Un algorithme de découverte de chroniques pertinentes pour le diagnostic par identification et reconstitution

Alexandre Sahuguède

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :
l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue le *12/03/2020* **par :**

ALEXANDRE SAHUGUÈDE

Un algorithme de découverte de chroniques pertinentes pour le diagnostic par identification et reconstitution

---

JURY

SANDRA BRINGAY	Professeur des Universités	Rapporteuse
YANN LABIT	Professeur des Universités	Président du Jury
SÉBASTIEN LAHAYE	Professeur des Universités	Rapporteur
EURIELL LE CORRONC	Maître de Conférences	Directrice de Thèse
MARIE-VÉRONIQUE LE LANN	Professeur des Universités	Directrice de Thèse
THOMAS GUYET	Maître de Conférences HDR	Membre du Jury

École doctorale et spécialité :
EDSYS : Automatique 4200046
Unité de Recherche :
Laboratoire d'analyse et d'architecture des systèmes
Directrices de Thèse :
Euriell LE CORRONC et *Marie-Véronique LE LANN*
Rapporteurs :
Sandra BRINGAY et *Sébastien LAHAYE*

À mes chats.

Table des matières

Introduction

Le *diagnostic* est une fonction essentielle dans un processus de surveillance des systèmes dynamiques. Il s'agit de savoir détecter, identifier et isoler l'occurrence de fautes dans le système grâce à des observations issues de capteurs, de journaux d'alarmes ou encore de signaux de commande. Si cela est réalisé suffisamment tôt, la panne peut être réparée, le système peut être reconfiguré, l'impact de la défaillance peut être minimisé.

Actuellement, le domaine du diagnostic est en profonde mutation en raison de la complexification des systèmes et du développement des capacités de génération et de stockage des données. Il est de plus en plus difficile d'obtenir des modèles fiables des systèmes à diagnostiquer alors que nous disposons beaucoup plus facilement d'une multitude d'informations issues d'historiques opérationnels, de simulations ou encore d'expérimentations du système. La problématique réside alors dans l'extraction automatique des informations utiles au diagnostic à partir de cette masse de données.

C'est dans ce contexte que ce travail a pour ambition de réaliser la *génération automatique* des modèles formels représentant les différents phénomènes d'un système dynamique. Les modèles formels générés sont des motifs temporels appelés *chroniques*. Il s'agit d'un modèle formel temporel permettant de représenter le comportement des systèmes dynamiques à un niveau abstrait en termes d'événements. Les événements sont contraints temporellement les uns par rapport aux autres. Ce processus de génération automatique de chroniques à partir de données est appelé *découverte de chroniques*.

Dans le domaine du diagnostic, les chroniques sont particulièrement efficaces en raison de leur degré d'abstraction et de l'aspect temporel qu'elles apportent. Chaque chronique caractérise un phénomène précis du système, qu'il soit nominal ou de défaut. Lorsqu'une chronique représente une situation de défaut, la reconnaître, c'est-à-dire observer les événements qui la forment dans le bon ordre et aux dates respectant les contraintes temporelles, signifie que le système est dans cette situation de défaut. Si le diagnostic est réalisé en temps réel, cette identification est immédiate ce qui permet de lever des alarmes ou de réaliser des actions afin de palier ce défaut dans un laps de temps très court après son occurrence.

Dans ce mémoire, plusieurs contributions aux chroniques en général et au processus de découverte de chroniques en particulier sont données :

— une approche innovante à la problématique de découverte de chroniques avec un nouvel algorithme adoptant cette approche intitulée CDIRe (*Chronicle Discovery by Identification and Reconstitution*),

— une implémentation de CDIRe en `C++` ainsi qu'une analyse des performances et des résultats de cet algorithme,

— une nouvelle représentation des chroniques dans un espace euclidien k-dimensionnel.

CDIRe, la principale contribution de ce mémoire, est un algorithme de découverte de chroniques qui repose sur une identification de chroniques élémentaires et une reconstitution de chroniques plus complexes. Une chronique élémentaire est une chronique avec deux événements et une contrainte temporelle entre ces événements. Cette approche innovante au processus de découverte de chroniques permet de construire des chroniques avec un minimum de connaissances a priori sur le système considéré et en particulier sans critère de fréquence. En effet, la fréquence est un critère de choix qui est très largement utilisé pour résoudre la problématique de découverte de chroniques.

Cette approche est fondée sur des notions d'*identification* de chroniques élémentaires et de *reconstitution*. L'identification de chroniques élémentaires est la tâche de découverte de chroniques simples et pertinentes dans une séquence temporelle d'entrée à l'aide d'un algorithme de partitionnement des données. Quant à la reconstitution de chroniques, c'est le processus d'assemblage de ces chroniques élémentaires en chroniques plus complexes afin de décrire au mieux le ou les phénomènes sous-jacents aux données d'entrée.

La quantité souvent importante de chroniques générées par le processus de découverte de chroniques nécessite une étape d'analyse. En effet, une grande partie de ces chroniques est inutilisable car représentant du bruit. Les outils actuels fournis par l'analyse de chroniques sont inexistants ou inadaptés à la tâche de comparaison et de classification de chroniques. Dans un espace euclidien, ces tâches deviennent aisés grâce aux nombreux travaux traitant de comparaisons entre points ou ensembles dans un espace euclidien. Une méthode mathématique permettant d'associer une chronique à un objet dans un espace euclidien est avantageuse pour le processus de découverte de chroniques.

Une autre contribution de cette thèse est la création d'une telle méthode mathématique : la projection de chronique dans un espace euclidien k-dimensionnel. Ce travail démontre qu'il est possible d'associer une chronique à un volume géométrique où les informations événementielles et temporelles d'une chronique sont conservées. Cette projection ouvre la porte à de nombreuses perspectives pour l'analyse de chroniques et en particulier pour la découverte de chroniques.

Ce mémoire est structuré en quatre chapitres :

- Le **chapitre 1** présente le contexte d'étude de ce mémoire. Le cadre formel des chroniques et son utilité dans un environnement de diagnostic y est décrit. Un tour d'horizon dans la littérature des différents modèles temporels d'une part, et des algorithmes d'extraction de connaissance à partir de données temporelles d'autre part est fourni. De plus, un soin particulier aux algorithmes de découverte de chroniques dans cet état de l'art est offert. Enfin,

le positionnement des contributions de ce mémoire est donné.

- Le **chapitre 2** détaille l'algorithme de découverte de chroniques par identification et reconstitution. La problématique concernée par CDIRe est posée. Puis, les clefs de la conception de cet algorithme ainsi qu'une étude de sa complexité sont détaillés.
- Le **chapitre 3** offre une analyse des performances et des résultats de CDIRe. Les différents paramètres de celui-ci sont analysés un à un et leurs effets sur les résultats sont analysés sur un jeu de données provenant d'une application réelle. De plus, un bilan des influences de ces paramètres sur les résultats est fournie sous forme de tableaux récapitulatifs guidant l'utilisateur de CDIRe dans ses choix des valeurs des paramètres.
- Le **chapitre 4** concerne les travaux sur la projection de chroniques dans un espace euclidien k-dimensionnel. Un aperçu de la littérature sur cette technique de projection aléatoire est donné. La méthodologie de la projection aléatoire appliquée aux chroniques est fournie ainsi que quelques propriétés de cette projection. Une analyse sur le nombre k de dimensions de l'espace euclidien considéré est offerte. Puis, plusieurs exemples d'utilisation possible de cette méthode dans un contexte de découverte de chroniques sont détaillés et montrent le potentiel de celle-ci.

Chapitre 1

Contexte scientifique & état de l'art

Sommaire

Ce chapitre présente le contexte d'étude de nos contributions. Il permet de décrire le cadre formel du modèle considéré : les chroniques. En plus de définir le contexte scientifique des chroniques, un des objectifs de ce chapitre est de positionner notre travail par rapport aux travaux dans la littérature. C'est pourquoi un tour d'horizon des motifs temporels ainsi que des méthodes d'extraction de connaissance à partir de données temporelles y est proposé. L'état de l'art proposé dans ce chapitre présente un résumé des techniques existantes par rapport aux objectifs de ce mémoire.

Le reste de ce chapitre est organisé de la manière suivante. La section 1.1 introduit le modèle des chroniques et l'usage de celles-ci tel qu'il est considéré dans ce mémoire. La section 1.2 définit le cadre formel des chroniques et donne un tour d'horizon des chroniques dans le domaine du diagnostic. La section 1.3 présente

un état de l'art autour de quelques motifs temporels pertinents pour le diagnostic. La section 1.4 offre une étude de quelques algorithmes d'extraction de modèles fréquents. La section 1.5 s'intéresse plus particulièrement aux algorithmes de découverte de chroniques. Enfin, la section 1.6 présente le positionnement de notre algorithme CDIRe.

1.1 Préambule : qu'est-ce qu'une chronique ?

Une chronique est une représentation d'une information qui est à la fois temporelle et événementielle. Elle modélise l'apparition d'événements dans un ordre défini par des contraintes temporelles. En raison de cette abstraction des événements, les chroniques offrent un formalisme qui est particulièrement bien adapté au diagnostic de fautes. En effet, la reconnaissance d'une chronique modélisant la signature d'une faute, un comportement symptomatique de la faute connue dans un flux d'événements provenant d'un système à diagnostiquer permet de détecter rapidement celle-ci et d'appliquer les actions adéquates. Un diagramme de l'usage des chroniques tel qu'il est considéré dans ce document est représenté sur la figure 1.1. Quatre modules importants peuvent être identifiés :

— Le module *modélisation* correspond à l'élaboration d'une base de chroniques par un expert à partir de sa connaissance du système ou d'autres modèles existants.

— Le module *découverte de chroniques* est la construction ou l'enrichissement d'une base de chroniques à partir d'un ensemble de données, le plus souvent sous la forme de séquences temporelles. Ce document traite en particulier de cet axe de travail.

— Le module *analyse* correspond à l'étude des différentes propriétés que peuvent posséder une chronique.

— Le module *reconnaissance de chroniques* est le processus de diagnostic de fautes en temps réel à partir d'un flux d'événements provenant du système complexe et dynamique modélisé.

La *modélisation* de chroniques par un expert du système est l'approche la plus simple et directe pour la construction d'une base de chroniques pertinentes pour le diagnostic. Il s'agit de construire un ensemble de chroniques qui représentera chacun un phénomène dynamique d'intérêt. Ces chroniques sont bien souvent très adaptées au système modélisé car elles sont élaborées à partir des connaissances d'un expert ou encore à partir d'autres modèles démontrés (comme un modèle de commande par exemple). Malheureusement, la complexification des systèmes dynamiques modernes rend une telle modélisation de plus en plus ardue. De plus, cette approche ne permet pas de décrire des phénomènes inconnus qui peuvent être très pertinents du point de vue du diagnostic.

Une approche différente de construction de chroniques qui est basée sur les données générées par le système devant être modélisé propose une solution aux problématiques liées à la modélisation par un expert. Cette approche qui est appelée

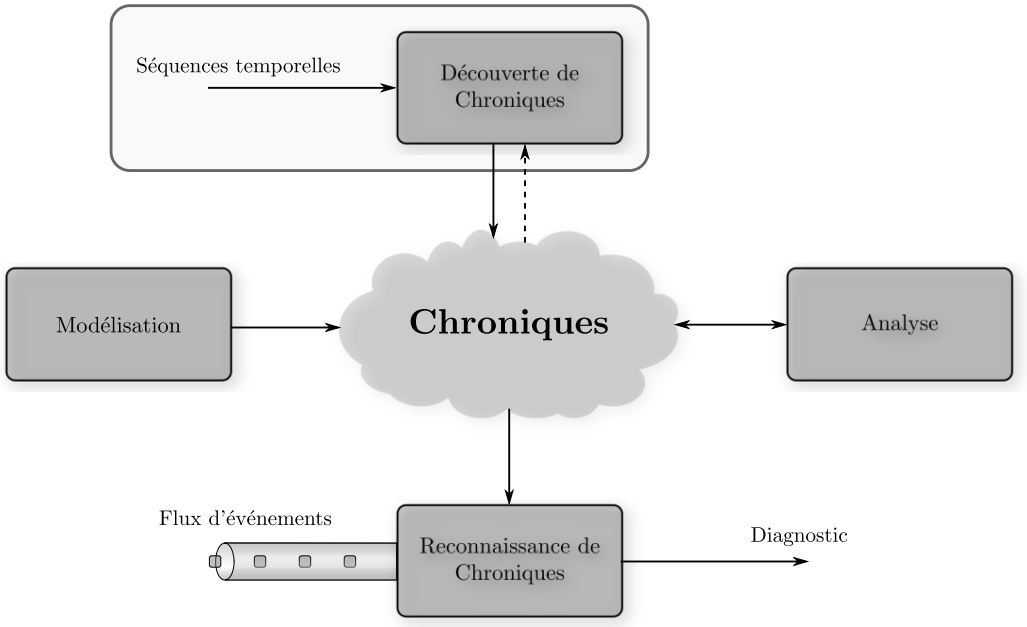


FIGURE 1.1 – Diagramme récapitulatif des différents modules en lien avec les chroniques.

découverte de chroniques s'appuie sur des méthodes d'apprentissage automatique. C'est un outil qui permet de faciliter le travail des experts en lui proposant un ensemble de chroniques respectant un critère de choix (bien souvent, le critère de *fréquence* est choisi) dans les données d'entrée. Les données d'entrée utilisables sont sous la forme de *séquences temporelles*, un ensemble d'événements datés ordonnés suivant leurs dates d'exécution, sur une période de temps où le système à modéliser engendre des observations de phénomènes intéressants. La découverte de chroniques peut être utilisée afin de construire une base de chroniques utile pour le diagnostic en partant de zéro, ou encore enrichir et améliorer une base de chroniques déjà implémentée sur un système. La force de ce type d'approche est la capacité à découvrir des schémas temporels complexes qui peuvent passer inaperçus aux yeux des experts en raison de leur complexité ou du comportement apparemment décorrélé de deux sous-systèmes différents. Malheureusement, il reste encore des problématiques importantes à résoudre, notamment, le temps de calcul qui peut être important en fonction du volume des données d'entrée. De plus, ce type d'approche peut générer une quantité importante de chroniques où une grande partie de celles-ci sont inutilisables car représentant du bruit. Néanmoins, la découverte de chroniques est un outil d'aide à la modélisation de chroniques très utile pour l'expert. Un état de l'art dans le domaine de la découverte de chroniques est proposé dans la section 1.5.

Dans une base de chroniques efficace pour le diagnostic, quels que soient les outils utilisés pour la modéliser, les chroniques doivent respecter un certain nombre de propriétés. En particulier, la propriété de *cohérence* garantit qu'une chronique peut être reconnue dans une séquence temporelle. D'autres propriétés permettent

de donner des éléments de comparaison entre deux chroniques, comme l'*équivalence* (deux chroniques qui sont toujours reconnues par la même séquence temporelle) ou encore la *couverture* (lorsqu'une des chroniques est reconnue, la seconde est garantie d'être reconnue). Ces propriétés ainsi que le cadre formel des chroniques sont définis en détail dans la section 1.2.2.

