ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-901

Dynamic Backtracking for Distributed Constraint Optimization[1]

Redouane Ezzahir[2] and **Christian Bessiere**[3] and **Imade Benelallam**[2] and **El Houssine Bouyakhf**[2] and **Mustapha Belaissaoui**[4]

Abstract. We propose a new algorithm for solving Distributed Constraint Optimization Problems (DCOPs). Our algorithm, called DyBop, is based on branch and bound search with dynamic ordering of agents. A distinctive feature of this algorithm is that it uses the concept of valued nogood. Combining lower bounds on inferred valued nogoods computed cooperatively helps pruning dynamically unfeasible sub-problems and speeds up the search. DyBop requires a polynomial space at each agent. Experiments show that DyBop has significantly better performance than other DCOP algorithms.

1 INTRODUCTION

The Distributed Constraint Optimization Problem (DCOP) is a powerful formalism to model a wide range of applications in multi-agent coordination. The major motivation for research on DCOPs is that they are an elegant model for many every day combinatorial problems that are distributed by nature, such as distributed resource allocation, distributed scheduling, or sensor networks.

In this paper, we present a new distributed algorithm, called DyBop: Dynamic Backtracking search for DCOPs. It is based on branch and bound search and uses the concept of valued nogood introduced in [2, 7]. DyBop is guaranteed to terminate and requires polynomial space. The agents assign their variables sequentially and compute asynchronously a lower bound to the cost of the current context. Whenever an agent is successfully assigned, it sends copies of the current context to all unassigned agents concurrently. These unassigned agents send back the cost of their cheapest assignment wrt this context. If the aggregation of the costs received by the current agent is greater than the current upper bound, the current agent changes its value. If no value remains available, a valued nogood is sent from the current agent to the lowest agent involved in the nogood. Experimental results on random Max-DisCSPs and a real structured problem (Distributed Meeting Scheduling) show that DyBop outperforms both AFB-BJ [5], NCBB [1] and ABFS [4].

2 BACKGROUND

The Distributed Constraint Optimization Problem is a tuple $(\mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, F)$, where $\mathcal{A}$ is a set of agents $\{A_1, A_2, ..., A_k\}$, $\mathcal{X}$ is a set of variables $\{X_1, X_2, ..., X_n\}$, and $\mathcal{D} = \{D_1, D_2, ..., D_n\}$ is a set of domains, where each D_i in $\mathcal{D}$ is a finite set containing the values to which its associated variable X_i may be assigned. Only the agent who contains a variable has control of its value, and knowledge of its domain. $\mathcal{C} = \{c_{ij} : D_i \times D_j \rightarrow \mathbb{R}^+$, with $i, j \in 1...n, i \neq j\}$ is a set of constraints, represented by a cost function c_{ij} for each pair of variables X_i and X_j. The goal is to find a global assignment I of values to variables in $\mathcal{X}$ such that the objective function F is minimized. The valued nogood [2] is an extension of the classical nogood for Valued CSP. Recently, Silaghi et al. have introduced the inference power of valued nogoods in DCOP solving [7].

Definition 1 (Valued Nogood [2]) *A valued nogood has the form (I, v, C). It specifies that given a set of constraints C, a global assignment extending the partial assignment $I = \{(X_1, v_1), \ldots, (X_k, v_k)\}$ has cost at least v. C is a set of reference constraints called justification in [7].*

3 DYBOP

In DyBop, each agent stores a nogood per value. During search, each agent holds its view of the current state of search in a data structure called current context $CCTX$.

