Distributed Diagnosis of Faults in Circuits and Biological Systems

Ana L. C. Bazzan
Instituto de Informática, UFRGS
Caixa Postal 15064, CEP 91.501-970 Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

Abstract. Model-based diagnosis aims at inferring abnormalities of components in a system, given its behavior. One specific goal is to determine the minimal set of candidates, which is useful e.g. to efficiently perform further measurements. A key difference between the centralized and the distributed versions of this problem is that the former uses knowledge of the designer. In the latter case we cannot count on any agent knowing all inputs and outputs of the system. Distributed modelling means that only local knowledge is available to the agents. This paper focuses on distributed troubleshooting of model based devices in which outputs are produced from given inputs. The consequences of distributed representation of the knowledge and of distributed reasoning are discussed.

1 Introduction

Model-based diagnosis (MBD) infers *abnormalities of internal components* of a system given the *behavior of the system's input(s) and output(s).* This kind of reasoning has several applications and can be approached in different ways: some resemble (distributed) constraint satisfaction problems (CSP) while others involve incremental, nonmonotonic reasoning over constraints. This combination of propagation of constraints and assumption-based truth maintenance (ATMS) has been used in diagnosis as discussed in the next section.

In this paper a research line which dates from the 1970s and 1980s [1, 2, 3, 5, 9] is used. In these works, the authors propose techniques to compute the diagnosis of electronic circuits based on the model of the devices. Apart from electronic circuits, these frameworks can be used in other domains as well, such as biological systems and industrial plants in which outputs are produced from given inputs. Nowadays, due to the size and complexity of the biological processes, physical devices or industrial plants, it makes sense to do *diagnosis in a distributed way.* Hints of this can be found both in the practice of industrial automation (which uses distributed control systems) as well as in early works in AI which show how local knowledge about physical devices can be used to determine which measurement to make next [2].

There is a key difference between the centralized and the distributed versions of that problem. The former assumes the knowledge of the designer, so it is more or less straightforward to compute the faulty values, given the designer's knowledge of the device. In the latter case we cannot count on any agent knowing all inputs and outputs of the system.

In traditional centralized MBD of hardware devices, symptoms are easily detected from the designer's knowledge. For example, in Figure 1 the output **F** should be 12 but it is measured at 10. Symptoms are used to guide the computation of conflicts and candidate sets. Henceforth the term model-based diagnosis of devices (MBDD) is used for this class of problems.

The rest of this paper is organized as follows: the next section summarizes the research developed in the 1980s, as well as the work regarding MBD for MAS scenarios related to planning and coordination failure. Section 3 discuss the centralized solution for a classical example (circuit). In Section 4 the approach for the distributed diagnosis is presented. Section 5 presents two examples of the usage of the approach: one deals with the circuit presented in Section 3, where the second shows an example of a biological system (gene regulation). Section 6 concludes the paper discussing the examples, shortcomings, and future work.

2 Related Work

2.1 Diagnosis of devices from first principles

Diagnosis from first principles is a natural way of dealing with the problems meaning that the model normally mirrors the causal structure of components [1, 2, 3, 9]. MBD needs a description of the system under study and an observation of the system's behavior. If this observation conflicts with the expected behavior, the diagnosis problem is to determine which component(s) of the system, if malfunctioning, would explain the discrepancy between the observed and expected behavior. In this paper the consistency-based approach is followed: a diagnosis $\mathcal{D}$ is a set of faulty components such that the observed behavior is consistent with the assumption that exactly those components are behaving abnormally. Following [9], a diagnosis $\mathcal{D}$ of $(SD, OBS, COMP)$ is a set $\Delta \subseteq COMP$ such that $SD \cup OBS \cup \{AB(c) | c \in \Delta\} \cup \{\neg AB(c) | c \in COMP - \Delta\}$ is consistent, where SD is the set of formulae describing the system, OBS is the set of formulae describing the observations, and $COMP$ is the set of components of the system.

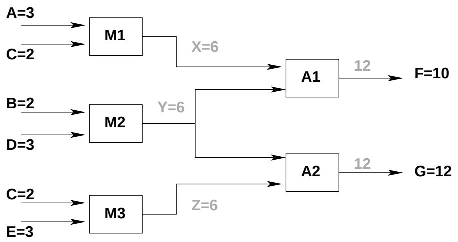

Figure 1. Circuit Adder/Multiplier – Two Outputs

2.2 Single and Multiagent Diagnosis

In [3], propagation of constraints and ATMS are used to cope with the problem of diagnosing multiple faults in a single agent setting. In the case of several agents reasoning nonmonotonically, pursuing logical consistency is an important issue. However, exchange of beliefs causes new concepts to be derived and consequently assuring global truth maintenance becomes problematic. In [7], a distributed truth maintenance system (DTMS) is proposed in which data is exchanged, affecting the global consistency. The authors use justification-based TMS, with data labeled as IN (believed) or OUT (disbelieved). In the present paper, ATMS is used so that those labels are substituted by sets of assumptions underlying the derivation of other data. The advantage is that ATMS deals with multiple states of belief, and not just IN or OUT.

