the characteristics of the encoded distribution [17, 18]. Thus, it is important to be able evaluate a number of algorithms to see which are most appropriate for DSI/DD models and to see whether the algorithms can be tuned for particular probabilistic computations. It may even be the case that different types of DSI/DD domain models (e.g., independent or correlated) may best be accommodated by different algorithms. Several classes of inference algorithms will be considered, including sampling [18], search-based [7, 18], and factoring approaches [17].

[32] In larger domains there are often a very large number of data cases, most with very small probabilities. Instead of considering all data cases, those with the highest probability should be processed first. The assessment can stop when cases that constitute a suitably high probabilistic fraction have been examined. The other obvious way to speed up the computations is to randomly sample from the subsets for each data case instead of exhaustively considering them.

[33] www-nt.cs.berkeley.edu/home/nir/public_html/Repository/.

Another extension that is under active development will allow the use of *attribute SCFGs* to represent domains (rather than just SCFGs). The addition of attributes is necessary to be able to model complex, real-world SI domains, where the sensor data and the events often involve multiple continuous/semi-continuous-valued attributes (like position). Though it is possible to represent such situations with basic SCFGs, the resulting models quickly become impractical. As we noted earlier, the IDP work [28] has demonstrated how attribute SCFGs can be used to model complex SI domains. The key issue in moving to attribute SCFGs is updating the tools. In some cases this will be fairly straightforward, as with the tools that generate simulated data sets. A difficult tool to convert will be the one that converts SCFG representations to BN representations. The effects of the attribute functions must be represented in a BN via the conditional probability tables. When one combines this with the need to deal with effects that are not conditionally independent, direct computation of BN probabilities appears to be very difficult (unless attribute functions are highly restricted in form). One approach that is under consideration is to use the attribute SCFGs to generate simulated data sets, and then use a method for *learning* BNs from data (e.g., [23]) to construct a corresponding BN.

Ultimately, we hope that this research will lead to the development of a set of generic strategies for CDPS-based DSI/DD (and then CDPS in general) along with tools that can evaluate domain characteristics to predict how effective each strategy will be. This would provide system designers with a very useful starting point when building a CDPS system: an abstract strategy whose likely level of performance is known. As we have done with local solutions strategies and domain monotonicity, previous CDPS research will be considered to suggest abstract strategies for evaluation. Obvious examples that we intend to consider are strategies that involve meta-level planning or the communication of other meta-level information. We also expect that our own research will suggest additional strategies and domain characteristics. For example, the research reported on here has made it clear that the nature of subproblem independence is an important domain/system characteristic in determining whether local solutions strategies can be effective in large-scale DSI/DD systems. It remains to be determined, however, how one should assess subproblem independence in order to predict CLSS performance. Furthermore, the use of approximate strategies will raise issues (such as what "consistency" of local solutions should mean) that have not been satisfactorily investigated in previous research, but which can be investigated via simulations.

9 Acknowledgments

This work was supported in part by a Special Projects grant from Southern Illinois University at Carbondale.

References

[1] Yaakov Bar-Shalom and Thomas Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[2] Richard Brooks and S. S. Iyengar, *Multi-Sensor Fusion*, Prenctice Hall, 1998.

[3] Norman Carver, Zarko Cvetanovic, and Victor Lesser, "Sophisticated Cooperation in FA/C Distributed Problem Solving Systems," *Proceedings of AAAI-91*, 191–198, 1991.

[4] Norman Carver, Victor Lesser, "The DRESUN Testbed for Research in FA/C Distributed Situation Assessment: Extensions to the Model of External Evidence," *Proceedings of the International Conference on Multiagent Systems* (ICMAS-95), 33–40, 1995.

[5] Norman Carver, Victor Lesser, and Robert Whitehair, "Nearly Monotonic Problems: A Key to Effective FA/C Distributed Sensor Interpretation?," *Proceedings of AAAI-96*, 88–95, 1996.

[6] Norman Carver, Frank Klassner, and Victor Lesser, *Sensor Interpretation in Complex Domains using RESUN: Experiences with the IPUS Sound Understanding Testbed*, Technical Report 99-1, Computer Science Department,

Southern Illinois University, 1999.
(Can be obtained from: "http://cs.siu.edu/~carver/")

[7] Eugene Charniak and Solomon Shimony, "Cost-Based Abduction and MAP Explanation," *Artificial Intelligence*, 66, 345–374, 1994.

