Synthesis of a Distributed and Accurate Diagnoser

Priscilla Kan John [1], Alban Grastien [1], and Yannick Pencolé [2]

[1] *NICTA*and Australian National University; Canberra, ACT 02 00, Australia*
priscilla.kanjohn@anu.edu.au
alban.grastien@nicta.com.au
[2] *CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse, France*
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
ypencole@laas.fr

ABSTRACT

The complex behaviour of large discrete event systems makes such systems difficult to diagnose. Using decentralised techniques helps limit combinatorial explosion but is not sufficient. Often, the complexity of the diagnosis is dependent on how components in the system are connected and the number of connections between them. We propose to augment a decentralised junction tree-based approach by ignoring some connections on the system. This helps reduce the complexity, and hence the cost, of the diagnostic reasoning required. However accuracy of the diagnosis is also reduced. We get around this problem by performing an off-line analysis to determine which connections can be safely ignored.

1 INTRODUCTION

The supervision of large decentralised systems, such as telecommunication networks, web services and electricity networks, is a complex task. One malfunction in the system can cause a series of cascading events and alarms that are difficult to interpret and potentially dangerous for the system. We are interested in discrete event systems (Cassandras and Lafortune, 1999). To allow for flexibility, we consider model-based diagnosis. In practice, this means that observations on the system are compared to the model to determine whether faults have occurred in the system.

The complexity of reasoning on a model increases exponentially with respect to the number of components in the system. This implies that it is not possible to employ simple diagnosis techniques when the system is made up of tens of components. To be able to handle systems with hundreds, or even thousands, of components, a range of techniques have been developed to get over this hurdle; *e.g.* (Schumann *et al.*, 2004),(Pencolé and Cordier, 2005).

The components of a system are physically and logically linked by *connections* that restrict their individual behaviours and cause them to display complex collective behaviour. Thus, for a system of low connectivity, it is sufficient to reason locally on small subsystems, whereas for a system with high connectivity, this does not apply, making if difficult to track the global behaviour.

We implement a diagnosis algorithm that decomposes the network into a junction tree (Kan John and Grastien, 2008). The latter representation allows us to deduce the complexity of the diagnosis which depends on the number of connections between components of the system and how they are connected.

The idea presented in this paper is to ignore certain connections in the system. This makes it possible to reason on smaller subsystems such that diagnosis can be obtained in reasonable time. However, this could lead to a loss of accuracy of the diagnosis. This results from not taking into account information from the ignored connections that could have helped in eliminating certain diagnostic scenarios. Therefore, we perform a prior *accuracy analysis* on the model to determine which connections can be ignored without having a negative impact on the global accuracy of the diagnoser. In (Pencolé *et al.*, 2006) only diagnosis on subsystems is considered.

The rest of the paper is divided as follows. Notations are presented in the next section. Diagnosis is then defined and distributed approaches are presented. Diagnosis on a subconfiguration is defined and finally the determination of the optimal subconfiguration is discussed.

2 PRELIMINARIES

We are interested in Discrete Event Systems (DES, (Cassandras and Lafortune, 1999)) and we use the no-

*NICTA is funded by the Australian Government as represented by the *Department of Broadband, Communication and the Digital Economy* and *the Autralian Research Council through the ICT Centre of Excellence program*.

tation of languages to model such systems and to define diagnosis.

We note Σ a set of *symbols* (modeling events on the system). A word σ is a finite sequence of sets of symbols $s_1.\cdots.s_n$ such that $\forall i,\ s_i \subseteq \Sigma, s_i \neq \emptyset$. So, if $\Sigma = \{a,b,c,d\}$, then $\{a\}.\{b,c\}.\{d\}.\{a\}$ is a word on Σ where b and c appear simultaneously.Generally, a word is defined simply as a sequence of symbols; we use an augmented notation so that we can represent the occurrence of simultaneous events and gain more flexibility to describe the main contribution of the paper that is about the relaxation of connections. The empty sequence is denoted ε. We abuse notation and write $s_i \in \sigma$ to denote that $s_i \subseteq \Sigma$ is in the sequence σ. $w(\Sigma) = ((2^{\Sigma})^{\star} - \{\emptyset\})$ is the set of words on Σ. A *language* $\mathcal{L}$ on Σ is a subset of words $\mathcal{L} \subseteq w(\Sigma)$. The *projection* operation can be used to focus on specific events of $\Sigma' \subseteq \Sigma$.

Definition 1 (Projection) *The projection on Σ' of a word σ on $\Sigma \supseteq \Sigma'$, denoted $\mathrm{P}_{\Sigma\rightarrow\Sigma'}(\sigma)$, or simply $\mathrm{P}_{\Sigma'}(\sigma)$, is the word on Σ' that only retains the symbols of Σ' and removes empty symbol sets. Formally,* $\mathrm{P}_{\Sigma\rightarrow\Sigma'}(\sigma)$ =

$$\begin{cases} \varepsilon & \textit{if } \sigma = \varepsilon, \\ \mathrm{P}_{\Sigma\rightarrow\Sigma'}(\sigma') & \textit{if } (\sigma = s.\sigma') \wedge (s \cap \Sigma' = \emptyset), \\ s \cap \Sigma'.\,\mathrm{P}_{\Sigma\rightarrow\Sigma'}(\sigma') & \textit{if } s \cap \Sigma' \neq \emptyset. \end{cases}$$

The projection on Σ' of the language $\mathcal{L}$ on $\Sigma \supset \Sigma'$, denoted $\mathrm{P}_{\Sigma\rightarrow\Sigma'}(\mathcal{L})$, is the set of words in $\mathcal{L}$ projected onto Σ': $\mathrm{P}_{\Sigma\rightarrow\Sigma'}(\mathcal{L}) = \{\mathrm{P}_{\Sigma\rightarrow\Sigma'}(\sigma) \mid \sigma \in \mathcal{L}\}$. The inverse operation gives the set of words on Σ whose projection on Σ' is included in the language of origin: $\mathrm{P}^{-1}_{\Sigma\rightarrow\Sigma'}(\mathcal{L}) = \{\sigma \in w(\Sigma) \mid \mathrm{P}_{\Sigma\rightarrow\Sigma'}(\sigma) \in \mathcal{L}\}$.

Synchronisation
Each local entity has its own specific language to represent its behaviour. When several entities are concerned, we need to *synchronise* their languages to generate a globally consistent language. Each language has its own symbol set, disjoint from the symbol sets of other languages. However, some symbols from different local sets could be different representations of the same physical reality. The synchronisation operation coordinates these equivalent symbols by forcing their simultaneity. Equivalent symbols on different languages are represented by synchronous sets.

Definition 2 (Synchronous Set) *Given two disjoint sets of symbols Σ_1 and Σ_2, a synchronous set $\mathcal{S}$ is a set of symbol pairs coming from the two sets: $\mathcal{S} \subseteq \Sigma_1 \times \Sigma_2$.*

An element $\langle a,b\rangle \in \mathcal{S}$ indicates that a and b are describing the same physical reality and we have to ensure that they are considered simultaneously.