Avec une base de chroniques représentative du système à diagnostiquer et validée par une analyse de celle-ci, la *reconnaissance de chroniques* peut être mise en place en ligne. La reconnaissance, c'est expliquer le flux d'événements d'entrée venant d'un système à l'aide de la base de chroniques modélisant ce système. Le cadre formel des chroniques est particulièrement efficace dans un contexte de supervision. La reconnaissance de chroniques représentant un phénomène spécifique (qu'il soit nominal ou représentatif d'une faute) en temps réel peut enchaîner sur la génération d'un événement ou sur une action externe, et ce de manière rapide. Néanmoins, des limitations techniques sur le volume de chroniques pouvant être suivies simultanément doivent être prises en compte lors de la génération de la base de chroniques. En effet, une base de chroniques trop importante aura un impact négatif significatif sur le temps de reconnaissance d'une chronique, ce qui se répercutera sur le temps du diagnostic. La reconnaissance de chroniques est plus détaillée dans la section 1.2.3.

Ces quatre axes de travail dans le cadre formel des chroniques dans un contexte de diagnostic sont interconnectés. En effet, des propriétés comme la cohérence sont nécessaires à une reconnaissance de chroniques efficace. De plus, certaines propriétés peuvent être utiles au processus de découverte de chroniques. Enfin, en raison des contraintes liées aux algorithmes de reconnaissance de chroniques, et en particulier de leur complexité algorithmique, la base de chroniques générée doit rester d'une taille limitée. Or, les algorithmes existant dans la littérature génèrent une quantité importante de chroniques qui ne va que croissant avec la taille des données d'entrée. Ces chroniques doivent par la suite être évaluées sur des critères arbitraires ou par un expert, ce qui limite l'utilité de tels algorithmes.

1.2 Contexte scientifique : les chroniques

Dans cette section, le cadre formel des chroniques dans un contexte du diagnostic est défini. Dans un premier temps, les *réseaux de contraintes temporelles*, dont le modèle des chroniques hérite de beaucoup de propriétés, est détaillé. Puis, le modèle des chroniques ainsi que les propriétés de celui-ci sont définis. Enfin, le bénéfice des chroniques dans un contexte de diagnostic est mis en avant à l'aide des travaux récents dans la littérature.

1.2.1 Réseaux de contraintes temporelles

Les réseaux de contraintes temporelles possèdent un cadre formel particulièrement riche et développé notamment par [Dechter 1991, Meiri 1996]. Ce cadre formel

donne des solutions pour résoudre des problèmes de satisfaction de contraintes temporelles où des variables représentent des informations spatiales contraintes temporellement entre elles. L'exemple suivant, issu de [Dechter 1991], met en avant les questions intéressantes qui découlent de ce type de modélisation.

Exemple 1.1. John se rend au travail en voiture, ce qui lui prend entre 30 et 40 minutes. Fred, quant à lui, met entre 40 et 50 minutes pour aller au travail en faisant du covoiturage. Aujourd'hui, John est parti de chez lui entre $7h10$ et $7h20$, et Fred est arrivé au travail entre $8h$ et $8h10$. De plus, John est arrivé au travail environ 10 à 20 minutes après que Fred soit parti de son domicile. Ce problème peut être modélisé par un graphe de contraintes, où chaque variable temporelle représente un des événements de ce problème et où chaque contrainte temporelle est définie par un arc entre les deux événements contraints. Ce graphe est représenté sur la figure 1.2. Cette modélisation permet de répondre à différentes requêtes, comme : *Est-ce que les informations de ce problème sont cohérentes ?* ou encore *Quelles sont les heures possibles auxquelles Fred ai quitté son domicile ?*

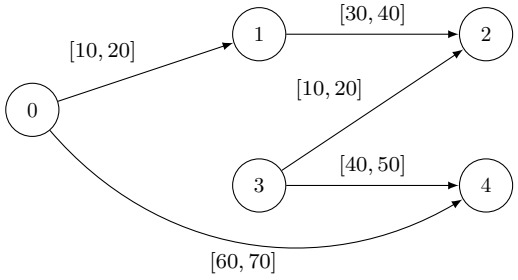


FIGURE 1.2 – Un graphe de contraintes qui représente le problème décrit dans l'exemple 1.1.

Les différentes requêtes abordées dans l'exemple précédent sont satisfaites à l'aide de problèmes temporels simples où chaque couple d'événements possède au plus une seule contrainte temporelle. À chaque problème temporel simple, est associé un graphe orienté pondéré sur les arcs appelés graphe de distance. Un exemple de graphe de distance est représenté sur la figure 1.3. C'est grâce à ces graphes que plusieurs algorithmes sont décrits qui, entre autres :

— cherchent tous les instants d'occurrence possibles d'un événement donné,

— cherchent toutes les relations possibles entre deux événements donnés,

— génèrent un ou plusieurs scénarios cohérents avec les informations données.

Ces algorithmes peuvent être effectués en temps polynomial [Dechter 1991].

Le cadre formel des réseaux de contraintes temporelles a été le sujet de plusieurs recherches exploratoires. Par exemple, dans [Vila 1994], la notion de *flou* est introduite dans les réseaux de contraintes temporelles permettant ainsi de modéliser des incertitudes de temps comme "il y a quelques jours" ou encore "plus ou moins 30 minutes". Dans [Álvarez 2013], un algorithme d'apprentissage de réseaux

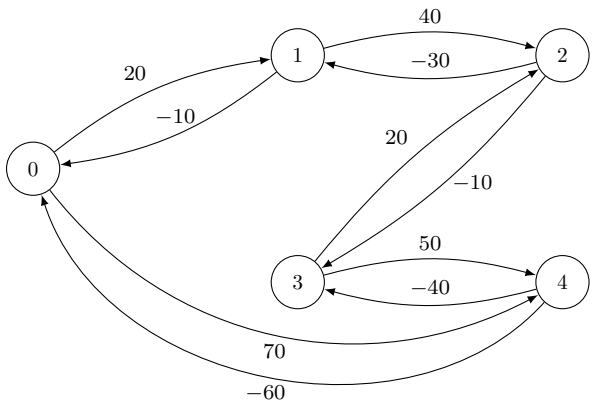


FIGURE 1.3 – Le graphe de distance associé au problème temporel simple qui représente le problème décrit dans l'exemple 1.1.

de contraintes temporelles appelé ASTPminer est présenté et est appliqué dans un contexte médical, en particulier sur des résultats de polysomnographies, un examen complet du sommeil, de patients atteint d'apnée du sommeil.

1.2.2 Cadre formel des chroniques

Dans cette section, le cadre formel des chroniques est défini. Seule la partie nécessaire à la compréhension du contexte scientifique de ce document est donnée. Il est nécessaire de distinguer l'information spatiale, ce qui se produit, et l'information temporelle, quand cela se produit. Par exemple, dans la phrase "Marc s'est levé à $7h$", "se lever" est l'information spatiale alors que "$7h$" est l'information temporelle. Une chronique est un modèle temporel qui est constitué d'une composante spatiale, représentée par les événements, et d'une composante temporelle, représentée par les contraintes temporelles.

Définition 1.1. Un **événement** est une information spatiale se produisant instantanément.

Définition 1.2. Un **domaine spatial** est l'ensemble des informations spatiales possibles noté $\mathbb{E}$. L'ensemble $\mathbb{E}$ est ordonné par sa relation d'ordre $<_{\mathbb{E}}$ qui est étendu à $\leq_{\mathbb{E}}$ suivant la formule suivante, $e \leq_{\mathbb{E}} e' \Leftrightarrow e <_{\mathbb{E}} e' \vee e =_{\mathbb{E}} e'$.

La relation d'ordre $<_{\mathbb{E}}$ est arbitraire. Généralement, et dans le reste de ce document, la relation d'ordre choisie est l'ordre lexicographique.

Définition 1.3. Un **domaine temporel** est l'ensemble des informations temporelles possibles noté $\mathbb{T}$. $\mathbb{T}$ est un domaine temporel où $\mathbb{T} \subseteq \mathbb{N}$.

Définition 1.4. Un **événement daté** est un couple (e, t) d'un événement e et d'un instant d'occurrence t tel que $e \in \mathbb{E}$ et $t \in \mathbb{T}$.

Définition 1.5. Une **contrainte temporelle** est un tuple $\tau_{(i,j)} = (e_i, e_j, t^-, t^+)$, aussi noté $\tau_{(i,j)} = e_i[t^-, t^+]e_j$, où $e_i, e_j \in \mathbb{E}$, $e_i \leq_{\mathbb{E}} e_j$, $t^-, t^+ \in \mathbb{T}$, et $t^- \leq t^+$. Une contrainte temporelle $\tau_{(i,j)} = e_i[t^-, t^+]e_j$ est dite satisfaite par un couple d'événements datés distincts $\langle (e,t), (e',t') \rangle$ si et seulement si $e = e_i$, $e' = e_j$ et $(t' - t) \in [t^-, t^+]$.

Définition 1.6. Une **chronique** $\mathcal{C}$ est un couple $(\mathcal{E}, \mathcal{T})$ où : $\mathcal{E} = \{e_1, \ldots, e_n\}$ avec $\forall i, j,\ 1 \leq i < j \leq n$, $e_i, e_j \in \mathbb{E}$, $e_i \leq_{\mathbb{E}} e_j$; $\mathcal{T} = \{\tau_{(i,j)}\}_{1 \leq i < j \leq n}$ est un ensemble de contraintes temporelles. $\mathcal{E}$ est un multi-ensemble, c'est-à-dire qu'il peut contenir plusieurs fois le même événement. Une chronique de taille n est une chronique telle que $|\mathcal{E}| = n$.

Définition 1.7. Le **nœud** d'un événement e_i dans une chronique $\mathcal{C}$ est l'indice i et est noté $\nu_{\mathcal{C}}(e_i) = i$.

Une chronique peut être visualisée par un graphe orienté. Les événements de $\mathcal{E}$ sont représentés par les nœuds du graphe et les contraintes temporelles de $\mathcal{T}$ sont représentées par les arcs.

Définition 1.8. Une **chronique élémentaire** $\mathcal{C}^\alpha = (\mathcal{E}^\alpha, \mathcal{T}^\alpha)$ est une chronique de taille 2 où $\mathcal{E}^\alpha = \{e_1, e_2\}$ et $\mathcal{T}^\alpha = \{\tau_{(1,2)}\}$. Le nœud $\nu_{\mathcal{C}^\alpha}(e_1)$ est le **nœud source** alors que le nœud $\nu_{\mathcal{C}^\alpha}(e_2)$ est le **nœud destination**.

Exemple 1.2. Soit $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ une chronique de taille 4 où $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(2,3)} = e_2[3,4]e_3, \tau_{(1,4)} = e_1[8,10]e_4\}$. La chronique $\mathcal{C}$ est représentée graphiquement sur la figure 1.4. La figure 1.5, quant à elle, représente une chronique élémentaire $\mathcal{C}^\alpha = (\mathcal{E}^\alpha, \mathcal{T}^\alpha)$ où $\mathcal{E}^\alpha = \{e_1 = a, e_2 = b\}$ et $\mathcal{T}^\alpha = \{\tau_{(1,2)} = e_1[1,10]e_2\}$.

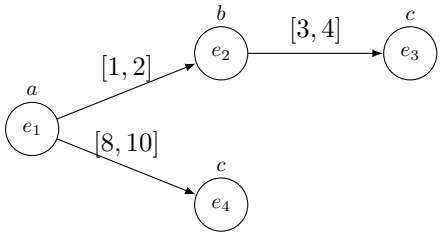


FIGURE 1.4 – Une représentation graphique de la chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ de taille 4 avec $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(2,3)} = e_2[3,4]e_3, \tau_{(1,4)} = e_1[8,10]e_4\}$.

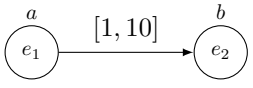


FIGURE 1.5 – Une représentation graphique de la chronique élémentaire $\mathcal{C}^\alpha = (\mathcal{E}^\alpha, \mathcal{T}^\alpha)$ avec $\mathcal{E}^\alpha = \{e_1 = a, e_2 = b\}$ et $\mathcal{T}^\alpha = \{\tau_{(1,2)} = e_1[1,10]e_2\}$.

Un aspect important des chroniques réside dans la reconnaissance de celles-ci dans une séquence temporelle [Dousson 1993, Dousson 1994]. Un sous-ensemble d'une séquence temporelle qui satisfait toutes les contraintes d'une chronique est une occurrence de celle-ci. On parle aussi dans la littérature d'instance de chronique.

Définition 1.9. Une **séquence temporelle** est un ensemble ordonné fini d'événements datés noté $\mathcal{S} = \{(e_1, t_1), \ldots, (e_m, t_m)\}$. Les événements datés sont ordonnés par $\preceq$ tel que :

$$\forall i, j \in [1, m], (e_i, t_i) \preceq (e_j, t_j) \Leftrightarrow t_i < t_j \vee (t_i = t_j \wedge e_i \leq_{\mathbb{E}} e_j). \tag{1.1}$$

L'ensemble des événements de $\mathcal{S}$ est noté $\mathbb{E}_{\mathcal{S}}$.

Exemple 1.3. Soit $\mathcal{S} = \{(a, 2), (b, 3), (d, 5), (c, 7), (a, 8), (b, 10), (c, 11), (c, 13), (d, 13), (c, 16)\}$ une séquence temporelle avec dix événements datés représentée sur la figure 1.6. L'ensemble des événements est $\mathbb{E}_{\mathcal{S}} = \{a, b, c, d\}$.

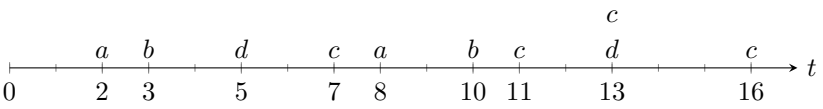


FIGURE 1.6 – Une séquence temporelle $\mathcal{S} = \{(a, 2), (b, 3), (d, 5), (c, 7), (a, 8), (b, 10), (c, 11), (c, 13), (d, 13), (c, 16)\}$ avec son ensemble des événements $\mathbb{E}_{\mathcal{S}} = \{a, b, c, d\}$.

Définition 1.10. Une **occurrence** d'une chronique $\mathcal{C} = (\mathcal{E} = \{e'_1, \ldots, e'_n\}, \mathcal{T})$ dans une séquence temporelle $\mathcal{S} = \{(e_1, t_1), \ldots, (e_m, t_m)\}$ est une sous-séquence $o_{\mathcal{C}} = \{(e_{f(1)}, t_{f(1)}), \ldots, (e_{f(n)}, t_{f(n)})\}$ qui respecte les conditions suivantes :

$$\begin{cases} f : [1, n] \mapsto [1, m] \text{ est une fonction injective,} \\ \forall i \in [1, n],\ e'_i = e_{f(i)}, \\ \forall \tau'_{(i,j)} = e'_i[t^-, t^+]e'_j \in \mathcal{T}, t_{f(j)} - t_{f(i)} \in [t^-, t^+]. \end{cases} \tag{1.2}$$

L'ensemble de toutes les occurrences d'une chronique $\mathcal{C}$ dans une séquence temporelle $\mathcal{S}$ est appelé $\mathcal{O}_{\mathcal{C}}(\mathcal{S})$.

