# Diagnosticabilité des Systèmes à Évènements Discrets: Une Nouvelle Variante de l'Approche Diagnostiqueur

Abderraouf Boussif, Mohamed Ghazel

▶ **To cite this version:**

Abderraouf Boussif, Mohamed Ghazel. Diagnosticabilité des Systèmes à Évènements Discrets: Une Nouvelle Variante de l'Approche Diagnostiqueur. MSR 2017 - 11ème Colloque sur la Modélisation des Systèmes Réactifs, Nov 2017, Marseille, France. 15p. ⟨hal-01647847⟩

**HAL Id: hal-01647847**

**https://hal.science/hal-01647847v1**

Submitted on 24 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

# Diagnosticabilité des Systèmes à Évènements Discrets: Une Nouvelle Variante de l'Approche Diagnostiqueur

Abderraouf Boussif[1,2] and Mohamed Ghazel[1,2]

[1] Univ. Lille Nord de France, F-59000 Lille, France
[2] IFSTTAR, Cosys/Estas, F-59650 Villenveuve d'Ascq, France
{abderraouf.boussif,mohamed.ghazel}@ifsttar.fr

**Résumé**

Dans ce papier, nous nous intéressons à l'analyse de la diagnosticabilité des systèmes à évènements discrets modélisés par des automates à états finis. En particulier, une variante de l'approche diagnostiqueur initiée par Sampath et co. [1, 2] est présentée. Cette variante repose sur une nouvelle structure qui consiste à séparer explicitement les états normaux de ceux fautifs à l'intérieur de chaque nœud du diagnostiqueur. Une telle distinction permet de suivre séparément l'évolution des traces normales et fautives dans le diagnostiqueur. Différentes caractéristiques de la nouvelle structure sont ensuite exploitées pour (*i*) raffiner la condition nécessaire et suffisante de la diagnosticabilité [1], (*ii*) développer une nouvelle condition nécessaire vérifiable directement sur le diagnostiqueur sans revenir au modèle, (*iii*) proposer une version simplifiée de la condition nécessaire et suffisante, et enfin (*iv*) développer une procédure systématique pour l'analyse de la diagnosticabilité basée sur un algorithme de vérification à la volée. L'évaluation de cette approche est faite à travers une série d'expérimentations et de comparaisons avec des approches classiques de référence.

## 1 Introduction

La diagnosticabilité représente une propriété essentielle dans la phase de conception des systèmes sûrs de fonctionnement. En effet, cette proporiété garantit la détection, l'identification et la localisation des fautes dans des délais finis. Dans le formalisme des systèmes à évènements discrets (SED) [2], la diagnosticabilité représente la capacité de la fonction de diagnostic (i.e., le diagnostiqueur) à déterminer, *avec certitude*, la présence (*ou non*) d'un type de fautes (*prédéfini*) à partir d'une séquence (*finie*) d'évènements observés [1].

Bien que récente comparée à la notion de diagnostic, la diagnosticabilité a été largement étudiée par les deux communautés du diagnostic : la communauté DX (pour *Data eXpert system*), issue de l'intelligence artificielle et la communauté FDI (pour *Fault Detection and Isolation*) issue de l'automatique (voir le travail présenté dans [3]).

Dans la littérature relative au diagnostic des SED, on distingue principalement deux [1] familles d'approches pour l'analyse de la diagnosticabilité : les approches dites à base de diagnostiqueurs [1–5], et les approches à base de vérificateurs (*verifiers*) [6–8]. La première famille d'approches est basée sur la construction d'un automate déterministe (*augmenté par des étiquettes associées aux états du système*) appelé *diagnostiqueur*. Ces approches permettent l'analyse de la diagnosticabilité et pour les modèles diagnosticables, les diagnostiqueurs sont utilisés pour assurer le diagnostic en ligne. Les approches à base de vérificateurs sont, quant à elles, basées sur la construction d'automates non-déterministes appelés *verifiers* ou *twin-plant*. Ces approches ont l'avantage d'analyser la diagnosticabilité avec une complexité (théorique) réduite comparée aux approches à base de diagnostiqueurs. Cependant, elles ne fournissent pas de moyen pour le diagnostic en ligne (dû à la nature non-déterministe des vérificateurs).

1. D'autres familles d'approches existent mais elles sont un peu loin de notre contexte d'étude.

Dans cet article, nous nous intéressons à la diagnosticabilité des SED modélisés par des automates à états finis (AF) avec les approches dites '*à base de diagnostiqueurs*'. Nous proposons une variante du diagnostiqueur classique de Sampath et co. [1, 2] avec une nouvelle structure qui distingue explicitement l'ensemble des états normaux de l'ensemble des états fautifs à l'intérieur de chaque nœud du diagnostiqueur. Nous montrons qu'une telle structure permet un suivi efficace de l'évolution des traces normales et fautives à travers le diagnostiqueur. A partir des différentes caractéristiques de cette structure, nous établissons quelques résultats théoriques pertinents pour l'analyse de la diagnosticabilité :

1. Nous proposons, tout d'abord, une version raffinée de la condition nécessaire et suffisante de la diagnosticabilité introduite dans [1] et utilisée par la plupart des approches à base de diagnostiqueurs. Dans cette version, nous prouvons que l'une des deux conditions, pour qu'un cycle $F$-incertain soit $F$-indéterminé, est toujours satisfaite.
2. Nous développons, en suite, une condition nécessaire pour la diagnosticabilité qui peut être vérifiée directement sur le diagnostiqueur sans revenir au modèle du système. Cette condition est d'autant plus intéressante pour les techniques de vérification à la volée de la diagnosticabilité.
3. Nous proposons, au final, une formulation simple de la condition nécessaire et suffisante de la diagnosticabilité basée sur la notion de *séquence indicative* associées au cycle ambigu.

D'un point de vue pratique, nous proposons une procédure systématique pour l'analyse de la diagnosticabilité (sans passer par un modèle intermédiaire comme dans [1]), qui est implémentée avec un algorithme de vérification à la volée. L'algorithme à la volée permet de construire notre diagnostiqueur et d'analyser les différentes conditions de diagnosticabilité en parallèle. Dans le but d'évaluer l'efficacité et la mise à l'échelle de notre approche, nous présentons une étude expérimentale et comparative avec des approches de référence.

Le reste de cet article est organisé comme suit : Dans la section 2, nous introduisons le formalisme de modélisation et nous rappelons la définition formelle de la diagnosticabilité. La section 3 présente la variante du diagnostiqueur. Nous détaillons, par la suite, les résultats théoriques du papier dans la section 4. Une étude expérimentale est présentée en section 5. L'extension de l'approche pour traiter le cas des classes de fautes multiples fait l'objet de la section 6. Nous terminerons avec une conclusion et des perspectives dans la section 7.

# 2 Préliminaires

## 2.1 Modélisation des SED

Nous nous intéressons dans ce papier aux SEDs modélisés par des automates à états finis $G = \langle X, \Sigma, \delta, x_0 \rangle$, où $X$ est un ensemble fini d'états ; $\Sigma$ est un ensemble fini d'évènements ; $\delta : X \times \Sigma \to 2^X$ est la relation de transition, et $x_0 \in X$ est l'état initial. $\langle x, \sigma, x' \rangle \in X \times \Sigma \times X$ est une *transition* si $x' \in \delta(x, \sigma)$. Nous considérons que le modèle $G$ produit un langage régulier clos par préfixe $L \subseteq \Sigma^*$ où $\Sigma^*$ est la fermeture de Kleene de $\Sigma$.