Definition 3 (Language Synchronisation) *Given two languages $\mathcal{L}_1$ on Σ_1 and $\mathcal{L}_2$ on Σ_2, and a synchronous set $\mathcal{S}$ defined on Σ_1 and Σ_2. The synchronous product of $\mathcal{L}_1$ and $\mathcal{L}_2$ on $\mathcal{S}$, denoted $\mathcal{L}_1 \bigotimes_{\mathcal{S}} \mathcal{L}_2$,is defined as the set of words on $(\Sigma_1 \cup \Sigma_2)$ whose projection on each local symbol set is the local language, and satisfies the constraint of simultaneity introduced by $\mathcal{S}$. Formally: $\{\sigma \in (\Sigma_1 \cup \Sigma_2) \mid \big(\forall i \in \{1,2\},\ \mathrm{P}_{\Sigma_1\cup\Sigma_2\rightarrow\Sigma_i}(\sigma) \in \mathcal{L}_i\big) \wedge \big(\forall\langle a,b\rangle \in \mathcal{S},\ \forall s \in \sigma,\ a \in s \Leftrightarrow b \in s\big)\}$.*

It is possible to prove that these notations preserve the properties of commutativity and associativity of the more traditional notations (although this would imply redefining synchronous sets). For simplicity and where it is obvious, the set $\mathcal{S}$ can be dropped from the notation: $\mathcal{L}_1 \otimes \mathcal{L}_2$.

We introduce one last notion here: *local consistency*. The local consistency operation between two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ builds the smallest language $\mathcal{L}'_1 \subset \mathcal{L}_1$ that maintains $\mathcal{L}'_1 \otimes \mathcal{L}_2 = \mathcal{L}_1 \otimes \mathcal{L}_2$. This operation can be implemented by: $\mathcal{L}'_1 = \mathrm{P}_{\Sigma_1}(\mathcal{L}_1 \otimes \mathcal{L}_2)$.

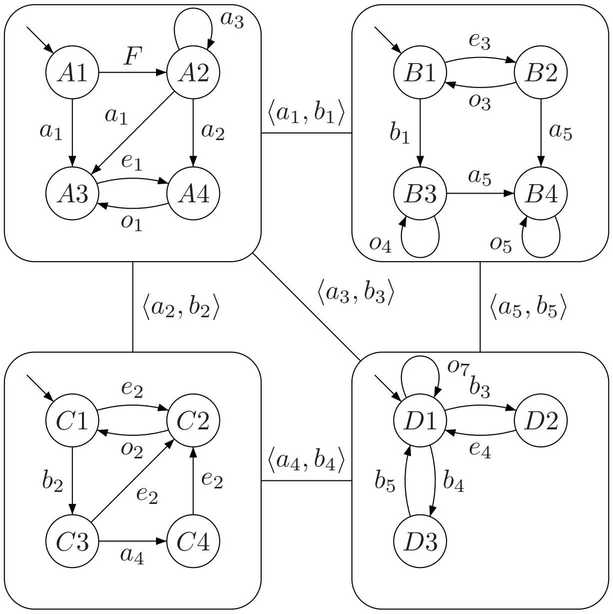

Figure 1: Example of network

Example Figure 1 gives an example of language represented in a distributed fashion. The language $\mathcal{L}$ is defined by four languages $\mathcal{L}_A$ to $\mathcal{L}_D$ that are synchronised through five set of simultaneous events $\langle a_1, b_1\rangle$ to $\langle a_5, b_5\rangle$; each local language is represented by an automaton. The synchronisation of languages $\mathcal{L}_B$ and $\mathcal{L}_D$ is represented on Figure 2.

3 FAULT DIAGNOSIS IN DES

Diagnosis is the reasoning process that determines what happened on a system from observing its behaviour. It helps detecting and identifying faults in a system. We consider model-based diagnosis.

Model We suppose we have a complete model of a system Γ captured by a language Mod on a finite set of events Σ (*i.e.* $Mod \subseteq w(\Sigma)$). The set of faulty events is denoted $\Sigma_F \subseteq \Sigma$.

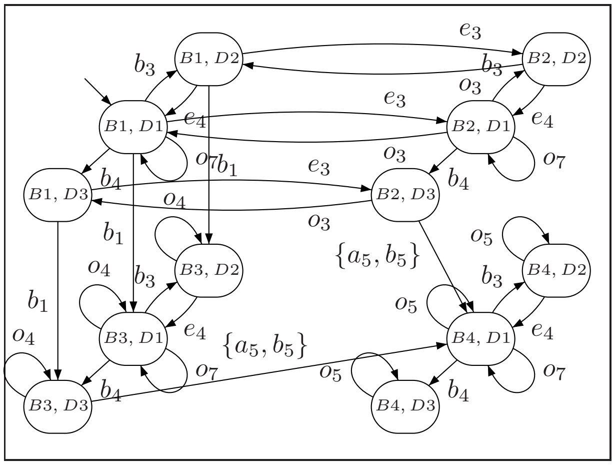

Figure 2: Synchronisation of $\mathcal{L}_B$ and $\mathcal{L}_D$ (we omit curly braces around single symbols for clarity)

Some events generate the emission of an observation. These *observable* events are denoted $\Sigma_o \subseteq \Sigma$. The observations are represented by a language $Obs \subseteq w(\Sigma_o)$. We consider there is no noise on the observations, so that Obs contains only one element that is a sequence of observable events (see (Cordier and Grastien, 2007)).

Diagnoser A diagnoser is an agent in charge of monitoring the observations generated by a system to provide diagnosis reports. A diagnoser dedicated to a specific fault $F \in \Sigma_F$, denoted Δ^F, may return one of the three following results:

1. F-sure: the fault occurrence is asserted;
2. F-safe: the fault occurrence is disproved;
3. F-$\mathrm{ambiguous}$: the fault occurrence is unknown.

By definition, the fault F is considered permanent. For each fault $F \in \Sigma_F$, we define an agent Δ^F responsible for the detection of F. The rest of the article focusses on a single fault event F and we therefore simplify the notation Δ^F to Δ.

Explanatory language The explanatory language is the set of behaviours accepted by the model and consistent with the observations. It can be defined as follows:

$$\mathit{Expl} = Mod \otimes Obs \,. \tag{1}$$

From *Expl*, it is possible to compute the global F-diagnoser Δ_{Mod} associated with F: that is, for any Obs, $\Delta_{Mod}(Obs) =$

$$\begin{cases} \text{F-sure} & \text{if } \forall \sigma \in \mathit{Expl}, \exists s \in \sigma : F \in s \\ \text{F-safe} & \text{if } \forall \sigma \in \mathit{Expl}, \exists s \in \sigma : F \notin s \\ \text{F-ambiguous} & \text{otherwise.} \end{cases} \tag{2}$$

Whichever representation is chosen for languages (automaton, Petri nets, etc.), diagnosis faces the problem of search-space explosion. The reasoning is exponentially complex with the number of components in the system, which makes trivial techniques impossible to apply for systems with a few dozens components. Distributed techniques aim at tackling this issue.