Definition 2 (Context) *For a partial assignment PA that we try to extend, we associate a current context $CCTX$ of the form $\langle PA, \mathcal{N}, CS\rangle$, where $\mathcal{N} = \{N_{X_1}, \ldots, N_{X_k}\}$ is the set of all nogoods associated to variable assignment in PA, and $CS = \{CS_{X_1}, \ldots, CS_{X_n}\}$ is a list of conflict sets. Each conflict set CS_{X_i} contains all agentID which are used to identify the assignments in any nogood stored by X_i.*

In DyBop, only one agent performs an assignment on the current context $CCTX$ at a time. Whenever the agent was successfully assigned, it sends copies of $CCTX$ to all unassigned agents concurrently and awaits for response messages. All unassigned agents compute asynchronously a valued nogood with a valuation equal to the lower bound of the cost of assigning a value to their variables, and send this nogood back to the agent which performed the assignment. The assigning agent accumulates these valued nogoods using *sum-inference*, an aggregation operator based on the objective function F. Once the valuation of accumulated nogood exceeds that of the best known solution found so far, the agent prunes its current value. The accumulated nogood is stored as explanation of value removal. On the other hand, when the cost of the aggregation of all valued nogoods coming from unassigned agents is less than the cost of the best known full assignment, the agent sends the current context to the agent selected as next. So, the current context is propagated forward sequentially. Whenever the current agent cannot find a valid value, it

[1] This work has been supported by the Maroc-France PAI project no. MA/05/127 and the ANR project ANR-06-BLAN-0383-02.
[2] LIMIARF/FSR, Morocco, email: ezzahir@lirmm.fr, bouyakhf@fsr.ac.ma, imade.benelallam@ieee.org
[3] LIRMM (CNRS/U. Montpellier), France, email: bessiere@lirmm.fr
[4] Université Hassan I, Morocco, email: m.belaissaoui@encg-settat.ma

performs the min-resolution of its stored set of nogoods, and sends back the resulting nogood to the last assigned agent in this nogood. When an agent receives such a valued nogood due to a backtrack, before storing it, the agent performs a partial reduction [2] of this nogood by using the last stored nogood related to the current value and the nogood related to the current assignment $CCTX$.

The communication among DyBop agents is performed by five types of messages. **CTX**: A message that carries the current context $CCTX$. **FB_CTX**: A forward bounding message that is an exact copy of a $CCTX$. Every agent that assigns its variables on a $CCTX$ creates an exact copy in the form of a FB_CTX and sends it forward to all unassigned agents. An agent receiving an FB_CTX message computes a valued nogood with a valuation equal to the lower bound on the cost increment caused by adding an assignment for its variables to the $CCTX$. This estimated nogood is sent back to the agent which sent the FB_CTX message via an **ESTIMATE** message. **BACK**: A message that is sent back when dynamic backtracking is performed. It carries a valued nogood that justifies the conflict and the current context $CCTX$. The receiver of this message is chosen as the last assigned agent in the carried nogood.

Theorem 1 *DyBop is correct and terminates.*

Proof. During DyBop search, all operations on valued nogoods are logically sound. Thus, if DyBop terminates, the upper bound is optimal, so solution is found. Termination can be proved if we consider a simple version of DyBop where FB_CTX messages are not used. This version is a complete algorithm because the nogoods produced by min-resolution are similar to classical nogoods rejecting an assignment. Therefore, it is sufficient to apply the method used by Ginsberg to show the termination of centralized dynamic backtracking. When we add FB_CTX messages, the stored nogoods coming with these messages cannot break the termination because they follow the same inference principle used for nogoods coming from Back messages. Thus, DyBop terminates. □

4 EXPERIMENTAL RESULTS

We considered two different problems for our experiments. The first was a random Max-DisCSP with 10 agents containing a single variable, in which all constraint costs are equal to 1, and density of the constraint graph is 40%. The second was a real structured problem, the Distributed Meeting Scheduling problem (DMS). We have tested three DMS problem classes. Each DMS problem class is represented by the pair (m, p) = (#meetings, #participants) in which there are p agents with multiple variables (m variables to the maximum). There are 5 values in each domain, each of them representing a possible meeting start time. All experiments were performed on the DisChoco platform [3] in which agents are simulated by threads which communicate only through message passing. We evaluate the algorithms performance by the mean of non obsolete messages (NO-MSGs), and the Equivalent Non Concurrent Constraint Checks (ENCCCs) [1] on 10 instances. ENCCCs are a weighted sum of processing and communication time. For random problems, we simulate two scenarios of communication: fast communication (message delay cost = 0 CCCs), and slow communication (message delay cost = 1000 CCCs). Experimental results are shown in Fig. 1. We observe that for almost all parameter settings, DyBop is significantly better than both AFB-BJ and ABFS. For DMS problems, Table 1 presents the results in a slow communication system. It shows that DyBop is even better than on random. On instances $(5, 7)$, with a cutoff set at 1,800 seconds,