Nous considérons que le système est partiellement observable, i.e., l'ensemble des évènements $\Sigma$ est réparti en deux sous-ensembles disjoints $\Sigma_o$ et $\Sigma_u$ où $\Sigma_o$ est l'ensemble des évènements observables et $\Sigma_u$ est l'ensemble des évènements inobservables ($\Sigma = \Sigma_o \biguplus \Sigma_u$). Pour une étude du diagnostic, on considère que les fautes sont également des évènements inobservables et on les regroupe dans l'ensemble $\Sigma_f$ ($\Sigma_f \subseteq \Sigma_u$). L'ensemble des évènements fautifs peut être

partitionné en plusieurs sous-ensembles représentant les différentes classes de fautes. Nous avons ainsi $\Sigma_f = \Sigma_{f_1} \biguplus \Sigma_{f_2} \biguplus \ldots \biguplus \Sigma_{f_n}$ avec $\Sigma_{f_i}$ $(i = 1, \ldots, n)$ est la classe de fautes $f_i$.

Une séquence d'évènements $s = \sigma_1\sigma_2 \ldots \sigma_n$, où $\sigma_i \in \Sigma$ est associée avec une séquence d'états $\pi = (x_1, x_2, \ldots, x_{n+1})$ si $\forall\, 0 < i \leq n, x_{i+1} \in \delta(x_i, \sigma_i)$. La fonction de transition $\delta$ peut être étendue à une séquence d'évènements, i.e., $x_{n+1} \in \delta(x_1, s)$. On note par $s_i$ le $i^{eme}$ évènement de la séquence $s$ et par $L/s$ l'ensemble des suffixes de $s$ dans $L$, i.e., $L/s := \{t \in \Sigma^* \,|st \in L\}$.

Nous désignons par $\psi(\Sigma_f)$ l'ensemble des séquences d'évènements dans $L$ qui se terminent avec un évènement fautif dans $\Sigma_f$, c'est à dire, $\psi(\Sigma_f) := \{s.\sigma_f \ \in L \ : \sigma_f \in \Sigma_f\}$. Soit $\sigma \in \Sigma$ et $s \in \Sigma^*$, la notation $\sigma \in s$ indique que $\exists\, 1 \leq i \leq |s| : s_i = \sigma$. Par abus de notation, on écrit $\Sigma_f \in s$ pour exprimer le fait que $\exists\, \sigma_f \in \Sigma_f$ tel que $\sigma_f \in s$.

On définit la *fonction de projection* sur l'ensemble des évènements observables par $P : \Sigma^* \to \Sigma_o^*$, i.e., $P$ supprime tous les évènements inobservables d'une séquence d'évènements donnée. D'une manière générale, $P(\sigma) = \sigma$ si $\sigma \in \Sigma_o$ et $P(\sigma) = \epsilon$ si $\sigma \in \Sigma_u$ ; $P(s.\sigma) = P(s)P(\sigma)$ avec $s \in \Sigma^*$ et $\sigma \in \Sigma$. On étend la fonction de projection aux langages par $P(L) = \{P(s) \mid s \in L\}$. Le langage $P(L)$ représente donc le comportement observable du système $G$. Inversement, on définit la *fonction de projection inverse* $P_L^{-1}$ telle que $P_L^{-1}(y) = \{s \in L \mid P(s) = y\}$.

### 2.2 Définition de la Diagnosticabilité

Dans le présent article, nous considérons un seul mode de défaillance, ce qui revient à se donner une seule classe de fautes, notée $\Sigma_f$. La généralisation à $m$ modes de défaillances sera discutée dans la section 6. Nous supposons également que le modèle $G$ satisfait les deux hypothèses classiques du diagnostic des SED :

($H_1$) *Le langage de $G$ est vivant, i.e. il existe une transition à partir de tout état $x$ de $X$ ;*

($H_2$) *Le modèle ne contient pas de cycle formé exclusivement d'évènements inobservables.*

La définition formelle de la diagnosticabilité a été introduite dans [1] comme suit :

**Définition 1.** *Un langage $L$ préfixe clos et vivant est dit diagnosticable par rapport à une fonction de projection $P$ et un ensemble de fautes $\Sigma_f$ si et seulement si :*

$$(\exists n \in \mathbb{N}\, [\forall s \in \Psi(\Sigma_f)]\, (\forall t \in L/s)\, [|t| \geq n \Rightarrow D]$$

*telle que la condition de diagnosticabilité $D$ est : $\omega \in P_L^{-1}[P(s.t)] \Rightarrow \Sigma_f \in \omega$.* ◇

Cette définition signifie qu'un langage $L$ est diagnosticable *si et seulement si*, pour toute séquence d'évènements $s$ se terminant par un évènement fautif et toute continuation $t$ de $s$ suffisamment longue ($|t| \geq n$), alors toute autre séquence d'évènements $\omega$ ayant la même projection observable ($P(s.t) = P(\omega)$), contient nécessairement un évènement fautif de $\Sigma_f$.

**Exemple 1.** *On considère l'AF $G$ de la figure 1. $\Sigma_o = \{a, b, c, d, e\}$ et $\Sigma_u = \Sigma_f = \{f\}$ sont respectivement les ensembles d'évènements observables et inobservables (qui est aussi l'ensemble des évènements fautifs). Le modèle $G$ est diagnosticable parce qu'il est possible de détecter l'occurrence de l'évènement fautif $f$ dans un délai fini (*6 *évènements après son occurrence).*

## 3 Une Nouvelle Variante du diagnostiqueur

Avant de présenter la variante du diagnostiqueur initialement proposé dans [2], nous introduisons les notions suivantes :

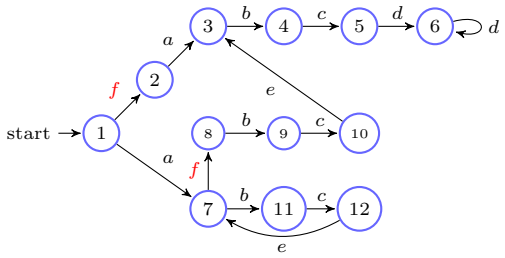



FIGURE 1 – Modèle $G$ de l'exemple 1

1. $Enable_{\Sigma}(x) = \{\sigma \in \Sigma | \delta(x,\sigma) \neq \emptyset\}$ est l'ensemble des évènements appartenant à $\Sigma$ et autorisés à partir de l'état $x$. La généralisation pour un sous-ensemble d'états $X' \subseteq X$ et un sous-ensemble d'évènements $\Sigma' \subseteq \Sigma$ est désignée par l'ensemble $Enable_{\Sigma'}(X') = \{\sigma \in \Sigma' | \exists\, x \in X' : \delta(x,\sigma) \neq \emptyset\}$, i.e. $Enable_{\Sigma'}(X') = \bigcup_{x \in X'} Enable_{\Sigma'}(x)$.
2. $Img(X',\sigma) = \bigcup_{x \in X'} \delta(x,\sigma)$ où $\sigma \in \Sigma$ est la généralisation de la fonction de transition $\delta$ pour un sous-ensemble d'états $X' \subseteq X$. Nous désignons par $Img(X',\Sigma') = \bigcup_{x \in X'} \bigcup_{\sigma \in \Sigma'} \delta(x,\sigma)$ la généralisation de $\delta$ pour un sous-ensemble d'évènements $\Sigma' \subseteq \Sigma$.
3. $Reach_{\Sigma_u}(x) = \{x\} \cup \{x' \in X | \exists\, t \in \Sigma_u^*, x' \in \delta(x,t)\}$ représente l'ensemble des états accessibles par l'exécution des séquences d'évènements inobservables à partir de l'état $x$ (*the unobservable reachability*). La généralisation de cette notion pour un sous-ensemble d'états est $Reach_{\Sigma_u}(X') = \bigcup_{x \in X'} Reach_{\Sigma_u}(x)$.

