How (not) to Choose Mediators for Distributed Constraint Satisfaction

Michael Benisch and Norman Sadeh
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave. Pittsburgh, PA 15213
{mbenisch, sadeh}@cs.cmu.edu

ABSTRACT

Many successful algorithms, such as Asynchronous Partial Overlay (APO), have recently been developed for cooperative distributed problem solving based on the notion of coordinated mediation. In this paper we examine the impact of different strategies for choosing mediators with respect to the complexity of distributed problem solving and the difficulty in merging decentralized solutions. We present experimental results which challenge previously held beliefs suggesting that the appointment of highly constrained agents leads to a decrease in problem solving complexity. We show that, instead, choosing loosely constrained agents as mediators in order to minimize the expected size of mediation sessions can lead to an overall improvement in system performance.

1. INTRODUCTION

Distributed constraint satisfaction continues to serve as a valuable paradigm for studying cooperative problem solving techniques in domains such as resource allocation [2], and distributed timetabling and transportation routing [10]. Perhaps more importantly, DCSPs have helped researchers understand some of the deeper issues surrounding cooperative problem solving and reasoning [14, 3], such as trade offs between the effectiveness of cooperation and the drawbacks due to the resulting communication. There have been significant efforts dedicated to designing and testing various algorithms and protocols for addressing problems in this domain such as Asynchronous Backtracking (ABT) [13], Asynchronous Weak-Commitment (AWC) [12], and most recently Asynchronous Partial Overlay (APO) [6].

For the most part, much of the problem solving effort in these algorithms involves mitigating the effects of local decisions made by individual agents that destabilize the system. This process usually requires a significant amount of inter-agent communication, because agents must inform each other of conflicting intentions and cooperate to rectify them. The most effective algorithms, such as APO, have adopted the notion of cooperative mediation and manage to cut down significantly on unnecessary communication. Mediation-based algorithms involve the coordination of parts of the problem by mediator agents, who are charged with the responsibility of solving some centralized problem and supplying information to other agents that rectifies some or all of their conflicts. In this paper we explore the impact of different strategies for choosing mediators, and discuss the relationship between these strategies and the overall complexity of the distributed problem solving process. As of yet this kind of analysis has not been discussed, and we find that it provides valuable insight into the nature of cooperation.

Our analysis focuses on the Distributed Constraint Satisfaction Problem (DCSP) and helps to clarify the relationship between the computational properties of solving a centralized version of the problem, the difficulties in merging or overlaying mediated solutions, and the potential arising from the parallel and distributed nature of the problem domain. In Section 4 we describe the experimental results of applying our analysis to the APO algorithm by examining three different mediator selection strategies on distributed 3-color graph coloring problems. Our results suggest that on solvable 3-color graph coloring problems mediators should be selected in order to minimize the number of agents involved in each mediation, which is contrary to the previously suggested strategies of choosing mediators to centralize as much of the problem as possible.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

2. BACKGROUND

2.1 Distributed Constraint Satisfaction

The distributed constraint satisfaction problem was first discussed by Yokoo *et.al.* as a way of formalizing Cooperative Distributed Problem Solving (CDPS) [13], building on previous exploration into the realm of specific distributed constraint satisfaction problems [11, 4, 9]. A DCSP is formally defined as a constraint satisfaction problem (CSP) of the following form:

- a set of n variables, $V = \{x_1, \ldots, x_n\}$
- a set of discrete finite domains for each variable, $D = \{D_1, \ldots, D_n\}$
- a set of constraints $R = \{R_1, \ldots, R_m\}$ where each $R_i(d_{i1}, \ldots, d_{ij})$ is a predicate on the Cartesian product of the domains of all the variables referenced by that constraint. The constraint is said to be *satisfied* if the assignments of each referenced variable satisfy the constraint.

