Distributed Dynamic Branch-and-Bound

For DisCOPs

Imade Benelallam*, Mustapha Belaissaoui∓, Redouane Ezzahir* and El Houssaine Bouyakhf*

*LIMIARF, Faculty of sciences, Rabat, Mohammed V-agdal University, Morocco,
imade.benelallam@ieee.org, ezzahir@lirmm.fr and bouyakhf@fsr.ac.ma
∓ENCG, Settat, Hassan I University, Morocco,
m.belaissaoui@encg-settat.ma

ABSTRACT. Some distributed constraint satisfaction problems (DisCSPs) are based on reasoning with nogoods, but only few optimization protocols get the most out of valued nogoods's power [14]. In this paper, we present a new search algorithm for distributed constraint optimization problems (DisCOPs), called Distributed Dynamic Branch-and-Bound (DisDBnB). Agents assigns their variables sequentially and forwards valued nogoods (lower bounds) synchronously to the next uninstantiated agent. Combining lower bounds on inferred valued nogoods can help to speed up the search, and prune infeasible sub-problems. We show that the algorithm proposed here (DisDBnB), is optimal and guaranteed to terminate, having polynomial space complexity. Detailed experimental results show that on benchmark problems (random Max-DisCSPs and graph coloring problems) the proposed algorithm achieve over an order of magnitude reduction in the messages exchanged between agents. DisDBnB results are compared to Synchronous Branch-and-Bound (SynBnB) algorithm.

RÉSUMÉ. Certains problèmes de satisfaction de contraintes distribués (DisCSPs) sont basés sur le principe des nogoods, mais rare sont les protocoles d'optimisation qui tirent le meilleur parti de la puissance des nogoods valués [14]. Dans cet article, nous proposons un nouvel algorithme de recherche pour les problèmes d'optimisation de contraintes distribués (DisCOPs), appelé Distributed Dynamic Branch-and-Bound (DisDBnB). Les agents assignent leurs variables d'une manière séquentielle et transmettent en avant leurs nogoods valués. La combinaison des bornes inférieures induites á partir des nogoods valués contribue assurément á l'accélération de la recherche et á l'élimination des sous problèmes irréalisables. Dans cet article, nous montrons que notre algorithme (DisDBnB) est optimal et garanti la terminaison. L'analyse des résultats expérimentaux montrent en comparaison avec l'algorithme Synchronous Branch-and-Bound (SynBnB) sur des problèmes de benchmarck (Max-DisCSPs et coloriage de graphe), que l'algorithme proposé atteint un ordre de performance trés en avance en terme des messages échangés entre les agents.

KEYWORDS : Distributed Constraint Optimization Problems DisCOPs, Valued nogoods recording, Branch-and-Bound search, dynamic backtracking.

MOTS-CLÉS : Problèmes d'Optimisation de Contraintes Distribués DisCOPs, apprentissage par nogoods valués, Branch-and-Bound, backtrack dynamique.

1. Introduction

Distributed Constraint Optimization Problems (DisCOPs) have recently attracted a lot of interest because of their ability to model several real life scenarios where information and control are distributed among a set of different agents. Differently from COP, in DisCOP collaborative agents must find solutions over a distributed set of constraints. In [2, 3, 6] asynchronous complete method (AFB, AFB-bj and ABFS) for distributed constraint optimization are proposed to find the optimal solution, these algorithms are based solely on a simple Branch-and-Bound with additional backjumping and forward checking mechanisms. However, they have been shown to outperform existing distributed protocols as ADOPT [4]. Only one previous recognized distributed optimization protocol ADOPT-ng [11] is known to us whose inference is based on valued nogood notion. The purposes of our ideas are to merge valued nogood learning with Synchronous Branch-and-Bound (SynBnB), to make the pruning mechanism more efficient and to qualify SynBnB algorithm to be extended to more sophisticated optimization algorithms.