Plusieurs propriétés intéressantes découlent des occurrences d'une chronique dans une séquence temporelle, en particulier la fréquence d'une chronique qui est communément utilisée comme critère de choix dans les différents algorithmes de découverte de chroniques présent dans la littérature. Ces algorithmes seront vus plus en détail dans la section 1.5.

Définition 1.11. La **fréquence** $f(\mathcal{C}, \mathcal{S})$ d'une chronique $\mathcal{C}$ dans une séquence temporelle $\mathcal{S}$ est la taille de l'ensemble $\mathcal{O}_{\mathcal{C}}(\mathcal{S})$:

$$f(\mathcal{C}, \mathcal{S}) = |\mathcal{O}_{\mathcal{C}}(\mathcal{S})|. \tag{1.3}$$

Exemple 1.4. Soit $\mathcal{C}$ la chronique décrite dans l'exemple 1.2 et $\mathcal{S}$ la séquence temporelle décrite dans l'exemple 1.3. Deux occurrences de $\mathcal{C}$ existent dans $\mathcal{S}$:

$$\mathcal{O}_{\mathcal{C}}(\mathcal{S}) = \{o^1_{\mathcal{C}}, o^2_{\mathcal{C}}\} = \begin{cases} o^1_{\mathcal{C}} = \{(a,2),(b,3),(c,7),(c,11)\}, \\ o^2_{\mathcal{C}} = \{(a,8),(b,10),(c,13),(c,16)\}. \end{cases} \tag{1.4}$$

La fréquence de $\mathcal{C}$ dans $\mathcal{S}$ est $f(\mathcal{C},\mathcal{S}) = |\mathcal{O}_{\mathcal{C}}(\mathcal{S})| = 2$. Une seconde chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ est une chronique de taille 2 où $\mathcal{E}' = \{e'_1 = c, e'_2 = d\}$ et $\mathcal{T}' = \{\tau'_{(1,2)} = e'_1[-2,1]e'_2\}$. Les occurrences de la chronique $\mathcal{C}'$ dans la séquence temporelle $\mathcal{S}$ sont :

$$\mathcal{O}_{\mathcal{C}'}(\mathcal{S}) = \{o^1_{\mathcal{C}'}, o^2_{\mathcal{C}'}\} = \begin{cases} o^1_{\mathcal{C}'} = \{(c,7),(d,5)\}, \\ o^2_{\mathcal{C}'} = \{(c,13),(d,13)\}. \end{cases} \tag{1.5}$$

Une troisième chronique $\mathcal{C}'' = (\mathcal{E}'', \mathcal{T}'')$ est une chronique de taille 2 où $\mathcal{E}'' = \{e''_1 = c, e''_2 = d\}$ et $\mathcal{T}'' = \{\tau''_{(1,2)} = e''_1[-3,2]e''_2\}$. Les occurrences de la chronique $\mathcal{C}''$ dans la séquence temporelle $\mathcal{S}$ sont :

$$\mathcal{O}_{\mathcal{C}''}(\mathcal{S}) = \{o^1_{\mathcal{C}''}, o^2_{\mathcal{C}''}, o^3_{\mathcal{C}''}, o^4_{\mathcal{C}''}\} = \begin{cases} o^1_{\mathcal{C}''} = \{(c,7),(d,5)\}, \\ o^2_{\mathcal{C}''} = \{(c,11),(d,13)\}, \\ o^3_{\mathcal{C}''} = \{(c,13),(d,13)\}, \\ o^4_{\mathcal{C}''} = \{(c,16),(d,13)\}. \end{cases} \tag{1.6}$$

Cette troisième chronique montre que plusieurs occurrences d'une même chronique peuvent partager un même événement daté (ici, l'événement daté $(d,13)$).

En prenant en compte une chronique $\mathcal{C}$ et une séquence temporelle $\mathcal{S}$, plusieurs propriétés intéressantes de chroniques peuvent être définies. Ces propriétés sont établies formellement dans [Maitre 2015, Maitre 2014]. Elles sont particulièrement utiles dans l'objectif de définir si une chronique est pertinente ou non et permettent de répondre à différentes questions :

— *Une chronique peut-elle être reconnue ?*

— *Une chronique est-elle plus stricte/générale qu'une autre ?*

— *Deux chroniques sont-elles équivalentes ?*

Ces questions sont importantes lors de la manipulation des chroniques et en particulier sont un prérequis indispensable dans un contexte de découverte de chroniques. En effet, ce type d'algorithme utilise intensément ces notions.

Définition 1.12. Une chronique **cohérente** est une chronique $\mathcal{C}$ telle qu'il existe une séquence temporelle $\mathcal{S}$ ayant au moins une occurrence de $\mathcal{C}$:

$$\exists \mathcal{S}, \text{ telle que } \mathcal{O}_{\mathcal{C}}(\mathcal{S}) \neq \emptyset. \tag{1.7}$$

Exemple 1.5. La chronique $\mathcal{C}$ décrite dans l'exemple 1.4 est cohérente car il existe une séquence temporelle $\mathcal{S}$ telle que $\mathcal{O}_{\mathcal{C}}(\mathcal{S}) \neq \emptyset$. La chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ avec

$\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 20]e_2, \tau_{(2,3)} = e_2[3, 5]e_3, \tau_{(1,3)} = e_1[30, 40]e_3\}$ représentée sur la figure 1.7 est incohérente car il est impossible de satisfaire les contraintes temporelles $\tau_{(1,2)}$ et $\tau_{(2,3)}$ en même temps que la contrainte temporelle $\tau_{(1,3)}$. Ainsi, quelle que soit la séquence temporelle $\mathcal{S}$, $\mathcal{O}_{\mathcal{C}'}(\mathcal{S}) = \emptyset$.

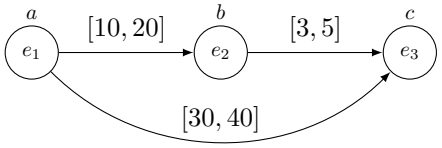
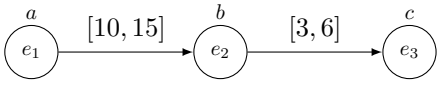


FIGURE 1.7 – Une chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ avec $\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 20]e_2, \tau_{(2,3)} = e_2[3, 5]e_3, \tau_{(1,3)} = e_1[30, 40]e_3\}$. Cette chronique est incohérente.

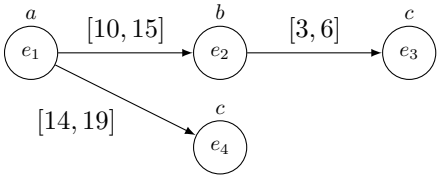
Définition 1.13. Une chronique $\mathcal{C}$ **couvre** une chronique $\mathcal{C}'$ si quelle que soit la séquence temporelle $\mathcal{S}$, toutes les occurrences de $\mathcal{C}'$ sont des sous-séquences des occurrences de $\mathcal{C}$:

$$\mathcal{C}' \sqsubseteq \mathcal{C} \Leftrightarrow \forall \mathcal{S}, \forall o_{\mathcal{C}'} \in \mathcal{O}_{\mathcal{C}'}(\mathcal{S}), \exists o_{\mathcal{C}} \in \mathcal{O}_{\mathcal{C}}(\mathcal{S}), o_{\mathcal{C}} \subseteq o_{\mathcal{C}'}. \tag{1.8}$$

Exemple 1.6. La chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ de taille 3 avec $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(2,3)} = e_2[3, 6]e_3\}$ représentée sur la figure 1.8a couvre la chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ de taille 4 avec $\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(1,4)} = e_1[14, 19]e_4, \tau_{(2,3)} = e_2[3, 6]e_3\}$ représentée sur la figure 1.8b, noté $\mathcal{C}' \sqsubseteq \mathcal{C}$.



(a) La chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ de taille 3 où $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(2,3)} = e_2[3, 6]e_3\}$.



(b) La chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ de taille 4 où $\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(1,4)} = e_1[14, 19]e_4, \tau_{(2,3)} = e_2[3, 6]e_3\}$.

FIGURE 1.8 – Deux chroniques représentant la propriété de couverture. La chronique $\mathcal{C}$ représentée sur la figure de gauche couvre la chronique $\mathcal{C}'$ représentée sur la figure de droite, noté $\mathcal{C}' \sqsubseteq \mathcal{C}$.

Définition 1.14. Deux chroniques $\mathcal{C}$ et $\mathcal{C}'$ sont **équivalentes** si les occurrences de $\mathcal{C}$ et $\mathcal{C}'$ sont identiques quelle que soit la séquence temporelle $\mathcal{S}$:

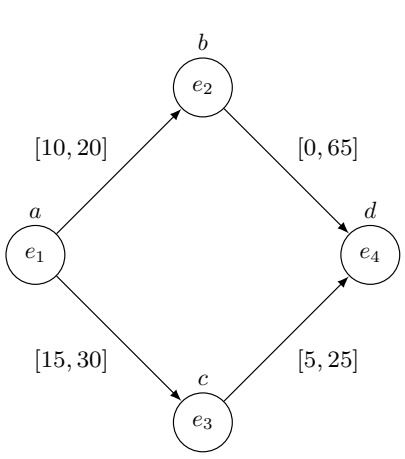
$$\mathcal{C} \equiv \mathcal{C}' \Leftrightarrow \forall \mathcal{S}, \mathcal{O}_{\mathcal{C}}(\mathcal{S}) = \mathcal{O}_{\mathcal{C}'}(\mathcal{S}). \tag{1.9}$$

Exemple 1.7. Les deux chroniques $\mathcal{C}$ et $\mathcal{C}'$ représentées sur la figure 1.9 sont équivalentes.

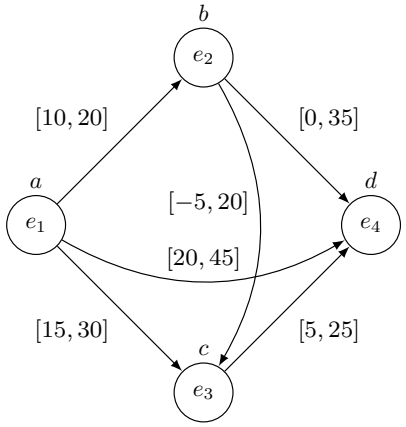
Définition 1.15. Une chronique $\mathcal{C}_{min}$ de taille n où tout couple de nœuds i, j a une contrainte temporelle $\tau_{(i,j)}$ est **minimale** si et seulement si elle respecte les conditions suivantes :

$$\begin{cases} \mathcal{T} = \{\tau_{(i,j)}\}_{1 \leq i < j \leq n} \\ \forall \mathcal{C}' = (\mathcal{E}', \mathcal{T}'), \mathcal{C}' \equiv \mathcal{C}, \tau'_{(i,j)} \in \mathcal{T}' \Rightarrow \tau_{(i,j)} \in \mathcal{T}, \tau_{(i,j)} \subseteq \tau'_{(i,j)}. \end{cases} \tag{1.10}$$

Exemple 1.8. La chronique $\mathcal{C}'$ représentée sur la figure 1.9b est minimale.



(a) La chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ où $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c, e_4 = d\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 20]e_2, \tau_{(1,3)} = e_1[15, 30]e_3, \tau_{(2,4)} = e_2[0, 65]e_4, \tau_{(3,4)} = e_3[5, 25]e_4\}$.



(b) La chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ où $\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c, e_4 = d\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 20]e_2, \tau_{(1,3)} = e_1[15, 30]e_3, \tau_{(1,4)} = e_1[20, 45]e_4, \tau_{(2,3)} = e_2[-5, 20]e_3, \tau_{(2,4)} = e_2[0, 35]e_4, \tau_{(3,4)} = e_3[5, 25]e_4\}$.

FIGURE 1.9 – Deux chroniques équivalentes. La chronique $\mathcal{C}'$ représentée sur la figure de droite est minimale.

La minimalité d'une chronique est très intéressante dans plusieurs domaines d'études des chroniques, et en particulier la reconnaissance. En effet, la chronique minimale est la chronique la plus descriptive parmi toutes ses chroniques équivalentes et est donc plus aisée à manipuler. Les chroniques minimales ainsi que le processus de minimisation de chroniques sont détaillés dans [Dousson 1994].

En raison de l'importance de ces propriétés dans les algorithmes de découverte de chroniques, il est crucial d'être en mesure de les vérifier rapidement. Grâce aux graphes de distance définis dans le cadre formel des réseaux de contraintes temporelles [Dechter 1991], ces propriétés peuvent être aisément vérifiées. En effet, plusieurs algorithmes décrits dans cet article vérifient ces propriétés en un temps polynomial.

1.2.3 Diagnostic à base de chroniques

La reconnaissance de chroniques en conjonction d'une base de chroniques suffisamment riche et bien modélisée est un outil efficace pour le diagnostic de faute. En effet, les chroniques possèdent un cadre formel riche décrit en partie dans la section précédente et parfaitement adapté à ce contexte. Dans cette section, le processus de reconnaissance de chroniques dans un flux d'événements datés en ligne est détaillé. De plus, plusieurs applications réelles des chroniques dans la littérature sont décrites.

CRS (*Chronicle Recognition System*) [Dousson 1993, Dousson 1994] est l'algorithme de reconnaissance de chroniques le plus commun dans la littérature. Nous le présentons ici de manière synthétique. En fonction du flux d'événements datés d'entrée, CRS maintient un ensemble d'*instances partielles*, qui est une partie d'une potentielle occurrence d'une chronique, en fonction des événements datés. Une instance partielle donne une *instance reconnue*, c'est-à-dire une occurrence d'une chronique, lorsque tous les événements datés respectent toutes les contraintes temporelles, et devient une *instance rejetée* lorsqu'au moins une des contraintes temporelles est violée. Afin de mieux comprendre le processus, les avantages et les défauts de CRS, une illustration d'une reconnaissance de chronique est donnée dans l'exemple suivant.

Exemple 1.9. Soit $\mathcal{C}^\alpha = (\mathcal{E}^\alpha, \mathcal{T}^\alpha)$ une chronique élémentaire représentée sur la figure 1.5 avec $\mathcal{E}^\alpha = \{e_1 = a, e_2 = b\}$ et $\mathcal{T}^\alpha = \{\tau_{(1,2)} = e_1[1,10]e_2\}$ et soit $\mathcal{S} = \{(a,10),(b,12),(a,20),(b,22),(b,27)\}$ une séquence temporelle avec cinq événements datés et $\mathbb{E}_\mathcal{S} = \{a,b\}$. La reconnaissance de la chronique $\mathcal{C}^\alpha$ dans la séquence temporelle $\mathcal{S}$ suivant le processus de CRS se fait par étapes qui sont mises en valeur dans la table 1.1. Un premier événement a se produit à l'instant 10, une instance partielle I_1 de $\mathcal{C}$ est créée. Puis, à l'instant 12, un événement b apparaît, la contrainte $\tau_{(1,2)}$ est respectée et l'instance partielle I_1 génère une instance reconnue I_2. Néanmoins, I_1 peut toujours générer des instances reconnues, elle est donc conservée dans l'ensemble des instances partielles. À l'instant 20, deux actions se produisent, un nouvel événement a se produit, ce qui créée une nouvelle instance partielle I_3, et la contrainte $\tau_{(1,2)}$ ne peut plus être respectée pour I_1, celle-ci est donc rejetée. Deux nouveaux événements b se produisent aux instants 22 et 27, donnant deux instances reconnues I_4 et I_5 générées par I_3. Enfin, il faut attendre l'instant 30 pour pouvoir rejeter l'instance partielle I_3.