Where each agent is said to *know* about a particular set of variables and constraints. The goal of each agent is to assign a value to its variables that satisfies all of its known constraints, in an effort to satisfy the entire problem. For the rest of this paper, for the sake of simplicity, we will restrict our problem domain to associate each agent with a single variable, and its known constraints to be all of the constraints that refer to that variable. We will also consider only binary constraints (constraints between two variables), although the ideas presented can be generalized to relax both of these restrictions. It is also useful to note that CSPs are often viewed as graphs or networks where vertices represent individual variables, and edges represent constraints between two or more (in the case of hyper-edges) variables. In the distributed domain the constraint graph also typically represents a communication network between agents, because agents are more likely to communicate with others connected via constraints.

2.2 Related Work

In their original paper formalizing DCSPs, Yokoo *et.al.* also described a simple extension of a commonly used CSP solving technique called Asynchronous Backtracking (ABT). This technique involves a trial-and-error type solution, where agents continually assign their variables random values and revise them, or backtrack, when they receive indication from others that they have destabilized the system.

Soon thereafter several other techniques were ported from the centralized CSP domain, such as Asynchronous Weak Commitment (AWC) [12]. This algorithm extends ABT by allowing agents to treat assignments from other agents as weak-commitments, rather than unchangeable instantiations. When an agent identifies a conflict, it can choose to temporarily ignore the offending variable if it has reason to believe that its value can be easily changed. Agents decide whether or not offending variables can be easily changed based on dynamically computed priorities, which reflect the frequency with which the corresponding agent has been involved in other trial and error sessions.

The most recent, and most successful framework for DCSP algorithms so far is called Asynchronous Partial Overlay (APO) developed by Roger Mailler and Victor Lesser [6]. APO is based on the concept of cooperative mediation, where agents that identify conflicts in local sub-problems choose a mediator to solve a centralized version of the sub-CSP that includes the conflict, and then abide by the decisions of the mediator.

The use of cooperative mediation for distributed problem solving first appeared in work involving air-traffic control [1], and has appeared in various other distributed domains such as supply chain manufacturing [7]. These algorithms have proven mediation to be an extremely useful technique for coordinating distributed decision making, however there has been little to no formal discussion of the effects of different mediator selection strategies.

3. METHODS

3.1 Mediation, Overlay and Complexity

For the purposes of this paper, we will define the terms mediation and overlay in the following fashion with respect to DCSPs:

Definition: *mediation* is the process of solving a centralized version of a subproblem in the DCSP to resolve conflicts among two or more agents (one of which may be the mediator).

Definition: *overlay* is the process of fitting together partial solutions to a DCSP like puzzle pieces.

The remainder of the analysis in this section describes the relationship between the complexity of a mediation procedure, the mediation process, and the overlay process. In the most basic terms, the initiation of a mediation session in DCSP solving involves the following process:

1. A mediator must be appointed either arbitrarily or based on some strategy implemented by the agents.

2. The mediator must collect any information deemed necessary to aid the mediation process, such as the constraints placed on other agents involved in the session.

3. The mediator must derive a solution that satisfies identifiable conflicts.

4. The solution must be relayed and adopted by the other agents involved in the session, and the solution must be overlayed with problem solving efforts in the rest of the DCSP.

Notice that if mediation is the only technique available to the agents for satisfying conflicts and the mediator's solution does not perfectly overlay onto the problem, the final step will require additional mediation sessions to stabilize the system. Mediation in DCSP solving is usually initiated when a conflict is identified by an agent based on currently assigned variable values. The goal of the session will be to re-assign values to each agent involved that alleviate the conflict.

The benefits of this process to DCSP solving arise from computational improvements associated with finding a solution to the centralized version of the sub-problem, and communication improvements arising from the lack of negotiation needed since the mediator's decisions are adopted immediately by all agents in the session. The drawbacks of employing mediation based techniques in DCSP solving result from the wasted efforts of non-mediating agents during a mediation session, and the difficulty of overlaying solutions from several different mediation sessions.

