A Complete Asynchronous Method for Solving any Distributed Constraint Satisfaction Problem

Ahlem Ben Hassine, Tu Bao Ho
School of Knowledge Science, Japan Advanced Institute of Science and Technology
Tel. : 81-(0)761-51-1111 - Fax. : 81-(0)761-51-1795
E-mail : {hassine, bao}@jaist.ac.jp

Résumé :

Le problème de satisfaction de contraintes (CSP) est un formalisme puissant pour représenter et résoudre n'importe quel problème combinatoire. Récemment, avec le développement et l'apparition de nouvelles technologies des systèmes distribués et des réseaux, particulièrement avec l'omniprésence des applications naturellement réparties au monde réel, le cadre distribué a attiré l'attention de plusieurs chercheurs. Cependant la plupart des approches proposées dans la littérature, qui sont basées sur la représentation principale d'un réseau de contraintes (CN), traitent directement des problèmes binaires, exigent particulièrement l'enregistrement des nogoods et nécessitent un nombre élevé de connexions réseau. Nous proposons dans cet article une nouvelle approche asynchrone de recherche multi-contraintes, basée sur la représentation duale des CSP et aussi sur le principe de Backtracking pour assurer la complétude de la recherche. Nous proposons aussi une nouvelle méthode distribuée pour ordonner les agents du système afin de réduire au minimum les connexions inutiles. L'approche est discutée en termes de terminaison et complétude. Une évaluation comparative expérimentale est également donnée.

Abstract :

Constraint satisfaction problem (CSP) is a powerful formalism to represent and to solve any combinatorial problem. Recently, with the advent of both distributed computing and networking technologies, especially with the omnipresence of naturally distributed real world applications, distributed framework has attracted the attention of many researchers. Most approaches proposed in the literature are based on the primal representation of a constraint network (CN), deal directly with binary problems, require nogood recording and require a high amount of connections. In this paper we describe a novel multi-constraint asynchronous search approach based on dual graph, although based on the principle of backtracking to ensure the completeness of search. We propose a new distributed method to set an ordering among the agents of the system in order to minimize the amount of connections. The approach is discussed in term of termination and completeness. An experimental comparative evaluation is also given.

1 Introduction :

Constraint satisfaction problem (CSP) formalism [Montanari, 1974] is widely used to formulate and to solve many combinatorial problems, such as planning, resource allocation, time tabling and scheduling. The great success of the CSP paradigm is due essentially to its natural expressiveness of real-world applications.

A CSP is triplet (X, D, C) composed of a finite set of *n* variables $X=\{X_1, ..., X_n\}$, each of which is taking values in an associated finite domain $D=\{D_1, ...,D_n\}$ and a set of *e* constraints between these variables $C=\{C_{ij}, ...\}$; C_{ij} is a constraint between X_i and X_j. The constraints restrict the values the

variables can simultaneously take. Solving a CSP consists in finding one or all-complete assignments of values to variables satisfying all the constraints. This type of problem is known as NP-Complete for which the solving task is hard, i.e. when a blind search often leads to a combinatorial explosion.

Recently, with the advent of distributed computing, netw orking technologies and especially with the omnipresence of naturally distributed real world CSP applications, the interest increases toward distributed computing leading to several distributed approaches in distributed CSP (DisCSP) . A DisCSP [Yokoo et al., 1990] is a CSP in which variables and constraints are distributed among a set of automated agents. Several applications in multi-agent systems (MAS) (e.g., distributed resource allocation problems [Conry et al., 1991], distributed scheduling problems [Sycara et al., 1991], and multi-agent truth maintenance tasks [Huhns and Bridgeland, 1991]) can be formalized as distributed CSPs.

Many significant efforts dealing with solving distributed constraint problems have been proposed in the literature. These works can be divided into two groups, complete and incomplete methods. The most dominant work is the complete variable-based distributed algorithm of Yokoo and colleagues, the Asynchronous Backtracking algorithm (ABT) [Yokoo et al., 1992]. This algorithm is a statically ordered asynchronous protocol in which the agents of higher level communicate their tentative value assignment to their lower level neighbor agents. In case of conflict, the concerned agent generates a nogood and sends it to its neighbor agent of higher level. Several extensions of the ABT algorithm were proposed, among which, the asynchronous weak commitment (AWC) [Yokoo, 1995]. This algorithm is based essentially on the min-conflict heuristic with dynamic reordering protocol in order to moderate the influence of bad decisions taken by a higher level agent. As mentioned in [Maestre and Bessiere, 2004], this algorithm is incomplete unless agents can store a potentially exponential number of nogoods. Another incomplete extension based on ABT approach was proposed by [Hamadi et al., 1998], the distributed backtracking algorithm (DIBT), which performs a graph-based backjumping behavior during failure phases without nogood storage.