La chronique élémentaire $\mathcal{C}^\alpha$ possède trois occurrences dans la séquence temporelle $\mathcal{S}$. En effet, les instances reconnues et les occurrences sont deux dénominations pour la même information. Les occurrences de $\mathcal{C}^\alpha$ sont les suivantes :

$$\mathcal{O}_{\mathcal{C}^\alpha}(\mathcal{S}) = \{o^1_{\mathcal{C}^\alpha}, o^2_{\mathcal{C}^\alpha}, o^3_{\mathcal{C}^\alpha}\} = \begin{cases} o^1_{\mathcal{C}^\alpha} = \{(a,10),(b,12)\}, \\ o^2_{\mathcal{C}^\alpha} = \{(a,20),(b,22)\}, \\ o^3_{\mathcal{C}^\alpha} = \{(a,20),(b,27)\}. \end{cases}$$

Ainsi, CRS génère de nombreuses instances partielles au cours de la reconnais-

TABLE 1.1 – Exemple de reconnaissance d'une chronique $\mathcal{C}^{\alpha}$ dans une séquence temporelle $\mathcal{S} = \{(a, 10), (b, 12), (a, 20), (b, 22), (b, 27)\}$ par l'algorithme CRS.

Événement	Instant d'occurrence	Instances partielles	Instances reconnues	Instances rejetées
a	10	$I_1 = \{(a, 10)\}$	-	-
b	12	-	$I_2 = \{(a, 10), (b, 12)\}$	-
a	20	$I_3 = \{(a, 20)\}$	-	$I_1 = \{(a, 10)\}$
b	22	-	$I_4 = \{(a, 20), (b, 22)\}$	-
b	27	-	$I_5 = \{(a, 20), (b, 27)\}$	-
-	30	-	-	$I_3 = \{(a, 20)\}$

sance. Ce nombre augmente exponentiellement lorsque plusieurs chroniques sont suivies par le reconnaisseur. CRS souffre du problème classique de l'explosion combinatoire. Néanmoins, ce problème peut être atténué [Dousson 2007] et CRS reste particulièrement efficace lorsque la base de chroniques à suivre est restreinte. Les autres algorithmes de reconnaissance de chroniques [Carle 2012] rencontrent malheureusement le même problème. Le maintien d'une base de chroniques de qualité est donc primordial à un diagnostic réussi.

Les chroniques trouvent notamment des applications dans le diagnostic en ligne de systèmes dynamiques. Une illustration est le domaine des services web, [Cordier 2007, Le Guillou 2008] utilisent une méthode à base de chroniques pour le diagnostic de systèmes distribués de services web. Dans [Pencolé 2009], les auteurs s'intéressent à la diagnosticabilité d'un système de diagnostic de services web à base de chroniques. Une autre application des chroniques peut être retrouvée dans la gestion d'alarmes dans un réseau web [Morin 2003] ou encore dans une usine pétrochimique [Vasquez Capacho 2017]. Pour plus d'applications des chroniques dans le diagnostic, [Subias 2013] offre un état de l'art des chroniques dans le domaine du diagnostic.

Plusieurs applications des chroniques dans le domaine du vivant existent également dans la littérature. Par exemple, [Carrault 1999] utilise des chroniques pour modéliser des arythmies cardiaques. Le parcours de soin est un aspect du domaine du vivant exploitant largement les chroniques. Un parcours de soin se compose des différentes procédures médicales, prise de médicaments et visites chez un médecin associées à un patient dans une durée de temps déterminée. [Huang 2012] utilise des chroniques pour modéliser des parcours de soin associés à plusieurs maladies : cancer du poumon, cancer de l'estomac, hémorragie cérébrale, cancer du sein, infarctus et cancer du côlon. [Dauxais 2017, Dauxais 2018] exploitent la base de donnée de l'assurance maladie française appelée SNIIRAM pour modéliser des chroniques associées à un parcours de soin.

1.3 Modèle des chroniques, un motif temporel parmi d'autres

Une chronique est un motif temporel particulièrement usité dans le domaine du diagnostic en raison de sa capacité d'abstraction des événements datés. Néanmoins, il existe dans la littérature de nombreux motifs temporels pertinents pour le diagnostic. Dans le reste de cette section, un état de l'art autour de quelques motifs temporels appropriés est réalisé.

Dans un premier temps, une sélection de motifs temporels simples avec uniquement des contraintes de temps qualitatives est étudiée. Un motif temporel de ce type est par exemple : *un événement A s'est produit avant un événement B*. Dans cet exemple, aucune information n'est donnée sur le temps écoulé entre les deux événements A et B. Néanmoins, une information temporelle est présente sous la forme d'une notion de précédence entre ces deux événements.

Puis, dans un second temps, des motifs temporels plus complexes où la valeur du temps est prise en compte sont définis. Contrairement aux motifs avec une information temporelle qualitative, ce type de motifs temporels possède une information supplémentaire. Par exemple : *un événement A s'est produit* 4 *unités de temps avant un événement B*. Ici, l'information sur la précédence est présente, on sait que A s'est produit avant B, et de plus le temps écoulé entre ces deux événements est donné, on sait que 4 unités de temps s'est écoulé entre A et B. Cette information quantitative rend ce type de modèles temporels plus descriptif mais également plus complexe et difficile à manipuler.

1.3.1 Information temporelle qualitative

Souvent, une information temporelle quantitative n'est pas nécessaire à l'efficacité d'un motif temporel. En effet, la manipulation d'une information temporelle qualitative est beaucoup plus simple que cela soit pour la commande, mais également pour le diagnostic. Dans le reste de cette section, quelques modèles temporels avec une information temporelle qualitative sont décrits.

Les treize relations d'intervalles d'Allen [Allen 1983], représentées sur la table 1.2, composent un motif temporel extrêmement simple et communément utilisé. En effet, les intervalles temporels offrent un outil puissant pour modéliser des incertitudes dans les systèmes dynamiques. Les treize relations d'intervalles d'Allen fournissent une première approche qui permet à une logique basée sur les intervalles temporels d'être efficace d'un point de vue de la complexité informatique. Cette logique permet de définir des motifs temporels basés sur ces intervalles.

Les *épisodes* [Mannila 1995, Mannila 1996, Mannila 1997] est un parfait exemple de motif temporel où l'information temporelle est présente uniquement sous la forme d'une notion de précédence. À l'instar des chroniques, les épisodes sont des ensembles partiellement ordonnés d'événements qui peuvent être représentés graphiquement sous la forme de graphes orientés acycliques. Ainsi, les épisodes peuvent être vus comme des chroniques sans contrainte temporelle quantitative. Suivant les

TABLE 1.2 – Les treize relations d'intervalles d'Allen [Allen 1983].

Relation	Symbole	Symbole inverse
X avant Y	$<$	$>$
X rencontre Y	m	mi
X chevauche Y	o	oi
X durant Y	d	di
X commence Y	s	si
X termine Y	f	fi
X égal à Y	$=$	

connexions entre les différents événements, les épisodes peuvent être catégorisés en trois types différents qui sont représentés sur la figure 1.10. Un épisode *série* représente un ensemble d'événements apparaissant séquentiellement, par exemple dans l'épisode (a) l'événement C se produit après l'événement B qui lui-même se produit après A. Un épisode *parallèle* est un ensemble d'événements sans contrainte entre eux, par exemple dans l'épisode (b) aucune contrainte sur l'ordre d'occurrence des événements A, B et C n'est donnée. Enfin, un épisode *hybride* est une combinaison d'épisodes *séries* et *parallèle*, par exemple dans l'épisode (c) les événements A et B doivent se produire avant les événements C et D, mais aucune contrainte n'est donnée sur l'ordre relatif des événements A et B (respectivement C et D). Comme les chroniques, les épisodes peuvent être appliqués dans le domaine du diagnostic. Par exemple, [Obry 2016, Obry 2017] utilise les épisodes pour le diagnostic dans le domaine automobile.

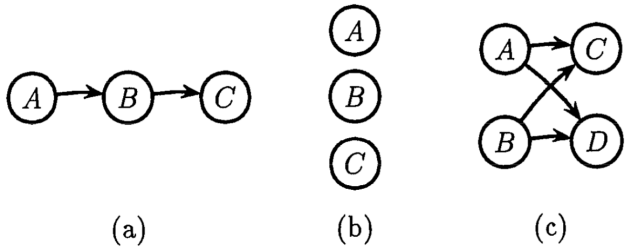


FIGURE 1.10 – Exemples des trois types possibles d'épisodes : (a) est un épisode *série*, (b) est un épisode *parallèle* et (c) est un épisode *hybride*. Image issue de [Mannila 1996].

Un autre motif avec une notion de précédence qui est particulièrement connu et utilisé dans la littérature, que ce soit pour la commande ou pour le diagnostic de systèmes, est le *réseau de Petri* [Peterson 1977]. Un réseau de Petri est un outil graphique et mathématique permettant de représenter des systèmes à événements discrets et possédant un cadre formel riche. Dans [Gougam 2015, Gougam 2017], les réseaux de Petri sont exploités pour la vérification de tels motifs.

1.3.2 Information temporelle quantitative

Les *automates temporisés* [Alur 1994] est un exemple de modèle temporel où l'information temporelle est quantifiée. Ils sont utilisés pour modéliser le comportement dynamique de systèmes informatiques. Les *réseaux de Petri temporisés* [Merlin 1976], contrairement aux réseaux de Petri classiques où seule l'information événementielle est nécessaire pour la progression du réseau, nécessitent également une information temporelle. Les réseaux de Petri temporisés sont particulièrement utilisés pour la capacité de vérification de modèles [Boufaied 2005]. Ce type de modèles peuvent être utilisés pour la vérification de l'information temporelle des chroniques [Pencolé 2009, Gougam 2012].

L'algèbre $(\max, +)$ [Baccelli 1992] a pour objectif de représenter des systèmes à événements discrets temporisés et en particulier les phénomènes de synchronisation entre équipements et des différents temps d'exécution des processus typiquement rencontrés dans les procédés industriels. Ce formalisme est basé sur une algèbre qui peut être représentée sous forme de GET (*Graphe d'Événements Temporisés*) qui est une sous-classe de réseaux de Petri temporisés. Un exemple d'un GET est représenté sur la figure 1.11. L'algèbre $(\max, +)$ est intéressant car il permet d'exploiter toutes les avancées dans le domaine de l'automatique continue, un domaine de recherche très actif. Pour ces raisons, l'algèbre $(\max, +)$ est très utilisé pour le contrôle de systèmes dynamiques. Par exemple, dans [Houssin 2007], une modélisation d'un réseau de bus urbain est faite grâce à l'algèbre $(\max, +)$.

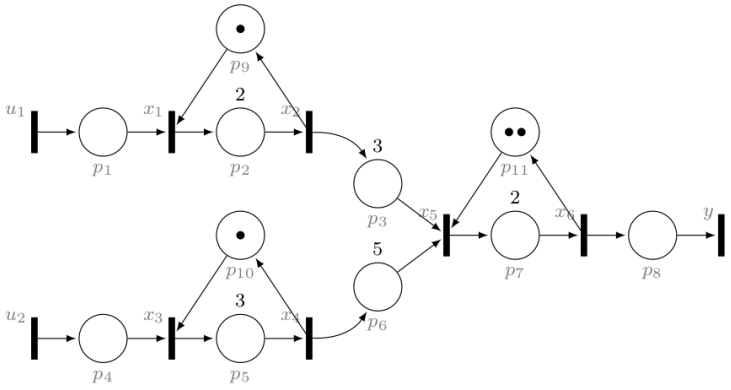


FIGURE 1.11 – Un exemple d'un GET représentant une ligne d'assemblage automatisée.

L'algèbre $(\max, +)$ est utilisé pour la détection [Sahuguède 2017] et la localisation [Sahuguède 2016, Le Corronc 2017, Le Corronc 2018] de décalages temporels fautifs dans des systèmes $(\max, +)$-linéaires. Ce travail est étendu aux systèmes incertains dans [Paya 2018]. Dans [Provan 2017], une algèbre similaire à $(\max, +)$ est exploitée dans le cadre du diagnostic de faute dans un système hybride autonome à temps discret. Enfin, les travaux de [Baniardalani 2013] utilisent une méthode basée sur $(\max, +)$ pour la prédiction de trajectoire dans des graphes de durées qui sont une sous-classe d'automates temporisés.

Un autre formalisme pour la modélisation de systèmes à événements discrets temporisés est appelé *condition template* [Pandalai 2000]. Ce formalisme permet de modéliser des processus où des comportements pouvant se produire une seule fois et des comportements pouvant se produire un nombre de fois non spécifié, et ce, de manière simultanée. Dans [De Smedt 2017], ce formalisme est utilisé pour la classification de séquences temporelles.

1.4 Extraction de connaissance à partir de données temporelles

Une chronique est un modèle temporel qui peut apparaître dans des données temporelles. La tâche de trouver des chroniques dans des données temporelles, qui est appelée découverte de chroniques, exploite des méthodes de fouille de données, et en particulier des méthodes d'extraction de modèles fréquents. L'extraction de modèles fréquents est le processus d'isoler des *modèles fréquents* dans des données. Un modèle fréquent est un ensemble d'éléments, une sous-séquence, ou une structure qui apparaît avec une fréquence supérieure à un seuil défini par l'utilisateur. Les modèles fréquents sont très variés dans leur nature. Quelques exemples de modèles fréquents communément rencontrés :

— un ensemble de produits, comme du *lait* et des *céréales*, fréquemment achetés ensemble dans une base de données de transactions de ventes,

— une séquence d'événements, comme la visite d'un patient chez un médecin généraliste, puis un centre de radio, et enfin une nouvelle visite chez le même médecin généralise, fréquemment retrouvée dans la base de données de parcours de soins.

Ainsi, un modèle fréquent n'est pas nécessairement un modèle temporel.

En raison de la multiplicité des formes d'un modèle fréquent, une étude des quelques algorithmes d'extraction de modèles fréquents pertinents au processus de découverte de chroniques est donnée dans cette section. Dans un premier temps, les modèles fréquents dont le temps n'est pas une caractéristique majeure sont définis. Puis, dans un second temps, les modèles temporels fréquents sont étudiés. L'extraction de modèles fréquents étant un domaine d'étude prolifique dans ces dernières années, seul un aperçu des travaux dans ce domaine est donné ici. Le lecteur intéressé peut se tourner vers [Mitsa 2010, Aggarwal 2014] qui donnent un échantillon représentatif de cet axe de travail.