4 DISTRIBUTED APPROACH

Modern technical systems usually consist of components that are each an individual system with simple behaviours, but interacts with other components to produce an overall complex behaviour. We refer to the overall system as a distributed system and model each of its component separately. Let Γ be a distributed system made up of a set of components: $\Gamma = \{\Gamma_1 \dots \Gamma_n\}$. Each component Γ_i can be described by the language Mod_i defined on the alphabet Σ_i. The implicit assumption of fairness is made, whereby components cannot become silent in the long run: on an infinite time-scale, the number of observations generated by a given component is always infinite. Fault events are intrinsic to a component's physical set-up which is responsible for causing failures on the component itself but also causing them to propagate over the system. The occurrence of a fault of type F is considered as an event that can only happen on a component Γ_i : $F \in \Sigma_i \wedge (i \neq j \Rightarrow F \notin \Sigma_j)$. Components in a system communicate through *connections*.

Definition 4 *(Connection) A connection $\mathcal{K}_{ij}$ exists between two components Γ_i and Γ_j if they have a physical or logical link between them that allows the exchange of information about the events occurring in each of them. A synchronous set $\mathcal{S}_{ij}$ can be used as abstract model for a connection $\mathcal{K}_{ij}$ where $\mathcal{S}_{ij} \subseteq \Sigma_i \times \Sigma_j$ and $\mathcal{S}_{ij} = \mathcal{S}_{ji}$.*

We make the assumption that an event can only be shared by two components. The way in which the components of a distributed system are connected defines the global *topology* of the system. The global model of the system is implicitly defined by synchronising the models for all components of the system ($Mod = Mod_1 \otimes \cdots \otimes Mod_n$), hence it is unnecessary to calculate it explicitly. Observations on the system can also be modeled in a decentralised fashion: $Obs = Obs_1 \otimes \cdots \otimes Obs_n$ (Cordier and Grastien, 2007).

The explanatory language Expl_i on a component Γ_i is given by $Mod_i \otimes Obs_i$. The global explanatory language *Expl* is calculated by obtaining the synchronous product of the local languages:

$$\begin{aligned} \mathit{Expl} &= (Mod_1 \otimes \ldots \otimes Mod_n) \otimes (Obs_1 \otimes \ldots \otimes Obs_n) \\ &= (Mod_1 \otimes Obs_1) \otimes \ldots \otimes (Mod_n \otimes Obs_n) \\ &= \mathit{Expl}_1 \otimes \ldots \otimes \mathit{Expl}_n . \end{aligned}$$

Calculating the language *Expl* by synchronising all components is often impossible if the system consists of a large number of components. *Distributed* methods of diagnosis helps avoiding this calculation. We use a junction-tree based implementation (Kan John and Grastien, 2008).

Consider a graph $\mathcal{G} = \langle \Gamma, \mathcal{K} \rangle$ on the components of system Γ where $\mathcal{K}$ is the set of all connections on the system. A junction tree (Huang and Darwiche, 1996) on $\mathcal{G}$ is a pair $(\mathcal{J}, \mathcal{C})$ where $\mathcal{J}$ is a tree and $\mathcal{C}$ is a function that associates each node $\mathcal{N}$ of $\mathcal{J}$ to a *cluster* of

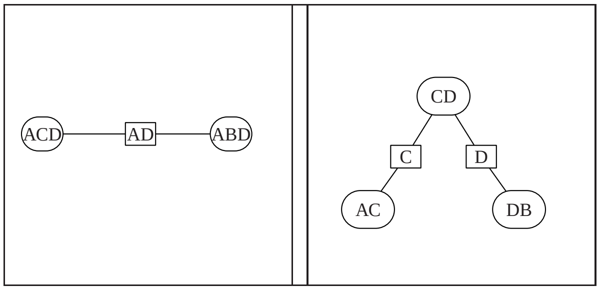

Figure 3: Junction Tree for whole system in Figure 1 (left) and same system with connections $\langle a_1, b_1\rangle$, $\langle a_3, b_3\rangle$ removed (right)

components $\mathcal{C}_i$ (see example figure 3). Moreover, for each connection $\langle i, j\rangle$, there exists a cluster containing the nodes : $\{i, j\} \subseteq \mathcal{C}(\mathcal{N})$. Finally, if two clusters of the tree contains the same node, every cluster between them will contain that node (see the node D between the clusters CD and DB).

To obtain the diagnosis on a distributed system, it is sufficient to calculate the local explanatory language of each cluster and to perform local consistency operations on the junction tree from the leaves to the root. If the root of the tree is chosen to contain the component on which an event occurs, the global diagnosis of the fault is obtained using formula (2) on the local explanatory language of the root.

This method allows us to circumvent the explicit calculation of *Expl*. However, calculating the explanatory language on each cluster is still necessary for each cluster, and the complexity of the representation of this language increases exponentially with the number of components in the cluster. The *tree-width* of the topology is the size of the biggest cluster of its junction tree minus one. This value serves as an *a priori* estimate of the algorithmic cost of diagnosis by this method. Hence, if we limit the tree-width we are potentially able to reduce the cost of diagnosis. We present in the next section a proposed method to handle this.

5 SUB-CONFIGURATION AND ACCURACY

The effectiveness of the diagnosis algorithm using junction trees is directly dependent on the connections between components in the system. We therefore propose to relax some of these connections with the goal of generating a tree on which reasoning can be carried out more effectively.

5.1 Relaxation of connections

The relaxation of connections is formalised with sub-topology and sub-configuration notions.

Definition 5 (Sub-topology) *A* sub-topology $\mathbb{T}$ *on a distributed system* Γ *is a subset of connections* $\mathcal{Y} \subseteq \mathcal{K}$*.*

A sub-topology $\mathbb{T}$ defines a language $\mathcal{L}(\mathbb{T})$ that corresponds to the synchronisation of local languages on the connections of the set $\mathcal{Y}$. We illustrate this using the example in Figure 1. This system consists of four components A to D and five connections $\langle a_i, b_i\rangle$. A possible word on the system is $\{F\}.\{e_3\}.\{a_2, b_2\}.\{a_4, b_4\}.\{a_5, b_5\}.\{o_5\}$ (we note that each a_i is synchronised with a corresponding b_i). We now consider the sub-topology where the connection $\langle a_2, b_2\rangle$ is ignored. The language of this sub-topology contains additional words, including $\{e_3\}.\{b_2\}.\{a_4, b_4\}.\{a_5, b_5\}.\{o_5\}$ (here b_2 appears on its own).