Distributed diagnosis is becoming increasing popular. Witteveen and colleagues [11] propose an adaptation of MBD to single and multi-agent *planning* systems in order to perform plan diagnosis. Similarly, in [6] a causal model is used to detect a failing task (component) in a plan. Kalech and Kaminka [8] deal with failures in the *coordination* of agents' actions (e.g. in the execution of a plan) regarding selection or assignment of incompatible plans of actions and/or values for variables in problems of selection of joint actions. Frölich and colleagues [4] introduced a consistency-based framework for MBD of spatially distributed technical systems. This was followed by Roos and colleagues [10] who developed a protocol for multi-agent diagnosis with spatially distributed knowledge. However, in both cases it is assumed that each agent knows all the context related to inputs and the connections of the system, i.e. each agent has information about points that connect components managed by other agents.

Some of these problems can be mapped to DisCSP as proposed in [12]. However, in MBDD, the issue is not to *select* values for variables in a joint action but to *find out which component(s) have failed* and/or come out with minimal candidate sets so that new measurements can be made to rule out some of the previous assumptions.

3 Diagnosis of Devices

In consistency-based MBDD, the goal is to determine the minimal set of candidates for the best next measurements leading to the overall discovery of the faulted set in a minimum number of measurements [3]. As seen in Section 2.1, this involves knowledge representation and computation of conflicts and of candidates. For the former, while some works use first order logic for representing the device and perform diagnosis reasoning as nonmonotonic reasoning using resolution, de Kleer and Williams [3] propose an approach which exploits the features of an ATMS and propagation of constraints. We follow the second research line as it is a good compromise between completeness and the computation costs.

To compute conflicts, we use the method proposed in [3]. In summary, one must define an inference strategy $\mathbf{C}(OBS, ENV)$ which, given the set of observations and a set of assumptions (an environment), determines whether or not they are consistent. The method starts with an empty environment and inserts one parent at time. If an environment is inconsistent, then it is a minimal conflict and the supersets are not explored. Otherwise it is enlarged by using other combinations of components. In order to make the inference, strategy $\mathbf{C}$ is implemented using a function $\mathbf{P}(OBS, ENV)$ which returns all behavioral predictions following OBS given the assumptions in ENV. $\mathbf{P}$ computes the outputs and/or inputs of component(s) *given that these components are assumed as not faulty (not abnormal)*. If, for a given ENV, two different values of $\mathbf{P}$ are found, then ENV is a conflict.

The classical example deals with the circuit shown in Figure 1: two adders and three multipliers. Given those inputs, while the outputs are measured at $G = 12$ and $F = 10$. The latter is faulty i.e. F is observed at 10, not 12 which is the correct value given the inputs (as in the original setting, the inputs are assumed to be 100% correct). For the circuit in Figure 1, $\mathbf{P}(\{A = 3, B = 2, C = 2, D = 3\}, \{A1, M1, M2\})$ yields $\{A = 3, B = 2, C = 2, D = 3, X = 6, Y = 6, F = 12\}$. The minimal conflict set is computed: $\{\{A1, M1, M2\}, \{A1, A2, M1, M3\}\}$. Intuitively, these are conflicts because if $A1, M1, M2$ were working correctly, F would have the value of 12, not 10. Similar reasoning is valid for $\{A1, A2, M1, M3\}$.

Henceforth, underline typeset represents the result of propagating some values through a component using $\mathbf{P}$. For instance, propagating A and C through $M1$, $\{X = 6, \{M1\}\}$ is obtained. Bold typeface means that the propagated values conflict. Finally, to simplify the formulas, OBS in P and the inputs in the propagation of results are omitted since these are assumed to be always correct. Below, the details of the computation of those conflict sets are shown for the single agent case. For the example in Figure 1, given the following system description:

at M1: $X = A \times C$ at M2: $Y = B \times D$
at M3: $Z = C \times E$ at A1: $F = X + Y$
at A2: $G = Y + Z$

Successive applications of $\mathbf{P}$ yield the conflict set (CS) as follows:

$P(\{A1\}) \rightarrow \{F = 10, F = X + Y\}$ (faulty)
$P(\{A2\}) \rightarrow \{G = 12, G = Y + Z\}$
$P(\{M1\}) \rightarrow \{\underline{X = 6}\}$
$P(\{M2\}) \rightarrow \{\underline{Y = 6}\}$
$P(\{M3\}) \rightarrow \{\underline{Z = 6}\}$
$P(\{A1, A2\}) \rightarrow \{F = 10, F = X + Y, G = 12, G = Y + Z\}$
$P(\{A1, M1\}) \rightarrow \{F = 10, F = X + Y, X = 6, \underline{Y = 4}\}$
$P(\{A1, M2\}) \rightarrow \{F = 10, F = X + Y, Y = 6, \underline{\overline{X = 4\}}}$
$P(\{A1, M3\}) \rightarrow \{F = 10, F = X + Y, Z = 6\}$
$P(\{A1, A2, M1\}) \rightarrow \{F = 10, F = X + Y, G = 12, G = Y + Z, X = 6, \underline{Y = 4}\}$
$P(\{A1, A2, M2\}) \rightarrow \{F = 10, F = X + Y, G = 12, G =$