1.4.1 Sans domaine temporel

La problématique de découverte de règles d'association entre produits dans une large base de données de transactions de vente est traitée dans [Agrawal 1993, Agrawal 1994, Agrawal 1995]. L'analyse des habitudes de transactions des clients dans un magasin permet de déduire de nombreuses règles d'association entre les différents produits placés dans le panier des clients. Par exemple, si un client achète

du lait, quelle est la probabilité que ce client achète également des céréales ? Et quel type de céréales ? Une règle d'association $X \Rightarrow Y$ est quantifiée par l'*indice de support* et l'*indice de confiance*. L'indice de support donne la proportion des transactions où les produits X et Y sont achetés (toutes les transactions où le client a acheté du lait et des céréales sur toutes les transactions). Alors que l'indice de confiance donne la proportion des transactions où le produit X est acheté et où le produit Y est acheté (toutes les transactions où le client a acheté du lait et des céréales sur toutes les transactions où le client a acheté du lait). La problématique de découverte de règles d'association est donc la suivante : *Soit un ensemble de transactions, le problème de découverte de règles d'association est de générer toutes les règles d'association ayant un indice de support et un indice de confiance supérieur aux minimums minsup et minconf donnés par l'utilisateur.*

Agrawal propose plusieurs algorithmes (AIS [Agrawal 1993], Apriori, AprioriTid [Agrawal 1994], AprioriSome, et AprioriAll [Agrawal 1995]) répondant à cette problématique et qui reposent sur une approche de type *générer & compter*. L'algorithme le plus connu est sans conteste Apriori qui est un algorithme classique dans le domaine de la fouille de données. Apriori construit itérativement des ensembles de plus en plus grands en deux étapes : une étape *générer* où des ensembles de taille i sont construits à partir des ensembles de taille $i-1$ prouvés fréquents dans l'itération précédente ; puis une étape *compter* où la fréquence des ensembles ainsi générés est calculée dans la base de transactions traitée.

C'est grâce à la propriété d'anti-monotonie des données que les approches générer & compter sont efficaces et populaires. En effet, cette propriété est très utile car elle permet de largement limiter le nombre de modèles construits dans l'étape générer. Elle signifie que si un ensemble est prouvé non-fréquent, c'est-à-dire qu'il ne respecte pas le critère de fréquence donné (ici l'indice de support minimal *minsup*), alors tous ses sur-ensembles sont non-fréquents. En effet, si du lait et du poulet sont rarement achetés ensemble, il n'est pas nécessaire de regarder si du lait, du poulet et des bananes sont fréquemment achetés ensemble car les produits de type "lait" et "poulet" ne respectent pas ce critère.

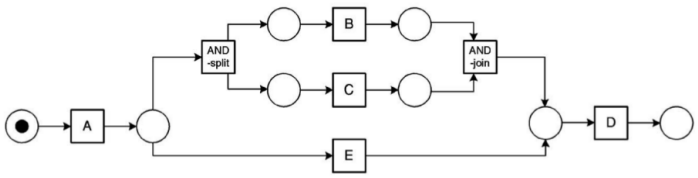
Dans [Agrawal 1995], Agrawal montre qu'il est aisé d'adapter Apriori à des séquences d'événements. Plusieurs sujets d'étude découlent des règles d'association. Notamment, la prise en compte d'une notion de précédence est une problématique très intéressante. Par exemple, *si un client achète un appareil photo numérique, quelle est la probabilité que ce même client achète une carte mémoire par la suite ?* Ainsi, la découverte de séquences peut être vue comme une généralisation du problème de découverte de règles d'association. SPADE [Zaki 2001] est un tel algorithme de découverte de séquences appliqué à un contexte de prédiction de fautes. Les épisodes peuvent être interprétés comme des séquences d'événements et une approche du type d'Apriori est appliquée dans [Mannila 1995, Mannila 1997]. Enfin, [Gunopulos 2003] propose un algorithme appelé *Dualize & Advance* qui est une solution possible à la problématique de la taille restreinte des résultats des algorithmes du type d'Apriori.

Dans le domaine de la gestion de procédés, où des processus complexes sont

constitués de nombreuses tâches différentes qui doivent être appliquées avec un ordonnancement précis, l'extraction de flux de travaux dans un journal de travail [Agrawal 1998, van der Aalst 2004] est utilisé. Un *flux de travaux* est un modèle constitué de plusieurs tâches qui doivent être effectuées dans un séquencement spécifique. Par exemple, un flux de travaux est représenté sur la figure 1.12b sous la forme d'un réseau de Petri. Ce modèle commence par une tâche A et se termine par une tâche D. Après exécution de la tâche A, soit les tâches B et C sont accomplies en parallèle, soit seule la tâche E est réalisée. Ainsi le problème d'extraction de flux de travaux consiste à retrouver ce type de flux dans un journal de travail. Un journal de travail consistant en une séquence d'événements datés ordonnés dans le temps où chaque événement daté est associé à une *tâche* (l'identifiant de l'action) et à un *cas* (l'identifiant du processus en cours). Un exemple de journal de travail est représenté sur la figure 1.12a où cinq cas sont répertoriés. Dans quatre de ces cas (les cas 1, 2, 3, et 4), les tâches A, B, C, et D sont exécutées. Par contre, dans le cinquième cas (le cas 5), seules les tâches A, E, et D sont exécutées. Ce journal de travail contient ainsi plusieurs occurrences du même flux de travaux représenté sur la figure 1.12b.

case identifier	task identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

(a) Un exemple de journal de travail.



(b) Un exemple d'un modèle de flux de travaux représenté sous la forme d'un réseau de Petri.

FIGURE 1.12 – Exemple de données d'entrée et de résultats pour un algorithme d'extraction de flux de travaux. Images issues de [van der Aalst 2004].

Les algorithmes d'extraction de flux de travaux [Agrawal 1998, van der Aalst 2004] peuvent être vus comme une généralisation des algorithmes de fouille de séquences. Ils sont basés sur une approche du même type que Apriori. Dans [van der Aalst 2007], une extraction de processus métiers est proposée. Les *processus métier* sont des versions plus détaillées des flux de travaux et sont notamment utilisées pour l'analyse des infrastructures sociales des entreprises. Elles permettent de répondre aux deux types de questions telles que : *Quel est l'interlocuteur privilégié d'Henri ? Quel est l'impact d'Henri sur une tâche*

donnée ? Quelle est la tâche la plus critique ? L'analyse du processus métier d'une organisation gouvernementale hollandaise est présentée dans [van der Aalst 2007] et met en valeur plusieurs problèmes organisationnels.

Outre l'approche Apriori, d'autres méthodes de fouille de séquences sont utilisées dans la littérature, en particulier dans le domaine de la requête de séquences dans une base de données [Mannila 2001, Wongsuphasawat 2012]. Notamment, [Mannila 2001] propose un tel algorithme via une projection aléatoire, où toutes les sous-séquences similaires à une séquence temporelle donnée doivent être retrouvée dans une longue séquence d'événements datés. Cet algorithme fonctionne en projetant aléatoirement les différentes sous-séquences dans un espace euclidien k-dimensionnel, puis ces projections sont comparées grâce à un simple calcul de distance euclidienne. La projection aléatoire est une méthode de réduction de dimension communément utilisée [Bingham 2001, Atkison 2009].

1.4.2 Avec domaine temporel

La prise en compte du temps dans la problématique d'extraction de connaissance à partir de données temporelles introduit de nombreuses problématiques intéressantes. En effet, la prise en compte du temps peut être vue comme une généralisation des problèmes traités dans la section précédente. Dans le reste de cette section, plusieurs algorithmes communs proposant une solution à cette nouvelle problématique sont décrits.

Les Δ-*patterns* [Yoshida 2000, Giannotti 2006] sont des schémas temporels simples qui offrent une solution à la problématique d'extraction de modèles prenant en compte l'information temporelle à partir de données provenant de systèmes dynamiques. Ils sont représentés sous la forme d'événements qui s'enchaînent de manière séquentielle avec une information temporelle entre chaque événement. Par exemple, le Δ-pattern suivant :

$$A \xrightarrow{[3,4]} B \xrightarrow{[0,2]} C, \tag{1.11}$$

représente un phénomène dynamique où un événement A se produit suivi d'un événement B entre 3 et 4 unités de temps, qui est lui-même suivi par un événement C entre 0 et 2 unités de temps. Ce type de modèle est une généralisation de séquences d'événements considérées dans la section précédente où le temps entre deux événements est quantifié. Néanmoins, les Δ-patterns restent un motif temporel plus simple que les chroniques car elles ne peuvent représenter qu'un ensemble totalement ordonné d'événements datés. [Yoshida 2000] utilise une variante d'Apriori pour extraire des Δ-patterns d'un ensemble séquences temporelles.

Dans [Mörchen 2010b], des motifs temporels avec des semi-intervalles sont proposés : les SISP (*Semi Interval Sequence Pattern*), un motif temporel totalement ordonné, et les SIPO (*Semi Interval Partial Order pattern*), un motif partiellement ordonné. Les SISP sont extraits d'une base de séquences de semi-intervalles à l'aide d'un algorithme d'extraction de motifs séquentiels clos tel que BIDE [Wang 2004].

Puis, les SISP sont combinés en SIPO. BIDE-Discriminative [Fradkin 2015] est une variante de BIDE proposant une solution à la problématique de classification de séquences temporelles.

Une autre méthode d'extraction de connaissance à partir de données temporelles est la fouille de motifs temporels dans un ensemble de séquences temporelles [Guyet 2008, Guyet 2011]. Dans ces articles, les motifs temporels recherchés correspondent à des séquences temporelles avec une date d'occurrence et une durée associées à chaque événement. Par exemple, $((A, [0.1, 3.4]), (B, [3.8, 5.3]))$ est un tel schéma temporel. Deux algorithmes sont proposés, QTempIntMiner [Guyet 2008] et QTIPrefixSpan [Guyet 2011]. QTempIntMiner cherche un ensemble d'événements fréquents grâce à Apriori, puis définit l'information quantitative à l'aide d'une estimation de la densité de distribution des intervalles de temps associée aux événements. QTIPrefixSpan, au contraire, mêle extraction de schéma séquentiel et caractérisation des intervalles de temps.

La quantification du temps dans la problématique de la fouille de données temporelles amène de nouvelles perspectives. Par exemple, [Ma 2001] propose une méthode en deux étapes pour extraire des schémas temporels, appelés *p-patterns*, sans avoir connaissance de la période de temps de ces motifs. Ces deux étapes consistent à : découvrir la période la plus adaptée grâce à un test du χ^2 [Pearson 1900] ; trouver les associations temporelles à l'aide d'un algorithme de type générer & compter. Une autre problématique est la gestion de schémas temporels sur différentes granularités du temps [Bettini 1998, Giannella 2002]. En effet, des modèles temporels qui sont représentatifs de phénomènes se produisant sur des échelles différentes (de l'ordre de la minute par rapport à un ordre de la journée par exemple) peut s'avérer utiles dans certains domaines d'études particuliers.

1.5 Découverte de chroniques

Comme vu dans la section précédente, l'algorithme Apriori est une approche communément utilisée dans le domaine de l'extraction de modèles fréquents. La plupart des algorithmes de découverte de chroniques reposent également sur cette base de travail, notamment, les algorithmes FACE [Dousson 1999] et HCDA [Cram 2012] qui sont les points de départ aux contributions proposées dans ce document. Ainsi, dans le reste de cette section, ces deux algorithmes sont décrits en détail. De plus, une vue d'ensemble des différents algorithmes de découverte de chroniques présents dans la littérature est proposée. Enfin, un comparatif des principales caractéristiques des algorithmes décrits est établi.

Avant d'étudier les algorithmes FACE et HCDA, certaines notions qui n'étaient pas nécessaires jusqu'à présent sont définies. Les définitions d'ensembles des occurrences et des distances temporelles d'un couple d'événements datés permettent un pré-traitement des séquences temporelles en entrée qui est utilisé dans ces algorithmes.

Définition 1.16. Une **distance temporelle** $d(t_i, t_j)$ est le temps écoulé entre deux instants $t_i, t_j \in \mathbb{T}$. Elle est définie par la formule suivante :

$$d(t_i, t_j) = t_j - t_i. \tag{1.12}$$

L'utilisation du terme "distance" est abusive ; on devrait parler de quasimétrique puisque que cette définition montre qu'elle ne satisfait pas à la propriété de symétrie d'une distance ($d(t_i, t_j) \neq d(t_j, t_i)$).

Définition 1.17. L'**ensemble des occurrences** $\mathcal{O}_{ab}$ d'un couple (a, b) dans une séquence temporelle $\mathcal{S}$ telle que $a, b \in \mathbb{E}$ et $a \leq_{\mathbb{E}} b$ est définie par la formule suivante :

$$\mathcal{O}_{ab}(\mathcal{S}) = \left\{ \langle (e_i, t_i), (e_j, t_j) \rangle \mid (e_i, t_i), (e_j, t_j) \in \mathcal{S}, \begin{cases} e_i = a,\ e_j = b & \text{si } a <_{\mathbb{E}} b \\ i < j,\ e_i = a,\ e_j = b & \text{si } a =_{\mathbb{E}} b \end{cases} \right\}. \tag{1.13}$$

Définition 1.18. L'**ensemble des distances temporelles** $\mathcal{D}_{ab}$ d'un couple (a, b) dans une séquence temporelle $\mathcal{S}$ telle que $a, b \in \mathbb{E}$ et $a \leq_{\mathbb{E}} b$ est défini par :

$$\mathcal{D}_{ab}(\mathcal{S}) = \{ d(t_i, t_j) \mid \langle (e_i, t_i), (e_j, t_j) \rangle \in \mathcal{O}_{ab}(\mathcal{S}) \}. \tag{1.14}$$

Exemple 1.10. Soit $\mathcal{S}$ la séquence temporelle décrite dans l'exemple 1.3. Il existe huit occurrences du couple (a, c) dans $\mathcal{S}$. Ces occurrences sont :

$$\begin{aligned} \mathcal{O}_{ac}(\mathcal{S}) = \{ &\langle (a,2), (c,7) \rangle, \langle (a,2), (c,11) \rangle, \langle (a,2), (c,13) \rangle, \langle (a,2), (c,16) \rangle, \\ &\langle (a,8), (c,7) \rangle, \langle (a,8), (c,11) \rangle, \langle (a,8), (c,13) \rangle, \langle (a,8), (c,16) \rangle \}. \end{aligned} \tag{1.15}$$

De plus, les distances temporelles du couple (a, c) sont :

$$\mathcal{D}_{ac}(\mathcal{S}) = \{5, 9, 11, 14, -1, 3, 5, 8\}. \tag{1.16}$$

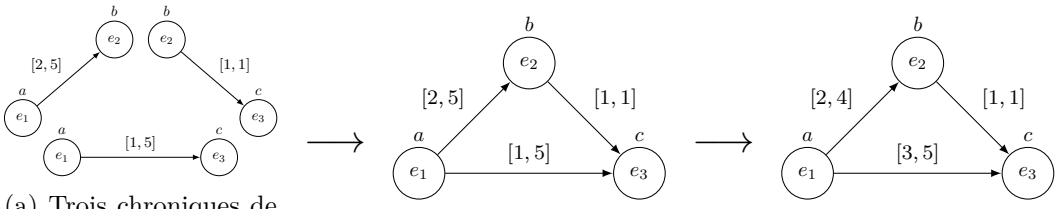
1.5.1 FACE (*Frequency Analyzer for Chronicle Extraction*)

L'algorithme FACE [Dousson 1999, Vu Duong 2001] repose sur la même propriété d'anti-monotonie des schémas temporels extraits que Apriori [Agrawal 1994] exploite. La sous-chronique de taille $i - 1$ couvrant celle de taille i, elle ne peut en effet être qu'au moins aussi fréquente que la chronique de taille i. Cette propriété d'anti-monotonie est au cœur de l'approche dite générer & compter. Ainsi, l'algorithme FACE construit itérativement des chroniques de taille i à partir des chroniques de taille $i-1$, et ce jusqu'à ne plus obtenir de chroniques candidates qui respectent le critère de fréquence.