## 3.1 La Structure Générale des Nœuds

Un noeud dans les diagnostiqueurs classiques [1, 2] représente un ensemble d'états du système qui peuvent être normaux et fautifs. Dans la variante du diagnostiqueur que nous proposons, on sépare explicitement l'ensemble des états normaux (noté $\mathcal{X}_N$) et l'ensemble des états fautifs (noté $\mathcal{X}_F$) à l'intérieur de chaque nœud. Il est possible que certains états fautifs de $\mathcal{X}_F$ soient accessibles à partir de certains états normaux de $\mathcal{X}_N$ dans le même nœud sous l'occurrence d'évènements fautifs. Ces transitions fautives sont appelées transitions transversales. La figure 2 représente la structure générale d'un nœud de notre diagnostiqueur.

De la même façon que les diagnostiqueurs classiques, il existe trois types de nœuds :

— **Un nœud N-certain** : est un nœud qui a l'ensemble d'états fautifs vide ($\mathcal{X}_F = \emptyset$) ;
— **Un nœud F-certain** : est un nœud qui a l'ensemble d'états normaux vide ($\mathcal{X}_N = \emptyset$) ;
— **Un nœud F-incertain** : est un nœud où les deux ensembles d'états sont non vides, i.e., $\mathcal{X}_N \neq \emptyset$ et $\mathcal{X}_F \neq \emptyset$.

Il existe différentes possibilités qu'une transition observable peut prendre à partir d'un nœud, e.g. l'évènement observable $\sigma_2$ sortant du nœud $a$ peut être autorisé par l'ensemble des états normaux, où par l'ensemble des états fautifs ou bien par les deux ensembles.

Pour simplifier la notation, nous désignons par $a.\mathcal{X}_N$ (resp. $a.\mathcal{X}_F$) l'ensemble d'états normaux (resp. fautifs) correspondant à un nœud $a$ dans le diagnostiqueur. Nous utilisons aussi une variable booléenne $\mathcal{B}_a$ pour indiquer l'existence de transitions fautives transversales dans $a$, i.e., $\mathcal{B}_a = \textit{vrai}$ si au moins une transition fautive existe, sinon $\mathcal{B}_a = \textit{faux}$.

## 3.2 La Construction du Diagnostiqueur

Pour un modèle de système $G$, nous définissons le diagnostiqueur comme suit :

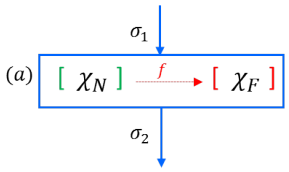



FIGURE 2 – La structure générale d'un nœud

**Définition 2.** *Soit $G = \langle X, \Sigma, \delta, x_0\rangle$ un modèle de système à diagnostiquer. Le diagnostiqueur associé à $G$ est un automate à états déterministe $\mathcal{D} = \langle \Gamma, \Sigma_o, \delta_{\mathcal{D}}, \Gamma_0\rangle$, avec :*

1. *$\Gamma$ est l'ensemble fini de nœuds ;*
2. *$\Gamma_0$ est le nœud initial du diagnositqueur avec :*
   *a) $\Gamma_0.\mathcal{X}_N = Reach_{\Sigma_u \setminus \Sigma_f}(x_0)$ ;*
   *b) $\Gamma_0.\mathcal{X}_F = Reach_{\Sigma_u}(Img(\Gamma_0.\mathcal{X}_N, \Sigma_f))$.*
3. *$\delta_{\mathcal{D}} : \Gamma \times \Sigma_o \to \Gamma$ est la relation de transition, définie comme suit : $\forall a, a' \in \Gamma$, $\sigma \in \Sigma_o$ :*
   *$a' = \delta_{\mathcal{D}}(a, \sigma) \Leftrightarrow$*
   *$a'.\mathcal{X}_N = Reach_{\Sigma_u \setminus \Sigma_f}(Img(a.\mathcal{X}_N, \sigma)) \wedge a'.\mathcal{X}_F = Reach_{\Sigma_u}(Img(a'.\mathcal{X}_N, \Sigma_f) \cup Img(a.\mathcal{X}_F, \sigma))$*

Le diagnostiqueur $\mathcal{D}$ est construit comme suit : soit $a$ un nœud du diagnostiqueur et $\sigma$ un évènements observable. Le nœud successeur $a'$ est calculé selon les règles suivantes :

1. Si $\sigma \in Enable(a.\mathcal{X}_N) \cap Enable(a.\mathcal{X}_F)$, alors :
   - $a'.\mathcal{X}_N = Reach_{\Sigma_u \setminus \Sigma_f}(Img(a.\mathcal{X}_N, \sigma))$.
   - $a'.\mathcal{X}_F = Reach_{\Sigma_u}(Img(a'.\mathcal{X}_N, \Sigma_f) \cup Img(a.\mathcal{X}_F, \sigma))$.
2. Si $\sigma \in Enable(a.\mathcal{X}_N) \setminus Enable(a.\mathcal{X}_F)$, alors :
   - $a'.\mathcal{X}_N = Reach_{\Sigma_u \setminus \Sigma_f}(Img(a.\mathcal{X}_N, \sigma))$.
   - $a'.\mathcal{X}_F = Reach_{\Sigma_u}(Img(a'.\mathcal{X}_N, \Sigma_f))$.
3. Si $\sigma \in Enable(a.\mathcal{X}_F) \setminus Enable(a.\mathcal{X}_N)$, alors :
   - $a'.\mathcal{X}_N = \emptyset$.
   - $a'.\mathcal{X}_F = Reach_{\Sigma_u}(Img(a.\mathcal{X}_F, \sigma))$.

Étant donné que tous les successeurs d'un nœud $F$-certain sont aussi des nœuds $F$-certain, alors il n'est pas nécessaire (*d'un point de vue analyse de la diagnostiquabilité*) de les construire (les sub-séquences d'un nœud $F$-certain). En effet, l'analyse de la diagnosticabilité concerne essentiellement le comportement ambigu (i.e., les cycles incertains et indéterminés [1]). En plus, du point de vue de diagnostic en ligne, on peut être certain qu'une fois un nœud $F$-certain est atteint, le système exécutera indéfiniment un comportement défaillant.

Afin de bien illustrer la procédure de construction du diagnostiqueur, nous considérons encore le modèle $G$ (figure 1). Le diagnostiqueur $\mathcal{D}$ correspondant à $G$ est représenté sur la figure 3(b). Le nœud initial ($a_0$) est composé de l'état initial du modèle $G$ (état 1) et l'état 2 accessible à partir de l'état 1 sur l'occurrence de l'évènement fautif $f$. Donc, on a $a_0.\mathcal{X}_N = \{1\}$, $a_0.\mathcal{X}_F = \{2\}$ et $\mathcal{B}_{a_0} = vraie$. Le nœud $a_1$ est le nœud successeur de $a_0$ sur l'occurrence de l'évènement observable $a$. Étant donné que l'évènement $a$ est autorisé par les deux ensembles (normaux et fautifs) du nœud $a_0$, alors $a_1$ certainement contient les deux ensembles normaux et fautifs non vides. $a_1.\mathcal{X}_N$ contient les états accessibles à partir de $a_0.\mathcal{X}_N$ sur l'occurrence de l'évènement $a$ et les séquences inobservables (non fautives) qui suivent l'évènement $a$ ($a_1.\mathcal{X}_N = \{7\}$). L'ensemble $a_1.\mathcal{X}_F$ est composé des états accessibles à partir de $a_0.\mathcal{X}_F$ sur

l'occurrence de l'évènement $a$ et les séquences inobservables qui le suivent. En plus, il contient les états accessibles à partir de $a_1.\mathcal{X}_N$ sous l'occurrence des évènements fautifs (i.e., les transitions fautives transversales). Donc, $a_1.\mathcal{X}_F = \{8, 3\}$ et $\mathcal{B}_{a_1} = vraie$. Le reste des nœuds sont construits de la même manière. On précise que les lignes discontinues dans la figure 3(b) sont ajoutées uniquement pour l'illustration.