$Y + Z, Y = 6, \underline{X = 4}\}$
$P(\{A1, A2, M3\}) \rightarrow \{F = 10, F = X + Y, G = 12, G = Y + Z, Z = 6, Y = 6\}$
$P(\{A1, M1, M2\}) \rightarrow \{\mathbf{F = 10}, F = X + Y, \mathbf{X = 6}, \mathbf{Y = 6}, \underline{\mathbf{Y = 4}}, \underline{\mathbf{X = 4}}, \underline{\mathbf{F = 12}}\}$
$\Rightarrow \mathbf{CS} = \{\{\mathbf{A1}, \mathbf{M1}, \mathbf{M2}\}\}$
$P(\{A1, M1, M3\}) \rightarrow \{F = 10, F = X + Y, X = 6, Z = 6, \underline{Y = 4}\}$
$P(\{A1, M2, M3\}) \rightarrow \{F = 10, F = X + Y, Y = 6, Z = 6, \underline{X = 4, G = 12}\}$
$P(\{A1, A2, M1, M2\}) \rightarrow$ superset of $\{A1, M1, M2\}$
$P(\{A1, A2, M1, M3\} \rightarrow \{F = 10, F = X + Y, G = 12, G = Y + Z, X = 6, Z = 6, \mathbf{Y = 4}, \mathbf{Y = 6}\}$
$\Rightarrow \mathbf{CS} = \{\{\mathbf{A1}, \mathbf{M1}, \mathbf{M2}\}, \{\mathbf{A1}, \mathbf{A2}, \mathbf{M1}, \mathbf{M3}\}\}$
$P(\{A1, A2, M2, M3\}) \rightarrow \{F = 10, F = X + Y, G = 12, G = Y + Z, Y = 6, Z = 6, \underline{X = 4}\}$
$P(\{A1, M1, M2, M3\}) \rightarrow$ superset of $\{A1, M1, M2\}$
$\mathbf{CS} = \{\{\mathbf{A1}, \mathbf{M1}, \mathbf{M2}\}, \{\mathbf{A1}, \mathbf{A2}, \mathbf{M1}, \mathbf{M3}\}\}$

Once the conflict set $CS = \{c_1, ..., c_k\}$ is computed, the diagnosis candidates can be found. In order to identify the minimal candidate set, here the approach by Reiter [9] is used, which is based on the computation of the *minimal hitting set* for the conflict set CS. A hitting set for CS is a set $H \subseteq \bigcup_{S \in CS}$ such that $H \cap S \neq \{\}$ for each $S \in CS$. A hitting set for CS is minimal if and only if no proper subset of it is a hitting set for CS. The diagnosis can then be computed:
$\mathcal{D} = \{\{\mathbf{A1}\}, \{\mathbf{M1}\}, \{\mathbf{A2}, \mathbf{M2}\}, \{\mathbf{M2}, \mathbf{M3}\}\}$

4 Distributed Diagnosis

In distributed MBDD, due to lack of global knowledge, no single agent can infer that a component is faulty. Thus, when MBDD is tackled as a distributed process, there are two possible scenarios: either one assumes that agent $\mathcal{A}_i$ receives information regarding a symptom (e.g. F observed as 10, not 12), or this assumption is dropped and agents just try to find conflicts by exchanging values. Here the latter is assumed, i.e. no external communication regarding symptoms is made. Agents communicate to propagate the constraints (values of outputs of components). This has the shortcoming of looking as a non-purpose communication but seems more reasonable than having an external entity telling symptoms to agents.

As said, ATMS allows a more flexible labeling of the data: instead of using just IN or OUT, we label the data according to whether it is internally known or was communicated by other agent. In the latter case, the data is *believed* to be true. Finally, here we assume that each component is associated with one agent.

Agents are represented by calligraphic letters, as for example $\mathcal{A}_1$ is the agent in charge of component $A1$, $\mathcal{M}_2$ the one in charge of $M2$, etc. Each agent $\mathcal{A}_i$ has a knowledge base $KB(i)$ to store the knowledge $K(i)$, where $K(i)$ denotes local information which is 100% trustworth. Also computations performed by $\mathcal{A}_i$ over $K(i)$ are stored in $KB(i)$, as well as the list of agents which are related to $\mathcal{A}_i$. This relation is both physical (agents physically connected) and logical (agents inserted in the list because they can help in the detection of conflicts and faults).

Both the knowledge and the beliefs are based on messages received from other agents. Values can be computed locally due to the knowledge of an agent about its component. For instance if agent $\mathcal{A}_i$ is in charge of *adder* A1, it knows that the output of A1 is the *sum* of both inputs to A1. Agents can perform the following actions: collect *input* and *output values* (only for agents in charge of input and output components respectively); compute output values (only if both inputs are known) and input value (given the output and one input); request knowledge and communicate values to agents in the *agent_list*; and communicate diagnosis (via broadcast to those in the agent_list).