Understanding the general form of mediation techniques helps us formalize their complexity in DCSP solving. Throughout our analysis we will make the following assumptions:

- For simplicity sake, and to focus on the primarily on the impact of mediation, we assume that agents have no manner of resolving conflicts other than mediation.

- We will assume that the complexity of the mediation process dwarfs all other agent processing, such as the computation required to choose a mediator. This assumption is justified considering that mediation usually requires performing some kind of search, which is likely to overshadow the complexity of other agent tasks.

- For a particular problem instance and a particular mediation framework we will assume that M provides a finite upper bound on the number of mediations required to reach a stable solution, or identify that the problem is infeasible. An upper bound exists for any complete algorithm such as APO, as mediation sessions can continually grow in size until the entire finite sized problem is mediated by a single agent in the worst case.

- We assume that the function $f(x)$ describes the worst-case complexity of mediating x agents, such that the computation involved by the individual mediator is $O(f(x))$.

Let $T(\omega)$ be a function such that the worst-case computational complexity of the entire mediation system can be described as $O(T(\omega))$, where $\omega \in \Omega$ defines a vector of size M whose i'th entry is the number of agents involved in the i'th mediation session, and Ω is the set of all such possible vectors that lead to a solution with $\leq M$

sessions (note that $|\Omega| \leq n^(M+1)$). Different chains of mediation sessions will result in different ω vectors, and any chain which requires fewer than M sessions will have values of zero for all entries beyond the number of necessary sessions. To clarify, the differences between chains can result from the selection of different mediators, and differences in solutions chosen by each mediator. Using these definitions and the assumptions above, the function $T(\omega)$ can be defined as the a linear combination of the complexity of each individual mediation session as follows:

$$T(\omega) = \sum_{i=1}^{M} f(\omega_i) \tag{1}$$

The mediation procedure that minimizes computational complexity[1] of the system, is the one that produces a vector, ω, which minimizes the value of T in Equation 1.

However, determining the number and size of mediation sessions required for the procedure to lead to a stable state often involves carrying out the entire problem solving process. Instead it is useful to use probabilistic analysis and empirical investigation to design mediation procedures that minimize the expected worst-case complexity of the process.

To that end we can describe the expected worst-case complexity of our mediation system by introducing a probability density function $p(\omega)$, which describes the probability that our mediation procedure involves the mediation sizes in the chain specified by ω. Using the probability function, we can describe the expected complexity of our system as $O(E[T(\omega)])$, where $E[T(\omega)]$ is equal to the function $\tau(p)$. This function involves summing the complexity over all possible values of $\omega \in \Omega$, multiplied by their probability. We can derive τ as follows:

$$\begin{aligned} E\left[T(\omega)\right] &= E\left[\sum_{i=1}^{M} f(\omega_i)\right] \\ \tau(p) &= \sum_{\omega\in\Omega}\left(\sum_{i=1}^{M} f(\omega_i)\right) p(\omega) \end{aligned} \tag{2}$$

We can conclude our analysis and reach the desired intuition by introducing one final set of assumptions.

- Let us assume that the probability function, $p(\omega)$, can be estimated with a function $p'(j)$, which satisfies Markov and independence properties; such that the probability of observing a mediation of size j depends only on the size of j and not on its previous or future values. This assumption allows us to reason about and measure the probability of producing sessions of a particular size, rather than specific chains.

- We will also assume that the function $g(p')$ relates the probability distribution over the size of mediation sessions to an estimate of the number of mediations required to reach a stable state, or the state beyond which the values of ω would be zero. This function can also be reasoned about and experimentally measured, and we believe introducing it helps to make our discussion more insightful.

