Grid-based SensorDCSP*

R.Bejar[1], C.Domshlak[2], C.Fernandez[1], C. Gomes[2], B. Selman[2] and M. Vails[3]

[1] Dept. d'Informatica
Universitat de Lleida
Jaume II, 69. Lleida, Spain

[2]Dept. of Computer Science
Cornell University
Ithaca, NY 14853, USA

[3]Dept. de Matematica
Universitat de Lleida
Jaume II, 69. Lleida, Spain

Abstract

We introduce Grid-based SensorDCSP, a geometrically structured benchmark problem for the study of distributed CSP algorithms. This domain provides realistic structure of the communication and tracking constraints. We formally define this problem, and perform its worst-case complexity analysis. Likewise, we provide an average case empirical analysis of the AWC algorithm, studying its behavior on tractable and intractable sub-classes of our problem.

1 Introduction

In recent years we have seen an increasing interest in Distributed Constraint Satisfaction Problem (DisCSP) formulations to model combinatorial problems arising in multi-agent environments [Yokoo, 2001 J. Recently, an interesting benchmark for DisCSP algorithms, called SensorDCSP, has been introduced in [Fernandez *et al,* 20021. Inspired by distributed applications arising in networked systems, SensorDCSP involves a network of distributed sensors simultaneously tracking multiple mobile objects. An instance of SensorDCSP is defined by a set of sensors, a set of mobiles which are to be tracked by the sensors, and two sets of visibility and compatibility constraints specifying which sensors can communicate one with another and which sensors can track which mobiles, respectively. The goal is to allocate three sensors to track each mobile, such that all these triplets of sensors are pairwise disjoint, and the sensors in each such triplet are mutually compatible and can track the mobile they are assigned to. The analysis in [Fernandez *et al*., 2002] addresses the complexity of SensorDCSP, as well as its empirical evaluation, considering the performance and scalability of alternative DisCSP algorithms on various scenarios in a truly networked environment.

In this work we extend SensorDCSP by providing it with an underlying spatial structure. In SensorDCSP, the constraints of communication between sensors and visibility of mobiles are independent of their physical location. In contrast to SensorDCSP, the spatial limitations for both communication between sensors and visibility of the mobiles are captured by a geometrical structure in our new benchmark, providing a more realistic framework. We refer to our geometrically enriched benchmark as Grid-based SensorDCSP or just *GSensorDCSP.* We provide both an analytical complexity analysis of GSensorDCSP and an empirical study of a characteristic DisCSP algorithm on various instances of this problem.

2 GSensorDCSP: Definition and Complexity

GSensorDCSP is a specific variant of the general SensorDCSP, enriched by a spatial structure: we have multiple sensors $S = \{s_1, \cdots, s_m\}$, multiple objects (mobiles) $T = \{t_1, \cdots, t_n\}$ which are to be tracked by the sensors subject to visibility and compatibility constraints, and the goal is to decide whether there exists an allocation of three sensors to each object, while keeping these triplets of sensors pair-wise disjoint. However, the visibility and compatibility constraints in GSensorDCSP relate to the physical limitations of the sensors and the terrain on which the sensors are located.

In GSensorDCSP, the sensors are located on the nodes of a

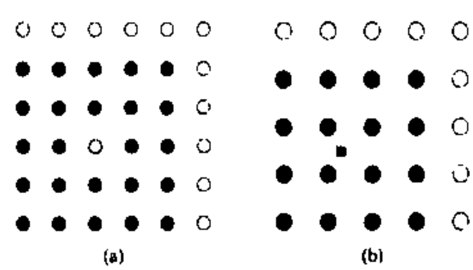

Figure 1: Grid of 6 x 6 sensors; black sensors stand for the (a) 2-compatibility window of the gray sensor, and (b) 2-visibility window of the square mobile.