The algorithm to perform the diagnosis needs a list of input and output components ($\mathcal{IN}$ and $\mathcal{OUT}$ respectively), a list of agents (assuming that each component is associated with an agent). All components/agents are collected in the $\mathcal{AG}$ list. The schema of the approach is as follows:

1. *representation of knowledge*: knowledge bases keep the variables values and constraints between variables;
2. *computation of conflicts*: this process occurs in a finite number of cycles (in the worst case some agents have to communicate with all others). In cycle 0, for each agent, collect local knowledge to the agent_list (distinguishing between agents connected backwards and forwards) and expand the knowledge via the function $\mathbf{P}(OBS, ENV)$ (Section 3). In cycles 1 and 2, communicate with agents backwards and forwards in the agent_list respectively. During this process, new agents can be added to a list (new_in_agent_list). While this list is not empty this means that agent i still has new relationships to explore. At the end of the communications, some agents have collected a list of conflicts. This is exchanged and the agent with the most complete one computes the set of candidates for diagnosis;
3. *computation of candidates for diagnosis*: the hitting set is computed (Section 3).

An important remark is that this is a concurrent process within each cycle, i.e. all agents can communicate concurrently. This is so because the order in which communication is performed *within a cycle* does not matter. This order does not alter the amount of information each agent has at the end of each cycle.

Next, examples of the diagnosis procedure are presented. The first deals with the circuit with two outputs discussed before; the second with a simplified case of gene regulation. As before, local computations (function P) are shown as underline text, whereas conflicts appear in boldface.

5 Examples

5.1 Circuit with two outputs

Following the algorithm scheme outlined in the previous section, the knowledge base for each agent is as follows:

agent_list at each agent:
$agent\_list(\mathcal{A}_1, backwards(\{\mathcal{M}_1, \mathcal{M}_2\}))$
$agent\_list(\mathcal{A}_2, backwards(\{\mathcal{M}_2, \mathcal{M}_3\}))$
$agent\_list(\mathcal{M}_1, forwards(\{\mathcal{A}_1\}))$
$agent\_list(\mathcal{M}_2, forwards(\{\mathcal{A}_1, \mathcal{A}_2\}))$
$agent\_list(\mathcal{M}_3, forwards(\{\mathcal{A}_2\}))$
Cycle 0 – Knowledge of isolated agents
$K(\mathcal{A}_1) : \{\{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}\}$
$K(\mathcal{A}_2) : \{\{G = 12, \{\mathcal{A}_2\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}\}$