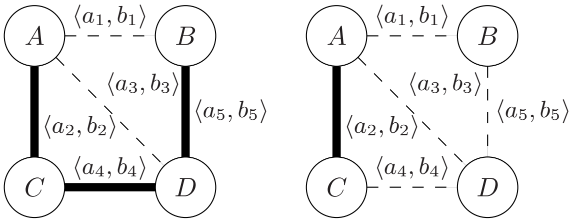
A
B
C
D
$\langle a_1, b_1\rangle$
$\langle a_3, b_3\rangle$
$\langle a_5, b_5\rangle$
$\langle a_2, b_2\rangle$
$\langle a_4, b_4\rangle$
A
B
C
D
$\langle a_1, b_1\rangle$
$\langle a_3, b_3\rangle$
$\langle a_5, b_5\rangle$
$\langle a_2, b_2\rangle$
$\langle a_4, b_4\rangle$

Figure 4: Two different sub-topologies - solid lines represent connections under consideration and dotted lines represent connections that are ignored

Lemma 1 *Words defined on a sub-topology* $\mathbb{T} \subseteq \mathbb{T}'$ *need to satisfy less constraints than those of* $\mathbb{T}'$*:* $\mathcal{L}(\mathbb{T}') \subseteq \mathcal{L}(\mathbb{T})$*, where* $\mathcal{L}$ *represents either the system model or the explanatory language.*

In practice, a sub-topology can isolate components of the system. In that case, it becomes unnecessary to keep track of the observations of those components since the model indicates that they are functioning independently from the other components. This is encompassed by the notion of a sub-configuration.

Definition 6 (Sub-configuration) *A* sub-configuration $\mathbb{C}$ *is a tuple* $(\{\Gamma_{p_1}, \ldots, \Gamma_{p_m}\}, \mathcal{Y}_\mathbb{C}, \overline{\mathcal{Y}_\mathbb{C}})$ *where* $\{\Gamma_{p_1}, \ldots, \Gamma_{p_m}\}$ *is a set of components,* $\mathcal{Y}_\mathbb{C}$ *is a set of connections between the components of* $\mathbb{C}$*, and* $\overline{\mathcal{Y}_\mathbb{C}}$ *is the set of connections between the components of* $\mathbb{C}$ *that are not found in* $\mathcal{Y}_\mathbb{C}$*.*

Figure 4 illustrates two different sub-topologies from the example of Figure 1 (on the left hand side the sub topology is $\{\langle a_2, b_2\rangle, \langle a_4, b_4\rangle, \langle a_5, b_5\rangle\}$ and on the right hand side the sub-topology is $\{\langle a_2, b_2\rangle\}$). The corresponding sub-configuration on the left hand side (resp. on the right hand side) involves the components $\{A, B, C, D\}$ (resp. $\{A, C\}$).

5.2 Diagnosis within a sub-configuration

The basic idea of this paper is to perform diagnosis based on a model (denoted Mod' here) that is simpler than the model Mod. Table 1 represents what can be expected by doing so. Each cell indicates the diagnosis result of model-based diagnosis using the original model Mod compared to the simplified model Mod'. The diagonal (labels ✓) represents the cases where $\Delta_{Mod'}$ returns the same result as the original diagnoser Δ_{Mod}. The cell labeled A shows an accuracy reduction: diagnoser Δ_{Mod} can decide whether a fault occurred while $\Delta_{Mod'}$ cannot. The label × indicates inconsistent cases: the simplified model Mod' is inconsistent with the model Mod which means the diagnoser $\Delta_{Mod'}$ returns inconsistent results with regards to Δ_{Mod}. Regarding this table, it is better to determine simplified models Mod' such that diagnostic results correspond to cells labeled by ✓. Cells labeled

		Δ_{Mod}		
		F-sure	F-safe	F-amb
$\Delta_{Mod'}$	F-sure	✓	×	×
	F-safe	×	✓	×
	F-amb	A	A	✓

Table 1: Comparison between diagnosis results

by A are acceptable in the sense that they only betray loss of accuracy. However, the model Mod' should be chosen such that the cells labelled × are unreachable.

It is easy to demonstrate that model-based diagnosis using, as a simplified model, a sub-topology $\mathbb{T}$ (or its equivalent sub-configuration $\mathbb{C}$) falls in the acceptable category. Indeed, as stated in 1, the generated language by $\mathbb{T}$ always contains the initial language, so the corresponding diagnoser cannot provide inconsistent results but in the worst case less accurate results.

5.3 Accurate diagnoser on $\mathbb{C}$

We define a diagnoser $\Delta_{\mathbb{C}}$ on a sub-configuration $\mathbb{C}$ to be the diagnosis result obtained by using $\mathbb{C}$ as simplified model Mod'. $\Delta_{\mathbb{C}}$ is obtained by synchronising the language defined on $\mathbb{C}$, $\mathcal{L}(\mathbb{C})$, with observation Obs on the system: $\Delta_{\mathbb{C}} = \mathcal{L}(\mathbb{C}) \otimes Obs$.
The challenge is now to determine a sub-configuration $\mathbb{C}$ based on which the diagnoser $\Delta_{\mathbb{C}}$ maintains the accuracy with respect to the global diagnoser Δ_{Mod}. Formally,

Definition 7 (Accuracy) *The diagnoser $\Delta_{\mathbb{C}}$ is said to be* accurate *if for every observable σ_o emitted from the system such that $\Delta_{Mod}(\sigma_o) = \text{F-sure}$, and for every continuing observable σ'_o of the system, there exists a bound $n \in \mathbb{N}$ such that $|\sigma'_o| \geq n$, $\Delta_{\mathbb{C}}(\mathrm{P}_{\mathcal{L}(\mathbb{C})}(\sigma_o.\sigma'_o)) = \text{F-sure}$ (see Figure 5).*

Diagnoser accuracy is possible only under the assumption of observability fairness in the system (see section 4).

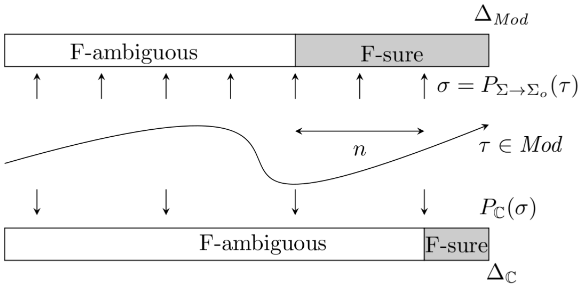

Figure 5: Accurate diagnoser $\Delta_{\mathbb{C}}$.