* Research supported by AFOSR, grant F49620-01-1-0076 (Intelligent Information Systems Institute) and F49620-01-1-0361 (MURI grant on Cooperative Control of Distributed Autonomous Vehicles in Adversarial Environments), CICYT, TIC2001-1577-C03-03 and DARPA, F30602-00-2-0530 (Controlling Computational Cost: Structure, Phase Transitions and Randomization) and F30602-00-2-0558 (Configuring Wireless Transmission and Decentralized Data Processing for Generic Sensor Networks). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR, DARPA, or the U.S. Government.

uniform grid, and the mobiles are located within the surface enclosed by this grid; this way the grid specifies the generally trackable region. The physical limitations of the sensors are modeled by the notions of k_c*-compatibility* and k_v*-visibility.* The k_c-compatibility window for a sensor S_i, denoted $\mathbb{C}^{k_c}(s_i)$, corresponds to the set of all sensors that are at most k_c general (rectilinear and/or diagonal) hops from s_i. For example, the black sensors in Figure 1 (a) correspond to 2-compatibility window for the gray sensor. Similarly, the k_v-visibility window for a mobile t_j, denoted $\mathbb{V}^{k_v}(t_j)$, corresponds to the set of all sensors that are at most k general hops around t_j. For example, the black sensors in Figure 1(b) correspond to 2-visibility window for the square mobile. Finally, if each sensor s_i of a given GSensorDCSP problem instance can communicate only with some sensors within $\mathbb{C}^{k_c}(s_i)$, then the set of compatibility constraints of this instance is called k_c*-restricted.* The notion of k_v-restricted visibility is defined similarly.

Studying the worst-case complexity of GSensorDCSP we *formally* show that:

1. Any GSensorDCSP instance with 1-restricted visibility is solvable in low polynomial time, and
2. GSensorDCSP with k_v-restricted visibility is NP-complete for all $k_v \geq 2$.

The former is shown by a reduction to the problem of feasible integer flow in bipartite graphs, while the latter is shown by a non-trivial reduction from 3-SAT and is valid even for the problems with 1-restricted compatibility.

3 Connecting Locality and Constrainedness

While the physical limitations of the sensors in GSensorDCSP are modeled via the locality windows, the terrain limitations are modeled via incomplete compatibility and visibility *within* the windows. Problem instances of any GSensorDCSP sub-class (k_c, k_v) can be ordered according to the *local constrainedness,* i.e., the average number of sensors that a sensor can communicate with and the average number of sensors that can track a mobile. In our experiments, for each pair of locality parameters (k_c, k_v), we define a random distribution of GSensorDCSP instances via two parameters $P_v, P_c \in (0, 1]$ that control the density of visibility and compatibility: The probability that two sensors s_i and s_j will be compatible is given by P_c if $(s_i, s_j) \in \mathbb{C}^{k_c}(s_i)$, otherwise, it is equal to 0. Clearly, higher value for P_c and P_v correspond to better conditions for communication and tracking, respectively.

For our first experiments with the AWC [Yokoo, 1994] algorithm, we consider different sets of instances with 25 sensors (grid 5 x 5), and 5 mobiles, with every set generated with different values for the parameters P_c and P_v. The parameters P_c and P_v are ranging from 0.1 to 1 with an increment of 0.1, giving a total number of 100 data sets, where every set contains 50 instances. Given our formal complexity results, we consider several hard subclasses of GSensorDCSP corresponding to k_v = 2. For instance, Figure 3(a) shows the ratio of the satisfiable instances *(P_{sat})* as a function of P_c and P_v for k_v = 2 with k_c = 1 and k_c = 2. As in the case of general SensorDCSP [Fernandez *et al,* 2002], when both probabilities are low, the instances generated are mostly unsatisfiable, while for high probabilities most of the instances are satisfiable. Both for k_c = 1 and k_c — 2, the transition between the satisfiable and unsatisfiable regions occurs within a narrow range of the density parameters. Observe that, for k_c = 1 this range corresponds to significantly higher values of P_c and P_v, comparatively to these for k_c = 2 and k_c = 2. However, the form of the transition for various values of k_c is very similar.