$K(\mathcal{M}_1) : \{X = 6, \{\mathcal{M}_1\}\}$
$K(\mathcal{M}_2) : \{Y = 6, \{\mathcal{M}_2\}\}$
$K(\mathcal{M}_3) : \{Z = 6, \{\mathcal{M}_3\}\}$
Cycle 1 – Knowledge after communication (backwards) and local computations:
$\mathcal{A}_1$ communicates with $\mathcal{M}_1 \rightarrow K(\mathcal{M}_1)$:
$\{\{X = 6, \{\mathcal{M}_1\}\}, \{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{Y = 4, \{\mathcal{A}_1, \mathcal{M}_1\}\}\}$
$\mathcal{A}_1$ communicates with $\mathcal{M}_2 \rightarrow K(\mathcal{M}_2)$:
$\{\{Y = 6, \{\mathcal{M}_2\}\}, \{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{X = 4, \{\mathcal{A}_1, \mathcal{M}_2\}\}\}$
$\mathcal{A}_2$ communicates with $\mathcal{M}_2 \rightarrow K(\mathcal{M}_2)$:
$\{\{Y = 6, \{\mathcal{M}_2\}\}, \{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{X = 4, \{\mathcal{A}_1, \mathcal{M}_2\}\}, \{G = 12, \{\mathcal{A}_2\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}, \{Z = 6, \{\mathcal{A}_2, \mathcal{M}_2\}\}\}$
$\mathcal{A}_2$ communicates with $\mathcal{M}_3 \rightarrow K(\mathcal{M}_3)$:
$\{\{Z = 6, \{\mathcal{M}_3\}\}, \{G = 12, \{\mathcal{A}_2\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}, \{Y = 6, \{\mathcal{A}_2, \mathcal{M}_3\}\}\}$
Cycle 2 – Knowledge after communication (forwards):
$\mathcal{M}_1$ communicates with $\mathcal{A}_1 \rightarrow K(\mathcal{A}_1)$:
$\{\{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{X = 6, \{\mathcal{M}_1\}\}, \{Y = 4, \{\mathcal{A}_1, \mathcal{M}_1\}\}\}$
$\mathcal{M}_2$ communicates with $\mathcal{A}_1 \rightarrow K(\mathcal{A}_1)$:
$\{\{\{\mathbf{F = 10}, \{\mathcal{A}_\mathbf{1}\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{\mathbf{X = 6}, \{\mathcal{M}_\mathbf{1}\}\}, \{\mathbf{Y = 4}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{1}\}\}, \mathbf{Y = 6}, \{\mathcal{M}_\mathbf{2}\}\}, \{\mathbf{X = 4}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}, \{G = 12, \{\mathcal{A}_2\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}, \{Z = 6, \{\mathcal{A}_2, \mathcal{M}_2\}\}, \{\mathbf{F = 12}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}, \{\mathbf{G = 10}, \{\mathcal{A}_\mathbf{1}, \mathcal{A}_\mathbf{2}, \mathcal{M}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}, \{\mathbf{Z = 4}, \{\mathcal{A}_\mathbf{1}, \mathcal{A}_\mathbf{2}, \mathcal{M}_\mathbf{1}\}\}, \{\mathbf{F = 8}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}\}$
$\mathcal{A}_1$ perceives conflict among agents: $CS_{\mathcal{A}_1} = \{\mathcal{A}_1, \mathcal{M}_1, \mathcal{M}_2\}$
$\mathcal{A}_1$ includes $\mathcal{A}_2$ in the agent_list
$\mathcal{M}_2$ communicates with $\mathcal{A}_2 \rightarrow K(\mathcal{A}_2)$:
$\{\{G = 12, \{\mathcal{A}_2\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}, \{Y = 6, \{\mathcal{M}_2\}\}, \{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{X = 4, \{\mathcal{A}_1, \mathcal{M}_2\}\}, \{Z = 6, \{\mathcal{A}_2, \mathcal{M}_2\}\}\}$
$\mathcal{M}_3$ communicates with $\mathcal{A}_2 \rightarrow K(\mathcal{A}_2)$:
$\{\{G = 12, \{\mathcal{A}_2\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}, \{Y = 6, \{\mathcal{M}_2\}\}, \{F = 10, \{\mathcal{A}_1\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{X = 4, \{\mathcal{A}_1, \mathcal{M}_2\}\}, \{Z = 6, \{\mathcal{A}_2, \mathcal{M}_2\}\}, \{Z = 6, \{\mathcal{M}_3\}\}, \{Y = 6, \{\mathcal{A}_2, \mathcal{M}_3\}\}\}$
Cycle 3 – since $\mathcal{A}_1$ has included $\mathcal{A}_2$ in its view, $\mathcal{A}_1$ requests information from $\mathcal{A}_2$:
$\mathcal{A}_2$ communicates with $\mathcal{A}_1 \rightarrow K(\mathcal{A}_1)$:
$\{\{\{\{\mathbf{F = 10}, \{\mathcal{A}_\mathbf{1}\}\}, \{F = X + Y, \{\mathcal{A}_1\}\}, \{\{\mathbf{X = 6}, \{\mathcal{M}_\mathbf{1}\}\}, \{\{\mathbf{Y = 4}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{1}\}\}, \{\{\mathbf{Y = 6}, \{\mathcal{M}_\mathbf{2}\}\}, \{\{\mathbf{X = 4}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}, \{\mathbf{G = 12}, \{\mathcal{A}_\mathbf{2}\}\}, \{G = Y + Z, \{\mathcal{A}_2\}\}, \{\{\mathbf{Z = 6}, \{\mathcal{A}_\mathbf{2}, \mathcal{M}_\mathbf{2}\}\}, \{\mathbf{Z = 6}, \{\mathcal{M}_\mathbf{3}\}\}, \{\{\mathbf{Y = 6}, \{\mathcal{A}_\mathbf{2}, \mathcal{M}_\mathbf{3}\}\}, \{\{\mathbf{F = 12}, \{\mathcal{M}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}, \{\{\mathbf{F = 8}, \{\mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}, \{\mathbf{Z = 8}, \{\mathcal{A}_\mathbf{2}, \mathcal{A}_\mathbf{1}, \mathcal{M}_\mathbf{1}\}\}, \{\mathbf{G = 10}, \{\mathcal{A}_\mathbf{1}, \mathcal{A}_\mathbf{2}, \mathcal{M}_\mathbf{1}, \mathcal{M}_\mathbf{2}\}\}\}$
$\mathcal{A}_1$ perceives conflict among agents:
$CS_{\mathcal{A}_1} = \{\mathcal{A}_1, \mathcal{M}_1, \mathcal{M}_2\}, \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{M}_1, \mathcal{M}_2\}, \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{M}_1, \mathcal{M}_3\}$ but the second one is a superset of the former, so that $CS_{\mathcal{A}_1} = \{\mathcal{A}_1, \mathcal{M}_1, \mathcal{M}_2\}, \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{M}_1, \mathcal{M}_3\}$.

$\mathcal{A}_1$ is the only agent which has collected a conflict set so it now solves this using the minimal hitting set method achieving the same $\mathcal{D}$ computed in the centralized version.

5.2 Biological System

The second example is a system in which five genes are regulated. Each can be ON or OFF (hereafter represented by 1 and 0 respectively). This example derives from another classical circuit [9], in which both outputs are faulty as in Figure 2. The idea here is slightly different from the previous example. Adders and multipliers are men-made device, whose behavior is well-known. In biology, we have just theories or assumptions about how the components should behave. Thus, the goal here is to verify whether the theory leads to the observed behavior and, when this is not the case, to point out the candidates which could explain how or where the theory fails. In summary, the proposed diagnosis method delivers a set of components to be better observed in order to find where and/or how the theory fails.

