

Towards Efficient Multi-Agent Abduction Protocols

Gauvain Bourgne
National Institute of Informatics
Tokyo, Japan
Email: bourgne@nii.ac.jp

Katsumi Inoue
National Institute of Informatics
Tokyo, Japan
Email: ki@nii.ac.jp

Nicolas Maudet
LAMSADE,
Paris Dauphine University, France
Email: nicolas.maudet@lamsade.dauphine.fr

Abstract—What happens when distributed sources of information (agents) hold and acquire information locally, and have to communicate with neighbouring agents in order to refine their hypothesis regarding the actual global state of this environment? This question occurs when it is not be possible (*e. g.* for practical or privacy concerns) to collect observations and knowledge, and centrally compute the resulting theory. In this paper, we assume that agents are equipped with full clausal theories and individually face abductive tasks, in a globally consistent environment. We adopt a learner/critic approach. We present the Multi-agent Abductive Reasoning System (MARS), a protocol guaranteeing convergence to a situation “sufficiently” satisfying as far as consistency of the system is concerned. Abduction in a full clausal theory has however already a high computational cost in centralized settings, which can become much worse with arbitrary distributions. We thus discuss ways to use knowledge about each agent’s theory language to improve efficiency. We then present some first experimental results to assess the impact of those refinements.

I. INTRODUCTION

In multi-agent systems, the inherent distribution of autonomous entities, perceiving and acting locally, is the source of many challenging questions. To overcome the limitation of their own knowledge, usually local and incomplete, agents are driven to form some hypotheses and share information with other agents. Especially, abductive reasoning is a form of hypothetical reasoning deriving the possible causes of an observation. It can be used to complete an agent’s understanding of its environment by explaining its observations, or, more proactively, for planning, as one can try to find the possible actions that might cause the completion of a goal. However reasoning in a sound manner with distributed knowledge rises interesting problems, as one cannot ensure locally the consistency of an information. Moreover, the system often comes with severe communication restrictions, due to physical (*e. g.* the limited scope of a communication device) or reasoning (*e. g.* the mere impossibility to consider all the potential communications) limitations of agents populating it. For such situations, we presented in [1] a sound mechanism that is guaranteed to find an abductive hypotheses with respect to distributed full clausal theories whenever one exists. This Multi-agent Abductive Reasoning System, MARS, is based on a consequence finding tools named SOLAR [2], that serves as a main reasoning engine. We are concerned in this paper with the efficiency of this mechanism, and thus want to evaluate and improve its average computational and communicational cost.

Distributed abduction has been considered in recent years in the ALIAS system [3]. They distribute the abductive programming algorithm of [4], using abductive logic program to represent each agent’s theory. More recently, DARE [5] addressed a similar problem, but consider possible dynamicity of the system by allowing agents to enter or exit some proof cluster. In none of these works however is the issue of communication constraints explicitly raised. Another related work is the peer-to-peer consequence finding algorithm DeCA [6]. Based on a different method (splitting clauses), it is to our knowledge the only other work in this domain taking into account restrictions of communication between peers. It is however restricted to propositional theories. The work on partition-based logical reasoning presented [7] is of particular interest for our present study as it investigates efficient theorem proving in partitioned theories. It relies on communication languages describing the common symbol in the individual languages of pairs of agents. However, this approach and the previous one explore all the consequences of the distributed theories, whereas when we are only concerned with some *new* consequences of the theories with respect to some knowledge (namely the negated observations when computing a hypothesis through inverse entailment, or the hypothesis itself when ensuring its consistency). As a result, while inspirational, they cannot be directly applied to our approach.

The rest of this paper is as follows. Section II gives the necessary background on abduction and consequence finding. Then, Section III describe formally a multi-agent abduction problem, and present the MARS protocol, giving details about the communications exchanged over its execution. Efficiency is then discussed, and we describe two improvement on the previous protocol. These variants are then experimentally tested in Section IV, and we conclude in Section V.

II. ABDUCTIVE REASONING

A. Preliminaries

First, we review some notions and terminology to represent our problem in a logical setting. A *literal* is an atom or the negation of an atom. A *clause* is a disjunction of literals, and is often denoted by the set of literals. A clause $\{A_1, \dots, A_m, \neg B_1, \dots, \neg B_n\}$, where A_i and $\neg B_j$ are respectively positive and negative atoms is also written as $A_1 \vee \dots \vee A_m \leftarrow B_1 \wedge \dots \wedge B_n$. Any variable in a clause is assumed to be universally quantified at the front. A *clausal theory* is a finite set of clauses which can be identified with the

conjunction of the clauses. Let S and T be clausal theories. S *logically implies* T , denoted as $S \models T$, if and only if for every interpretation I such that S is true under I , T is also true under I . \models is called the *entailment relation*. For a clausal theory T , a *consequence* of T is a clause entailed by T . We denote by $Th(T)$ the set of all consequences of T . Let C and D be two clauses. C *subsumes* D , denoted $C \succeq D$, if there is a substitution θ such that $C\theta \subseteq D$. C *properly subsumes* D if $C \succeq D$ but $D \not\subseteq C$. For a clausal theory T , μT denotes the set of clauses in T not properly subsumed by any clause in T .