The main attraction of an accurate diagnoser $\Delta_{\mathbb{C}}$ is its ability to eventually obtain the same result, albeit with a finite delay, as a global diagnoser Δ_{Mod} if the fault F has occurred. In fact, as soon as a fault F has occurred on the system, Δ_{Mod} has two possible answers to explain the current sequence σ_o of observations: either it responds F-ambiguous or F-sure. By the fairness property of the system, $\Delta_{\mathbb{C}}$ also responds as soon as a new observation o is available on $\mathbb{C}$. Let $\sigma''_o.o$ be this finite continuation of σ_o, if $\Delta_{Mod}(\sigma) = \text{F-ambiguous}$, there are two possible scenarios:

1. either the ambiguity is still present $\Delta_{Mod}(\sigma_o\sigma''_o.o) = \text{F-ambiguous}$, then by construction, $\Delta_{\mathbb{C}}(\mathrm{P}_{\mathbb{C}}(\sigma_o\sigma''_o.o)) = \text{F-ambiguous} = \Delta_{Mod}(\sigma_o\sigma''_o.o)$;
2. or the ambiguity is no longer present $\Delta_{Mod}(\sigma_o\sigma''_o.o) = \text{F-sure}$ and then, by waiting a finite number n of observations σ'_o, $\Delta_{\mathbb{C}}(\mathrm{P}_{\mathbb{C}}(\sigma_o\sigma''_o.o\sigma'_o)) = \text{F-sure}$, and therefore in the end, $\Delta_{\mathbb{C}}$ returns the same result as Δ_{Mod} but by only observing $\mathbb{C}$.

5.4 Characterisation of an accurate diagnoser

In order to determine whether the diagnoser $\Delta_{\mathbb{C}}$ is accurate or not for a given sub-configuration $\mathbb{C}$, it is thus sufficient to analyse *a priori* if the sub-configuration $\mathbb{C}$ contains the characteristics that are required to implement an accurate diagnoser on it. Before describing these characteristics, some notations are introduced.

We consider a sub-configuration $\mathbb{C} = \{\{\Gamma_{p_1}, \ldots, \Gamma_{p_m}\}, \mathcal{Y}_{\mathbb{C}}, \overline{\mathcal{Y}_{\mathbb{C}}}\}$. We assume that the fault F has to occur on one of the components $\Gamma_F = \Gamma_{p_i}$ of $\mathbb{C}$. We also introduce the sub-configuration $\mathbb{C}_{\max} = \{\{\Gamma_{p_1}, \ldots, \Gamma_{p_m}\}, \mathcal{Y}_{\mathbb{C}} \cup \overline{\mathcal{Y}_{\mathbb{C}}}, \emptyset\}$ that is associated with $\mathbb{C}$ in which no connection is relaxed. $\mathbb{C}_{\max}$ therefore takes into consideration all connections of the system that involve the components $\{\Gamma_{p_1}, \ldots, \Gamma_{p_m}\}$. The language defining the events generated by the sub-configuration $\mathbb{C}$ (resp. $\mathbb{C}_{\max}$) is denoted $\mathcal{L}_{\mathbb{C}}$ (resp. $\mathcal{L}_{\mathbb{C}_{\max}}$). By definition, $\mathcal{L}_{\mathbb{C}_{\max}} \subseteq \mathcal{L}_{\mathbb{C}}$. In this section, to simplify, Σ is constrained to the set of events of $\mathbb{C}$ (and therefore of $\mathbb{C}_{\max}$). Among the events of Σ we distinguish in particular: the set Σ_o of observable events, the set Σ_r^{ext} of interactive events of $\mathbb{C}$ associated with external relaxed connections (*i.e.* a connection of the system where only one of the components belong to $\mathbb{C}$). Finally, as it will be explained later on, the characterisation of an accurate diagnoser relies on the notion of *traces* and *observable traces*.

Definition 8 (Trace) *Let $F \in \Sigma$ be a fault and $\mathbb{C}$ a sub-configuration, the set of* traces *of F in $\mathbb{C}$ is the language :*

$$\mathcal{T}(\mathbb{C}, F) = \{\tau = s_1....s_m \in \mathcal{L}_{\mathbb{C}}, \exists s_i, F \in s_i\}$$

with $s_i \subseteq \Sigma, i \in \{1, \cdots, m\}$.

Similarly, the complement of $\mathcal{T}(\mathbb{C}, F)$ in $\mathcal{L}_{\mathbb{C}}$ (denoted $\mathcal{T}(\mathbb{C}, \neg F)$) consists of the set of traces where the fault F is not present. Figure 6 illustrates the traces associated to fault F in the sub-configuration consisting only of the component A of Figure 1.

Definition 9 (Observable Trace) *Let $F \in \Sigma$ be a fault and $\mathbb{C}$ a sub-configuration, an* observable trace *of F in $\mathbb{C}$ is a sequence of observable events of the language :*

$$Obs(\mathbb{C}, F) = \mathrm{P}_{\Sigma\to\Sigma_o}(\mathcal{T}(\mathbb{C}, F)).$$

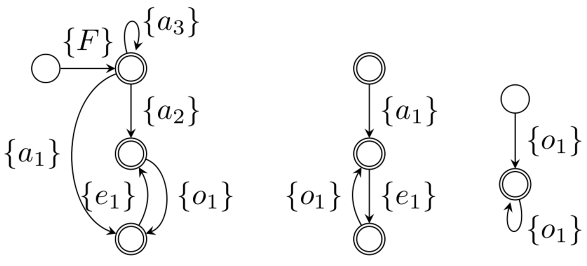

Figure 6: Traces $\mathcal{T}(A, F)$, $\mathcal{T}(A, \neg F)$ and $Obs(A, F)$.

Similarly, $Obs(\mathbb{C}, \neg F)$ represents the set of observable traces of $\mathbb{C}$ where F is not present (see Figure 6).

We first explore the reasons why the diagnoser $\Delta_{\mathbb{C}}$ is not accurate for a given sub-configuration $\mathbb{C}$. We then describe the necessary criteria for making $\Delta_{\mathbb{C}}$ accurate.

What are the sources of inaccuracy in $\mathbb{C}$?

Let $\sigma_o.o$ be an observable sequence of the system which ends in the observable event o from $\mathbb{C}$ and for which the global diagnoser returns $\Delta_{Mod}(\sigma_o.o) =$ F-sure. Let $\sigma'_o.o$ be the observable projection of $\sigma_o.o$ on $\mathbb{C}$. Firstly, the answer $\Delta_{\mathbb{C}}$ to the observation $\sigma'_o.o$ can only be F-sure or F-ambiguous as $\sigma'_o.o$ has to be an observable trace of F in $\mathbb{C}$. Secondly, if $\Delta_{\mathbb{C}}(\sigma'_o.o) =$ F-sure, there is no accuracy problem. There only remains the problematic case of $\Delta_{\mathbb{C}}(\sigma'_o.o) =$ F-ambiguous while $\Delta_{Mod}(\sigma_o.o) =$ F-sure. In this case, $\sigma'_o.o$ is an ambiguous observable trace ($\sigma'_o.o \in Obs(\mathbb{C}, F) \cap Obs(\mathbb{C}, \neg F)$) and this ambiguity is always due to the following situations.

1. The set of components of $\mathbb{C}$ are not sufficiently observable locally and only observations emitted from components external to $\mathbb{C}$ can eliminate the ambiguity (this problem is intrinsic to $\mathbb{C}_{\max}$).
2. There are too many relaxed connections in $\mathbb{C}$. The diagnosis is ambiguous because $\Delta_{\mathbb{C}}$ assumes the existence of behaviours that are not possible in $\mathbb{C}_{\max}$.