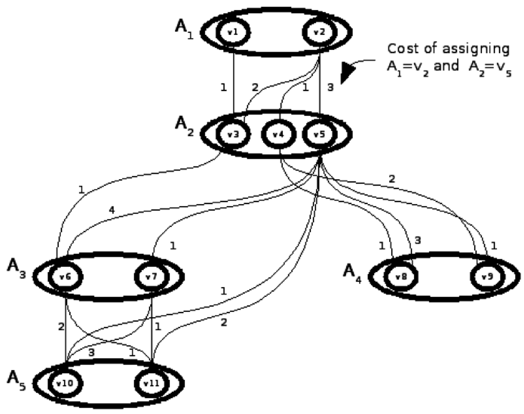


Figure 1. *MultiAgent Constraint graph on DisCOPs*

In this paper, we extend Ginsberg's original algorithm [7, 8] to solve DisCOPs via extension of Valued Dynamic Backtracking [12]. This extension enable memory-bounded, conflict-directed, and optimal search in DisCOPs by utilizing valued nogoods recording in order to quickly prune infeasible and suboptimal regions of the search space.

The rest of the paper is organized as follows: Section tow gives an overview of the preliminary concepts. Section three describes the Distributed Dynamic Branch-and-Bound Search algorithm (DisDBnB). Section four presents an experimental evaluation of the presented algorithm. Comparing DisDBnB and SynBnB on DisChoco plateform [10] show that DisDBnB performs better than SynBnB [15].

2. Preliminary concepts

2.1. Distributed Constraint Optimization problems : DisCOPs

Formally, a DisCOP is composed of a set of n agents $\{A_1, A_2, ..., A_n\}$. a set X of variables: $\{x_1, x_2, ..., x_n\}$, and a set of constraints giving by a set of cost functions $\{c_1,$

$c_2, ..., c_i, ..., c_k\}$, $c_i : X_i \rightarrow \Re^+$, $X_i \subseteq X$, where only agents involved in X_i knows c_i. We assume that ϑ_i is the valuation of the constraint c_i and that x_i can only take values from a domain $D_i = 1, ..., d$, and that each agent contains only one variable, Figure 1.

An assignment is a pair $< A_i, v_i >$, where v_i is a value from x_i's domain that is assigned to it. A partial assignment PA is a set of assignments of values to a set of variables. A global assignment GA, is the selection of one value for each agent (variable) $\in X$. An optimal solution is a global assignment which minimises the total cost.

In this paper, We will consider that constraints are at most binary and the delay in delivering a message is finite [5, 4]. Furthermore, we assume an initial order on the agents, known to all agents participating in the search process.

2.2. Valued nogood concept

Currently, new optimization approaches began to use inference power of valued nogoods [11]. In order to apply nogood-based algorithms, we define the notion of nogoods as follows. First, we attach a value to each nogood obtaining a valued nogood [14].

Definition (Valued Nogood) *[9, 14]: A valued nogood has the form (PA, ϑ, C), and specifies that the (global) problem has cost at least the valuation ϑ, given the set of assignments $PA = \{A_1^{v_1}, ..., A_k^{v_k}\}$ for distinct agents. C is a set of constraints called justification.*

Example : For the graph DisCOPs in Figure 1, a possible valued nogood is $(A_1^{v_2}, 1, C_{12})$, it specifies that if $A_1^{v_2}$ then there exists no solution with a cost lower than 1.

Given a valued nogood $(\{A_1^{v_1}, ..., A_{k-1}^{v_{k-1}}, A_k^{v_k}\}, \vartheta, C)$, one can infer an implication for the value v_k from the domain of A_k given the assignments $\{A_1^{v_1}, ..., A_{k-1}^{v_{k-1}}\}$. This implication is semantically equivalent to an applied valued nogood, (i.e., the inference): $< \{A_1^{v_1}, ..., A_{k-1}^{v_{k-1}}\} > \longrightarrow A_k^{v_k}$ has cost value ϑ with justification C.

Many propositions and properties have been proposed in [9, 14]. In the rest of the paper we consider that all these proposals are accepted, especially : (1) min-resolution, (2) sum-inference and (3) Partial reduction.