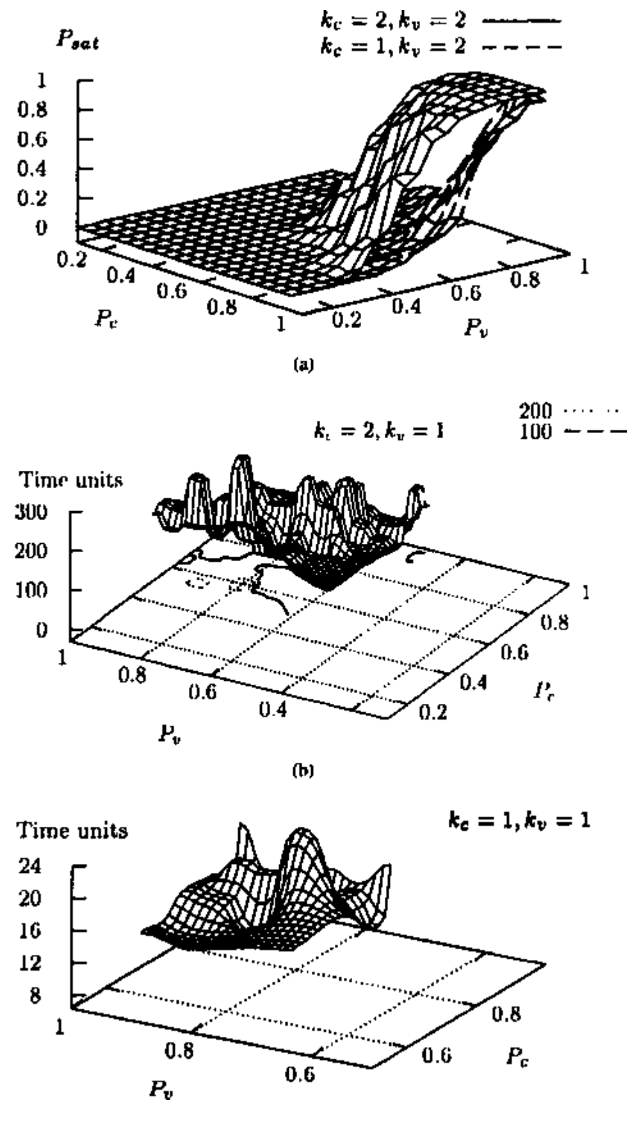

Figure 2: Some experiments with AWC on instances of GSensorDCSP: (a) ratio of satisfiable instances for k_v = 2, and mean solution time for (b) k_v = 2 and (c) k_v = 1, as a function of the density parameters P_c and P_v.

Consistently with the general SensorDCSP, we observe that the phase transition of GSensorDCSP coincides with the region where the hardest instances occur. For instance, Figure 3(b) shows the mean solution time with respect to the density parameters P_v and P_c for the problem instances with 25 sensors, 5 mobiles, k_v = 3, and k_r = 1. Somewhat

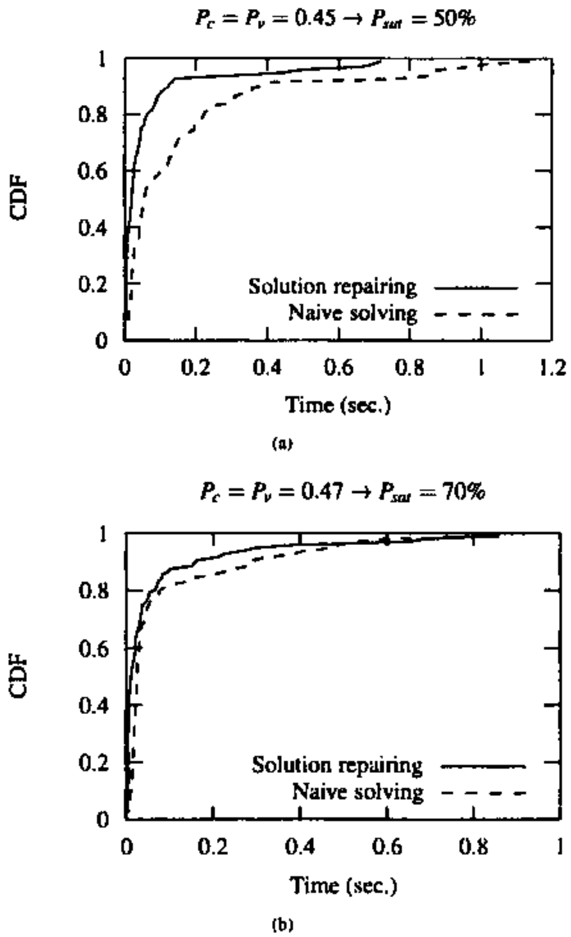

Figure 3: Solving dynamic GSensorDCSP problems with the naive and the solution repairing approaches: (a) performance for a problem generated from an instance in the 70 % P_{sat} (b) performance for a problem generated from a hard instance in the 50% P_{sat} point.