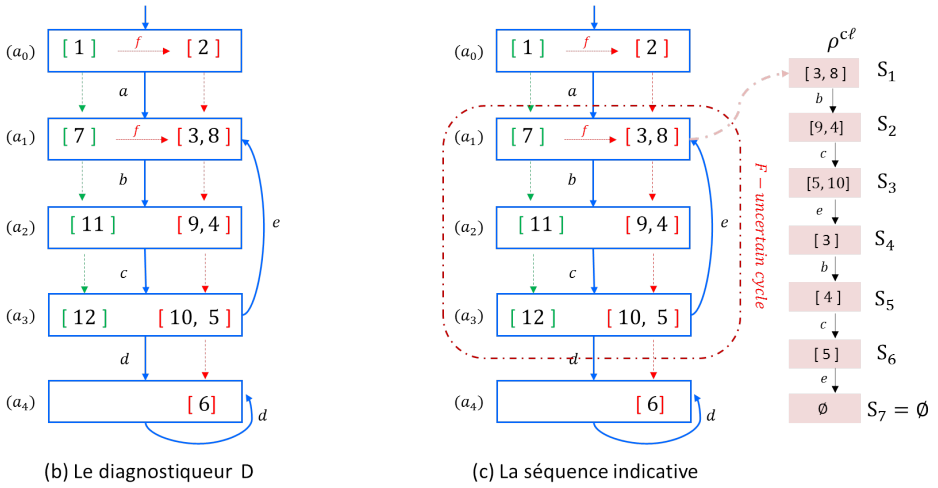



FIGURE 3 – Construction du diagnostiqueur et analyse de la diagnosticabilité

# 4 Analyse de la Diagnosticabilité

La nouvelle structure des nœuds que nous proposons est riche d'informations par rapport aux nœuds classiques [1, 2]. En effet, la distinction entre l'ensemble des états normaux et des états fautifs à l'intérieur des nœuds, nous permet de bien comprendre et de suivre l'évolution des traces normales et fautives séparément. En exploitant ces informations, nous proposons dans cette section des résultats théoriques qui permettent d'améliorer le processus d'analyse de la diagnosticabilité. Nous précisons que par manque d'espace, nous présentons uniquement les résultats. Les preuves détaillées des propositions et théorèmes peuvent être consultées dans [9].

## 4.1 Une Condition Nécessaire et Suffisante Raffinée

Un cycle $F$-incertain dans le diagnostiqueur est défini comme un cycle composé exclusivement de noeuds $F$-incertain [1]. Nous avons montré que pour chaque cycle $F$-incertain $c\ell$ dans le diagnostiqueur, il existe au moins un cycle normal (i.e., non-fautif) dans le modèle du système qui partage la même observation avec $c\ell$.

**Proposition 1.** *Soit $c\ell = a_1, a_2, \ldots, a_n$ un cycle $F$-incertain dans le diagnostiqueur $\mathcal{D}$, où $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$* [2] *$(\forall 1 \leq i \leq n)$. Alors, il existe au moins un cycle non-fautif dans $G$, qui partage la même séquence observable $(\sigma_1, \sigma_2, \ldots, \sigma_n)^*$.* ◇

2. $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$ veut dire : $\forall i < n$ : $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{i+1}$ et $\delta_{\mathcal{D}}(a_n, \sigma_n) = a_1$.

Ce résultat est très intéressant pour l'analyse de la diagnosticabilité avec les diagnostiqueurs classiques [1, 2]. En effet, la condition nécessaire et suffisante pour l'analyse de la diagnosticabilité avec l'approche diagnostiqueur a été établie dans [1] et correspond à l'absence des cycles dites *F-indéterminés* dans le diagnostiqueur. Cette même condition a été ensuite reformulée dans [5] pour l'analyse de la diagnosticabilité des réseaux de Petri. A notre connaissance, toutes les approches à base de diagnostiqueurs (*définis comme automates*) utilisent cette condition pour l'analyse de la diagnosticabilité. Ci-après, nous rappelons la notion de cycle $F$-indéterminé.

**Définition 3.** (*Cycle F-indéterminé* [1])
*Un cycle F-indéterminé dans le diagnostiqueur est un cycle F-incertain, qui correspond à des cycles dans le modèle du système G, tels que :*

*C1) il existe au moins un cycle dans G qui contient uniquement des états normaux (i.e., un cycle non-fautif) ;*

*C2) il existe au moins un cycle dans G qui contient uniquement des états fautifs (i.e., un cycle fautif).* ◇

La proposition 1 montre que la condition *C1* est toujours satisfaite. Donc, pour l'analyse de la diagnosticabilité avec les approches à base de diagnostiqueurs, il suffit uniquement de vérifier la condition C2. La notion du cycle $F$-indéterminé peut être raffinée comme suit :

**Définition 4.** (*cycle F-indéterminé − Réécriture*)
*Un cycle F-indéterminé $c\ell$ dans le diagnostiqueur est un cycle F-incertain qui correspond au moins à un cycle fautif dans G qui partage la même observation que $c\ell$.* ◇

## 4.2 Une Condition Necessaire pour la Diagnosticabilité

Une question importante qui dérive de la propostion 1 est la suivante : *Pourquoi un cycle F-incertain dans le diagnostiqueur ne correspond pas* ***toujours*** *à un cycle fautif dans le modèle ?* En observant l'évolution des traces fautives dans un cycle $F$-incertain, nous remarquons que les transitions fautives transversales à l'intérieur du nœuds créent des cycles fautives *fictives*, i.e., une forme de cycle dans le diagnostiqueur qui ne correspond pas à un cycle dans le modèle. Par conséquent, on remarque que l'absence de ces transitions fautives transversales révèle l'existence de cycles fautifs ***réels*** dans le modèle. Ceci est le résultat de la proposition suivante.

**Proposition 2.** *Soit $c\ell = a_1, a_2, \ldots, a_n$ un cycle F-incertain dans le diagnostiqueur $\mathcal{D}$, où $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$ $(\forall 1 \leq i \leq n)$. Si $\forall i : 1 \leq i \leq n$, $\mathcal{B}_{a_i} = faux$, alors il existe au moins un cycle fautif dans G, qui partage la même séquence observable $(\sigma_1, \sigma_2, \ldots, \sigma_n)^*$.* ◇

Nous rappelons que $\mathcal{B}_{a_i}$ est la variable booléenne associée au nœud $a_i$ et qui indique l'existence ou non de transitions fautives transversales de $a_i.\mathcal{X}_N$ vers $a_i.\mathcal{X}_F$. Il est facile de déduire que le résultat de la proposition 2 correspond exactement à la condition *C2* dans la définition du cycle $F$-indéterminé (Définition 3).

**Proposition 3.** *Soit $c\ell = a_1, a_2, \ldots, a_n$ un cycle F-incertain dans $\mathcal{D}$, avec $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$ pour $1 \leq i \leq n$. Si $\forall i : 1 \leq i \leq n$, $\mathcal{B}_{a_i} = faux$, alors $c\ell$ est un cycle F-indéterminé.*

La proposition 3 indique qu'un cycle $F$-incertain est un cycle $F$-indéterminé s'il n'existe pas de transitions fautives transversales dans tous les nœuds du cycle $F$-incertain. Une condition *nécessaire* de la diagnosticabilité peut être reformulée à partir du résultat de de la proposition 3 comme suit :

**Proposition 4.** *Un modèle G est non-diagnosticable si la proposition 3 est satisfaite.* ◇

L'avantage de cette condition nécessaire est qu'elle est vérifiable à partir des nœuds des cycles $F$-incertain, sans avoir besoin de revenir vers le modèle du système, ce qui améliore nettement le processus d'analyse dans le cas où les modèles sont non-diagnosticables.

### 4.3 Reformulation de la Condition Nécessaire et Suffisante

En se basant sur la nouvelle structure du diagnostiqueur, nous reformulons une version simplifiée de la condition nécessaire et suffisante de la diagnosticabilité. Cette condition est basée sur la notion des séquences indicatives.