3. Distributed Dynamic Branch-and-Bound : DisDBnB

We now present a distributed optimization algorithm whose efficiency is improved by exploiting the increased flexibility brought by the use of valued nogoods. The algorithm can be seen as an extension of a Valued Dynamic Backtracking, and will be denoted Distributed Dynamic Branch-and-Bound (DisDBnB). In this algorithm, we show that a way to achieve correctness is to use at each agent a separate nogood storage for each value and position that the agent holds in the order on assigned agents. To manage this task, data structure will be organised as follow :

– $A_i^{v_i}$: an assignment $< A_i, v_i >$.

– $N_{A_i}^{v_i}$: a valued nogood associated to $A_i^{v_i}$.

– $CCTX$: a current context, it contains three data structures. A list of assignments $PA = \{A_1^{v_1}, ..., A_k^{v_k}\}$, a list of valued nogoods $\{N_{A_1}^{v_1}, ..., N_{A_k}^{v_k}\}$ associated to each instantiated agent and a global valued nogood N_{CCTX} corresponding solely to totally instantiated constraints.

In order to apply a min-conflicts (minimum conflicts) values reordering heuristic, each value in each agent is initialized with a valued nogoods $N^v_{A_i}$, examples are depicted in Figure 2, and formally denoted by :

$$N^v_{A_i} = (A^v_i, h(v), C_v) \tag{1}$$

with $h(v) = \sum_{A_j \prec A_i} mincost(v, u)$ and $u \in D_j$.

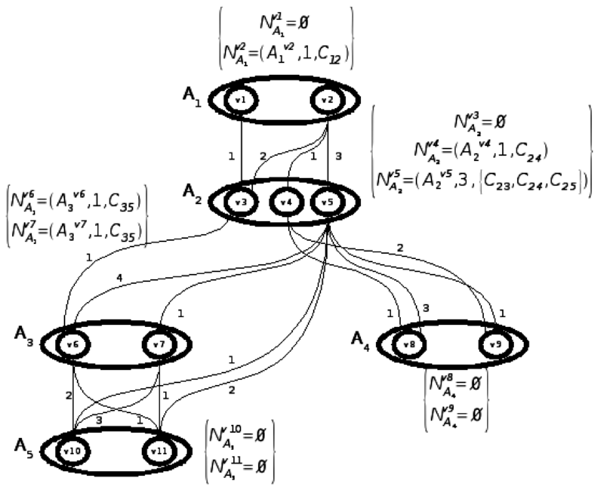


Figure 2. *Initialized valued nogoods.*

The DisDBnB algorithm combines the advantage of assigning values consistent with all former assignments and using future bounds (valued nogoods) memorized and transmitted forward within $CCTX$. Assignments in DisDBnB are performed by one agent at a time. Agents assign their variables only when they hold the current context ($CCTX$). The $CCTX$ is a unique data structure that is passed between agents, using two types of messages.

– $CCTX\_MSG$ message : invites next unassigned agent to assign their variable synchronously, it carries the current context $CCTX$.

– $BACKTRACK\_MSG$ message : announces a valued nogood to the culprit agent inviting him to change his assignment, it carries $CCTX$ and a valued nogood Ng.

3.1. DisDBnB Description

The main procedures and functions of the DisDBnB algorithm are presented in (Algorithm 1 and 2) and perform the following tasks. In the first step *Initialize()* procedure is executed, each agent initializes their valued nogoods (line 4) according to their current position in the order (equation (1)), and awaits for incoming messages. The upper bound B is initialized at max value (infinity). If the procedure is run by the initializing [1] agent (line 5), it initiates the search by generating an empty $CCTX$ data structure, and then calls function *assign_CCTX()* (line 7).