Using these final assumptions (also recall that n is the number of agents and variables in the DCSP) we can re-write the expected worst-case complexity formula as:

[1] It is worth noting that other goals in mediator selection may be desirable, such as minimizing the amount of communication involved in reaching a stable state, which require different analysis.

$$\begin{aligned} \tau(p') &= \sum_{j=1}^{n}\left(\sum_{i=1}^{g(p')} f(j)\right) p'(j) \\ \tau(p') &= g(p')\sum_{j=1}^{n} f(j)p'(j) \end{aligned} \tag{3}$$

Notice that the function g describes the relationship between difficulty in overlaying solutions and likely session size, because the number of mediations required to reach a solution is related to how many conflicts are created by each session. The function f describes the relationship between difficulty in finding a mediation solution and session size. Thus, Equation 3 clearly illustrates the trade off between mediation procedures which are likely to involve coordinating large parts of the problem space, and procedures that are likely to coordinate smaller parts of the problem at the expense of involving more sessions. In terms of designing mediation procedures, this model suggests that during the execution of the procedure, optimal mediator selection strategies should be employed to adjust the probability distribution, p', over the size of mediation sessions and minimize the expected system complexity, τ.

3.2 APO Overview

The APO algorithm provides a basic framework for utilizing mediation in DCSPs, which we will employ to explore our theoretical model of distributed problem complexity. For the details of the APO algorithm the reader is directed to [6] (for additional details see [5]). The algorithm can be summarized as follows:

1. Agent i begins by assigning a random value, $d_i \in D_i$, to its variable, x_i.

2. It then calculates its local priority, p_i, based on the number of agents it share constraints with (its degree in the constraint graph).

3. The agent then communicates information about its local assignment, d_i, to all of its neighbors and when a conflict is identified by an agent that cannot be rectified by the agent alone, a mediator is chosen with the highest priority in the group of agents known be part of the conflict.

4. The agents who are part of the conflict communicate all the information they know about their local sub-problems, including:
   - the set of constraints, C_i, that apply to their variable,
   - the entire domain of their variable, D_i,
   - and information about other agents that are known to be in conflict with each of the values in their domain (this is referred to as a *labeled* domain).

5. This information is then used by the mediator to perform a branch-and-bound search guaranteed to find a feasible solution, if one exists, to the sub-problem pertaining to agents that are part of the mediation session that minimizes the number of constraints violated for agents outside of the session (the mediator determines the number of external violations using the labeled domains).

6. The mediator's solution is communicated to each of the agents involved in the session, and any external agent (not involved in the elapsed mediation session) who was violated by the solution is added to the mediator's neighborhood. Next time the same agent mediates, the violated agent will be included in the session.

APO	p_i = |`neighborhood(i)`|
Random APO (RAPO)	p_i = random number $\in [0, 1]$
Inverse APO (IAPO)	$p_i = \frac{1}{\lvert\texttt{neighborhood(i)}\rvert}$

Figure 1: Summary of Mediator Selection Strategies

3.3 Mediator Selection

In order to validate our theoretical model of mediation complexity, we will explore the performance of different mediator selection strategies within the framework of APO. As described above, mediators are selected in APO based on local priorities which are proportional to the number of agents in the mediator's neighborhood. To explore different mediator selection strategies we will change the way these priorities are generated.

Notice that the mediation aspect of the APO algorithm involves a complete branch-and-bound search of the centralized sub-problem. The search algorithm is well known to be exponentially complex in the number of variables on 3-color graph coloring problems. Thus in Equation 3 the function f is $O(2^j)$, which minimizes complexity when sessions are likely to be smaller. However, g will likely be inversely related to expected mediation size, because larger mediations will lead to fewer sessions. This suggests increasing the size of mediations and potentially mitigates the effects of the branch-and-bound complexity. An important insight is that because f is exponential in expected mediation size, if g is inversely *sub-exponential* in the expected mediation size, then τ will be minimized when p' favors smaller mediations.