We can now introduce the notion of *characteristic clauses*, which represents “interesting” consequences of a given problem [8]. Each characteristic clause is constructed over a sub-vocabulary of the representation language called a *production field*, and represented as $\langle \mathcal{L} \rangle$, where \mathcal{L} is a set of literal closed under instantiation. A clause C *belongs* to $\mathcal{P} = \langle \mathcal{L} \rangle$ if every literal in C belongs to \mathcal{L} . For a clausal theory T , the set of consequences of T belonging to \mathcal{P} is denoted $Th_{\mathcal{P}}(T)$. Then, the characteristic clauses of T wrt to \mathcal{P} are defined as $Carc(T, \mathcal{P}) = \mu Th_{\mathcal{P}}(T)$, where μ is subsumption minimality¹. When a set of new clauses S is added to a clausal theory, some consequences are newly derived with this additional information. The set of such clauses that belong to the production field are called *new characteristic clauses* of S wrt T and \mathcal{P} ; they are defined as $Newcarc(T, S, \mathcal{P}) = Carc(T \cup S, \mathcal{P}) \setminus Carc(T, \mathcal{P})$.

B. Abductive hypothesis

The logical framework of hypothesis generation in abduction for the centralized case can be expressed as follows. Let T be a clausal theory, which represents the *background theory*, and O be a set of literals, which represents *observations*. Also let \mathcal{A} be a set of literals representing the set of *abducibles*, which are candidate assumptions to be added to T for explaining O . Given T , O and \mathcal{A} , the abduction problem is to find a *hypothesis* H such that:

- (i) $T \cup H \models O$ (accountability),
- (ii) $T \cup H \not\models \perp$ (consistency), and
- (iii) H is a set of instances of literals from \mathcal{A} (bias).

In this case, H is also called an *explanation* of O (with respects to T and \mathcal{A}). A hypothesis is *minimal* if no proposer subset of H satisfies the above three conditions (which is equivalent to subsumption minimality for ground clauses). A hypothesis is *ground* if it is a set of ground literals (literals containing no variable). This restriction is often employed in applications whose observations are also given as ground literals. In the following, we shall indeed assume that observations are grounded, and that we are only searching for minimal ground hypotheses.

C. Computation through hypothesis finding

Given the observations O , each hypothesis H of O can be computed by the principle of *inverse entailment* [8], [9],

¹meaning that μX represents the clauses of X that are not properly subsumed by any other clause of X .

which converts the accountability condition (i) to $T \cup \{\neg O\} \models \neg H$, where $\neg O = \bigvee_{L \in O} \neg L$ and $\neg H = \bigvee_{L \in H} \neg L$. Note that both $\neg O$ and $\neg H$ are clauses since O and H are sets of literals. Similarly, consistency condition (ii) is equivalent to $T \not\models \neg H$. Hence, for any hypothesis H , its negated form $\neg H$ is deductively obtained as a “new” theorem of $T \cup \{\neg O\}$ that is not an “old” theorem of T alone. Moreover, to respect the bias condition (iii), every literal of $\neg H$ has to be an instance of a literal in $\bar{\mathcal{A}} = \{\neg L \mid L \in \mathcal{A}\}$. Then the negation of minimal hypotheses are the new characteristic clauses of O with respect to T and $\bar{\mathcal{A}}$, that is, $Newcarc(T, \{\neg O\}, \bar{\mathcal{A}})$.

SOLAR [2] is a sophisticated deductive reasoning system based on SOL-resolution [8], which is sound and complete for finding *minimal* consequences belonging to a given language bias (a *production field*). Consequence-finding by SOLAR is performed by *skipping* literals belonging to a production field \mathcal{P} instead of resolving them. Those skipped literals are then collected at the end of a proof, which constitute a clause as a logical consequence of the axiom set. Using SOLAR, we can implement an abductive system that is *complete* for finding minimal explanations due to the completeness of consequence-finding. SOLAR is designed for *full clausal theories* containing non-Horn clauses, and is based on a *connection tableau* format [10]. In this format, many redundant deductions are avoided using various state-of-the-art pruning techniques [2], thereby hypothesis-finding is efficiently realized.

Once possible hypotheses have been computed, a ranking process can be applied to select a *preferred hypothesis* (e.g. hypothesis ranking such as in [11]). We will not dwell on this part here, and instead assumed that a *preference relation* \succeq_p over the hypothesis is given as a total order between sets of grounded literals.

III. DISTRIBUTED ABDUCTION

A. Problem setting

We propose here a new formalization of our problem as a *multi-agent abductive system*, which is defined as a tuple $\langle \mathcal{S}, \{\Gamma_t\}, \mathcal{A}, \succeq_p \rangle$, where:

- $\mathcal{S} = \{a_0, \dots, a_{n-1}\}$ is a set of agents. Each agent a_i has its own *individual theory* \mathcal{T}_i and its own *observations* O_i . It will also form its own preferred hypothesis H_i , though it can also adopt it from other agents. In fact, in the end of the process, all agents will share the same hypothesis.
- $\Gamma_t = \langle \mathcal{S}, E_t \rangle$ is the *communicational constraint graph* at time t , an undirected unlabeled graph whose nodes are the agents in \mathcal{S} and whose edges E_t represent the communicational links between the agent. An agent a_i can only communicate with another agent a_j at time t if $(a_i, a_j) \in E_t$.
- \mathcal{A} is the common set of *abducibles* that represents the language bias of the abductive process.
- \succeq_p is the *common preference relation*, a total order over hypotheses.