Next, steps similar to those of the previous example are shown. The conflict and candidate sets are calculated as below (steps omitted due to lack of space):

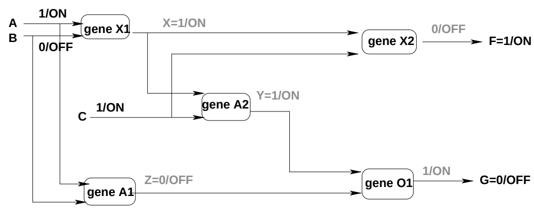

Figure 2. Genetic Network

agent_list at each agent:
$agent\_list(\mathcal{X}_1, forwards(\{\mathcal{X}_2, \mathcal{A}_2\}))$
$agent\_list(\mathcal{X}_2, backwards(\{\mathcal{X}_1\}))$
$agent\_list(\mathcal{A}_1, forwards(\{\mathcal{O}_1\}))$
$agent\_list(\mathcal{A}_2, \{forwards(\{\mathcal{O}_1\}), backwards(\{\mathcal{X}_1\})\})$
$agent\_list(\mathcal{O}_1, backwards(\{\mathcal{A}_1, \mathcal{A}_2\}))$
Cycle 1 – Knowledge after communication (backwards): $\mathcal{X}_2$ communicates with $\mathcal{X}_1 \rightarrow K(\mathcal{X}_1)$:
$\{\{\mathbf{X = 1}, \{\mathcal{X}_1\}\}, \{X = A \oplus B, \{\mathcal{X}_1\}\}, \{C = 1, \{\mathcal{X}_2\}\}, \{\mathbf{F = 1}, \{\mathcal{X}_\mathbf{2}\}\}, \{F = X \oplus C, \{\mathcal{X}_2\}\}, \{\mathbf{X = 0}, \{\mathcal{X}_\mathbf{2}\}\}, \{\mathbf{F = 0}, \{\mathcal{X}_1, \mathcal{X}_2\}\}\}$
$\mathcal{X}_1$ perceives conflict among agents: $CS_{\mathcal{X}_1} = \{\mathcal{X}_1, \mathcal{X}_2\}$
$\mathcal{O}_1$ communicates with $\mathcal{A}_2$ and $\mathcal{A}_1$ (omitted).
$\mathcal{A}_2$ communicates with $\mathcal{X}_1 \rightarrow K(\mathcal{X}_1)$:
$\{\{\mathbf{X = 1}, \{\mathcal{X}_1\}\}, \{X = A \oplus B, \{\mathcal{X}_1\}\}, \{C = 1, \{\mathcal{X}_2\}\}, \{\mathbf{F = 1}, \{\mathcal{X}_2\}\}, \{F = X \oplus C, \{\mathcal{X}_2\}\}, \{\mathbf{X = 0}, \{\mathcal{X}_2\}\}, \{\mathbf{F = 0}, \{\mathcal{X}_1, \mathcal{X}_2\}\}, \{C = 1, \{\mathcal{A}_2\}\}, \{Y = X \wedge C, \{\mathcal{A}_2\}\}, \{\mathbf{G = 0}, \{\mathcal{O}_1\}\}, \{G = Y \vee Z, \{\mathcal{O}_1\}\}, \{\mathbf{Y = 0}, \{\mathcal{A}_\mathbf{2}, \mathcal{X}_\mathbf{2}\}\}, \{\mathbf{Y = 1}, \{\mathcal{A}_\mathbf{2}, \mathcal{X}_\mathbf{1}\}\}, \{\mathbf{G = 1}, \{\mathcal{A}_\mathbf{2}, \mathcal{X}_\mathbf{1}, \mathcal{O}_\mathbf{1}\}\}\}$
The new conflict set is $CS_{\mathcal{X}_1} = \{\mathcal{A}_2, \mathcal{X}_1, \mathcal{O}_1\}$ ($\{\mathcal{A}_2, \mathcal{X}_1, \mathcal{X}_2\}$ is a superset of $\{\mathcal{X}_1, \mathcal{X}_2\}$). $\mathcal{X}_1$ includes $\mathcal{O}_1$ in its agent_list.
Cycle 2 – Knowledge after communication (forwards): $\mathcal{X}_1$ communicates with $\mathcal{X}_2 \rightarrow K(\mathcal{X}_2)$:
$\{\{C = 1, \{\mathcal{X}_2\}\}, \{F = 1, \{\mathcal{X}_2\}\}, \{F = X \oplus C, \{\mathcal{X}_2\}\}, \{\mathbf{X = 0}, \{\mathcal{X}_2\}\}, \{\mathbf{X = 1}, \{\mathcal{X}_1\}\}, \{X = A \oplus B, \{\mathcal{X}_1\}\}, \{F = 0, \{\mathcal{X}_1, \mathcal{X}_2\}\}, \{\mathbf{C = 1}, \{\mathcal{A}_2\}\}, \{Y = X \wedge C, \{\mathcal{A}_2\}\}, \{\mathbf{G = 0}, \{\mathcal{O}_1\}\}, \{G = Y \vee Z, \{\mathcal{O}_1\}\}, \{\mathbf{Y = 0}, \{\mathcal{A}_2, \mathcal{X}_\mathbf{2}\}\}, \{\mathbf{Y = 1}, \{\mathcal{A}_\mathbf{2}, \mathcal{X}_\mathbf{1}\}\}, \{\mathbf{G = 1}, \{\mathcal{A}_\mathbf{2}, \mathcal{X}_\mathbf{1}, \mathcal{O}_\mathbf{1}\}\}, \{\mathbf{C = 0}, \{\mathcal{X}_1, \mathcal{X}_2\}\}, \{Z = 0, \{\mathcal{A}_2, \mathcal{O}_1, \mathcal{X}_2\}\}\}$
$\mathcal{X}_2$ deduces that $\mathcal{X}_1$ knows about these conflicts because the information came from $\mathcal{X}_1$ itself thus $\mathcal{X}_2$ does nothing about this. Notice also that the new conflict set $\{\mathcal{A}_2, \mathcal{X}_1, \mathcal{X}_2\}$ is a superset of $\{\mathcal{X}_1, \mathcal{X}_2\}$.