Furthermore, during mediation, rather than choosing the first discovered feasible solution, mediators continue to search for the solution with the minimum number of external conflicts. This is intended to increase the likelihood that mediated solutions will overlay properly with existing solutions, and thus decrease the inverse complexity of the g function. Because of this fact, and the fact that on solvable graph coloring instances there will always be a way to overlay feasible solutions, our model predicts the expected complexity of the system on 3-color graph coloring problems will be minimized when the expected size of the mediations is smallest. This prediction is contrary to suggestions and intuitions presented in the description of the APO framework, which involves choosing mediators who are most constrained and thus will have the largest expected mediation sessions.

4. RESULTS

4.1 Distributed 3-color Graph Coloring

The experiments presented in this paper involve solving solvable instances of a distributed 3-color graph coloring (D3GC) problem using different mediator selection strategies with the APO algorithm. The D3GC problem is a DCSP where all of the constraints are *not-equal* constraints, and the domain of each variable contains 3 different colors. We generate random solvable instances according to the algorithm presented in [8], which has been used for benchmarking previous research in this area.

4.2 Experimental Setup

Figure 1 summarizes the different mediation strategies we examine in our experiments. The different strategies include the strategy suggested with the APO framework of choosing priorities proportional to an agent's number of constraints, a random strategy, which we will refer to as Random APO (RAPO), that assigns priorities completely randomly, and a strategy that assigns priorities inversely proportional to an agent's number of constraints. This last strategy is intended to minimize the expected mediation size, and will be referred to as Inverse APO (IAPO).

In our experiments we varied the number of variables in the problem, n, and the problem density, or number of constraints per variable, m. We generated 10 random solvable problems for each combination of n and m, n = 15, 30, 36, 45, 51, 60 and m = 2.0 (low density) and 2.7 (high density). For each of the 10 problems we generated 10 different random starting assignments which were shared across all algorithms, for a total of 1200 runs per algorithm.

4.3 Experimental Results

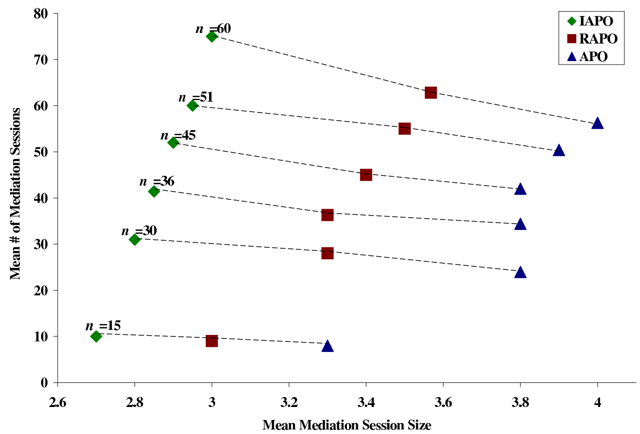
The relationship between the average size of mediation sessions and number of mediation sessions required for each algorithm is shown for each of the values of n and m in Figure 2. The running time results are shown in Figure 3 as percentage improvement graphs over the APO algorithm of the other two algorithms, with 95% confidence intervals. We are aware that the running time of an algorithm can be greatly effected by its implementation specifics, however these algorithms were all implemented in the exact same framework ensuring that differences were only due to the difference in mediator selection strategies. We also measured the number of cycles each algorithm required, and the number of messages passed between agents. During a cycle all messages are delivered to agents, they are allowed to process the information contained in the messages, and all messages emitted during the cycle are placed on the queue to be handled during the subsequent cycle. Both of these metrics revealed no significant differences between any of the algorithms or the results reported for APO in [6].

As expected, the IAPO algorithm has the smallest mediation sessions on average and the APO algorithm has the largest for all values of n and m. The mediation measurements confirm that there is an inverse relationship between the average size of the mediations and the number of sessions required for all algorithms and all values of n and m, however the relationship appears to be inversely sub-exponential. This relationship also appears to be more significant on problems with larger values of n and m, which can be explained by the increasing difficulty in overlaying solutions. The running time results confirm, that IAPO is significantly faster for all values of n and m than the APO algorithm, and appears to scale more effectively. The RAPO algorithm falls directly in between the other two algorithms on all reported metrics.