Theories and observations are considered to be certain knowledge. As such, they are assumed to be consistent,

meaning that $\bigcup_{i < n} \mathcal{T}_i \cup \bigcup_{i < n} \mathcal{O}_i \not\models \perp$. To ensure termination, it will also be assumed that $\text{Carc}(\bigcup_{i < n} \mathcal{T}_i, \langle \mathcal{L} \rangle)$ is finite, and that both hypotheses and observations are ground (*i.e.* contain no variable). Moreover, the system will be assumed to be *temporally connected*, meaning that at any time t , the graph $\Gamma_{t^+} = \langle \mathcal{S}, \bigcup_{t' > t} E_{t'} \rangle$ is a connected graph.

Our aim is then to ensure the formation of an abductive explanation of $\bigcup_{i < n} \mathcal{O}_i$ with respects to $\bigcup_{i < n} \mathcal{T}_i$ and \mathcal{A} . Given a group of agents $G = \{a_i, i \in J\} \subset \mathcal{S}$, we shall say that a hypothesis H is *group-consistent* with G iff it is consistent with the union of all the individual theory of the agents of the group, that is, iff $\bigcup_{i \in J} \mathcal{T}_i \cup H \not\models \perp$. Likewise, we shall say that H ensures *group-accountability* for G iff it can explain all observations of the agents of the group when it is associated with the union of their theories, that is iff $\bigcup_{i \in J} \mathcal{T}_i \cup H \models \bigcup_{i \in J} \mathcal{O}_i$. If $G = \mathcal{S}$, we shall say that the hypothesis is *mas-consistent* or that it ensures *mas-accountability*. Finally we shall say that a set of literals is *acceptable* for a group G iff it is a set of grounded literals of \mathcal{A} that is group-consistent with G and ensures group-accountability for G . The objective of a multi-agent abductive system is thus to find a hypothesis that is *acceptable* for the whole system.

While consistency or accountability of a hypothesis with respect to both $(\mathcal{T}_i, \mathcal{O}_i)$ and $(\mathcal{T}_j, \mathcal{O}_j)$ is not equivalent to consistency or accountability wrt $(\mathcal{T}_i \cup \mathcal{T}_j, \mathcal{O}_i \cup \mathcal{O}_j)$, we still can ensure some relation between them in classical logic. Specifically, group-inconsistency of H with G implies group-inconsistency of H with any superset of G , which ensures that hypothesis inconsistent with a sub-group of agents (possibly a single agent) can be ruled out as a potential solution. Moreover, group-accountability of H for both G and G' implies group-accountability of H for $G \cup G'$ (but not reciprocally), which ensures that accountability can be checked locally.

In order for a learner agent to propose a hypothesis to a critic, it is necessary that his agent can produce such a hypothesis. However, given only a few clauses of the whole clausal theory, it might not be able to find an explanation for the observations using only abducibles. Therefore, we shall allow an agent to build *partial hypothesis*, which contains some non-abducible literals. Those literals might be the unexplained observations, or preferably some other literals of the language that would explain it. While interacting with other agents, they will share knowledge to expand these hypotheses in order to progressively build a fully abducible one. Note that of course, a hypothesis respecting the bias condition will always be favored over one who does not.

We shall now present MARS, a mechanism for solving multi-agent abductive problems based on SOLAR.

B. Bilateral interaction

To deal with distributed hypothesis formation in multi-agent systems, we take a learner-critic approach, in which learner agents aim at producing a globally adequate hypothesis through internal computations and local interactions with other agents acting as critics. In our abductive setting, however, critic agent cannot ensure the consistency of the hypothesis

by itself, and needs to interact with the learner in order to find incoherence (computing the context of a hypothesis) and produce complete hypotheses (exchanging useful information by justifying partial hypotheses). The underlying mechanism was presented and proved correct in [1]. Here, we shall introduce the actual protocol based on that procedure, recapitulating its main steps while giving an exact account of the communications involved.

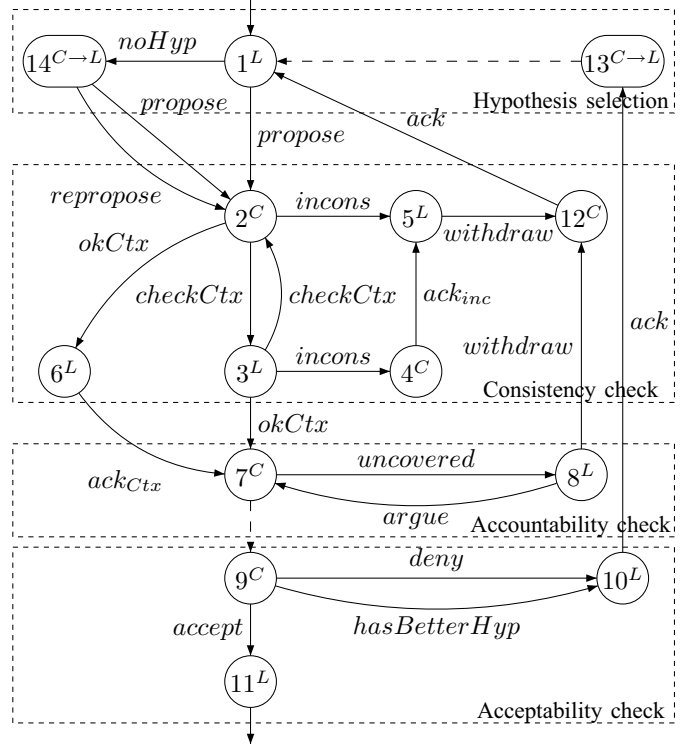


Fig. 1. Multi-Agent Learner-Critic Abductive Protocol.