Recently, Silaghi and colleagues proposed new constraint-based model algorithm, the asynchronous search aggregation (AAS) [Silaghi et al., 2000]. This work is considered as an ABT [Yokoo et al., 1992] for dual graph. The AAS technique consists in propagating aggregated tuples of Cartesian product of values rather than individual values themselves. The agents are assigned priority basically based on the lexicographic order. A link is set between each pair of agents if they share at least one variable. AAS works in exactly the same manner as ABT, except that messages refer to Cartesian products. If an agent find no combination in the Cartesian product $\{X_i=\{a_1, \ldots, a_l\}\}X\{X_j=\{b_1, \ldots,b_k\}\}$ is compatible with its constraints, it generates a nogood for this combination and sends it to a higher order agent.

Most works are based on the primal representation [Dechter and Pearl, 1988] of a CN, where the nodes are the variables of the problem and the arcs are the constraints relating these variables. Nevertheless, the use of primal representation for non-binary problems requires the addition of several links and/or nodes to express the n-ary constraints. Although, these transformations, may lead to the loss of a part of the semantic of the underlying constraints [Regin, 94]. Only the work discussed in [Silaghi et al., 2000] is a constraint-based model. However, this approach requires a high amount of connections between agents, i.e., each pair of constraints sharing one variable should be connected. In addition, the search for a viable Cartesian product increases the number of constraint checks. Hence, the cost of the search for a consistent assignment grows with the amount of connections and with the required constraint checks.

Our main contribution is to propose a novel complete and generic multi-agent algorithm for any constraint network, i.e. n-ary constraints, to solve any CSP. The proposed approach is based in a part on a lazy version the DRAC [BenHassine and Ghedira, 2002] approach (*Distributed Reinforcement of Arc Consistency*), without adding any new links and without recording any

nogoods. The main idea of using a lazy version of DRAC is to save as many fruitless backtracking as possible and consequently to enhance the efficiency of the proposed approach. In addition, We describe a novel generic distributed method to compute a static constraint ordering, in which we save as many links as possible in order to decrease the number of exchanged messages. In addition, information about variables may belong to different agents while information about constraints belong only to the owner agent and kept confidential.

This paper is organized as follows. In section 2, we present the proposed dual based asynchronous solving approach. In section 3, we give the experimental results. Finally, section 4 concludes the paper.

2 Dual-graph based Asynchronous Solving Approach:

2.1 Multi-agent Architecture and Global dynamic:

The proposed multi-agent architecture is based on the dual representation of a CN. This model involves two kinds of agents: Constraint agents and an Interface agent. The later agent is added to the system in order to inform the user of the result. Each agent has simple structure formed by its acquaintances (the agents that it knows), a local memory composed of its static and dynamic knowledge, a *mailbox* where the agent stores the received messages and a local behavior. All the agents communicate by exchanging asynchronous point-to-point messages containing only relevant information. An agent can send a message to another one only if it belongs to its acquaintances. For transmission between agents, we assume that messages are received in the order they were sent. The messages delivering time is finite.

The main common global objective of all the agents is to solve any constraint problem. This dynamic is divided into two steps:

- First step, a "partial" enforcement of arc consistency [Mackworth, 1977], consists in pruning some non-viable values and propagating them to high level agents in order to decrease the amount of backtracking and hence to reduce the complexity of the solver. This step can be viewed as a *lazy* version of DRAC approach.

- Second step, the solving process, consists in solving the obtained problem via interactions and negotiations between all the agents of the system. Each agent searches the suitable tuple that, on the one hand, satisfies its associated constraints and, on the other hand, satisfies all the agents belonging to its parents and children.