La première itération de génération de chroniques est particulière. Cette étape de génération de chroniques de taille i, où $i = 2$, se fait suivant un procédé similaire à celui vu précédemment. La principale différence se fait lors de l'étape de génération des contraintes temporelles. Dans ce cas, l'unique contrainte temporelle peut être déduite des instants des différentes occurrences de la chronique non contrainte $\mathcal{C}$. En effet, dans le cas d'une chronique non contrainte de taille 2, leurs occurrences

peuvent être obtenues grâce à l'équation 1.13. Si une contrainte temporelle $[t^-, t^+]$ est satisfaite par une certaine proportion des occurrences $\mathcal{O}_\mathcal{C}(\mathcal{S})$, appelée la *note minimale de couverture* $\mathcal{Q}$, alors cette contrainte temporelle est dite *acceptable*. Les contraintes temporelles peuvent être aisément obtenues en utilisant une fenêtre glissante de taille définie par $\mathcal{Q} \times |\mathcal{O}_{ab}(\mathcal{S})|$ sur l'ensemble des distances temporelles obtenues à l'aide de l'équation 1.14. Puis deux mesures sur les contraintes temporelles permettent de choisir une unique [1] contrainte temporelle parmi ces contraintes temporelles acceptables. Les mesures utilisées sont : la distance temporelle $d(t^-, t^+)$, puis en cas d'égalité, la durée maximale de l'intervalle $\max(|t^-|, |t^+|)$.

Puis, une fois les chroniques de taille 2 générées, la génération des chroniques candidates de taille i à partir des sous-chroniques de taille $i-1$ se fait en deux étapes : la première étape consiste à calculer une chronique fréquente qui ne possède aucune contrainte temporelle ; la seconde étape permet d'établir les contraintes temporelles des différentes chroniques fréquentes. Dans la première étape, seuls les événements sont considérés. Pour chaque chronique de taille $i-1$ fréquente, une chronique candidate est générée en rajoutant un événement appartenant à $\mathcal{E}$, et ce, pour chaque événement de $\mathcal{E}$. Ainsi, si $\mathcal{E}$ contient trois événements, chaque chronique de taille $i-1$ fréquente génère trois chroniques candidates qui ne possèdent pas de contrainte temporelle. La fréquence des chroniques non contraintes ainsi générées est calculée grâce à la formule suivante $f(\mathcal{C}, \mathcal{S}) = \left\lfloor \min_{a \in \mathbb{E}} \left(\frac{f(a,\mathcal{S})}{f(a,\mathcal{C})} \right) \right\rfloor$ où $f(a, \mathcal{S})$ est la fréquence de l'événement a dans la séquence temporelle $\mathcal{S}$ et $f(a, \mathcal{C})$ est la fréquence de l'événement a dans la chronique $\mathcal{C}$. La seconde étape consiste à établir les contraintes temporelles des chroniques de taille i à partir des différentes sous-chroniques de taille $i-1$ qui sont fréquentes. Les chroniques ainsi générées sont ensuite minimisées (cf. définition 1.15). La figure 1.13 est un exemple de la génération d'une chronique de taille 3 à partir de chroniques de taille 2. Une nouvelle étape de vérification de la fréquence des chroniques est ensuite effectuée grâce à un algorithme de reconnaissance de chroniques tel que CRS [Dousson 1993].



(a) Trois chroniques de taille 2 qui sont fréquentes.

(b) Une chronique de taille 3 fusionnée.

(c) Une chronique de taille 3 candidate.

FIGURE 1.13 – Un exemple de génération d'une chronique de taille 3 à partir de chroniques de taille 2 issu de l'article [Dousson 1999] par l'algorithme FACE.

Une propriété intéressante de cet algorithme est qu'il est tout à fait possible d'omettre cette étape de découverte de chroniques de taille 2. Dans ce cas, l'utili-

1. Dans le pire des cas, deux contraintes temporelles ne sont pas différenciables. Deux chroniques de taille 2 sont alors générées.

sateur fournit une base de chroniques de taille $i-1$ qui servira comme départ pour la génération de chroniques de taille i. FACE n'est pas un algorithme complet car il impose un choix lors de la génération de chroniques de taille 2 par le choix de la note de couverture $\mathcal{Q}$.

Des travaux plus récents ont permis de proposer un prétraitement permettant de ne s'intéresser qu'aux sections de la séquence temporelle les plus riches en événements. Par l'utilisation d'une méthode de prétraitement sur la séquence temporelle d'entrée réduisant significativement la taille de celle-ci, les performances de FACE sont améliorées [Fessant 2004, Fessant 2006]. Ce prétraitement utilise des cartes d'auto-organisation [Vesanto 2000] pour extraire les sous-séquences les plus riches en événements et recomposer une séquence temporelle composée uniquement de ces riches sous-séquences.

1.5.2 HCDA (*Heuristic Chronicle Discovery Algorithm*)

L'algorithme HCDA [Cram 2008, Cram 2010, Cram 2012] est également basé sur une approche générer & compter. Néanmoins, la propriété la plus intéressante de cet algorithme est sa complétude. En effet, un algorithme complet garantit que toutes les chroniques qui respectent le critère de fréquence requis sont extraites de la séquence temporelle examinée. HCDA est découpé en deux grandes étapes : la première consiste en la construction d'une base de contraintes temporelles à partir d'une séquence temporelle ; la seconde exploite le principe de générer & compter pour construire itérativement des chroniques de plus en plus complexes. La construction des chroniques est faite grâce à deux opérations : l'ajout d'un événement à la chronique ou le renforcement d'une des contraintes temporelles avec l'exploration de la base de contraintes temporelles construite.

La construction d'une base de contraintes temporelles complète est la première étape d'HCDA. Tout d'abord, les distances temporelles sont calculées pour chaque couple d'événements présents dans la séquence temporelle explorée par le biais des équations (1.13) et (1.14). Chaque ensemble de distances temporelles est ensuite trié. Puis, une fenêtre glissante parcourt chaque ensemble de manière itérative avec une largeur croissante, en partant de $f_{seuil}-1$ jusqu'à la taille de chaque ensemble $n-1$. Chaque fenêtre de largeur différente donne un niveau différent dans la base de contraintes temporelles. Une relation d'ordre partielle est définie sur la base de contraintes temporelles. La racine correspond à la contrainte définie par la fenêtre de taille $n-1$ et où chaque nœud possède au plus deux enfants qui correspondent aux restrictions de chaque borne de la contrainte temporelle du nœud considéré.

Exemple 1.11. Soit $\mathcal{D}_{ab} = \{-2,-1,0,1,1,12,12,13,15,20\}$ un ensemble de distances temporelles du couple d'événement (a,b). En choisissant la fréquence objective f_{seuil} d'HCDA à 5, la base de contraintes temporelles générée par HCDA est représentée sur la figure 1.14. Chaque niveau possède une fréquence différente, avec au sommet une fréquence de 10 et dans la partie inférieure, une fréquence égale à la fréquence seuil de 5.

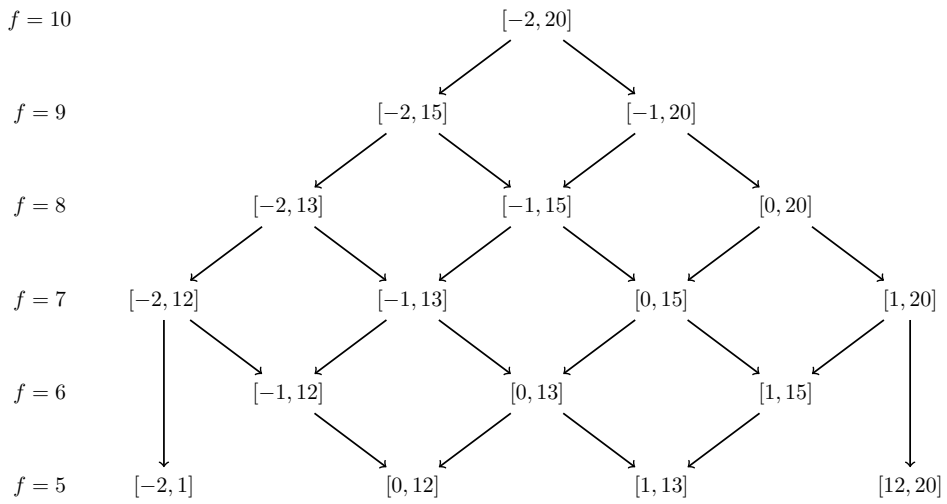


FIGURE 1.14 – Un exemple d'une base de contraintes temporelles générée par HCDA pour le couple d'événement (a, b). Cette base de contraintes temporelles est générée à partir de l'ensemble de distance temporelles $\mathcal{D}_{ab} = \{-2, -1, 0, 1, 1, 12, 12, 13, 15, 20\}$.

La figure 1.15 représente un autre exemple d'une base de contraintes temporelles simples où seuls les couples d'événements (a, b), (b, c) et (a, c) sont présents.

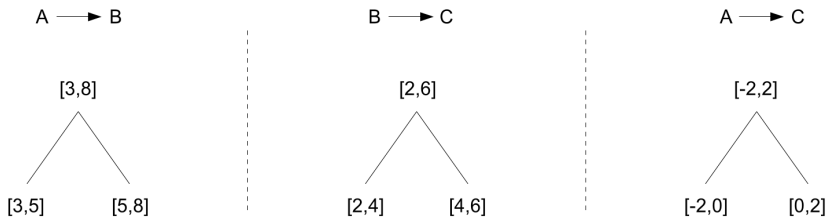


FIGURE 1.15 – Un exemple de base de contraintes temporelles utilisé par HCDA. Seuls trois couples d'événement sont présents dans cet exemple : (a, b), (b, c) et (a, c). Cette figure provient de [Cram 2008].

Une autre méthode de construction de cette base de contraintes temporelles est possible et reprend l'approche de [Dousson 1999]. Cette méthode, appelée *méthode de Dousson & Duong*, est décrite dans [Cram 2012] et permet d'obtenir des résultats similaires à FACE.

L'étape d'extraction complète de chroniques commence par la génération de chroniques de taille 2 à partir des racines des différentes bases de contraintes temporelles générées dans l'étape précédente. Puis, les chroniques sont itérativement rendues plus contraintes à l'aide de deux opérations : l'ajout d'un événement ou la restriction d'une des contraintes temporelles en explorant la base de contraintes temporelles. Chaque opération produit une nouvelle chronique candidate, celle-ci est alors comparée aux chroniques déjà connues qui sont regroupées dans deux bases de chroniques différentes : une base de chroniques *fréquentes* et une base de chroniques

non fréquentes. Trois situations peuvent émerger :

- Si la chronique candidate couvre une des chroniques *fréquentes*, alors elle est forcément fréquente et est rajoutée à la base des chroniques *fréquentes*.
- Au contraire, si une des chroniques *non fréquentes* couvre la chronique candidate, alors elle ne peut pas être fréquente et est rajoutée à la base des chroniques *non fréquentes*.
- Enfin, si la chronique candidate ne rentre dans aucun des deux cas cités précédemment, alors la chronique candidate est *comptée* à l'aide d'un algorithme de reconnaissance de chroniques, tel que CRS [Dousson 1993].

Remarque 1.1. Dans [Cram 2012], lorsqu'une chronique $\mathcal{C}_2$ couvre une chronique $\mathcal{C}_1$ (respectivement $\mathcal{C}_1$ couvre $\mathcal{C}_2$), on dit que $\mathcal{C}_1$ est *plus stricte* (respectivement *moins stricte*) que $\mathcal{C}_2$.

L'utilisateur peut, à chaque itération, contrôler les résultats obtenus et appliquer les opérations suivantes : supprimer des chroniques qui lui semblent peu intéressantes, effectuer une nouvelle itération de l'algorithme ou encore arrêter l'algorithme car les résultats lui semblent suffisants.

Exemple 1.12. Pour mieux comprendre le processus itératif mis en place par HCDA, prenons l'exemple représenté sur la figure 1.16. En considérant la base de contraintes temporelles représentée sur la figure 1.15, la chronique de taille 2 $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ avec $\mathcal{E}_1 = \{e_1 = a, e_2 = b\}$ et $\mathcal{T}_1 = \{\tau_{(1,2)} = e_1[3, 8]e_2\}$ est générée à l'étape (1). Puis dans la première itération de l'algorithme, deux opérations de restriction de la contrainte temporelle sont possibles (voir la base de contraintes à gauche sur la figure 1.15) et trois opérations d'ajout d'un nouvel événement sont possibles (l'ajout de a, de b ou de c). Néanmoins, les ajouts des événements a et b ne sont pas possibles car il n'y a pas de contrainte temporelle sur le couple d'événement (a, a) ou (b, b). Donc, trois nouvelles chroniques directement *plus strictes* sont obtenues à l'étape (2). Les deux chroniques de gauche sont obtenues par restriction et celle de droite par l'ajout de l'événement c et des contraintes associées issues de la base. Puis, en partant de la première chronique générée, une nouvelle chronique *plus stricte* est construire dans l'étape (3) par l'ajout de l'événement c. Enfin, dans l'étape (4), quatre nouvelles chroniques sont extraites, ces chroniques sont toutes obtenues par restriction.

Malgré sa complexité exponentielle, cet algorithme reste encore aujourd'hui très intéressant en raison de sa complétude. En effet, en répétant suffisamment d'itérations, les chroniques les plus pertinentes respectant le critère de fréquence imposé sont garanties d'être extraites. HCDA est le seul algorithme de découverte de chroniques complet à notre connaissance.