Fig. 1 illustrate this protocol. Nodes indicate states of the agents (steps of the mechanism), with superscript L or C indicating whether it concerns the *Learner* agent or the *Critic* agent. Note that states 13 and 14 indicate a switching of the roles, as the critic becomes the learner. Labeled arcs indicate that a given message can be sent by an agent in a given state, making the other agent go to the target state upon reception. Dashed arcs indicate an internal change of state without communication. This mechanism is divided in four main steps that we shall now detail.

1) *Hypothesis selection*: An interaction is initiated by a learner agent a_0 , in state 1^L , proposing its hypothesis and its validity context to a critic agent a_1 ($\text{propose}(H_0)$). If learner's information has changed since it last computed its possible hypotheses, it will recompute them through inverse entailment, using $\bar{\mathcal{A}}$ as a production field. In case it cannot find a hypothesis this way, it will compute a *partial hypothesis* by using an *extended set of abducibles* (possibly the whole language). If the proposed hypothesis h_0 is a new one, the first context Ctx_0 will be computed as the new consequences of

$h_0 \cup \mathcal{T}_0$ wrt h_0 , that is $Newcarc(\mathcal{T}_0, h_0, \mathcal{P}_{\mathcal{L}})$ where $\mathcal{P}_{\mathcal{L}} = \langle \mathcal{L} \rangle$. Otherwise, the previously computed context will be used as initial context Ctx_0 . Then, when receiving such proposal, a_1 will start its critic, which consists of three steps: consistency check, accountability check and admissibility check. As the interaction continues, new hypotheses might have to be proposed. If the current learner cannot propose a hypothesis (which can only happen if it has blocked all its possible hypotheses during previous admissibility check), it will send *noHyp* to the other agent to switch the roles (state 14). If this one has also exhausted all its hypotheses, then it will unblock all its hypotheses and propose again the best one (*repropose*). Note that the new critic agent will also unblock all its hypotheses when receiving such a message.

2) *Consistency check*: When receiving a proposed hypothesis and context, the first step of the critique is to check the group-consistency of the hypothesis with both agents involved. A context is progressively built for a given hypothesis H to compute the new consequences of $H \cup \mathcal{T}_0 \cup \mathcal{T}_1$ wrt to $\mathcal{T}_0 \cup \mathcal{T}_1$. If the hypothesis is incoherent, then it will have \perp as a consequence. The occurrence of a contradiction between the context and the agents' theories will thus enable detection of incoherent hypotheses. This relies on the fact that the global theory itself is assumed to be consistent, so any inconsistency can only arise from the hypotheses. Indeed, if \mathcal{T} is consistent and $\mathcal{T} \cup H$ is inconsistent, then $Newcarc(\mathcal{T}, U, \mathcal{P}_{\mathcal{L}}) = \{\perp\}$. The process is as follow :

1. First, remember that during the hypothesis selection step, learner agent a_0 retrieve context Ctx_0 of its hypothesis. If no context have been memorized from previous iteration, then a new one is computed as $Newcarc(\mathcal{T}_0, H_0, \mathcal{P}_{\mathcal{L}})$ (note that we should thus have $H_0 \in Ctx_0$).

2. When receiving H_0 and Ctx_0 (state 2^C), a_1 first check if it already has some context Ctx' for this given hypothesis. If it is the case, then it replaces Ctx_0 by $Ctx'_0 = Ctx_0 \cup Ctx'$. Context Ctx_1 is then computed as $Newcarc(\mathcal{T}_1, Ctx'_0, \mathcal{P}_{\mathcal{L}})$, and sent back to a_0 with message *CheckCtx*(Ctx_1) (unless a contradiction is found).

3. The process continues. At each step Ctx_i is computed by agent a_α as the new consequences $Newcarc(\mathcal{T}_\alpha, Ctx_{i-1}, \mathcal{P}_{\mathcal{L}})$, where $\alpha = 0$ if i is odd (state 3^L), and $\alpha = 1$ otherwise (state 2^C), and sent with a *checkCtx* message.

4. This computation stops when either a contradiction is found or Ctx_i is included in either Ctx_{i-1} or Ctx_{i-2} , in which case all consequence have been computed.

- If an inconsistency is discovered, the part p_0 of the hypothesis responsible for it is sent to the other agents with message *incons*(p_0). leading eventually to state 5^L . Both agents rule out p_0 (and any hypothesis containing it) by adding its negation to their theory. The learner agent then move on to its next hypothesis and propose it, triggering a new critic phase (states 12^C and 1^L).
- Otherwise, the end of the computation is acknowledged by sending *okCtx*. Both agents memorize the final context $Ctx_f = Ctx_i \cap Ctx_{i-1}$ (where i is the final

step) of this hypothesis. Any element in respectively $Ctx_i \setminus Ctx_{i-1}$ and $Ctx_{i-2} \setminus Ctx_{i-1}$ are added to $\mathcal{T}_{1-\alpha}$ and \mathcal{T}_α where again $\alpha = 0$ if i is odd and 1 otherwise. Indeed an element will only be removed from the context if it is a direct consequence of one of the agent's theory. The critic phase move to the next step (state 7^C).