2.2 Generic Parallel New Method for Static Constraint Ordering:

The complexity of the CN and the number of exchanged messages are highly dependent on the existing connections between the agents of the system. In this section we propose a new distributed method to define an optimal global order, i.e. optimal in term of connections, between the agents. In our system, each agent will compute locally its position in the ordering according to its variables. The first variable of an agent A_i, $Var^{Ai}[1]$, defines its level and will be used to determine both its set of higher level acquaintances, i.e., $Parents^{Ai}$, and its set of lower level acquaintances, i.e., $Children^{Ai}$,. The agent A_i responsible of the constraint C_{ij} will be in the level i. The obtained graph should satisfy the following property in order to ensure the completeness of the solving approach.

***Property* 1.** For each variable $X_i \in X$, for all the agents A_k such that $X_i \in Var^{Al}$, there is *only one* continue path relating them.

Initially we assume that for each agent A_i the set of children is all the constraints with which the agent shares at least one variable (basic of the dual graph). Each agent C_{ij} will reduce the set of its $Children^{Ai}$ and this by using the following rules:

Rule 1. Remove all A_l, $Var^{Al}=\{X_i, X_k\}$, from $Children^{Ai}$ ($Var^{Ai}=\{X_i, X_j\}$) such that $A_l \pi_{lo} A_i$, i.e., $A_l \pi_{lo} A_i$ if and only if $k < j$.

Rule 2. Remove all A_h, $Var^{Ah}=\{X_f, X_j\}$, from $Children^{Ai}$ ($Var^{Ai}=\{X_i, X_j\}$) such that $f > i+1$ and there is no $A_l \in Children^{Ai}$ with $Var^{Al}=\{X_m, X_j\}$ and $m \in \{i, ..., (f\text{-}1)\}$.

Once the set of the children is reduced each agent will inform each agent responsible of constraint belonging to its set of children that it is the father. Then each agent receiving the above message will add this agent to its set of parents, $Parents^{Ai}$.

It is noteworthy that for a full connected graph, using n variables, the total number of constraints is $n(n\text{-}1)/2$. For each constraint we will have $2(n\text{-}2)$ links then the total number of links for the dual graph is n^3. In case of ordered dual graph, each constraint C_{ij} have $(2n\text{-}(i+j)\text{-}1)$ ordered links. The total number of ordered links is $n(n\text{-}1)/2*(2n\text{-}(i+j)\text{-}1)$. As for our ordering method, each constraint of level $i \in \{2, \ldots, (n\text{-}1)\}$ will be connected to $2(n\text{-}i)$ other constraints in the next level, only the first level $i=1$ needs more $(n\text{-}2)$ ordered connections. Thus the remaining ordered connections for the proposed method is $n(n\text{-}2)$. We can easily see that our method save much more connections and consequently decrease the complexity of the exchanged messages in a real distributed computers architecture.

In addition, we assume to affect a leader for each variable, which will be responsible of this variable. Each agent A_i that has no parent for *at least* one of its variables, it will be the leader of this variable.

3 Experimental Comparative Evaluation:

We have developed the multi-agent dynamic with Actalk [Briot et al., 1997], an object oriented concurrent programming language using the Smalltalk-80 environment. In our experiment, we generated random constraint problems. The parameters used for a meeting problem are: n variables in the system, d size of the maximal domain, p the density of the problem, q the tightness of the constraints.

Our goal is to evaluate the performance of our approach especially on hard problems. For this purpose, we generated random problems near the peak of difficulty [Cheesman et al., 1991] with $n=15$, $d=5$, $p=30\%$ and q varied from 25% to 85%. For each $<p, q>$ we generated 5 instances. Then we measured the average of the obtained results. These results are expressed in terms of three criterions: the number of constraint checks, the CPU time and the number of exchanged messages. It is noteworthy that these experiments are the first ones, we plan to conduct more exhaustive `ones on more complicated and largest problems.

Table 1. Results in mean of constraints checks and CPU time.