$\mathcal{X}_1$ communicates with $\mathcal{A}_2 \rightarrow K(\mathcal{A}_2)$:
$\{\{C = 1, \{\mathcal{A}_2\}\}, \{Y = X \wedge C, \{\mathcal{A}_2\}\}, \{G = 0, \{\mathcal{O}_1\}\}, \{G = Y \vee Z, \{\mathcal{O}_1\}\}, \{\mathbf{X = 1}, \{\mathcal{X}_1\}\}, \{X = A \oplus B, \{\mathcal{X}_1\}\}, \{C = 1, \{\mathcal{X}_2\}\}, \{F = 1, \{\mathcal{X}_2\}\}, \{F = X \oplus C, \{\mathcal{X}_2\}\}, \{\mathbf{X = 0}, \{\mathcal{X}_2\}\}, \{F = 0, \{\mathcal{X}_1, \mathcal{X}_2\}\}, \{Y = 0, \{\mathcal{A}_2, \mathcal{X}_2\}\}, \{\mathbf{Y = 1}, \{\mathcal{A}_2, \mathcal{X}_1\}\}, \{G = 1, \{\mathcal{A}_2, \mathcal{X}_1, \mathcal{O}_1\}\}, \underline{\{Z = 0, \{\mathcal{A}_2, \mathcal{O}_1, \mathcal{X}_2\}\}}\}$
Also $\mathcal{A}_2$ deduces that $\mathcal{X}_1$ knows about these conflicts (same reasoning as $\mathcal{X}_2$) thus $\mathcal{A}_2$ does nothing about this.
$\mathcal{A}_1$ communicates with $\mathcal{O}_1 \rightarrow K(\mathcal{O}_1)$:
$\{\{G = 0, \{\mathcal{O}_1\}\}, \{G = Y \vee Z, \{\mathcal{O}_1\}\}, \{Z = 0, \{\mathcal{A}_1\}\}, \{Z = A \wedge B, \{\mathcal{A}_1\}\}\}$
$\mathcal{A}_2$ communicates with $\mathcal{O}_1 \rightarrow K(\mathcal{O}_1)$:
$\{\{\mathbf{G = 0}, \{\mathcal{O}_1\}\}, \{G = Y \vee Z, \{\mathcal{O}_1\}\}, \{Z = 0, \{\mathcal{A}_1\}\}, \{Z = A \wedge B, \{\mathcal{A}_1\}\}, \{C = 1, \{\mathcal{A}_2\}\}, \{Y = X \wedge C, \{\mathcal{A}_2\}\},$
$\{\mathbf{X = 1}, \{\mathcal{X}_1\}\}, \{X = A \oplus B, \{\mathcal{X}_1\}\}, \{C = 1, \{\mathcal{X}_2\}\}, \{F = 1, \{\mathcal{X}_2\}\}, \{F = X \oplus C, \{\mathcal{X}_2\}\}, \{\mathbf{X = 0}, \{\mathcal{X}_2\}\},$
$\{F = 0, \{\mathcal{X}_1, \mathcal{X}_2\}\}, \{\mathbf{Y = 1}, \{\mathcal{A}_2, \mathcal{X}_1\}\}, \{\mathbf{Y = 0}, \{\mathcal{A}_2, \mathcal{X}_2\}\}, \{Z = 0, \{\mathcal{A}_2, \mathcal{O}_1, \mathcal{X}_2\}\}, \{\mathbf{G = 1}, \{\mathcal{A}_2, \mathcal{X}_1, \mathcal{O}_1\}\}\}$
$\mathcal{O}_1$ deduces that $\mathcal{A}_2$ knows about these conflicts because the information came from $\mathcal{A}_2$ thus $\mathcal{O}_1$ does nothing about this.
Cycle 3: since $\mathcal{X}_1$ has included $\mathcal{O}_1$ in its view, $\mathcal{X}_1$ requests information from $\mathcal{O}_1$
$\mathcal{O}_1$ communicates with $\mathcal{X}_1 \rightarrow K(\mathcal{X}_1)$:
$\{\{\mathbf{X = 1}, \{\mathcal{X}_1\}\}, \{X = A \oplus B, \{\mathcal{X}_1\}\}, \{C = 1, \{\mathcal{X}_2\}\},$
$\{\mathbf{F = 1}, \{\mathcal{X}_2\}\}, \{F = X \oplus C, \{\mathcal{X}_2\}\}, \{\mathbf{X = 0}, \{\mathcal{X}_2\}\},$
$\{\mathbf{F = 0}, \{\mathcal{X}_1, \mathcal{X}_2\}\}, \{C = 1, \{\mathcal{A}_2\}\}, \{Y = X \wedge C, \{\mathcal{A}_2\}\},$
$\{\mathbf{G = 0}, \{\mathcal{O}_1\}\}, \{G = Y \vee Z, \{\mathcal{O}_1\}\}, \{\mathbf{Y = 0}, \{\mathcal{A}_2, \mathcal{X}_2\}\},$
$\{\mathbf{Y = 1}, \{\mathcal{A}_2, \mathcal{X}_1\}\}, \{\mathbf{G = 1}, \{\mathcal{A}_2, \mathcal{X}_1, \mathcal{O}_1\}\},$
$\{Z = 0, \{\mathcal{A}_1\}\}, \{Z = A \wedge B, \{\mathcal{A}_1\}\}, \{Z = 0, \{\mathcal{A}_2, \mathcal{O}_1, \mathcal{X}_2\}\}\}$