3) *Accountability check*: In this step, the critic agent checks if all its observations are explained by $H_0 \cup \mathcal{T}_1$. If an unexplained observation o is found, the message *uncovered*(o) is sent to the learner agent, now in state 8^L . We then have two possibilities.

- If o is not explained by $H_0 \cup \mathcal{T}_0$, it is a true counter-example. The learner agent then computes a new hypothesis that will also cover o , and propose it, triggering a new critic phase (states 12^C and 1^L).
- If o is already explained by $H_0 \cup \mathcal{T}_0$, then the learner agent will notify the critic of this fact with *argue*(p_0), where p_0 is the part of the hypothesis that is used in explaining o with \mathcal{T}_0 . The critic agent will add the clause $\{o \vee \neg p_0\}$ in its theory². This new information will ensure that the critic agent can find the hypothesis on its own in further steps, or build up upon it. It will then proceed to the next unexplained observation.

If there is no unexplained observation, the critic proceeds to the next step (state 9^C).

4) *Acceptability check*: Any hypothesis that reaches this step is consistent and accounts for the observations, but it might include some non-abducible literals, or unnecessary parts. This step ensures that alternative hypotheses are explored if needed.

1. If the critic has a hypothesis H_c that is preferred to H_0 (according to \geq_p), it will reverse roles (*hasBetterHyp*) and submit it. This will finally either result in the acceptance of a better hypothesis, or cause the former critic agent to learn why its hypothesis cannot be used.

2. Otherwise, if the hypothesis contains non-abducibles (partial hypothesis), the critic agent will temporarily block it, and ask the other agent to do the same (*deny*). I will then also switch roles (state $13^{C \rightarrow L}$). This ensures that all partial hypotheses that could provoke information exchange leading to building an abducible hypothesis are explored if needed.

3. If the hypothesis is acceptable, or if a partial hypothesis has been reproposed (meaning the exploration is complete), then the critic send an *accept* message. The final outcome of the interaction is thus chosen. Hypotheses that were temporarily blocked are unblocked, and the best hypothesis is chosen as the final hypothesis. It is adopted and memorized by both agents, ending the interaction.

C. Group of agents

Each interaction allows the participants to refine their hypotheses and augment their knowledge concerning their consequences. The protocol described before is enough to allow two agents to form a hypothesis that is group-consistent and

²Note that since p is a conjunction of literals, $\neg p_0$ is indeed a clause.

ensures group-accountability for the pair of agents. When more agents are involved, it is possible to chain such interactions to converge towards a consistent state of the system. To take into account possibly variable communication constraints in the system, we propose a rumor-like approach, ensuring the local behaviour and interactions of the agents make the system converges to a state in which all agents have a mas-consistent hypothesis ensuring mas-accountability.

An agent is motivated by the will to ensure it has an explanation with respect to its neighbours. As such, it will attempt to have local interactions with them whenever needed to ensure that, memorizing the result of their last interaction with each of their neighbours. In practice, an agent a_i will engage in a local interaction with a neighbour a_j whenever its hypothesis and context (h_i, Ctx^i) differ from those obtained during its last interaction with a_j .

In [1], this process was proved to be sound, and to guarantee that a solution is found if there is one.

D. Improving efficiency

The main computational cost of our mechanism lies in the multiple calls to a consequence-finding tools, which is used in the various steps to conduct the logical reasoning, especially for computing possible hypotheses through inverse entailment, computing the context of these hypotheses, and checking their accountability. To improve efficiency, it is thus crucial to reduce as much as possible the computational cost of each of these calls, as well as to reduce their number.

The tools we are using in our implementation, SOLAR, is based on tableaux methods. We assess the computational cost of a call by counting the number of inferences performed during it. Without entering in the details of the procedure, we will discuss here the factors that influence the cost of the computation of $NewcArc(F, T, \mathcal{P}_L)$. The number of inferences is directly related to the number of clauses used in the procedure. Used clausal are clauses which can be resolved with one of the top clauses (elements of F), or with a consequence of them. Thus, reducing the number of clauses in T and more importantly in F can both help to reduce the computations steps. Note that $Carc(T, \mathcal{P}_L)$ is in practice computed as $NewcArc(T, \emptyset, \mathcal{P}_L)$, so computing new consequences rather than all consequences is already a good step to ensure better efficiency. Reducing the number of literals of the top clauses also helps as it limits the number of clauses they can be resolved with. Then, another factor that can affect the computations is the size of the production field. A small production field limits the number of options to be explored and as a results, the number of inferences to be done.

With respects to our mechanism, these considerations means that we should keep each agent's individual theory as small as possible, which is ensured by adding single clauses with just the necessary parts of the hypothesis to memorize inconsistencies (when sending or receiving $incons.(p_0)$) or accountability arguments (when sending $argue(o \vee \neg p_0)$ in state 8^L). Moreover, during consistency check, we should minimize the computations for the context. We shall see in next subsection

how to reduce size of top clauses during this step by doing incremental computations. Then, we should also find ways to minimize the number of consequences computed during this consistency step by focusing on consequences that could lead to a contradiction, and even more importantly, to limit the number of partial hypotheses computed by focusing on those partial hypotheses that could trigger information exchanges leading to the formation of an acceptable hypothesis (as it would also reduce the number of applications of SOLAR).