**Définition 5.** (*séquence indicative*) [3]

*Soit* $c\ell = a_1, a_2, \ldots, a_n$ *un cycle* $F$*-incertain dans* $\mathcal{D}$*, avec* $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$ *pour* $1 \leq i \leq n$*. La séquence indicative (notée* $c\ell$*-indicative), associée à un cycle* $F$*-incertain* $c\ell$*, est définie comme une séquence infinie d'ensembles d'états* $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots$*, tels que :*

- $\mathcal{S}_1 = a_1.\mathcal{X}_F$ ;
- $\forall\, i > 1 : \mathcal{S}_i = Reach_{\Sigma_u}(Img(\mathcal{S}_{i-1}, \sigma_{(i-1)_{mod_n}}))$ ; ◇

Le rôle de la séquence indicative est de suivre l'évolution des traces fautives dans le cycle $F$-incertain à partir d'un nœud de départ (choisi arbitrairement) sans prendre en compte les nouvelles occurrences de fautes, i.e., les transitions fautives transversales à l'intérieur des nœuds. Concrètement, la séquence indicative a pour but de détecter les cycles fautifs correspondant au cycle $F$-incertain (susceptibles d'exister dans le modèle du système).

**Proposition 5.** *Soit* $c\ell = a_1, a_2, \ldots, a_n$ *un cycle* $F$*-incertain dans* $\mathcal{D}$*, avec* $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$ *pour* $1 \leq i \leq n$*. Soit* $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots$ *la séquence indicative associée à* $c\ell$*. Alors, la propriété suivante est toujours vraie :* $\exists\, k \in \mathbb{N}$ *tel que* $\forall\, i \in \mathbb{N} : \mathcal{S}_{(1+(k+i)n)} = \mathcal{S}_{1+(kn)}$.

La propostion 5 montre qu'à partir d'un certain rang, la séquence indicative présente un bloc répétitif de longueur fixe (possible vide) $[\mathcal{S}_{(i+1)}, \mathcal{S}_{(i+2)}, \ldots, \mathcal{S}_{(i-1+n)}, \mathcal{S}_{(i+n)}]$ où $\mathcal{S}_{(1+i+n)} = \mathcal{S}_i$ (i.e., un cycle). Autrement dit, la séquence indicative $\rho^{c\ell}$ prend toujours une de ces deux formes :

1. *une séquence primaire :* une séquence élémentaire non-cyclic (possible vide) $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_i$, suivie par un cycle élémentaire $(\mathcal{S}_{(i+1)}, \mathcal{S}_{(i+2)}, \ldots, \mathcal{S}_{(i-1+n)}, \mathcal{S}_{(i+n)})^*$, où $\mathcal{S}_{(i+1+n)} = \mathcal{S}_{i+1}$ ;
2. *une séquence finie de termes non vides suivie par des termes vides :* $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_i$ pour $i \in \mathbb{N}^*$, où $\mathcal{S}_i \neq \emptyset$ et $\mathcal{S}_{(i+k)} = \emptyset$, $\forall k \in \mathbb{N}^*$.

La première forme de la séquence indicative (i.e., une séquence primaire) révèle la présence d'au moins un cycle fautif dans le modèle, qui est illustré par le cycle élémentaire dans la séquence indicative. Quant à la deuxième forme, elle montre que le cycle $F$-incertain ne correspond à aucun cycle fautif dans le modèle. Donc, la condition nécessaire et suffisante peut être reformulée comme suit :

**Théorème 1.** *Pour un cycle* $F$*-incertain* $c\ell = a_1, a_2, \ldots, a_n$ *dans* $\mathcal{D}$*,* $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots$ *est une séquence indicative associée à* $c\ell$*, alors* $c\ell$ *est un cycle* $F$*-indéterminé si et seulement si :*

$$\forall i \in \mathbb{N}^* : \mathcal{S}_i \neq \emptyset.$$ ◇

Théorème 1 revendique qu'un cycle $F$-incertain $c\ell$ est considéré comme un cycle $F$-indéterminé si la séquence indicative associé à $c\ell$ n'atteint jamais un terme vide (i.e., un point fixe vide). Autrement dit, elle prend la forme d'une séquence primaire.

3. Cette notion a été utilisée initialement dans notre papier [10] paru en 2015. Aussi, nous remercions le reviewer (anonyme) qui a indiqué dans son rapport qu'une définition similaire (séquence raffinée) a été utilisée par le Professeur A. Giua dans ces notes de cours à l'université de Cagliari (http ://www.diee.unica.it/giua/ARP/) et Aix-Marseille (http ://www.lsis.org/giuaa/ACS/) pour l'analyse de la diagnosticabilité à base de l'approche classique de Sampath et al. [1].

**Exemple 2.** *Nous considérons encore le diagnostiqueur $\mathcal{D}$ du modèle $G$ (figure 3). Il existe un cycle $F$-incertain $c\ell = a_1, a_2, a_3$ dans $\mathcal{D}$. $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5, \mathcal{S}_6, \mathcal{S}_7, \ldots$ est la séquence indicative associée à $c\ell$ (voir figure 3(c)). Étant donné que le terme $\mathcal{S}_7 = \emptyset$, alors selon le théorème 1, le cycle $F$-incertain n'est pas un cycle $F$-indéterminé et donc le modèle $G$ est diagnosticable.*

### 4.4 Procédure de Vérification

Pour l'implémentation de l'approche, nous avons développé un algorithme à la volée basé sur une recherche en profondeur (*Depth-First Search*). L'algorithme permet la construction de notre diagnostiqueur et la vérification des différentes conditions développées dans la section précédente en parallèle (*Voir* [9] *pour une présentation détaillée de l'algorithme*). Nous présentons ici uniquement un pseudo-code de la procédure de vérification de la condition du théorème 1.

Durant la construction à la volée du diagnostiqueur, une fois un cycle $F$-incertain $c\ell$ est généré alors :

— générer les termes successifs de la séquence indicative $\rho^{c\ell}$. Pour chaque terme $\mathcal{S}_i$ :

1. Si $\mathcal{S}_i = \emptyset$, alors $c\ell$ n'est pas $F$-indéterminé et on arrête la procédure.
2. Sinon, si $\mathcal{S}_i \neq \emptyset$ et $\exists k \in \mathbb{N} : i = 1 + kn$ (où $n = |c\ell|$), alors :
   (a) Si $\mathcal{S}_i = \mathcal{S}_{(i-n)}$, alors $c\ell$ est cycle $F$-indéterminé et on arrête la procédure.
   (b) sinon, on continue.

On répète la procédure pour chaque cycle $F$-incertain généré à la volée. On a déjà prouvé que la procédure se termine toujours (*un point fixe est toujours atteignable après un délai fini*).

## 5 Évaluation Expérimentale

Dans le but d'évaluer notre approche, nous avons implémenté l'algorithme de construction et de vérification en langage de programmation $C\#$ et nous avons mené une série d'expérimentations sur un benchmark SED. Les résultats obtenus sont comparés avec les approches de référence dans l'analyse de la diagnosticabilité des SED, soit l'approche de diagnostiqueur classique [1] et l'approche de vérificateur (verifier) [6], implémentées en C++ dans la librairie UMDES [11].

### 5.1 Présentation du Benchmark

Le benchmark illustré par la figure 4 représente une version modifiée du benchmark WODES [12]. Il décrit le fonctionnement d'un système de production caractérisé par trois paramètres :

- $k$ : le nombre de lignes de production ;
- $m$ : le nombre de produits fabriqués simultanément par les $k$ lignes de production ;
- $b$ : le nombre d'opérations dans les lignes de production.