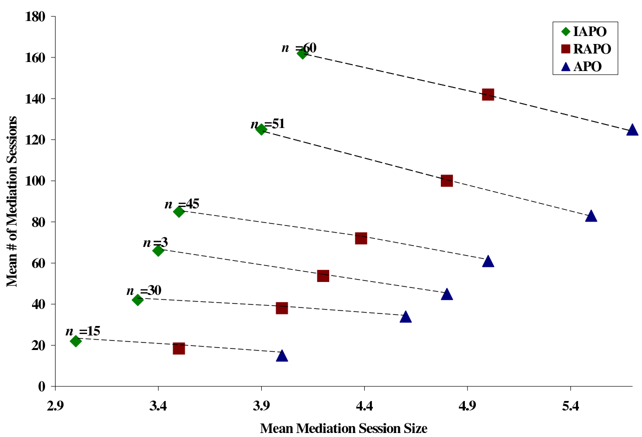
We performed similar experiments on random (not guaranteed to be solvable) 3-color graph coloring problems and found the results showed little difference in performance on average between the different mediator selection strategies. We attribute this to the fact that infeasible problem instances require larger mediation sessions to identify. In order to identify an infeasible problem the subset of variables which together prevent a feasible solution must be centralized by a mediator, which is more likely to happen when mediation sizes are larger. Thus, improvements on solvable instances by the IAPO algorithm are balanced by a more rapid recognition of infeasible problems by the APO algorithm.

5. DISCUSSION

In this paper we presented a theoretical model for understanding the computational complexity of mediation procedures for solving DCSPs. Our experimental results validate our theoretical model by examining different mediator selection strategies on solvable 3-color graph coloring problems. The results confirmed that the relationship between the expected size of a mediation session and the

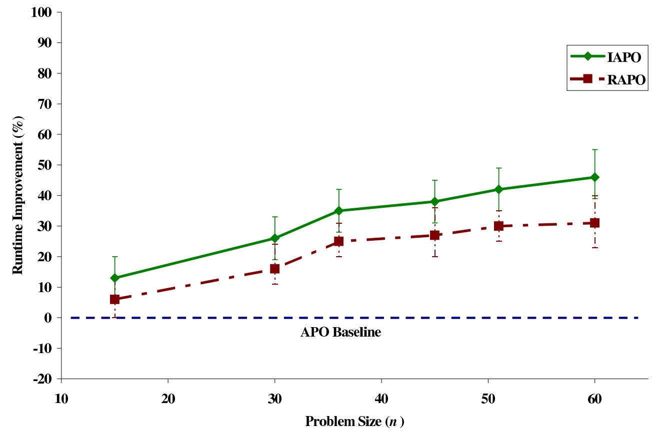


(a) Low Density Problems ($m = 2.0$)

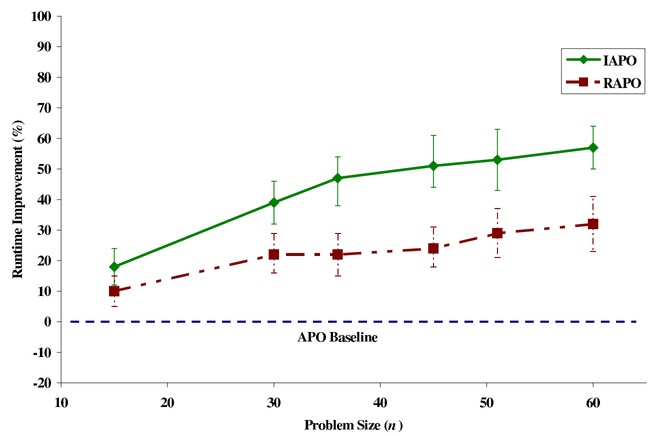


(b) High Density Problems ($m = 2.7$)

Figure 2: Mean mediation session size versus mean number of mediation sessions needed to solve random solvable D3GC instances.