E. Incremental consistency check

During consistency check, context is progressively computed until it does not evolve anymore, but sending the whole context at each step of the computation and using it as top clause for computing the next step. To avoid redundant communications and computations, we propose to communicate only the new consequences of the context, pruning consequence discovered in previous step. It does not change the computation and sending of Ctx_0 and Ctx_1 , but after computing Ctx_1 , the learner agent will only send back $ctxStep_1 = Ctx_1 \setminus Ctx_0$ (note that we use the original Ctx_0 here, and not Ctx'_0).

Then, when receiving $checkContext(ctxStep_i)$, agent a_α first computes $NC_{i+1} = NewCarc(ctxStep_i, \mathcal{T}_\alpha \cup Ctx_{i-1})$. It can then use it to compute the current context $Ctx_{i+1} = Ctx_{i-1} \cup NC_{i+1}$, and send the update $ctxStep_{i+1} = NC_{i+1} \setminus ctxStep_i$. If there is a clause c than is in $ctxStep_i$ that is not subsumed by any clause of NC_{i+1} , it means that it is a consequence of \mathcal{T}_α . It should then be sent to the other agent (with message $inform(c)$) to ensure that both agents will have the same final context. This replaces the theory adjustment with $Ctx_i \setminus Ctx_{i-1}$ and $Ctx_{i-2} \setminus Ctx_{i-1}$ that were made before. Note that the termination condition becomes much simpler, as the context can be confirmed as soon as $ctxStep_i$ is empty.

F. Language focus

1) *Languages*: Given a clausal theory T , we denote by $L(T)$ the set of non-logical symbols that occur in T , and by $\mathcal{L}(T)$ the language formed upon them. Each agent has its own theory \mathcal{T}_i , from which we can define its *individual language* $\mathcal{L}(\mathcal{T}_i)$. We can then compute for each pair of agent a_i, a_j ($i \neq j$), in the manner of [7], the *communications language* $\mathcal{L}_{i,j} = \mathcal{L}(\mathcal{T}_i) \cap \mathcal{L}(\mathcal{T}_j)$. It can be used to direct the focus of bilateral communications. If it is empty, a_i and a_j do not need to communicate together. However, it may be the case that a_i and a_j are never connected while $\mathcal{L}_{i,j}$ is not empty. For the sake of simplicity we will assume that the communicational links are static ³. We shall then adapt the communications languages by choosing a minimal path for all such pair of unconnected agents (a_i, a_j) , and add the $\mathcal{L}_{i,j}$ to the communication language of each pair of agent in this path. In the following, when referring to the communication

³otherwise, since the system is assumed to be temporally connected, it is possible to find a connected subgraph that is included in Γ_{t+} for all t and use it as a guaranteed basis.

language $\mathcal{L}_{i,j}$, we will assume that this modification has been done. Then the *restricted individual language* of an agent a_i is defined as $\mathcal{L}_i = \bigcup_{j \in N_i} \mathcal{L}_{i,j}$, where N_i is the set of the indexes of the neighbours of a_i . At last, the *common language* is the language $\mathcal{C} = \bigcup_{i < n} \mathcal{L}_i$, that is the union of all restricted individual languages (which is also the union of all the communication languages).

2) *Context narrowing*: When computing context, we want to ensure that any new consequence of the hypothesis that can be derived from the union of the agent’s theories is indeed found. It means a_0 need to send any consequence of H wrt \mathcal{T}_0 that could resolved with a clause of \mathcal{T}_1 . In practice, a_0 compute its context using its *restricted individual language* as a production field, and then retain in Ctx_0 only the one that contains at least a literal of the concerned communication language (here $\mathcal{L}_{0,1}$). Upon receiving it, a_1 will then temporarily add $\mathcal{L}(Ctx_0)$ to $\mathcal{L}_{0,1}$, and if it already has a context Ctx' , a_1 will use the same pruning before adding it to get Ctx'_0 and compute Ctx_1 (with his restricted individual language as a production field). The pruning (with updated language) is applied to Ctx_1 (or $ctxStep_1$) before sending it, and the process continue. Each time, contexts are pruned to exclude any clauses that do not have literals in the current communication language before being sent to the other agent. Note that since we compute only new consequences, we cannot directly use the communication language as a production field, as shown by the following example:

Example 1: Let’s take $\mathcal{T}_0 = \{\neg h \vee a \vee b, \neg h \vee o\}$, $\mathcal{T}_1 = \{\neg a\}$ and $\mathcal{T}_2 = \{\neg b\}$, all agents being connected. We have $\mathcal{L}_{0,1} = \mathcal{L}(\{a\})$, and $\mathcal{L}_{0,2} = \mathcal{L}(\{b\})$, so $\mathcal{L}_0 = \mathcal{L}(\{a, b\})$. We assume a_0 has observation o and wants to check hypothesis h with a_1 . If Ctx_0 was computed with $\mathcal{L}_{0,1}$, it would be empty, and no contradiction would be found when a_0 checks later with a_2 . However, using \mathcal{L}_0 as a production field, we get consequence $a \vee b$ that contains literal $a \in \mathcal{L}_{0,1}$. It is thus sent to a_1 that will give in return b . When proposing this context to a_2 later on, a_0 will thus be able to derive a contradiction.