The difficulty now lies in determining a criterion on the configuration $\mathbb{C}$ that guarantees that, if the diagnosis of $\Delta_{\mathbb{C}}$ is ambiguous then that of Δ_{Mod} also is. That criterion must guarantee that none of the two situations above hold in the sub-configuration $\mathbb{C}$. As the first situation is intrinsic to $\mathbb{C}_{\max}$ and the second is due to relaxation of connections, we first determine such a criterion on $\mathbb{C}_{\max}$ sub-configurations only.

Detection criterion of the accuracy of $\mathbb{C}_{\max}$

The sequence $\sigma_o.o$, introduced above, represents an observable sequence of Γ and therefore there exists at least one trace τ of Γ such that $Obs(\tau) = \sigma_o.o$. Let $\tau_{int} = \mathrm{P}_{\Sigma_r^{ext}}(\tau)$ be the interactive trace issued from τ and associated to the configuration $\mathbb{C}_{\max}$, the detection criterion depends on the following results.

Property 1 *If there exists in $\mathbb{C}_{\max}$ two traces τ_F and $\tau_{\neg F}$ such that :*

- $\mathrm{P}_{\Sigma_r^{ext}}(\tau_F) = \mathrm{P}_{\Sigma_r^{ext}}(\tau_{\neg F}) = \tau_{int}$
- $\mathrm{P}_{\Sigma_o}(\tau_F) = \mathrm{P}_{\Sigma_o}(\tau_{\neg F}) = \sigma'_o.o$
- $F \in \tau_F \wedge F \notin \tau_{\neg F}$

then $\Delta_{Mod}(\sigma_o.o) =$ F-ambiguous.

Proof : the result is immediate. Considering that τ_F forms part of a global trace that can explain $\sigma_o.o$, the trace $\tau_{\neg F}$ necessarily forms part of another global trace $\sigma_o.o$ (since they have the same observable and interactive projections). Finally, there indeed exist two global traces that explain $\sigma_o.o$, one containing F and one not. □

Property 1 describes the favourable case where there is no accuracy problem (*i.e.* $\Delta_{\mathbb{C}_{\max}}(\sigma'_o.o) = \Delta_{Mod}(\sigma_o.o) =$ F-ambiguous).

Property 2 *If* $\Delta_{Mod}(\sigma_o.o) =$ F-sure *and* $\Delta_{\mathbb{C}_{\max}}(\sigma'_o.o) =$ F-ambiguous *then there exists in $\mathbb{C}_{\max}$ at least two traces τ_F ($F \in \tau_F$) and $\tau_{\neg F}$ ($F \notin \tau_{\neg F}$) such that :*

- $\mathrm{P}_{\Sigma_o}(\tau_F) = \mathrm{P}_{\Sigma_o}(\tau_{\neg F}) = \sigma'.o$,
- $\forall \tau \in \mathcal{T}(\mathbb{C}_{\max}) | \mathrm{P}_{\Sigma_o}(\tau) = \sigma'.o \wedge \mathrm{P}_{\Sigma_r^{ext}}(\tau) = \mathrm{P}_{\Sigma_r^{ext}}(\tau_F) \implies F \in \tau$.

Proof : The first condition stems from the fact that $\Delta_{\mathbb{C}_{\max}}(\sigma'_o.o) =$ F-ambiguous if and only if there exists at least two traces τ_F ($F \in \tau_F$) and $\tau_{\neg F}$ ($F \notin \tau_{\neg F}$) such that $\mathrm{P}_{\Sigma_o}(\tau_F) = \mathrm{P}_{\Sigma_o}(\tau_{\neg F}) = \sigma'_o.o$. The second condition directly stems from the property 1 by contraposition and that allows for the fact that $\Delta_{Mod}(\sigma_o.o)$ can be F-sure. □

Property 2 states that accuracy problems come from both the presence of local faulty and non-faulty traces that emit the same observable sequence but do not interact with the neighbourhood of $\mathbb{C}_{\max}$ in the same manner (second condition of Property 2). Hence the following result, if such a problem occurs a finite number of time, the local diagnoser of $\mathbb{C}_{\max}$ is accurate.

Property 3 *For $\Delta_{\mathbb{C}_{\max}}$ to be accurate, it is sufficient that the set of couples $(\tau_F, \tau_{\neg F})$ defined by property 2 is finite.*

Proof : Consider an observable sequence $\sigma_o.o$ with o emitted from $\mathbb{C}_{\max}$ such that $\Delta_{Mod}(\sigma_o.o) =$ F-sure and let us suppose that $\Delta_{\mathbb{C}_{\max}}(\mathrm{P}_{\Sigma \to \Sigma_o}(\sigma o)) =$ F-ambiguous. If $\Delta_{\mathbb{C}_{\max}}$ is not accurate, then there exists at least one finite suite of observable continuations $\sigma_{o1} o_1$, $\sigma_{o1} o_1 \sigma_{o2} o_2 \ldots$ with o_i emitted from $\mathbb{C}_{\max}$ such that $\Delta_{\mathbb{C}_{\max}}(\mathrm{P}_{\Sigma \to \Sigma_o}(\sigma_o o \sigma_{o1} o_1)) =$ F-ambiguous, $\Delta_{\mathbb{C}_{\max}}(\mathrm{P}_{\Sigma \to \Sigma_o}(\sigma_o o \sigma_{o1} o_1 \sigma_{o2} o_2)) =$ F-ambiguous$\ldots$ hence the presence of an infinite set of couples $(\tau_F, \tau_{\neg F})$ according to property 2. □

Detection criterion of the accuracy of $\mathbb{C}$

The difference between any configuration $\mathbb{C}$ and the associated configuration $\mathbb{C}_{\max}$ is the relaxation of internal connections that leads the diagnoser $\Delta_{\mathbb{C}}$ to consider a set of traces that contains the set of traces of $\mathbb{C}_{\max}$. The consequence in terms of accuracy is the following. Given that $\Delta_{Mod}(\sigma_o.o) =$ F-sure, there necessarily exists a trace $\tau_F \in \mathcal{T}(\mathbb{C}_{\max}, \neg F)$ containing F that forms part of the explanation of $\sigma_o.o$ as described previously. Where $\mathbb{C}$ is concerned, the diagnoser $\Delta_{\mathbb{C}}$ answers not only in terms of the presence or absence of traces $\tau_{\neg F}$ of $\mathcal{T}(\mathbb{C}_{\max}, \neg F)$, producing the same observations as τ_F, but also in terms of the traces $\tau_{\neg F}$ of $\mathcal{T}(\mathbb{C}, \neg F) \setminus \mathcal{T}(\mathbb{C}_{\max}, \neg F)$ producing the same observations but coming from the relaxation of internal connections of $\mathbb{C}$. The consequence is an accuracy criterion for $\Delta_{\mathbb{C}}$ that is identical to that for a configuration $\mathbb{C}_{\max}$ (*i.e.* the property 3) but relies on the following property 4 that extends property 2.