[8] Keith Decker and Victor Lesser, "Quantitative Modeling of Complex Environments," *International Journal of Intelligent Systems in Accounting, Finance, and Management*, 215–234, 1993.

[9] Keith Decker and Victor Lesser, "Designing a Family of Coordination Algorithms," *Proceedings of ICMAS-95*, 73–80, 1995.

[10] Edmund Durfee and Victor Lesser, "Incremental Planning to Control a Time-Constrained, Blackboard-Based Problem Solver," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 5, 647–662, 1988.

[11] k. S. Fu, *Syntactic Methods in Pattern Recognition*, Academic Press, 1974.

[12] Donald Knuth, "Semantics of Context-Free Languages," *Mathematical Systems Theory*, vol. 2, no. 2, 127–146, 1968.

[13] Vipin Kumar and Laveen Kanal, "The CDP: A Unifying Formulation for Heuristic Search, Dynamic Programming, and Branch-and-Bound," in *Search in Artificial Intelligence*, L. Kanal and V. Kumar, editors, Springer-Verlag, 1988.

[14] Victor Lesser and Lee Erman, "Distributed Interpretation: A Model and Experiment," *IEEE Transactions on Computers*, vol. 29, no. 12, 1144–1163, 1980.

[15] Victor Lesser and Daniel Corkill, "Functionally Accurate, Cooperative Distributed Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 1, 81–96, 1981.

[16] Victor Lesser, "A Retrospective View of FA/C Distributed Problem Solving," *IEEE Transactions on Systems, Man, and Cybernetics*, special issue on Distributed Artificial Intelligence, vol. 21, no. 6, 1347–1362, 1991.

[17] Zhaoyu Li and Bruce D'Ambrosio, "Efficient Inference in Bayes Nets as a Combinatorial Optimization Problem," *International Journal of Approximate Reasoning*, vol. 11, no. 1, 55–81, 1994.

[18] Yan Lin and Marek Druzdzel, "Stochastic Sampling and Search in Belief Updating Algorithms for Very Large Bayesian Networks," *Working Notes of the AAAI-1999 Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information*, 77–82, 1999.

[19] Mortaon Nadler and Eric Smith, *Pattern Recognition Engineering*, John Wiley, 1993.

[20] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[21] Yun Peng and James Reggia, *Abductive Inference Models for Diagnostic Problem-Solving*, Springer-Verlag, 1990.

[22] M. V. Nagendra Prasad, Keith Decker, Alan Garvey, and Victor Lesser, "Exploring Organizational Design with TAEMS: A Case Study of Distributed Data Processing," *Proceedings of ICMAS-96*, 283–290, 1996.

[23] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.

[24] Sandip Sen, *Predicting Tradeoffs in Contract-Based Distributed Scheduling*, PhD Dissertation, Department of Electrical Engineering and Computer Science, University of Michigan, 1993.

[25] Sandip Sen and Edmund Durfee, "Unsupervised Surrogate Agents and Search Bias Change in Flexible Distributed Scheduling," *Proceedings of the International Conference on Multiagent Systems* (ICMAS-95), 336–343, 1995.

[26] Solomon Shimony, "Finding MAPs for Belief Networks is NP-Hard," *Artificial Intelligence*, vol. 68, 399–410, 1994.

[27] Robert Whitehair and Victor Lesser, *A Framework for the Analysis of Sophisticated Control in Interpretation Systems*, Technical Report, 93-53, Computer Science Department, University of Massachusetts, 1993. (Can be obtained from: "http://dis.cs.umass.edu/")

[28] Robert Whitehair, *A Framework for the Analysis of Sophisticated Control*, PhD Dissertation, Computer Science Department, University of Massachusetts, 1996.

[29] Michael Wooldridge and Nicholas Jennings "Formalizing the Cooperative Problem Solving Process," *Proceedings of the Thirteenth International Workshop on DAI* (DAI-94), 1994.

[30] Michael Wooldridge, "A Knowledge-Theoretic Approach to Distributed Problem Solving," *Proceedings of the 13th European Conference on Artificial Intelligence* (ECAI98), 308–312, 1998.

[31] Yang Xiang, "A Probabilistic Framework for Multi-agent Distributed Interpretation and Optimization of Communication," *Artificial Intelligence*, vol. 87, no. 1–2, 295–342, 1996.

[32] Makoto Yokoo and Edmund Durfee, "Distributed Search Formalisms for Distributed Problem Solving: Overview," *Proceedings of the Twelfth International Workshop on DAI* (DAI-92), 371–390, 1992