3) *Choice of partial hypotheses*: For computing partial hypotheses, and deciding whether to propose a given one to a neighbour or not, the same principles can be used. When no admissible hypothesis can be found, inverse entailment is performed again with an extended set of abducibles. To ensure that at least one solution can be found, the manifestations are added to the abducibles, enabling trivial explanations for some part of the hypothesis. Then, the idea is to use literals that can act as links between the theories. In practice, it means that we should include in the extended abducibles the literals in the restricted individual language of the agent. This should also be augmented with literals obtained through the arguments of other agents (when receiving $argue(o \vee p_0)$ in state τ^C). This allow us to compute all potentially useful partial hypothesis. For a given exchange, however, it is sufficient to propose those partial hypotheses that contains at least one literal of the communication language of the interacting agents.

IV. EXPERIMENTS

We describe here preliminary experimental results on a small set of problems⁴, testing our two improvements of the MARS protocol (namely, incremental context computation and restriction of the languages). Though it might be useful to assert the validity of our conclusions on a broader number of problems, we believe that the small problems used for evaluation highlight the main difficulties that can be encountered in a distributed abduction system.

The first problem, pb-1, is taken from [1], where it was used as a running example. It contains 10 clauses, distributed among 2 or 3 agents. With 3 agents, we tested two communicational constraint topologies: a line ($a_0 \leftrightarrow a_1 \leftrightarrow a_2$) and a completely connected system. This problem was designed to illustrate the MARS protocol, and thus make it go through all the possible states during its unfolding. The second problem, pb-fvar, is a toy problem with two observations that contains some clauses with unlinked variables. We pruned out hypotheses that contains variables in the resolution to respect our language bias. It is distributed among 3 agents, and here again, we tested it with line and completely connected graph topologies. The third problem, chain_n is a propositional problem designed to show a kind of worst case for distribution. It consists of three chains of implications linking respectively h_1 to o_1 (through k_{n-2}, \dots, k_0), h_1 to o_2 (through m_{n-2}, \dots, m_0) and h_2 to o_1 (through l_{n-2}, \dots, l_0). Moreover, one agent (agent $a_{n/2}$) has a constraint $\neg o_1 \vee \neg o_2$, which makes h_1 inconsistent. The aim is then to explain o_1 with abducibles $\{h_1, h_2\}$. Each agent knows 3 rules, one from each chains, and agent a_0 initially has observation o_1 . This chain was tested with $n = 8$, with either a line topology (from a_0 to a_7) or a circuit topology (as the line, with an additional link between a_0 and a_7). To check the influence of the number of agent, we also used a version of this problem with 4 agents, chain_8.4, in which a_0, a_1, a_2, a_3 are merged with respectively a_4, a_5, a_6 and a_7 , and a version with 2 agents, chain_8.2, in which a_0 are merged with respectivley even and odd indexed agents. At last, we used a more practical problem, schedule_var, which is an adaption from a scheduling problem presented in [5], with 8 agents. schedule_dir is a direct translation of the same problem from its original formalization as an abductive logic program (negation by default is dealt with by using additional abducibles).

Tables I gives the results for the four variants of our mechanism. Computational cost is given by the total number of operations performed by the consequence finding tool over the course of the protocol, whereas communicational cost is expressed as the total number of bits exchanged by the agents during the process. From these results, it is obvious that using individual communication languages does indeed greatly reduce both costs. It is especially true for the most complex problems, and the gain ratio is more important when there

⁴Complete description of all these problems can be found at <http://rjcia09.fr/MARS>.

TABLE I
EXPERIMENTAL RESULTS

Language focus		no	no	yes	yes
Incremental ctx comp.		no	yes	no	yes
Computational cost					
Pb-1	2 ag.	711	666	711	666
Pb-1 (line)	3 ag.	1 602	1 454	1 548	1 390
Pb-1 (clique)	3 ag.	2 216	2 003	1 713	1 673
Pb-fvar (line)	3 ag.	11 890	11 818	8 019	7 959
Pb-fvar (clique)	3 ag.	13 680	14 126	6 457	6 415
Chain_8.2	2 ag.	31 171	21 330	12 438	8 675
Chain_8.4 (line)	4 ag.	57 406	45 803	29 844	24 285
Chain_8.4 (circ.)	4 ag.	81 998	70 168	23 999	21 075
Chain_8 (line)	8 ag.	133 696	103 219	22 450	18 600
Chain_8 (circ.)	8 ag.	92 986	75 146	9 015	8 639
Schedule _{var}	8 ag.	53 607	50 571	39 391	39 725
Schedule _{dir}	8 ag.	381 992	372 998	95 909	94 963
Communicational cost					
Pb-1	2 ag.	617	552	617	552
Pb-1 (line)	3 ag.	1 832	1 700	1 624	1 511
Pb-1 (clique)	3 ag.	2 540	2 373	2 115	2 064
Pb-fvar (line)	3 ag.	3 177	3 080	2 659	2 622
Pb-fvar (clique)	3 ag.	3 568	3 494	2 139	2 106
Chain_8.2	2 ag.	12 230	8 924	5 746	4 238
Chain_8.4 (line)	4 ag.	25 606	14 312	22 278	12 620
Chain_8.4 (circ.)	4 ag.	35 072	12 576	35 531	11 883
Chain_8 (line)	8 ag.	65 582	57 305	14 625	13 383
Chain_8 (circ.)	8 ag.	50 749	47 317	6 509	6 627
Schedule _{var}	8 ag.	28 774	27 171	20 448	20 752
Schedule _{dir}	8 ag.	219 335	209 398	77 638	76 238