Property 4 *If* $\Delta_{Mod}(\sigma_o.o)$ = F-sure *and* $\Delta_{\mathbb{C}}(\sigma'_o.o)$ = F-ambiguous *then there exists in* $\mathbb{C}_{\max}$ *at least one trace* τ_F *(*$F \in \tau_F$*) and in* $\mathbb{C}$ *one trace* $\tau_{\neg F}$ *(*$F \notin \tau_{\neg F}$*) such that :*

- $\mathrm{P}_{\Sigma_o}(\tau_F) = \mathrm{P}_{\Sigma_o}(\tau_{\neg F}) = \sigma'_o.o$ *,*
- $\forall \tau \in \mathcal{T}(\mathbb{C}_{\max}) | \mathrm{P}_{\Sigma_o}(\tau) = \sigma'_o.o \wedge \mathrm{P}_{\Sigma_r^{ext}}(\tau) = \mathrm{P}_{\Sigma_r^{ext}}(\tau_F) \implies F \in \tau.$

5.5 Verification Algorithm

We are now ready to describe an algorithm that checks whether $\Delta_{\mathbb{C}}$ is accurate or not which relies on the properties described in section 5.4. The first remark is that the diagnoser $\Delta_{\mathbb{C}}$ is only accurate if the diagnoser $\Delta_{\mathbb{C}_{\max}}$ is itself accurate (this comes directly from the definition). The proposed algorithm is described in terms of languages and successively analyses the accuracy of $\Delta_{\mathbb{C}_{\max}}$, then of $\Delta_{\mathbb{C}}$. By consecutive operations of intersection and projection of languages, the algorithm eliminates the traces that do not lead to a problem of accuracy and retains at the end only traces that present problems. If this number of traces is finite, we conclude that $\Delta_{\mathbb{C}}$ is accurate. As stated by properties 2-4, only interactive events Σ_r^{ext} and observable events Σ_o come into consideration in the verification of accuracy. The other type of events are abstracted by projection of traces $\mathcal{T}(F)$ and $\mathcal{T}(\neg F)$ (lines 2–3). With lines 4–5, the objective is to calculate the sources of ambiguity that do not present a problem of accuracy (see property 1), by the intersection $\mathcal{T}(F) \cap \mathcal{T}(\neg F)$ and are thus eliminated from $\mathcal{T}'(F)$. Then, the algorithm checks that there does not exist in the remaining traces of $\mathcal{T}'(F)$ an infinite set of traces (loop detection) whose observable projection is also the same as that of traces coming from $\mathcal{T}(\neg F)$ (line 6). To this end, we calculate the set of observable projections common to $\mathcal{T}(F)$ and $\mathcal{T}(\neg F)$ and by inverse projection find the traces of $\mathcal{T}'(F)$ to preserve. Finally, if $\mathcal{T}'(F)$ is finite (lignes 7–11) then property 3 is verified and $\Delta_{\mathbb{C}_{\max}}$ is accurate. It is sufficient to iterate through the process (lines 12–20) to deal with the non-faulty traces of $\mathcal{L}_{\mathbb{C}}(\neg F) \setminus \mathcal{L}_{\mathbb{C}_{\max}}(\neg F)$ and compare them with the faulty traces of $\mathcal{L}_{\mathbb{C}_{\max}}(F)$ in order to establish if the extension of property 3 is also verified.

Algorithm 1 Verification of the accuracy of $\Delta_{\mathbb{C}}$

```
 1: Input : Sub-configuration $\mathbb{C}$, Fault $F$
 2: $\mathcal{T}(F) \leftarrow \mathrm{P}_{\Sigma \rightarrow \Sigma_o \cup \Sigma_r^{ext}}(\mathcal{L}_{\mathbb{C}_{\max}}(F))$
 3: $\mathcal{T}(\neg F) \leftarrow \mathrm{P}_{\Sigma \rightarrow \Sigma_o \cup \Sigma_r^{ext}}(\mathcal{L}_{\mathbb{C}_{\max}}(\neg F))$
 4: $Ambiguous(F) \leftarrow \mathcal{T}(F) \cap \mathcal{T}(\neg F)$
 5: $\mathcal{T}'(F) \leftarrow \mathcal{T}(F) \setminus (Ambiguous(F))$
 6: $\mathcal{T}'(F) \leftarrow \mathcal{T}'(F) \cap$
        $\mathrm{P}^{-1}_{\Sigma_o \cup \Sigma_r^{ext}}(\mathrm{P}_{\Sigma_o \cup \Sigma_r^{ext} \rightarrow \Sigma_o}(\mathcal{T}(F)) \cap$
        $\mathrm{P}_{\Sigma_o \cup \Sigma_r^{ext} \rightarrow \Sigma_o}(\mathcal{T}(\neg F)))$
 7: if $\mathcal{T}'(F)$ is finite then
 8:     {Property 2 does not occur indefinitely.}
 9:     if $\mathbb{C} = \mathbb{C}_{\max}$ then
10:         The diagnoser $\Delta_{\mathbb{C}}$ is accurate
11:     else
12:         $\mathcal{T}'(\neg F) \leftarrow \mathrm{P}_{\Sigma \rightarrow \Sigma_o \cup \Sigma_r^{ext}}(\mathcal{L}_{\mathbb{C}}(\neg F) \setminus$
                        $\mathcal{L}_{\mathbb{C}_{\max}}(\neg F))$
13:         $\mathcal{T}'(F) \leftarrow \mathcal{T}(F) \setminus (Ambiguous(F))$
14:         $\mathcal{T}'(F) \leftarrow \mathcal{T}'(F) \cap$
                $\mathrm{P}^{-1}_{\Sigma_o \cup \Sigma_r^{ext}}(\mathrm{P}_{\Sigma_o \cup \Sigma_r^{ext} \rightarrow \Sigma_o}(\mathcal{T}(F)) \cap$
                $\mathrm{P}_{\Sigma_o \cup \Sigma_r^{ext} \rightarrow \Sigma_o}(\mathcal{T}'(\neg F)))$
15:         if $\mathcal{T}'(F)$ is finite then
16:             {Property 4 does not occur.}
17:             The diagnoser $\Delta_{\mathbb{C}}$ is accurate
18:         else
19:             Problem of inaccuracy of $\Delta_{\mathbb{C}}$ to occur due
                to relaxed internal connections
20:         end if
21:     end if
22: else
23:     The accuracy of $\Delta_{\mathbb{C}}$ cannot be demonstrated at
        this stage
24: end if
```