less expected result is depicted in Figure 3(c) for the case of $k_v = 1$ (and $k_c = 1$), which we showed to be polynomial by a reduction to the problem of feasible integral flow in bipartite graphs. Despite the fact that AWC is a general purpose algorithm, Figure 3(c) shows that these tractable instances are relatively easy for AWC. able to "detect and exploit" the tractable without being aware of its existence.

Extending the evaluation, we consider different sets of instances for several orders of the problem (grids of 25-100 sensors), and several levels of decomposition (K_c, $k_v \in \{2,3,4,5\}$). The analysis of these sets of instances clearly shows both exponential dependence of the mean solution time on the size of the grid, and, somewhat more interestingly, growth of the slope of this dependence with k_v.

4 Dynamic GSensorDCSP

As an additional extension of the GSensorDCSP domain, we consider the dynamics of the mobiles, in order to determine under which conditions (number of sensors and mobiles, velocities, trajectories, etc.) generic DisCSP algorithms are able to track the mobiles in real time. More formally, a dynamic GSensorDCSP problem can be seen as an ordered sequence of static GSensorDCSP problems $\Pi_1, \cdots, \Pi_N$. The static problems of the sequence only differ in the positions of the mobiles and hence the visibility between sensors and mobiles. Observe that, if no assumptions are made about the relationship between the positions of the mobiles in Π_i and Π_{i+1}, it is natural to solve these problems independently (naive approach). However, mobile dynamics are typically far from being chaotic. A first approximation is to consider linear trajectories. In this case, changes between the subsequent problems are governed by a clear model. An approach that a priori seems to be promising for dealing with the dynamic problem is to initialize the search for Π_i by the solution already achieved for Π_{i-1} (solution repairing approach).

In order to compare the naive and the solution repairing approach, we have performed experiments where we solve dynamic GSensorDCSP problems with both approaches. Figure 3 depicts the cumulative probability distributions of solving *Ui* for two different problems. One of them is obtained from the 50 % P_{sat} point of GSensorDCSP and the other from the 70 % P_{8at} point.

Our results show that we can exploit the previous solution and that, assuming certain reasonable constraints on the mobile movements, we can benefit from solution repairing over the naive approach. More interestingly, it also follows that the relative attractiveness of solution repairing is higher in the region of hardest instances.

5 Conclusions and Future Work

We believe that GSensorDCSP provides a realistic framework for the analysis of the DisCSP algorithms, and we hope it will contribute to the further research in this area. For the dynamic version, a next step would be to consider the effect of several kinds of trajectories in the performance of the solution repairing approach. Finally, a further step towards an even more realistic DisCSP benchmark would be to consider its optimization version: maximizing the number of tracked mobiles. In this extended distributed benchmark, it would be interesting to study the existence of the easy-hard complexity patterns that have been observed in classical optimization problems [Slaney and Walsh, 2002].

References

[Fernandez *et* a/., 2002] C. Fernandez, R. Bejar, B. Krishnamachari, and C. P. Gomes. Communication and computation in distributed CSP algorithms. In *Proc. of CP-2002,* pages 664-679,2002.

[Slaney and Walsh, 2002] J. Slaney and T. Walsh. Phase transition behavior: from decision to optimization. In *Proceedings of the SAT-2002 Conference,* 2002.

[Yokoo, 1994] M. Yokoo. Weak-commitment search for solving constraint satisfaction problems. In *Proc. ofAAAI-94*$_y$ pages 313-318,1994.

[Yokoo, 2001] M. Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-Agent Systems.* Springer-Verlag, 2001.