1. initially, agents are totally ordered by priority

An agent receiving a $CCTX$ (When received $CCTX\_MSG$), first updates their valued nogoods (line 9), reorder domain (line 51) according to a Min-Conflicts heuristic, and then calls *assign_CCTX()* (line 10). Function *assign_CCTX()* tries to find an assignment for the agent's local variables, within lower bounds of the valued nogoods $N_{A_i}^{v_i}$ carried in the current $CCTX$ (lines 19-25). First *DisDBnB_estimate()* is called (line 21) for each value of the current agent, and a maximized valued nogood is returned (line 46) [2]. Next, either a value is found (line 23) and included to the $CCTX$ with the associated valued nogoods, or all values are determined to be nogoods. When an agent cannot find a value assignment, *backtrack()* function is called (lines 26-27). If the agent is the last assigned agent, a global assignment GA has been reached, with an accumulated cost lower than the upper bound B, and the cost of the current assignment becomes the new upper bound (line 58).
Function *backtrack()* is called when the agent cannot find a feasible assignment for its variables. However min-resolution is performed and a valued min-resolved nogood is returned (line 38). In the case when the resulting valued nogood is empty (i.e any culprit agent in the conclusion) (line 39), search process is terminated (line 40). Otherwise it's sent back via $BACKTRACK\_MSG$ to the culprit agent that has been chosen according to Ginsberg's Dynamic Backtracking policy [7].
When a $BACKTRACK\_MSG$ is received, current agent remove its inserted data from $CCTX$ using $Remove\_MySelf()$ function (line 12), update their valued nogoods and those stored in the received $CCTX$ (lines 13-14). In order to ensure an increased termination value [12], the rejected agent's value update his associated valued nogood using a partial reduction (line 15) [12] (e.i sum-inference of both Ng and $(N_{CCTX} \cap N_{A_i}^{crt\_v})$ constraints followed bye a reduction of $(N_{CCTX} - N_{A_i}^{crt\_v})$ related constraints). Then *assign_CCTX()* is called (line 16).
When a feasible solution is found then the $CCTX$ is broadcasted to all agents and search is continued (line 29-32).

Theorem 1: *Distributed Dynamic Branch-and-Bound is optimal and terminate.*

Proof. The optimality of DisDBnB is guaranteed by the fact that all used operations (inferences) are logically sound. If DisDBnB terminate, the upper bound B is optimal, so solution is founded. Let us now prove termination, If we consider that the cost of the rejected value of the culprit agent, we can easily see that this value is monotonically increased (partial reduction line 15), assuming that the constaint costs are finite, we deduce that DisDBnB terminate.

4. Experimental Results

We considered two different domains for our experiments. The first was a random Max-DisCSPs in which all constraint costs (weights) are equal to one [4]. The second was a Graph-coloring problem. All experiments were performed on DisChoco plateform [10] in which agents are simulated by threads which communicate only through message passing. We have used a uniform distribution of message delay. The measure of performance used to evaluate presented algorithms communication load, in the form of the total number of messages sent [5]. Each measure presents an average on 100 random problem

2. $\Uparrow^{N_{CCTX}} (N_{A_k}^u)$: denote the sum-inference of both N_{CCTX} and $N_{A_k}^u$.

Algorithm 1 Distributed Dynamic Branch-and-Bound pseudo-code (1/2)

```
Procedure initialize()
B ← ∞
foreach v ∈ D_i do
    N^v_{A_i} ←< A^v_i, h(v), C_v >
if A_i is the initializer then
    create_CCTX()
    assign_CCTX()

when received (CCTX_MSG, CCTX) do
    Update_MyNogoods()
    Assign_CCTX()

when received (BACKTRACK_MSG, CCTX, Ng) do
    CCTX ← Remove_MySelf(CCTX)
    Update_MyNogoods()
    Update_CCTX()
    N^{crt_v}_{A_i} ←⇓_{(N_{CCTX}, N^{crt_v}_{A_i})} (Ng)
    Assign_CCTX()

Procedure Assign_CCTX()
v ← empty
while D_i has not fully explored AND v is empty do
    v ← Choose_MyValue()
    tempN^v_{A_i} ← DisDBnB_estimate(CCTX, v)
    if Valuation(tempN^v_{A_i}) ≺ B then
        CCTX ← CCTX ∪ ≺ A^v_i, N^v_{A_i} ≻
    else
        v ← empty
if v is empty then
    Backtrack()
else
    if CCTX is a full assignment then
        Broadcast(SOLUTION, CCTX)
        B ← Valuation(N_{CCTX})
        Assign_CCTX()
    else
        A_k ← Choose_NextAgent()
        Send(CCTX_MSG, CCTX) to A_k
```