Sur la figure 4, les transitions en rouge sont fautives alors que les autres transitions sont nominales et qui, selon les différents tests, peuvent être observables ou non. Étant donné que le benchmark est modélisé par un réseau de Petri, nous utilisons l'outil TINA [13] pour générer le graphe de marquage (*qui sera considéré comme notre modèle automate de départ*). Nous considérons deux configurations pour ce benchmark :

**Configuration 1** : on considère que $m = 1$, $b = 4$ et $k = 3, \ldots, 8$. On considère aussi que les transitions $t_0$, $t_1$, et $t_{i,1}$, $t_{i,3}$ $(1 \leq i \leq k)$ sont observables et les transitions $f_i$ $(1 \leq i \leq k)$ sont fautives. Par rapport aux autres transitions deux cas sont considérés :

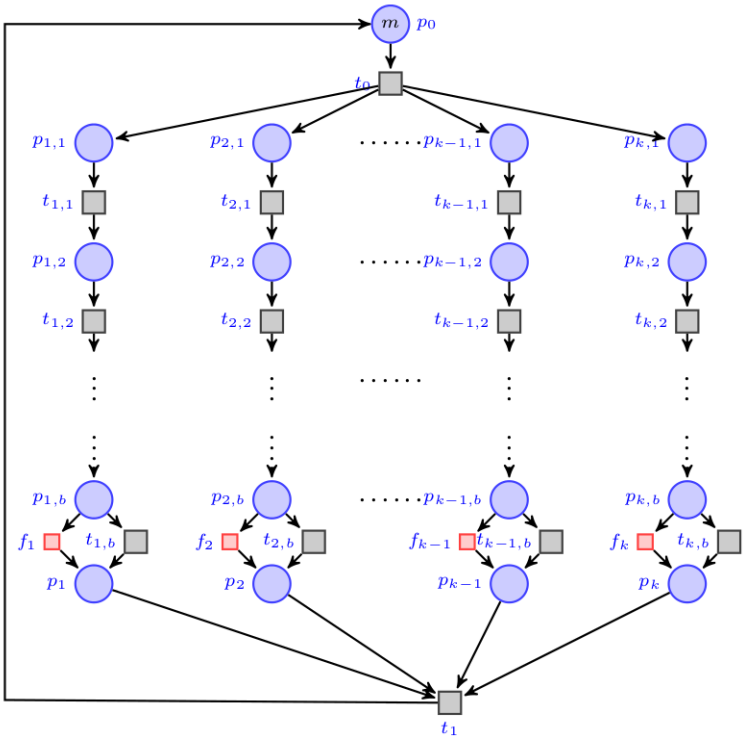



FIGURE 4 – Une version modifiée du benchmark de WODES [12].

1. les transitions sont inobservables (dans ce cas, le modèle est non-diagnosticable) ;
2. les transition sont inobservables à part les transitions $t_{i,4}$ $(1 \leq i \leq k)$ qui sont observables (dans ce cas, le modèle est diagnosticable).

**Configuration 2** : nous souhaitons évaluer notre approche par rapport aux nombres d'évènements (transitions) observables et inobservables dans le modèle. Nous considérons que $m = 1$, $k = 4$ et $b = 2, 3, \ldots, 10, 15, 18$. Les transitions $t_0$, $t_1$, et $t_{i,b}$ $(1 \leq i \leq 4)$ sont observables tandis que les transitions $f_i$ $(1 \leq i \leq 4)$ sont inobservables (et donc le modèle est toujours diagnosticable). Par rapport aux autres transitions deux cas sont aussi considérés :

1. les transitions sont inobservables ;
2. les transition sont observables ;

Autrement dit, pour cette configuration, on incrémente le nombre des évènements (transitions) observables/inobservables d'un test à l'autre, i.e., pour chaque test, on considère 4 transitions observables/inobservables de plus).

Les simulations sont effectuées sur un 64-bit PC, Intel Core i5, 4 cœurs cadencés à 2.5 GHz et 6 Go de RAM. Le durée de simulation est de 4h.

## 5.2 Discussion des Résultats

Les résultats expérimentaux sont donnés dans les tableaux 1 et 2 puis illustrés sur les figures 5 et 6 pour les configurations 1 et 2 respectivement.

TABLE 1 – Résultats expérimentaux pour la configuration 1.

| | | | | | Notre approche | | Diagnostiqueur [1] | | Verifier [6] | |
|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | $\|P\|$ | $\|T\|$ | $\|R_S\|$ | $\|R_T\|$ | $\|\mathcal{D}\|$ | $Te_{\mathcal{D}}$ (s) | $\|S\|$ | $Te_S$ (s) | $Te_V$ (s) | A |
| 3 | 16 | 17 | 126 | 377 | 16 | $\simeq 0$ | 56 | 0.3 | 0.2 | non-diagnosticable |
| 4 | 21 | 22 | 626 | 2502 | 20 | $\simeq 0$ | 164 | 0.4 | 4.4 | |
| 5 | 26 | 27 | 3126 | 15627 | 24 | 0.1 | 488 | 4.4 | o.t. | |
| 6 | 31 | 32 | 15626 | 93752 | 28 | 0.3 | 1460 | 1893 | o.t. | |
| 7 | 36 | 37 | 78126 | 546877 | 32 | 1.5 | * | o.t. | o.t. | |
| 8 | 41 | 42 | 390626 | 3125002 | 36 | 9.3 | * | o.t. | o.t. | |
| 3 | 16 | 17 | 126 | 377 | 65 | $\simeq 0$ | 270 | 0.2 | 0.1 | diagnosticable |
| 4 | 21 | 22 | 626 | 2502 | 257 | $\simeq 0$ | 1378 | 0.3 | 1.3 | |
| 5 | 26 | 27 | 3126 | 15627 | 1025 | 0.2 | 6686 | 1 | 163.4 | |
| 6 | 31 | 32 | 15626 | 93752 | 4097 | 1.8 | 31314 | 31.4 | o.t. | |
| 7 | 36 | 37 | 78126 | 546877 | 16385 | 41.6 | 143086 | 1471 | o.t. | |
| 8 | 41 | 42 | 390626 | 3125002 | 65537 | 921 | * | o.t. | o.t. | |

* : pas de résultat après 4 heures. o.t. : dépassement du temps de simulation.

- $m$, $k$, et $b$ sont les paramètres du benchmark ;
- $|P|$ et $|T|$ sont respectivement le nombre de places et transitions du du réseau de Petri ;
- $|R_S|$ et $|R_T|$, sont respectivement le nombre d'états et transitions du graphe des marquages ;
- $|\mathcal{D}_S|$ et $Te_{\mathcal{D}}$ sont respectivement le nombre d'états et le temps de simulation par notre approche ;
- $|S|$ et $Te_S$, sont respectivement le nombre d'états et le temps de simulation obtenus par l'approche diagnostiqueur classique [1] ;
- $V_e$ est le temps de simulation avec l'approche verifier [6] ;
- A est le verdict concernant la diagnosticabilité des modèles.