5.6 Illustrative Example

Going back to the example in Figure 1, if $\mathbb{C}$ only contains component A, then $\mathbb{C} = \mathbb{C}_{\max}$. In this case, $\Delta_{\mathbb{C}}$ still returns an ambiguous result. There exists an infinite number of traces for which the fault F has occurred and whose interactive behaviour is different from that of traces for which F has not occurred (these traces being with $\{F\}.\{a3\}.\cdots$ or $\{F\}.\{a2\}.\cdots$), thus property 2 is verified an infinite number of times. Note also that the other traces of F beginning with $\{F\}.\{a1\}.\cdots$ have the same interactive and observable projection as the traces in which F has not occurred. These traces are intrinsically ambiguous (see line 4 of algorithm 1). We now consider the sub-configuration $\mathbb{C} = \{\{A, B, C, D\}, \{\langle a_2, b_2\rangle, \langle a_4, b_4\rangle, \langle a_5, b_5\rangle\}, \{\langle a_1, b_1\rangle, \langle a_3, b_3\rangle\}\}$ (see Figure 4). $\mathbb{C}_{\max}$ is necessarily accurate here since $\mathbb{C}_{\max}$ is the complete system in this simple example, $\Sigma_r^{ext} = \emptyset$, and in this case $\mathcal{T}'(F)$ (line 7) that results from this algorithm is empty by construction. It remains to see if the relaxation of connections $\{\langle a_1, b_1\rangle, \langle a_3, b_3\rangle\}$ induces an accuracy problem. It then becomes sufficient to note that the relaxation does not cause an increase in the number of observable traces of $\neg F$ and therefore that the remaining set $\mathcal{T}'(F)$ is also empty (line 15). Relaxing connections $\{\langle a_1, b_1\rangle, \langle a_3, b_3\rangle\}$ is thus interesting as $\Delta_{\mathbb{C}}$ is accurate.

6 CHOOSING A SUB-CONFIGURATION

We discuss now how to choose a sub-configuration minimizing the cost of diagnosis (defined by the tree width) while ensuring an accurate diagnosis. A sub-topology $\mathbb{T}$ is *better* than another topology $\mathbb{T}'$ if the accuracy associated with $\mathbb{T}$ is stronger than the accuracy associated with $\mathbb{T}'$, or both accuracies are identical but the tree width of $\mathbb{T}$ is smaller than in $\mathbb{T}'$. To find an optimal sub-topology, we have to explore the set of sub-topologies $\Upsilon = 2^{\mathcal{K}}$ defined as the power set of the connections $\mathcal{K}$ in the system. The partially-ordered set $\langle \Upsilon, \subseteq, \supseteq \rangle$ forms a lattice. The cost and the accuracy have monotonicity properties in this lattice. Indeed, if $\mathbb{T} \subseteq \mathbb{T}'$, then

- since $\mathbb{T}'$ is a sub-topology of $\mathbb{T}$, the diagnosis with $\mathbb{T}$ is equal to or more accurate than the one with $\mathbb{T}'$, and
- any junction tree of $\mathbb{T}$ is also a junction tree of $\mathbb{T}'$, and the tree width of $\mathbb{T}'$ is equal to or smaller than that of $\mathbb{T}$.

These properties allow for an efficient search in Υ. A possible approach is to start from the physical topology and to remove connections as long as the accuracy is not affected. We work in a context with a huge number of components – possibly thousands – and expect to build sub-topologies with very small tree width – several units at most. Therefore, we recommend to start from the empty topology $\mathbb{T}_{\perp} = \emptyset$ and incrementally add connections; when an accurate sub-topology was determined, it is possible to refine it by removing connections that were added unnecessarily. This is illustrated Algorithm 2.

Algorithm 2 Exploration of Υ

```
1: Input: $\Gamma$, $F$
2: $\mathbb{T} := \emptyset$
3: while $\mathbb{T}$ is not accurate, do
4:     Add a connection to $\mathbb{T}$.
5: end while
6: while $\exists c \in \mathbb{T}$ s.t. $\mathbb{T} \setminus \{c\}$ is accurate, do
7:     Remove $c$ in $\mathbb{T}$.
8: end while
9: Return $\mathbb{T}$
```

It is possible to improve the exploration of Υ as follows:

- The accuracy testing may generate an explanation for non accuracy, and indicate which connections of $\mathcal{K} \setminus \mathbb{T}$ are responsible for non accuracy. The connection added line 4 may be chosen in this set of connections.
- When the junction tree of $\mathbb{T}$ is also a junction tree for $\mathbb{T}' \supseteq \mathbb{T}$, it is possible to test the accuracy immediately on $\mathbb{T}'$ since the cost associated with $\mathbb{T}$ and $\mathbb{T}'$ are identical.

The algorithm proposed here leads to a local optimal if the accuracy testing (line 3) is correct; if the condition for accuracy is sufficient but not necessary, the result may be not locally optimal.

7 CONCLUSION

This article proposes an original approach to reduce the complexity of the diagnosis of large discrete event systems by ignoring some connections in the system. Our work can be seen as a particular case of abstraction, similar to what is presented in (Sachenbacher and Struss, 2005).

We note that to choose a sub-configuration minimising the cost of diagnosis, we do not put a bound on the delay required for the sub-configuration to become accurate. The fairness assumption allows us to predict that in most cases it will be a reasonable delay. A way around the problem is to introduce a bounded delay in the definition of accuracy. This would imply that the sub-configuration to choose might be bigger, thus needing a trade-off between sub-configuration size and delay. This is part of future extensions.

Other future works include refining the cost function with additional factors such as the total number of nodes in the junction tree, the tree shape, the proportion of observable events in nodes, the longest line in the tree, etc. The accuracy criterion could also be improved, with the Boolean result replaced by a real value that could allow for a trade-off between accuracy and cost. Another interesting improvement is to consider several faults. Each fault can be diagnosed by a junction tree, but those trees may include identical nodes. The question is then how to combine these trees.

REFERENCES

(Cassandras and Lafortune, 1999) C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

(Cordier and Grastien, 2007) M.-O. Cordier and A. Grastien. Exploiting independence in a decentralised and incremental approach of diagnosis. In M. Veloso, editor, *Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 292–297. AAAI press, 2007.

(Huang and Darwiche, 1996) C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

(Kan John and Grastien, 2008) P. Kan John and A. Grastien. Local consistency and junction tree for diagnosis of discrete-event systems. In *European Conference on Artificial Intelligence (ECAI-08)*, 2008.

(Pencolé and Cordier, 2005) Y. Pencolé and M.-O. Cordier. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence (AIJ)*, 164:121–170, 2005.

(Pencolé *et al.*, 2006) Y. Pencolé, D. Kamenetsky, and A. Schumann. Towards low-cost fault diagnosis in large component-based systems. In *Sixth IFAC Symposium on Fault Detection, Supervision and Safety of Technical PRocess*, 2006.

(Sachenbacher and Struss, 2005) M. Sachenbacher and P. Struss. Task-dependent qualitative domain abstraction. *Artificial Intelligence (AIJ)*, 162(1–2):121–143, 2005.

(Schumann *et al.*, 2004) A. Schumann, Y. Pencolé, and S. Thiébaux. Symbolic models for diagnosing discrete-event systems. In *Sixteenth European Conference on Artificial Intelligence (ECAI'04)*, 2004.