instances. A random binary Max-DisCSPs generator is characterized by ($\#n$, $\#d$, $p1$, $p2$) where $\#n$ is the number of agents/variables, $\#d$ the number of values in each variable domain, $p1$ the probability of a constraint among any pair of variables and $p2$ the tightness values (probability for the occurrence of a violation (a non zero cost) among two assignments of values to a constrained pair of variables). Figure 3-(a) presents the average of a total number of messages sent for a graph 3-coloring problem. We illustrate that DisDBnB outperform SynBnB. The difference between performances DisDBnB and SynBnB increase when the number of agents increase. Figure 3-(b) presents the number average of total number of messages sent for a random Max-DisCSP with $p1 = 0.4$. We show that communication load of DisDBnB algorithm is smaller than the SynBnB's one.

Algorithm 2 Distributed Dynamic Branch-and-Bound pseudo-code (2/2)

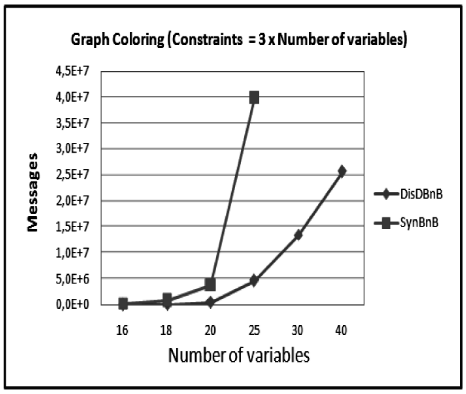
Procedure Backtrack()
$Restore\_MyDomain()$
$N \leftarrow \underset{u \, in \, D_i}{min\_resolution}(DisDBnB\_estimate(CCTX, u))$
if N *is empty* **then**
  $Broadcast(TERMINATE)$
else
  choose A_j from N such as $\forall A_k \in N,\ A_k \succ A_j$
  $Send(BACKTRACK\_MSG, CCTX, Ng)\, to\, A_j$

Function DisDBnB_estimate(CCTX, v)
$tempCCTX \leftarrow CCTX \cup \prec A_i^v, N_{A_i}^v \succ$
$return \underset{N_{A_k}^u \, in \, tempCCTX}{maximize}(\Uparrow^{N_{CCTX}} (N_{A_k}^u))$

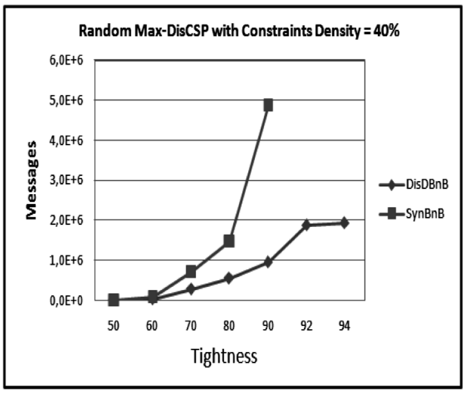
Function Update_MyNogoods()
foreach $v \in D_i$ **do**
  if $N_{A_i}^v$ *is not compatible with* PA **then**
    $N_{A_i}^v \leftarrow < A_i^v, h(v), C_v >$
Reorder domain according to Min-Conflicts

Function Update_CCTX()
foreach $N_{A_i}^v \in CCTX$ **do**
  if $N_{A_i}^v$ *contains* A_i **then**
    $N_{A_i}^v \leftarrow \emptyset$

when ***received (SOLUTION**, CCTX)* **do**
  $GA \leftarrow CCTX.PA$
  $B \leftarrow Valuation(N_{CTTX})$



(a)



(b)