	<0.3, 0.25>	**<0.3, 0.35>**	**<0.3, 0.45>**	**<0.3, 0.55>**
Const Checks	684.8	659.2	542	446
CPU Time	112.2	170.4	177.75	198.6
Nbr of Msg	112.4	305.2	322	363.4
	<0.3, 0.65>	**<0.3, 0.75>**	**<0.3, 0.85>**	
Const Checks	387	402.6	379.6	
CPU Time	194	241.4	260.6	
Nbr of Msg	326.6	398.6	438.6	

We used the same parameters as those given by most researchers in solving distributed complex problems. Table 1 shows that this approach required low amount of constraint checks and

consequently CPU tie and exchanged messages (compared to the results presented in [Hamadi et al., 1998], [Silaghi et al. 2000]). This can be vindicated by the use of the knowledge collected during the lazy enforcement of arc consistency (the set of supports).

4 Conclusion:

We have presented in this paper a new distributed asynchronous approach to solve any constraints network (n-ary constraints). The proposed multi-agent model is based on dual graph representation of CSP where each agent maintains a constraint of the problem. These agents will cooperate concurrently and asynchronously without any central control. However, we proposed in addition a new distributed method to establish a total order among agents with the minimum amount of connections. The main reason is to make it use easy in real distributed environment and also to decrease the required exchanged messages.

There are two main ideas underlying this approach. First is to perform lazy enforcement of arc consistency, in order to avoid basing high order agents' decisions on arc-inconsistent values. Second is that in case of conflict, a backjumping is performed to the leader of the concerned shared variable and not to the nearest parent. The reason is to decrease the number of useless backtracking that can be done between the agent source of conflict and the first agent responsible of the concerned variable.

In addition, we do not perform either nogood recordings or new links addition in the new approach. This approach includes an enhanced detection mechanism.

In our future work, we plan to improve the approach by exploiting more the knowledge collected during the solving process, and to evaluate the approach in really distributed environment.

References

1. Briot, J.P. Actalk: A Framework for Object-Oriented Concurrent Programming - Design and Experience. In *Proceedings 2nd France-Japan Workshop* (*OBPDC*-97), 23 pages, 1997.
2. BenHassine, A., Ghedira, K. How to Establish Arc-Consistency by Reactive Agents. In *Proceedings of the 15th ECAI*-02, pages 156-160, 2002.
3. Conry S.E., Kuwabara K., Lesser V.R., and Meyer R.A. Multistage Negotiation for Distributed Constraint Satisfaction. In *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1462-1477, 1991.
4. Cheesman P., Kanefsky B., and Taylor W. Where the Really Hard Problems are. In *Proceedings of the 12th International Joint Conference on AI*, 1991.
5. Dechter R. and Pearl, J. Tree-Clustering Schemes for Constraint-Processing. In *AAAI*, 1988.
6. Huhns M.N., and Bridgeland D. M. Multiagent Truth Maintenance. In *IEEE Transactions on Systems, Man and Cybernetics,* 21(6)*:*1437-1445*,* 1991.
7. Hamadi Y., Bessiere C., and Quinqueton J. Backtracking in Distributed Constraint Networks. In *Proceedings ECAI*, pages , 1998.
8. Montanari, U. Networks of Constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7: 95-132, 1974.
9. Maestre A., and Bessiere C. Improving Asynchronous Backtracking for Dealing with Complex Local Problems. In *Proceedings of ECAI*, pages , 2004.
10. Regin, J. C. A Filtering Algorithm for Constraints of Difference in CSPs. In *Proceedings AAAI'*94, pages 362-367, 1994.
11. Silaghi, M.-C., Sam-Haroud D., and Faltings, B.V. Asynchronous Search with Aggregations. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 917-922, 2000.
12. Sycara K.P., Roth S., Sadeh N., and Fox M.S. Distributed Constrained Heuristic Search. In *IEEE Transactions on Systems, Man and Cybernetics,* 21(6)*:*1446-1461*,* 1991.
13. Yokoo M., Ishida T., and Kuwabara K., Distributed Constraint Satisfaction for DAI Problems. In *Proceedings of the 10th Internatinal Workshop on Distributed Artificial Intelligence*, 1990.
14. Yokoo M., Durfee E.H., Ishida T., and Kuwabara K. Distributed Constraint Satisfaction Formalizing Distributed Problem Solving. In *Proceedings of DCS*, 1992.
15. Yokoo M. Asynchronous Weak-commitment Search for Solving Distributed Constraint Satisfaction Problems. In *Proceedings of CP*, 1995.