Dans [Vasquez Capacho 2017], une extension à HCDA appelée HCDAM est proposée et est appliquée dans un contexte de diagnostic d'une entreprise pétrochimique. Cette extension propose d'améliorer les chroniques obtenues en considérant non plus une seule séquence temporelle mais plusieurs. La phase de *comptage* est modifiée telle que chaque chronique est comptée dans chaque séquence temporelle et est considérée fréquente si sa fréquence est supérieure à la fréquence seuil dans

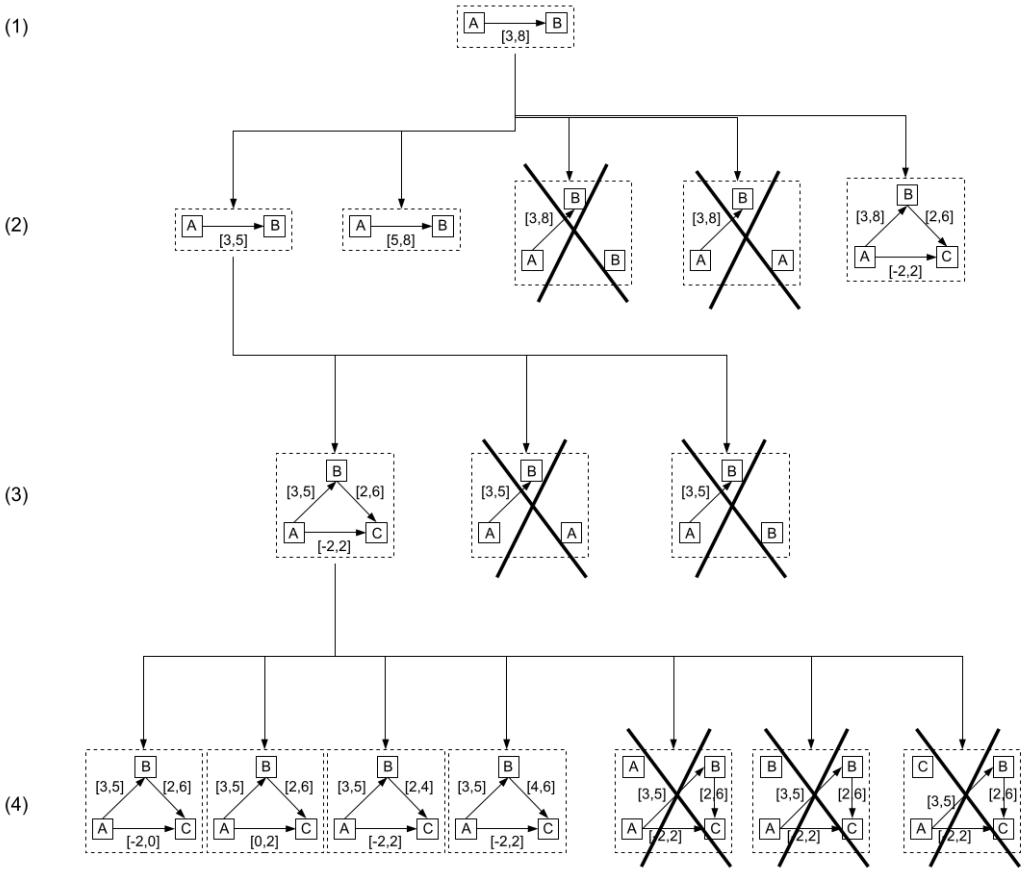


FIGURE 1.16 – Génération de chroniques *plus strictes* par le biais des opérations ajout d'un événement et restriction d'une contrainte temporelle d'HCDA. Cette génération est développée en utilisant la base de contraintes temporelles complète représentée figure 1.15. Ici, seule une chronique est définie dans l'étape (1) de génération de chroniques de taille 2 à partir d'une base de contraintes temporelles. Cette figure est issue de [Cram 2008].

toutes les séquences temporelles. Grâce à ce critère de choix supplémentaire, la quantité des chroniques découvertes est moindre par rapport à HCDA, mais les auteurs ne précisent pas si la propriété de complétude est conservée ou non.

1.5.3 Autres approches pour la découverte de chroniques

L'approche abordée par FACE et HCDA génère des chroniques respectant un critère de fréquence, toutes les chroniques découvertes ont une fréquence supérieure à un seuil défini par l'utilisateur dans une seule séquence temporelle. Une approche différente est de générer des chroniques avec un critère de support, les chroniques ont ainsi un support supérieur à un seuil défini dans une base de séquences temporelles. Ainsi, avec cette approche, le critère de choix des chroniques intéressantes n'est plus la fréquence mais la proportion d'occurrences de celles-ci dans un ensemble de

séquences temporelles.

Dans [Huang 2012], une telle approche est appliquée dans le domaine médical et en particulier dans la fouille de parcours de soin. Les données d'entrée considérées par cette approche consiste en un ensemble de séquences temporelles représentant un parcours de soin. Un exemple de ce type de données est représenté sur la figure 1.17. Ainsi, chaque séquence temporelle contient seulement une occurrence d'un phénomène d'intérêt. La méthodologie de découverte de chroniques proposée dans cet article est découpée en trois algorithmes distincts. Le premier algorithme, nommé SCP-Miner, permet la fouille de parcours de soin clos respectant le critère *minsupp* défini par l'utilisateur. Cet algorithme fonctionne suivant une approche générer & compter en générant de manière récursive des parcours de soin de plus en plus grand. Puis, les algorithmes MATC-Miner et CCP-Miner permettent de générer des chroniques à partir des séquences temporelles offertes par SCP-Miner. Ces deux algorithmes œuvrent de la manière suivante : MATC-Miner permet de générer les chroniques élémentaires les plus intéressantes contenant un ensemble d'événements jugés importants par l'utilisateur ; CCP-Miner utilise les chroniques élémentaires et les enrichit grâce à une méthode de type générer & compter pour obtenir des chroniques pertinentes dans les séquences temporelles fournies par SCP-Miner.

id	Sequence
σ_1	$\langle(a,1),(b,1),(c,1),(d,1),(j,1),(f,2),(q,2),(g,4),(h,4),(i,4),$ $(s,7),(r,8),(o,13),(t,15),(v,16)\rangle$
σ_2	$\langle(a,1),(b,1),(c,1),(d,1),(e,1),(f,9),(g,10),(h,10),(i,10),$ $(j,11),$ $(s,14),(r,18),(k,22),(t,22),(l,24),(v,24)\rangle$
σ_3	$\langle(a,1),(b,1),(c,1),(d,1),(e,1),(m,2),(n,2),(j,3),(g,4),(h,$ $4),(i,4),(s,7),$ $(r,13),(o,17),(p,17),(u,19),(k,21),(l,21),(t,21),(v,21)\rangle$
σ_4	$\langle(a,1),(b,1),(c,1),(d,1),(j,3),(q,3),(g,5),(h,5),(i,5),$ $(s,9),(r,12),(o,18),(t,21),(v,21)\rangle$

FIGURE 1.17 – Un exemple d'un ensemble de séquences temporelles représentant des parcours de soin d'un cancer du poumon. Par exemple, l'événement a correspond à l'admission du patient à l'hôpital, l'événement b est un examen d'ultrasons en couleur, et l'événement c est une électrocardiographie. L'intégralité des événements sont décrits dans [Huang 2012].

Suivant la même approche, [Dauxais 2017, Dauxais 2018] propose un algorithme de découverte de chroniques *discriminantes* nommé DCM. Une chronique discriminante est une chronique qui possède plus d'occurrences dans un ensemble de séquences temporelles donné que dans un autre. DCM exploite comme données d'entrée un ensemble de séquences temporelles étiquetées $+$, pour les séquences temporelles contenant un phénomène d'intérêt, ou $-$, pour les séquences ne contenant pas de phénomènes d'intérêt. Cet algorithme fournit des chroniques discriminantes en deux étapes. Dans un premier temps, seuls les multi-ensembles d'événements fréquents sont générés. Puis, dans un second temps, les chroniques discriminantes sont découvertes à partir de ces multi-ensembles fréquents et avec une extraction des

différentes contraintes temporelles entre ces événements. DCM peut être vu comme un algorithme de découverte de chroniques supervisé.

Enfin, d'autres méthodes de découverte de chroniques existent. Par exemple, [Quinqueton 1997] offre un apprentissage de chroniques par renforcement. Cet algorithme fonctionne en ligne et extrait des chroniques représentatives des situations menant à une action spécifique de l'utilisateur. Une méthode pour l'apprentissage de chroniques par programmation logique inductive est développée dans [Mayer 1998].

1.5.4 Bilan des algorithmes de découverte de chroniques

Dans cette section, une discussion autour des différentes caractéristiques des algorithmes de découverte de chroniques existant dans la littérature est offerte. Plusieurs points de comparaisons sont abordés et les différentes propriétés les plus importantes sont répertoriées dans la table 1.3.

Table 1.3 – Table récapitulative des différentes propriétés des algorithmes de découverte de chroniques de la littérature. Les propriétés sont les suivantes : *Complet*, renvoie toutes les chroniques respectant le critère de fréquence ; *Itératif*, l'utilisateur peut agir sur l'algorithme à chaque itération ; *Plusieurs séquences*, utilise plusieurs séquences temporelles en entrée ; *Départ à partir de chroniques connues*, l'utilisateur peut fournir une base de chroniques à enrichir ; *Approche générer & compter*, utilise une approche de ce type.

	Plusieurs séquences	Départ à partir de chroniques connues	Itératif	Complet	Approche générer & compter
FACE [Dousson 1999]		x			x
HCDA [Cram 2012]		x	x	x	x
HCDAM [Vasquez Capacho 2017]	x	x	x		x
ASTPminer [Álvarez 2013]	x	x	x		x
SCP-Miner, MATC-Miner & CCP-Miner [Huang 2012]	x				x
DCM [Dauxais 2017]	x				x

Malgré l'utilisation générale du format des séquences temporelles comme données d'entrée, les divers algorithmes de découverte de chroniques se découpent en deux familles : les algorithmes qui considèrent une seule longue séquence temporelle où les phénomènes d'intérêt se produisent de manière répétée dans ces données, par exemple, HCDA [Cram 2012] ou FACE [Dousson 1999] ; ou les algorithmes qui considèrent plusieurs séquences temporelles contenant ou non une occurrence du phénomène d'intérêt [Álvarez 2013, Huang 2012, Dauxais 2017]. Un algorithme, HCDAM [Vasquez Capacho 2017], propose un mélange entre ces deux méthodes.

Plusieurs algorithmes possèdent la fonctionnalité de fournir une base de chroniques à enrichir [Dousson 1999, Cram 2012, Vasquez Capacho 2017, Álvarez 2013]. Cela permet à l'utilisateur de fournir une direction initiale au processus de décou-

verte de chroniques. Néanmoins, cette fonctionnalité réduit également l'éventualité de découvrir un phénomène intéressant inconnu a priori. De plus, un bon nombre de ces algorithmes [Cram 2012, Vasquez Capacho 2017, Álvarez 2013] abordent une méthode itérative pour leurs algorithmes. Cela permet à l'utilisateur d'être actif sur la découverte de chroniques tout au long de ce processus ; ce qui améliore grandement les performances et la qualité des résultats fournis, mais en contrepartie, une plus grande connaissance du système étudié de l'utilisateur est nécessaire.

Au vu de l'étude bibliographique offerte dans ce chapitre, seuls les auteurs de l'algorithme HCDA [Cram 2012] démontrent la propriété de complétude de leur algorithme. Enfin, chacun de ces algorithmes utilise une approche ou une variation de l'approche générer & compter.

1.6 Introduction à CDIRe : une approche innovante à la découverte de chroniques

Dans la section précédente, une étude des divers algorithmes de découverte de chroniques existant dans la littérature montre que, malgré quelques variations, l'approche générer & compter est généralement adoptée. Malheureusement, la complexité algorithmique de ce type d'approche est problématique. En effet, l'étape *compter* est coûteuse en temps et prend bien souvent la majorité du temps d'exécution total de ces algorithmes. De plus, ce type d'approche impose un critère de choix des chroniques intéressantes. Bien souvent, le critère choisi est la fréquence. Ce choix implique une certaine connaissance du système sous-jacent aux données d'entrée par l'utilisateur et peut éclipser un certain nombre de phénomènes pouvant être pertinents.

Afin d'éviter les désavantages liés à l'approche générer & compter, une nouvelle approche est abordée pour répondre à la problématique de découverte de chronique. Cette approche repose sur des notions d'*identification* et de *reconstitution*. Dans ce document, un algorithme intitulé CDIRe (*Chronicle Discovery by Identification and Reconstitution*) et reposant sur ces notions est défini dans le chapitre 2 et analysé sur un jeu de données provenant d'une application réelle dans le chapitre 3. Plusieurs itérations précédentes de CDIRe sont déjà publiées dans la littérature [Sahuguède 2018b, Sahuguède 2018c]. Cet algorithme repose sur deux étapes fonctionnant de pair et où chaque étape dépend fondamentalement de ces nouvelles notions d'identification et de reconstitution.

L'étape d'*identification des chroniques élémentaires* permet dans un premier temps d'extraire des chroniques simples à l'aide d'un algorithme de partitionnement des données. Cette utilisation de ce type d'algorithmes efficaces et originaires d'un domaine de recherche particulièrement actif ces dernières années permet à cette étape de générer des chroniques élémentaires pertinentes dans un temps d'exécution performant. Ces chroniques élémentaires sont cruciales car elles servent par la suite comme composants simples pour la génération de chroniques plus complexes.

L'étape de *reconstitution de chroniques complexes* donne lieu à des chroniques

pertinentes à partir des chroniques élémentaires générées par l'étape précédente. Elle tente de reconstituer les phénomènes sous-jacents aux données d'entrée sous forme de chronique par l'assemblage des différentes chroniques élémentaires identifiées à disposition. Cet assemblage de chroniques élémentaires se fait grâce à une fusion des nœuds suivant un critère de similarité. Ce critère ainsi qu'un choix dans l'ordre des différentes fusions des nœuds permet une reconstitution de chroniques pouvant être d'une grande complexité avec un bon temps d'exécution.

Cette nouvelle approche par identification & reconstitution offre une alternative aux variations de l'algorithme Apriori proposant une solution aux problèmes inhérents à ce type d'algorithmes de découverte de chroniques.

1.7 Conclusion

Le thème principal de ce chapitre concerne le contexte scientifique de nos contributions. Les chroniques y sont présentées, aussi bien l'usage qui en est fait que les définitions formelles nécessaires à la compréhension de celles-ci. Un tour d'horizon des modèles temporels, des algorithmes d'extraction de connaissance et en particulier des algorithmes de découverte de chroniques permet de positionner notre travail dans la littérature. Enfin, une introduction à l'approche adoptée pour résoudre la problématique de découverte de chroniques ainsi que l'algorithme proposé intitulé CDIRe ouvre la voie au chapitre 2 où celui-ci y est décrit.

Chapitre 2

CDIRe : un algorithme de découverte de chroniques par identification et reconstitution

Sommaire

Le thème de ce chapitre est l'approche innovante du processus de découverte de chronique abordée dans ce mémoire : une approche par *identification* de chroniques

élémentaires et *reconstitution* de chroniques plus complexes. De nombreux exemples et détails des clefs algorithmiques de CDIRe permettent de décrire ces nouvelles notions. Cette approche nécessite une problématique de découverte de chroniques adaptée qui est posée dans ce chapitre.