are a greater number of communicational links. Incremental computation of context is however less convincing, as it only helps when there are several context computations step, which is not such a common occurrence, unless theories are really mixed (it is the case for `pb-1` and all `chain_8` problems, which do benefit from this improvement). In the end, this improvement is useful, but only marginally so in most situations. While more experiment would be required to say anything more definite, the present results give us some hint about the influence of topology and “encoding”. Having a topology with cycle can lead to redundant computations, but can also provide easier exchange of information by avoiding the extra cost of bringing back a crucial fact or rule (as demonstrated by the addition of the link a_0-a_7 in `chain_8`). Overall, using individual communication language allows us to reduce the cost of redundant computations, so that we can take more benefit from situations where additional links are helpful. Moreover, reducing the number of agents can be either detrimental (in `chain_8`) or beneficial (in `pb-1`): the size of the communication languages seem to be a more relevant factor. At last, the huge difference between `schedulevar` and `scheduledir` seems to indicate that our protocol is much more efficient for finding whether an abducible hypothesis is consistent than it is for finding an abducible hypothesis by exploring all partial hypotheses. When formalizing a given problem, it is thus more efficient to ensure one agent can easily generate candidate hypotheses, and express rules that constrain it.

V. CONCLUSION

We presented in this paper a formalization of a multi-agent abduction problem, and proposed a sound mechanism for computing an abductive explanation that is guaranteed to

find a solution whenever one exists. We then discussed way to improve the average efficiency of this protocol, called Multi agent Abductive Reasoning System (MARS). Two improvements were proposed. The first one reduce the costs of building a complex context by doing the computation incrementally. It only helps when several steps are needed and was therefore shown to have only a limited impact on efficiency by experimental results. The main improvement consist of using informations about the individual language of each agent to focus the exchanges on what can really advance the search for a hypothesis (or the inconsistency of a candidate hypothesis). Contrarily to [7], we do not need the communication graph to be made cycle-free. While our approach use a similar idea of using *communication language*, we are only interested in the new consequences of some formulas, and thus want to avoid computing all consequences of a theory. As a result, we showed that we needed to allow the exchanges of clauses that belongs only partially to the communication language. Nonetheless, it is still an important efficiency improvement compared to the more naive approach of using only the common language for all exchanges. Experimental results showed that it substantially reduces the number of computations as well as the size of the communications. More improvement should however be brought to the search of hypotheses that can only be produced by using the theories of several agents. The learner-critic assumption that hypothesis are produced locally might be unadapted in such situations. It might thus be better to design a collaborative hypothesis formation, though another lead could be to refine the current information exchange to ensure a better treatment of “sub-goals”.

REFERENCES

- [1] G. Bourgne, K. Inoue, and N. Maudet, “Abduction of distributed theories through local interactions,” in *Proc. of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, August 2010.
- [2] H. Nabeshima, K. Iwanuma, and K. Inoue, “Solar: A consequence finding system for advanced reasoning,” in *Autom. Reas. with Analytic Tableaux and Rel. Meth.* Springer, 2003, pp. 257–263.
- [3] A. Ciampolini, E. Lamma, P. Mello, F. Toni, and P. Torroni, “Cooperation and competition in ALIAS: a logic framework for agents that negotiate,” *Annals of Math. and AI*, vol. 37, no. 1–2, pp. 65–91, 2003.
- [4] A. C. Kakas and P. Mancarella, “Database updates through abduction,” in *Proc. of VLDB '90*. Morgan Kaufmann Pub., 1990, pp. 650–661.
- [5] J. Ma, A. Russo, K. Broda, and K. Clark, “DARE: a system for distributed abductive reasoning,” *JAAMAS*, vol. 16-3, pp. 271–297, 2008.
- [6] P. Adjiman, P. Chatalic, F. Goasdoué, M.-C. Rousset, and L. Simon, “Distributed reasoning in a peer-to-peer setting: Application to the semantic web,” *J. Artif. Intell. Res. (JAIR)*, vol. 25, pp. 269–314, 2006.
- [7] E. Amir and S. A. McIlraith, “Partition-based logical reasoning for first-order and propositional theories,” *AI*, vol. 162, no. 1-2, pp. 49–88, 2005.
- [8] K. Inoue, “Linear resolution for consequence finding,” *Artif. Intell.*, vol. 56, no. 2-3, pp. 301–353, 1992.
- [9] S. Muggleton, “Inverse entailment and progol,” *New Generation Comput.*, vol. 13, no. 3&4, pp. 245–286, 1995.
- [10] R. Letz, K. Mayr, and C. Goller, “Controlled integration of the cut rule into connection tableau calculi,” *JAR*, vol. 13, pp. 297–338, 1994.
- [11] K. Inoue, T. Sato, M. Ishihata, Y. Kameya, and H. Nabeshima, “Evaluating abductive hypotheses using an em algorithm on bdds,” in *Proc. of IJCAI'09*, 2009, pp. 810–815.