(a) Low Density Problems ($m = 2.0$)



(b) High Density Problems ($m = 2.7$)

Figure 3: Running time needed to solve random solvable D3GC instances as mean percentage improvement over APO.

number of mediation sessions needed to reach a stable state was inversely sub-exponential on these problems. As our model predicted, this led to a significant increase in system running time as average mediation size increased. These results are contrary to previously reported intuitions about effective ways of selecting mediators, and the resulting algorithm outperformed APO, the previously most successful technique in this domain.

The model presented in this paper formalizes the relationship between the benefits of mediation for cooperation and the difficulty in fitting the resulting solutions together in DCSPs. DCSPs have been shown as formalized instances of cooperative distributed problem solving, and we believe this generalizes our model to provide insight into the larger domain of mediated cooperative problem solving.

6. ACKNOWLEDGEMENTS

The research reported in this paper has been funded by the National Science Foundation under ITR Grant 0205435 and under IGERT grant 9972762.

7. REFERENCES

[1] S. Cammarata, D. MacArthur, and R. Steeb. Strategies of cooperation in distributed problem solving. In *Proceedings of the 8th Internation Joint Conference on Artificial Intelligence(IJCAI-83)*, volume 2, pages 767–770, 1983.

[2] S. Conry, K. Kuwabara, V. Lesser, and R. Meyer. Multistage Negotiation in Distributed Constraint Satisfaction. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1462–1477, November 1992.

[3] H. Jung, M. Tambe, and S. Kulkarni. Argumentation as distributed constraint satisfaction: applications and results. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 324–331. ACM Press, 2001.

[4] V. Lesser. An Overview of DAI: Viewing Distributed AI as Distributed Search. *Journal of Japanese Society for Artificial Intelligence-Special Issue on Distributed Artificial Intelligence*, 5(4):392–400, January 1990.

[5] R. Mailler. *A Mediation-Based Technique for Cooperative Distributed Problem Solving*. PhD thesis, The University of

Massachusettes, 2004.

[6] R. Mailler and V. Lesser. Using Cooperative Mediation to Solve Distributed Constraint Satisfaction Problems. In *Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, volume 1, pages 446–453, New York, 2004. IEEE Computer Society.

[7] F. Maturana, W. Shen, and D. Norrie. Metamorph: An adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37(10):2159–2174, 1999.

[8] S. Minton, M. D. Johnston, A. B. Phillips, and P. Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction problems. *Artificial Intelligence*, 58(1-3):161–205, 1992.

[9] H. Parunak, P. Lozo, R. Judd, and B. Irish. A distributed heuristic strategy for material transportation. In *Proceedings 1986 Conference on Intelligent Systems and Machines*, Rochester, Michigan, 1986.

[10] G. Solotorevsky and E. Gudes. Solving a Real-life Time Tabling and Transportation Problem Using Distributed CSP Techniques. In *Proceedings of CP '96 Workshop on Constraint Programming Applications*, pages 123–131, 1996.

[11] K. Sycara, S. F. Roth, N. Sadeh, and M. S. Fox. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1446–1461, December 1991.

[12] M. Yokoo. Asynchronous weak-commitment search for solving distributed constraint satisfaction problems. In *Proceedings of the First International Conference on Principles and Practice of Constraint Programming*, pages 88–102, 1995.

[13] M. Yokoo and E. H. Durfee. Distributed constraint satisfaction for formalizing distributed problem solving. In *12th IEEE International Conference on Distributed Computing Systems*, pages 614–621, 1992.

[14] M. Yokoo, T. Ishida, and K. Kuwabara. Distributed constraint satisfaction for DAI problems. In *10th International Workshop on Distributed Artificial Intelligence*, 1990.