Figure 3. *(a): Number of messages sent by DisDBnB and SynBnB for a distributed graph coloring with 3 colors, (b) Number of messages sent by DisDBnB and SynBnB for a Max-DisCSP with* $p1 = 0.4$ *.*

5. Conclusions and future Work

We have proposed DisDBnB, a novel synchronous protocol that is an extension of centralized valued dynamic backtracking algorithm [12]. DisDBnB uses a new distributed search strategy based on valued nogoods recording and dynamic backtracking. These techniques produce lower bounds inferred from valued nogoods, that allow portions of the search space to be pruned. We have shown through experimental analysis on random Max-DisCSPs and distributed graph coloring problems that DisDBnB offer an order of magnitude speed-up. One promising direction for future work is to consider that an asynchronous optimized algorithm based on Distributed Dynamic Branch-and-Bound search (DisDBnB) could potentially improve performance in contrast with recent algorithms who are based naively on Synchronous Branch-and-Bound search (SynBnB).

6. References

[1] P. J. MODI, W. M. SHEN, AND M. TAMBE, "An asynchronous complete method for distributed constraint optimization.", *In Proc. AAMAS*, 2003,

[2] A. GERSHMAN, A. MEISELS, AND R. ZIVAN, "Asynchronous forward-bounding for distributed constraints optimization", *In Proc. ECAI-06, pages 103-108, Riva Del Garda, Italy*, August 2006,

[3] A. GERSHMAN, A. MEISELS, AND R. ZIVAN, "Asynchronous Forward-Bounding with Backjumping.", *In IJCAI 07 proceding, 28–39*, January 2007,

[4] P. J. MODI, W. SHEN, M. TAMBE, AND M. YOKOO, "Adopt: asynchronous distributed constraints optimization with quality guarantees", *Artificial Intelligence*, vol. 161:1-2:149-180 January 2005.

[5] M. YOKOO, "Algorithms for distributed constraint satisfaction problems A review. Autonomous Agents", *Multi-Agent Sys*, vol. 3:198-212, 2000.

[6] R. EZZAHIR, C. BESSIERE, E. H. BOUYAKHF, I. BENELALLAM, AND M. BELAISSAOUI, "Asynchronous Breadth-First Search DisCOP Algorithm", *In EUMAS 07 proceding*, 2007,

[7] M. L. GINSBERG, "Dynamic backtracking", *Journal of AI Research*, vol. 1, 1993.

[8] M. L. GINSBERG, AND D. MCALLESTER, "Gsat and dynamic backtracking", *In J.Doyle, editor, Proceedings of the 4th IC on PKRR, pages 226-237. KR*, 1994,

[9] P. DAGO, AND G. VERFAILLIE, "Nogood recording for valued constraint satisfaction problems", *In ICTAI, pages 132-139*, 1996,

[10] C. BESSIERE, R. EZZAHIR, M. BELAISSAOUI, AND E.H. BOUYAKHF, "DisChoco: A platform for distributed constraint programming", *In Proceeding of Workshop DCR, IJCAI Hyderabad, India*, 2007,

[11] M. C. SILAGHI, M. YOKOO, "Nogood-based asynchronous distributed optimization (ADOPT-ng)", *In AAMAS, pages 1389-1396*, 2006,

[12] P. DAGO, "Backtrack dynamique valué", *In JFPLC, pages 133-148*, 1997,

[13] M. C. SILAGHI, J. LANDWEHR, AND J. B. LARROSA, "Asynchronous Branch and Bound and A* for DisWCSPs with heuristic function based on Consistency-Maintenance", *volume 112 of Frontiers in Artificial Intelligence and Applications, chapter. IOS Press*, 2004,

[14] M. C. SILAGHI, "Dynamic Branch-and-Bound, an optimization counterpart for dynamic backtracking", *Report*, 2006,

[15] K. HIRAYAMA, AND M. YOKOO, "Distributed partial constraint satisfaction problem", *In Proceedings of the Conference on Constraint Processing (CP-97), pages 222-236,* 1997,