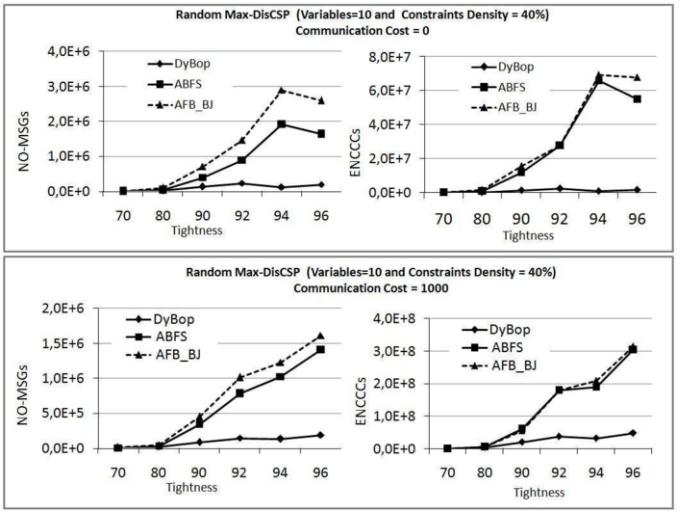

Figure 1. Results on random instances of max-DisCSP with 10 agents

DyBop provides optimal solutions for all instances against 70% for ABFS and AFB. We point out that Gershman et al. showed that AFB-BJ is faster than DPOP, which has been shown faster than NCBB [6].

Table 1. Results on DMS. #msg = #NO-MSG and #cc = #ENCCCs

	DyBop		ABFS		AFB-BJ	
(m, p)	#msg	#cc	#msg	#cc	#msg	#cc
$(5, 5)$	931	2,299	1,315	3,730	1,946	7,982
$(5, 6)$	1,676	3,797	3,365	12,450	5,403	34,188
$(5, 7)$	2,597	5,791	316K	1,513K	5,387K	80,345K

5 CONCLUSION

We have presented DyBop, a new algorithm for solving Distributed Constraint Optimization Problems (DCOPs). DyBop is based on branch and bound search with dynamic ordering of agents, and it uses the concept of valued nogood introduced in [2, 7]. Experiments show that the proposed approach of combining lower bounds on inferred valued nogoods computed cooperatively speeds up the search significantly wrt existing techniques.

REFERENCES

[1] A. Chechetka and K. Sycara., 'No-commitment branch and bound search for distributed constraint optimization', in *AAMAS*, pp. 1427–1429, (2006).
[2] P. Dago and G. Verfaillie, 'Nogood recording for valued constraint satisfaction problems', in *ICTAI*, pp. 132–139, (1996).
[3] R. Ezzahir, C. Bessiere, M. Belaissaoui, and E.H. Bouyakhf., 'Dischoco: A platform for distributed constraint programming.', in *IJCAI'07 Workshop on DCR*, pp. 16–21, (2007).
[4] R. Ezzahir, C. Bessiere, E. H. Bouyakhf, I. Benelallam, and M. Belaissaoui, 'Asynchronous breadth-first search DisCOP algorithm', in *EUMAS'07*, (2007).
[5] A. Gershman, A. Meisels, and R. Zivan, 'Asynchronous forward-bounding with backjumping', in *IJCAI'07 Workshop on DCR*, pp. 28–39, (2007).
[6] A. Gershman, R. Zivan, T. Grinshpoun, A. Grubshtein, and A. Meisels, 'Measuring distributed constraint optimization algorithms', in *AAMAS'08 Workshop on DCR*, (2008).
[7] M.C. Silaghi and M. Yokoo, 'Nogood based asynchronous distributed optimization (adopt-ng)', in *AAMAS*, pp. 1389–1396, (2006).