Nous discutons d'abord les résultats obtenus pour la configuration 1 :

- Pour les différents paramètres considérés, notre approche réussie l'analyse de la diagnosticabilité pour les deux types de modèles (diagnosticables et non-diagnosticables) tandis que les deux autres approches bloquent à partir de certains rangs (figure 5-2).
- Concernant le cas des modèles non-diagnosticables, le temps d'analyse de la diagnosticabilité reste dans l'ordre de millisecondes pour notre approche. Cela est dû essentiellement à la technique d'analyse à la volée qui arrête le processus de vérification une fois un cycle $F$-indéterminé est trouvé. Les deux autres approches construisent d'abord tout l'espace d'états du diagnostiqueur/vérificateur pour ensuite analyser la diagnosticabilité, ce qui affecte considérablement le temps de simulation (voir figure 5-2).
- Les tests effectués nous ont permis de remarquer que notre approche reste aussi efficace pour le cas des modèles diagnosticables. Cette efficacité peut être expliquée à travers trois points :
  1. Notre diagnostiqueur est construit directement à partir du modèle du système sans passer par un modèle intermédiaire (c.-à-d. le générateur), comme le cas du diagnostiqueur classique [1].
  2. La génération de l'espace d'états (du diagnostiqueur) est limitée à la partie pertinente pour l'analyse de la diagnosticabilité et qui peut être utilisé pour le diagnostic en ligne (*la partie certainement fautive n'est pas générée*) ;
  3. La procédure de vérification facilite l'analyse des cycle $F$-incertains et améliore nettement le temps de simulation.
- Bien que l'approche *verifier* présente une complexité réduite par rapport aux approches *diagnostiqueurs* (polynomiale vs exponentielle), les résultats obtenus ont montré qu'elle

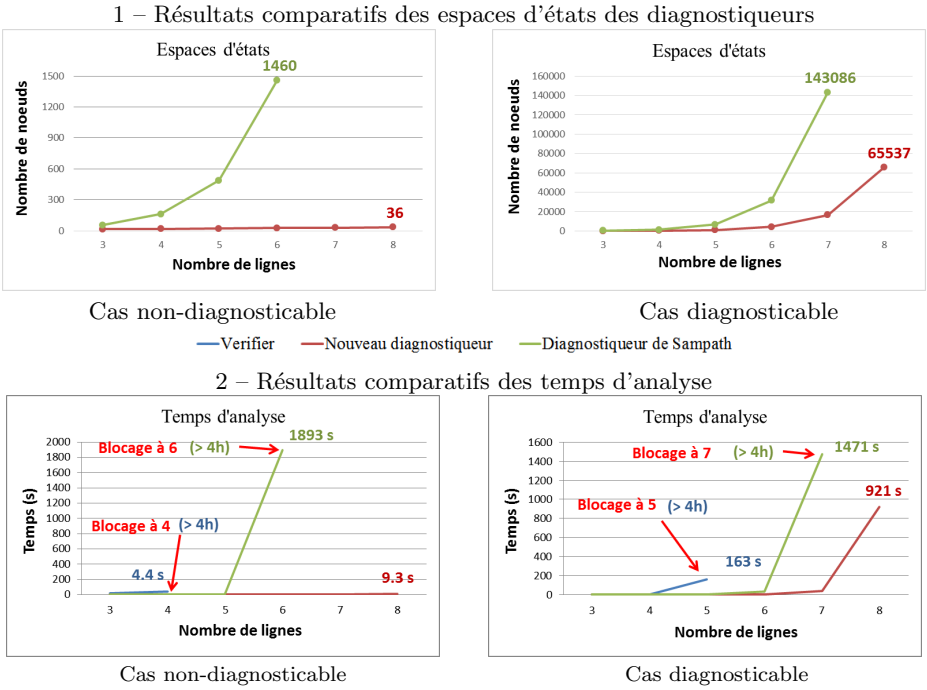



FIGURE 5 – Illustration graphique comparative – Tableau 1.

est légèrement moins efficace pour ce benchmark. En effet, l'efficacité des approches de diagnostic, en pratique, dépend plus des caractéristiques des systèmes (i.e., nombre d'évènements observables/inobservables, composantes fortement connexes, etc.).

Nous donnons ci-après quelques remarques concernant la configuration 2 :

- Les espaces d'états des diagnostiqueurs ne sont pas affectés par l'incrémentation du nombre d'évènements inobservables. Par contre, ils sont très sensibles à l'incrémentation du nombre d'évènements observables (voir figure 6-1). .
- Nous remarquons que l'efficacité des approches diagnostiqueur classique [1] et verifier [6] se dégrade drastiquement avec l'incrémentation du nombre d'évènements inobservables. Concernant l'approche diagnostiqueur, cette dégradation est dûe à l'opération $\epsilon$-*reduction* dans la construction du générateur (qui est fortement liée au nombre d'évènements inobservables). Concernant l'approche *verifier*, elle est dûe aux différentes règles de construction du vérificateur, i.e., une seule transition inobservable dans le modèle correspond à trois nouvelles transitions dans le vérificateur.
- Concernant notre approche, on constate une efficacité remarquable par rapport aux deux autres approches. À notre surprise, cette efficacité est nettement claire avec l'incrémentation des évènements inobservables, comme le montre la figure 6-2.

# 6 Extension : Multiples Classes de Fautes

L'approche proposée dans ce papier, peut être étendue pour traiter le cas de multiples classes de fautes, c.-à-d. l'ensemble d'évènements fautifs est partitionné en plusieurs classes distinctes

TABLE 2 – Résultats expérimentaux pour la configuration 2.

| | | | | | Notre approche | | Diagnostiqueur [1] | | Verifier [6] |
|---|---|---|---|---|---|---|---|---|---|
| $b$ | $\|P\|$ | $\|T\|$ | $\|R_S\|$ | $\|R_T\|$ | $\|\mathcal{D}\|$ | $Te_{\mathcal{D}}$ (s) | $\|S\|$ | $Te_S$ (s) | $Te_V$ (s) |
| 2 | 13 | 14 | 82 | 326 | 17 | $\simeq 0$ | 34 | 0.1 | 0.6 |
| 3 | 17 | 18 | 257 | 1026 | 17 | $\simeq 0$ | 34 | 3000 | 46 |
| 4 | 21 | 22 | 626 | 2502 | 17 | $\simeq 0$ | * | o.t. | 1440 |
| 5 | 25 | 26 | 1297 | 5186 | 17 | 0.1 | * | o.t. | o.t. |
| 6 | 29 | 30 | 2402 | 9606 | 17 | 0.2 | * | o.t. | o.t. |
| 7 | 33 | 34 | 4097 | 16386 | 17 | 0.5 | * | o.t. | o.t. |
| 8 | 37 | 38 | 6562 | 26246 | 17 | 0.7 | * | o.t. | o.t. |
| 9 | 41 | 42 | 10001 | 40002 | 17 | 1.7 | * | o.t. | o.t. |
| 10 | 45 | 46 | 14642 | 58566 | 17 | 3 | * | o.t. | o.t. |
| 14 | 61 | 62 | 50626 | 202502 | 17 | 8.4 | * | o.t. | o.t. |
| 18 | 77 | 78 | 130322 | 521286 | 17 | 28.8 | * | o.t. | o.t. |
| 2 | 13 | 14 | 82 | 326 | 82 | $\simeq 0$ | 164 | $\simeq 0$ | 0.4 |
| 3 | 17 | 18 | 257 | 1026 | 257 | $\simeq 0$ | 514 | $\simeq 0$ | 0.4 |
| 4 | 21 | 22 | 626 | 2502 | 626 | $\simeq 0$ | 626 | 0.1 | 0.5 |
| 5 | 25 | 26 | 1297 | 5186 | 1297 | 0.1 | 0.4 | 1.2 | 1.2 |
| 6 | 29 | 30 | 2402 | 9606 | 2402 | 0.3 | 4804 | 0.7 | 2.7 |
| 7 | 33 | 34 | 4097 | 16386 | 4097 | 1 | 8194 | 2.1 | 9.3 |
| 8 | 37 | 38 | 6562 | 26246 | 6562 | 3 | 13124 | 3.4 | 31 |
| 9 | 41 | 42 | 10001 | 40002 | 10001 | 8.9 | 20002 | 8.5 | 71.6 |
| 10 | 45 | 46 | 14642 | 58566 | 14642 | 15.4 | 29284 | 25.3 | 154.4 |
| 14 | 61 | 62 | 50626 | 202502 | 50626 | 206 | 101252 | 363 | 1590 |
| 18 | 77 | 78 | 130322 | 521286 | 130322 | 1382 | 260644 | 2340 | 8520 |