No new conflict is found so that $CS_{\mathcal{X}_1} = \{\mathcal{X}_1, \mathcal{X}_2\}, \{\mathcal{A}_2, \mathcal{X}_1, \mathcal{O}_1\}$. This means that the biologist would have to further investigate his/her assumptions about the behavior of genes $\mathcal{X}_1$ and $\mathcal{X}_2$ or $\mathcal{A}_2$, $\mathcal{X}_1$ and $\mathcal{O}_1$.

5.3 Brief Notes on the Computational Efficiency

It is known that stablishing *global* minimal diagnosis is NP-hard. Some simplifications have been proposed as in [10], but authors provide each agent information about the connection points managed by all other agents. In the present paper, each agent $\mathcal{A}_i$ exchange information only with others which are either present in $\mathcal{A}_i$'s agent_list or are included later due to logical relationships. In the worst case $\mathcal{A}_i$ has to collect information about all other agents, but in practice this does not happen because agents with more knowledge lead the process.

The procedure proposed always terminates because each time agent j communicates with i, after detecting conflicts (if any), j leaves the agent_list of i and do not communicate again since either one of them will lead the diagnosis process (because it perceives a super set of conflicts), or there will be another agent k which, upon communicating with i and j later will perceive conflicts between all of them. Thus, in the worst case one agent will collect all data, new_in_agent_list will be empty and the algorithm ends. After, the hitting set algorithm is called. For proofs regarding this algorithm, refer to [9]; in the worst case all components will be candidates. This algorithm is exponential in the number of potential faults. However, MBDD deals with candidates in terms of minimal space. Typically this space is small, so that in practice the approach tends to grow with the square or cube of the number of potential faults.

6 Conclusion and Future Work

This paper deals with distributed MBDD in which no agent has full knowledgment. The consequences of distributed representation of the knowledge and of distributed reasoning were discussed and algorithms were proposed to deal with the occurrence of conflicts. In the future, different granularities have to be explored, as for example, the case where agents manage more than one component, and, in case of biological systems, how to deal with cycles in the graph. Also, the approach used here has the same kind of shortcoming as some DisCSP methods: eventually one agent centralizes the process (and the knowledge). This issue will be investigated in the near future.

REFERENCES

[1] Davis, R. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence 24*, 1–3 (December 1984), 347–410.
[2] de Kleer, J. Local methods for localizing faults in eletronic circuits. Memo 394, MIT Art. Int. Lab., November 1976.
[3] de Kleer, J., and Williams, B. C. Diagnosing multiple faults. *Artificial Intelligence 32*, 1 (April 1987), 97–130.
[4] Frölich, P., de Almeida Móra, I., Nejdl, W., and Schroeder, M. Diagnostic agents for distributed systems. In *Formal Models of Agents: ESPRIT Project ModelAge Final Workshop* (1996), J.-J. C. Meyer and P.-Y. Schobbens, Eds., vol. 1760 of *Lecture Notes in Artificial Intelligence*, pp. 173–186.
[5] Genesereth, M. The use of design descriptions in automated diagnosis. *Artificial Intelligence 24* (December 1984), 411–436.
[6] Horling, B., Lesser, V., Vincent, R., Bazzan, A. L. C., and Xuan, P. Diagnosis as an integral part of multi-agent adaptability. In *DARPA Information Survivability Conference and Exposition* (South Carolina, January 2000), IEEE Computer Society, pp. 211–219. see also UMASS CSTR 1999-03.
[7] Huhns, M. N., and Bridgeland, D. M. Multiagent truth maintenance. *IEEE Transactions on Systems, Man, and Cybernetics 21*, 6 (November/December 1991), 1437–1445.
[8] Kalech, M., and Kaminka, G. A. Towards model-based diagnosis of coordination failures. In *AAAI '05: Proceedings of the 12th National Conference on Artificial Intelligence* (2005), M. M. Veloso and S. Kambhampati, Eds., AAAI Press / The MIT Press, pp. 102–107.
[9] Reiter, R. A theory of diagnosis from first principles. *Artificial Intelligence 32*, 1 (April 1987), 57–95.
[10] Roos, N., ten Teije, A., and Witteveen, C. A protocol for multi-agent diagnosis with spatially distributed knowledge. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems* (Melbourne, July 2003), vol. 1, ACM Computing Press, pp. 655–661.
[11] Witteveen, C., Roos, N., van der Krogt, R., and de Weerdt, M. Diagnosis of single and multi-agent plans. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, July 2005), F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, Eds., vol. II, pp. 805–812.
[12] Yokoo, M., Durfee, E. H., Ishida, T., and Kuwabara, K. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering 10*, 5 (1998), 673–685.