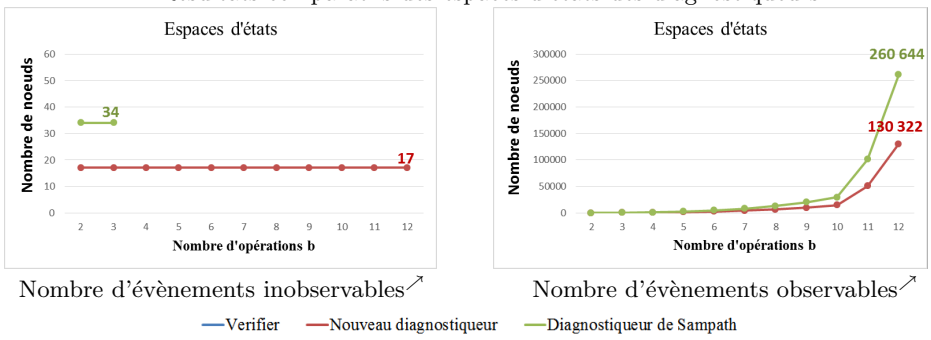
$\Sigma_f = \Sigma_{f_1} \biguplus \Sigma_{f_2} \biguplus \ldots \biguplus \Sigma_{f_m}$, où $\Sigma_{f_i} (i = 1, 2, \ldots, m)$ représente la $i^{eme}$ classe. Sampath et co. [1] ont proposé une généralisation de la condition nécessaire et suffisante de la diagnosticabilité pour traiter le cas de multiples classes de fautes. En effet, un modèle est diagnosticable s'il n'existe pas de cycle $F_i$-*indéterminé* (*i correspond à la classe de fautes* $\Sigma_{f_i}$) dans le diagnostiqueur.

Pour traiter ce cas en utilisant notre diagnostiqueur, nous avons besoin d'étendre la structure du nœuds. Pour un modèle contenant $m$ classes de fautes, un nœud du diagnostiqueur contient $m$+1 sous-ensembles d'états, tels que : $a.\mathcal{X}_N$ est l'ensemble d'états normaux, $a.\mathcal{X}_{F_i}$ est l'ensemble des états fautifs de classes $i$ (pour $i : 1 \leq i \leq m$). Les ensembles des états peuvent être liés entre eux par différentes transitions fautives transversales. La figure 7a illustre la forme générale d'un nœud du diagnostiqueur.

La Figure 7b illustre un nœud du diagnostiqueur correspondant à une partie d'un modèle contenant deux classes de fautes ($\Sigma_{f_1} = \{f_1\}$ et $\Sigma_{f_2} = \{f_2\}$). Les différents ensembles d'états sont $a.\mathcal{X}_N = \{1\}, a.\mathcal{X}_{F_1} = \{2, 3\}$, et $a.\mathcal{X}_{F_2} = \{3, 4\}$.

En ce qui concerne l'analyse de la diagnosticabilité, la condition nécessaire et suffisante à base de séquence indicative proposée dans ce papier peut être aussi généralisée pour traiter le cas de classes de fautes multiples. L'idée principale est de générer (*au maximum*) $m$ séquences indicatives pour chaque cycle $F$-incertain. De la même manière, on vérifie la condition nécessaire et suffisante du théorème 1.

1 – Résultats comparatifs des espaces d'états des diagnostiqueurs




2 – Résultats comparatifs des temps d'analyse

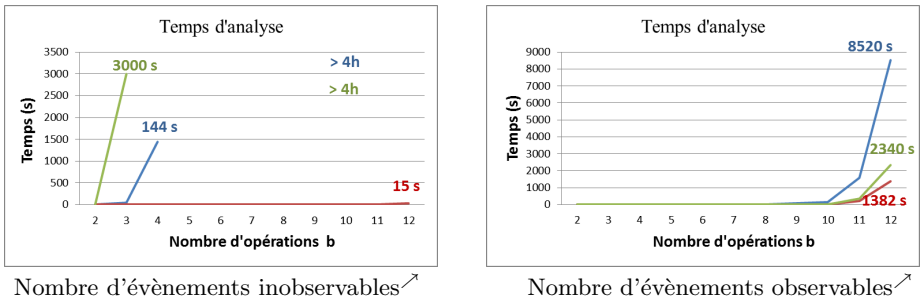
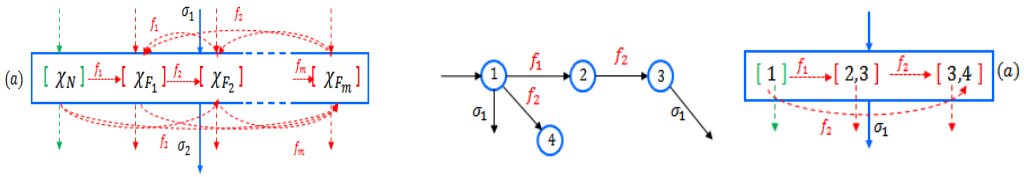



FIGURE 6 – Illustration graphique comparative – Tableau 2.



(a) La structure générale d'un nœud

(b) Un exemple illustratif

# 7 Conclusion

Cet article présente une variante efficace de l'approche diagnostiqueur pour l'analyse de la diagnosticabilité des SED. L'originalité de cette variante c'est qu'elle repose sur une nouvelle structure du diagnostiqueur qui consiste séparer entre l'ensemble des états normaux et l'ensemble des états fautifs dans chaque nœud. En se basant sur cette nouvelle structure, plusieurs résultats théoriques et pratiques ont été proposés dans le but d'améliorer le processus de l'analyse de la diagnosticabilité. L'approche proposée a été évaluée à travers une série d'expérimentations et de comparaisons avec des approches classiques de référence.

Nos travaux futurs viseront à étendre la nouvelle structure du diagnostiqueur pour traiter le cas des fautes complexes comme les fautes intermittentes et répétitives. En plus, nous souhaiterons étudier les problématiques liées au placement des capteurs et à la reconfiguration des modèles. En effet, nous estimons que la distinction entre les ensembles des états normaux et fautifs dans les nœuds, peut aider à identifier les évènements et les séquences fautifs respon-

sables de la non-diagnosticabilité des modèles ce qui présente une donnée intéressante pour le placement de capteurs.

## Remerciement

Ce travail a été réalisé dans le cadre du projet ELSAT2020. ELSAT2020 est cofinancé par l'Union Européenne avec le Fonds européen de développement régional, par l'État et la Région Hauts de France.

## Références

[1] M. Sampath, R. Sengupta, and S. Lafortune, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[2] C. Cassandras and S. Lafortune, "Introduction to discrete event systems," *Springer*, 2008.

[3] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[4] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems : framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, 2003.

[5] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of bounded Petri nets," *Proceedings of the* $48^{th}$ *IEEE Conference on Decision and Control, held jointly with the* $28^{th}$ *Chinese Control Conference*, pp. 1254–1260, 2009.

[6] T.-S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1491–1495, 2002.

[7] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, "A polynomial algorithm for testing diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1318–1321, 2001.

[8] M. Moreira, T. Jesus, and J. Basilio, "Polynomial time verification of decentralized diagnosability of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1679–1684, 2011.

[9] A. Boussif, "Contributions to fault diagnosis of discrete-event systems," *Ph.D. Thesis, IFSTTAR - Université de Lille 1, Science et Technologies*, 2016.

[10] A. Boussif, M. Ghazel, and K. Klai, "Combining enumerative and symbolic techniques for diagnosis of discrete-event systems," *Verification and Evaluation of Computer and Communication Systems*, pp. 1–15, 2015.

[11] S. Lafortune, "UMDES-Lib software library." http ://www.eecs.umich.edu/ umdes/toolboxes.html, 2000.

[12] A. Giua, "A benchmark for diagnosis.," *International Workshop on Discrete Event Systems*, 2008.

[13] B. Berthomieu, P.-O. Ribet, and M. Vernadat, "The tool TINA : Construction of abstract state spaces for Petri nets and time Petri nets," *International Journal of Production Research*, vol. 42, no. 14, pp. 2741–2756, 2007.