Ces deux notions d'*identification* et de *reconstitution* donnent naturellement naissance aux deux grandes étapes interconnectées de CDIRe. La première étape repose sur des notions de partitionnement de données, une technique dans un domaine de recherche largement exploré au cours de ces dernières années. La deuxième étape repose sur la notion d'*opération*, la fusion de deux nœuds d'une chronique. Les opérations permettent d'agglomérer les chroniques élémentaires obtenues dans l'étape précédente en une chronique plus complexe et plus descriptive d'un phénomène complexe sous-jacent aux données d'entrée. Une analyse de la complexité algorithmique confirme les interconnexions fortes entre ces deux étapes de CDIRe.

L'organisation de ce chapitre est la suivante. La section 2.1 fournit les préliminaires nécessaires à la compréhension de CDIRe, pose la problématique et décrit d'une manière générale la solution proposée. Dans la section 2.2, la première étape de CDIRe, l'étape d'identification de chroniques élémentaires est décrite. L'étape de reconstitution est détaillée dans les trois sections suivantes. La section 2.3 présente la méthodologie de cette étape ainsi que l'algorithme de celle-ci. La section 2.4 définit les opérations et la section 2.5 se penche sur l'ordonnancement de celles-ci. Enfin, la section 2.6 propose une étude de la complexité algorithmique de CDIRe.

2.1 Préliminaires : introduction à la problématique

Au vu de l'approche innovante au problème de la découverte de chroniques exploitée pour le fonctionnement de CDIRe, quelques informations supplémentaires doivent enrichir le contexte du travail de ce document. Dans cette section, des définitions complémentaires sont fournies. Puis, la problématique traitée par CDIRe est formulée. Enfin, une vue d'ensemble du fonctionnement algorithmique de CDIRe est présentée.

2.1.1 Définitions complémentaires

Dans cette section, un ensemble de définitions complémentaires au cadre formel des chroniques donné dans le chapitre précédent (cf. section 1.2.2, page 10, et section 1.5, page 25) et nécessaires à la compréhension de CDIRe sont fournies. Elles introduisent en particulier la notion d'opération, qui est une fusion de deux nœuds d'une chronique.

Définition 2.1. L'**ensemble des instants d'occurrence d'un nœud** $\nu_{\mathcal{C}}(e_i)$ d'une chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ dans une séquence temporelle $\mathcal{S}$ est le multi-ensemble des instants d'occurrences du nœud $\nu_{\mathcal{C}}(e_i)$ dans $\mathcal{S}$ et est noté $\mathcal{O}_{\mathcal{C}}(e_i, \mathcal{S})$. Cet ensemble est défini par la formule suivante :

$$\mathcal{O}_{\mathcal{C}}(e_i, \mathcal{S}) = \{t_i \mid (e_i, t_i) \in o_{\mathcal{C}}, o_{\mathcal{C}} \in \mathcal{O}_{\mathcal{C}}(\mathcal{S})\} \tag{2.1}$$

Exemple 2.1. Soient $\mathcal{C}_1$, $\mathcal{C}_2$ et $\mathcal{C}_3$ les chroniques suivantes :

— $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ avec $\mathcal{E}_1 = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T}_1 = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(2,3)} = e_2[3,4]e_3, \tau_{(1,4)} = e_1[8,10]e_4\}$,

— $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ avec $\mathcal{E}_2 = \{e_1 = c, e_2 = d\}$ et $\mathcal{T}_2 = \{\tau_{(1,2)} = e_1[-2,1]e_2\}$,

— $\mathcal{C}_3 = (\mathcal{E}_3, \mathcal{T}_3)$ avec $\mathcal{E}_3 = \{e_1 = c, e_2 = d\}$ et $\mathcal{T}_3 = \{\tau_{(1,2)} = e_1[-3,2]e_2\}$.

Soit $\mathcal{S}$ la séquence temporelle suivante : $\mathcal{S} = \{(a,2),(b,3),(d,5),(c,7),(a,8),(b,10),(c,11),(c,13),(d,13),(c,16)\}$. Les ensembles des occurrences des chroniques $\mathcal{C}_1$, $\mathcal{C}_2$ et $\mathcal{C}_3$ dans la séquence temporelle $\mathcal{S}$ sont :

$$\mathcal{O}_{\mathcal{C}_1}(\mathcal{S}) = \{o^1_{\mathcal{C}_1}, o^2_{\mathcal{C}_1}\} = \begin{cases} o^1_{\mathcal{C}_1} = \{(a,2),(b,3),(c,7),(c,11)\}, \\ o^2_{\mathcal{C}_1} = \{(a,8),(b,10),(c,13),(c,16)\}. \end{cases} \tag{2.2}$$

$$\mathcal{O}_{\mathcal{C}_2}(\mathcal{S}) = \{o^1_{\mathcal{C}_2}, o^2_{\mathcal{C}_2}\} = \begin{cases} o^1_{\mathcal{C}_2} = \{(c,7),(d,5)\}, \\ o^2_{\mathcal{C}_2} = \{(c,13),(d,13)\}. \end{cases} \tag{2.3}$$

$$\mathcal{O}_{\mathcal{C}_3}(\mathcal{S}) = \{o^1_{\mathcal{C}_3}, o^2_{\mathcal{C}_3}, o^3_{\mathcal{C}_3}, o^4_{\mathcal{C}_3}\} = \begin{cases} o^1_{\mathcal{C}_3} = \{(c,7),(d,5)\}, \\ o^2_{\mathcal{C}_3} = \{(c,11),(d,13)\}, \\ o^3_{\mathcal{C}_3} = \{(c,13),(d,13)\}, \\ o^4_{\mathcal{C}_3} = \{(c,16),(d,13)\}. \end{cases} \tag{2.4}$$

Les instants d'occurrence des nœuds de la chronique $\mathcal{C}_1$ dans la séquence temporelle $\mathcal{S}$ sont :

$$\begin{aligned} \mathcal{O}_{\mathcal{C}_1}(e_1,\mathcal{S}) &= \{2,8\}, \quad \mathcal{O}_{\mathcal{C}_1}(e_2,\mathcal{S}) = \{3,10\}, \\ \mathcal{O}_{\mathcal{C}_1}(e_3,\mathcal{S}) &= \{7,13\}, \quad \mathcal{O}_{\mathcal{C}_1}(e_4,\mathcal{S}) = \{11,16\}. \end{aligned} \tag{2.5}$$

Les instants d'occurrence des nœuds de la chronique $\mathcal{C}_2$ sont :

$$\mathcal{O}_{\mathcal{C}_2}(e_1,\mathcal{S}) = \{7,13\}, \quad \mathcal{O}_{\mathcal{C}_2}(e_2,\mathcal{S}) = \{5,13\}. \tag{2.6}$$

Ceux de la chronique $\mathcal{C}_3$ sont :

$$\mathcal{O}_{\mathcal{C}_3}(e_1,\mathcal{S}) = \{7,11,13,16\}, \quad \mathcal{O}_{\mathcal{C}_3}(e_2,\mathcal{S}) = \{5,13,13,13\}. \tag{2.7}$$

Dans la littérature, les indices de similarité permettent de quantifier le degré de similarité entre deux ensembles symboliques. Un tel indice est l'indice de Jaccard [Jaccard 1912] qui calcule le ratio entre la taille de l'intersection et la taille de l'union de deux ensembles A et B. Dans CDIRe, nous allons exploiter cet indice sur les ensembles d'instants d'occurrence de nœuds.

Définition 2.2. L'**indice de Jaccard** $J(\nu_{\mathcal{C}}(e_i), \nu_{\mathcal{C}}(e_j), \mathcal{S})$ entre les nœuds $\nu_{\mathcal{C}}(e_i)$ et $\nu_{\mathcal{C}}(e_j)$ dans une séquence temporelle $\mathcal{S}$ est donné par la formule suivante :

$$J(\nu_{\mathcal{C}}(e_i), \nu_{\mathcal{C}}(e_j), \mathcal{S}) = \frac{|\mathcal{O}_{\mathcal{C}}(e_i,\mathcal{S}) \cap \mathcal{O}_{\mathcal{C}}(e_j,\mathcal{S})|}{|\mathcal{O}_{\mathcal{C}}(e_i,\mathcal{S}) \cup \mathcal{O}_{\mathcal{C}}(e_j,\mathcal{S})|}. \tag{2.8}$$

Remarque 2.1. Notons que dans l'union de deux multi-ensembles $A \cup B$, la multiplicité d'un élément x de $(A \cup B)$ est le maximum des deux ordres de multiplicité de x dans A et B :

$$(A \cup B)(x) \Leftrightarrow \max(A(x), B(x)). \tag{2.9}$$

Pour l'intersection de deux multi-ensembles $A \cap B$, la multiplicité d'un élément x de $(A \cup B)$ est le minimum des deux ordres de multiplicité de x dans A et B :

$$(A \cap B)(x) \Leftrightarrow \min(A(x), B(x)). \tag{2.10}$$

Exemple 2.2. Soient $\mathcal{C}_1$, $\mathcal{C}_2$ et $\mathcal{C}_3$ les chroniques décrites dans l'exemple 2.1. Les ensembles des instants d'occurrence des événements de ces chroniques sont donnés par les équations (2.5), (2.6) et (2.7). L'indice de Jaccard entre les nœuds $\nu_{\mathcal{C}_1}(e_3)$ et $\nu_{\mathcal{C}_2}(e_1)$ est :

$$J(\nu_{\mathcal{C}_1}(e_3), \nu_{\mathcal{C}_2}(e_1), \mathcal{S}) = \frac{|\mathcal{O}_{\mathcal{C}_1}(e_3, \mathcal{S}) \cap \mathcal{O}_{\mathcal{C}_2}(e_1, \mathcal{S})|}{|\mathcal{O}_{\mathcal{C}_1}(e_3, \mathcal{S}) \cup \mathcal{O}_{\mathcal{C}_2}(e_1, \mathcal{S})|} = \frac{|\{7, 13\}|}{|\{7, 13\}|} = \frac{2}{2} = 1. \tag{2.11}$$

Ce résultat indique que tous les instants d'occurrence des nœuds $\nu_{\mathcal{C}_1}(e_3)$ et $\nu_{\mathcal{C}_2}(e_1)$ sont identiques. De la même manière, l'indice de Jaccard entre les nœuds $\nu_{\mathcal{C}_1}(e_3)$ et $\nu_{\mathcal{C}_2}(e_2)$ est :

$$J(\nu_{\mathcal{C}_1}(e_3), \nu_{\mathcal{C}_2}(e_2), \mathcal{S}) = \frac{|\mathcal{O}_{\mathcal{C}_1}(e_3, \mathcal{S}) \cap \mathcal{O}_{\mathcal{C}_2}(e_2, \mathcal{S})|}{|\mathcal{O}_{\mathcal{C}_1}(e_3, \mathcal{S}) \cup \mathcal{O}_{\mathcal{C}_2}(e_2, \mathcal{S})|} = \frac{|\{13\}|}{|\{5, 7, 13\}|} = \frac{1}{3} = 0.33. \tag{2.12}$$

Ici, seul un instant d'occurrence est en commun entre $\nu_{\mathcal{C}_1}(e_3)$ et $\nu_{\mathcal{C}_2}(e_2)$. Enfin, l'indice de Jaccard entre les nœuds $\nu_{\mathcal{C}_2}(e_2)$ et $\nu_{\mathcal{C}_3}(e_2)$ est :

$$J(\nu_{\mathcal{C}_2}(e_2), \nu_{\mathcal{C}_3}(e_2), \mathcal{S}) = \frac{|\mathcal{O}_{\mathcal{C}_2}(e_2, \mathcal{S}) \cap \mathcal{O}_{\mathcal{C}_3}(e_2, \mathcal{S})|}{|\mathcal{O}_{\mathcal{C}_2}(e_2, \mathcal{S}) \cup \mathcal{O}_{\mathcal{C}_3}(e_2, \mathcal{S})|} = \frac{|\{5, 13\}|}{|\{5, 13, 13, 13\}|} = \frac{2}{4} = 0.5. \tag{2.13}$$

Définition 2.3. Le **seuil sur l'indice de Jaccard** noté $seuil_{sim}$ est le seuil au-dessus duquel les indices de Jaccard sont utiles. Il est défini par :

$$seuil_{sim} \in [0, 1]. \tag{2.14}$$

$seuil_{sim}$ est un des paramètres de CDIRe.

Définition 2.4. Une chronique $\mathcal{C}$ est dite **compatible** à une fréquence f dans une séquence temporelle $\mathcal{S}$ lorsque sa fréquence $f(\mathcal{C}, \mathcal{S})$ est supérieure ou égale au produit entre la fréquence f et le seuil $seuil_{sim}$:

$$f(\mathcal{C}, \mathcal{S}) \geq f \times seuil_{sim}. \tag{2.15}$$

Définition 2.5. Une **sous-chronique** $\mathcal{C}'$ d'une chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ est une chro-

nique $(\mathcal{E}', \mathcal{T}')$ telle que :

$$\begin{cases} \mathcal{E}' \neq \emptyset, \\ \mathcal{E}' \subseteq \mathcal{E}, \\ \forall e_i, e_j \in \mathcal{E}', \tau_{(i,j)} \in \mathcal{T} \Leftrightarrow \tau_{(i,j)} \in \mathcal{T}'. \end{cases} \tag{2.16}$$

Définition 2.6. Une **sous-chronique indépendante** $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ d'une chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ est une sous-chronique de $\mathcal{C}$ telle que :

$$\forall \tau_{(i,j)} \in \mathcal{T}, e_i \in \mathcal{E} \setminus \mathcal{E}' \Rightarrow e_j \in \mathcal{E} \setminus \mathcal{E}'. \tag{2.17}$$

Ainsi, toutes les contraintes sur les événements de $\mathcal{E}'$ qui appartiennent à $\mathcal{T}$ appartiennent également à $\mathcal{T}'$.

Une sous-chronique indépendante est notée $s\mathcal{C}$ dans le reste de ce document. Une sous-chronique indépendante élémentaire est notée $s\mathcal{C}^\alpha$.

Exemple 2.3. Soit $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ une chronique de taille 8 où $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c, e_5 = d, e_6 = d, e_7 = d, e_8 = f\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(1,4)} = e_1[8,10]e_4, \tau_{(2,3)} = e_2[3,4]e_3, \tau_{(3,4)} = e_3[4,9]e_4, \tau_{(3,5)} = e_3[3,6]e_5, \tau_{(6,8)} = e_6[0,1]e_8, \tau_{(7,8)} = e_7[3,4]e_8\}$. Cette chronique est graphiquement représentée sur la figure 2.1 et contient deux sous-chroniques indépendantes qui respectent toutes les conditions définies par l'équation (2.17) :

- la chronique $s\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ de taille 5 avec $\mathcal{E}_1 = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c, e_5 = d\}$ et $\mathcal{T}_1 = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(1,4)} = e_1[8,10]e_4, \tau_{(2,3)} = e_2[3,4]e_3, \tau_{(3,4)} = e_3[4,9]e_4, \tau_{(3,5)} = e_3[3,6]e_5\}$,
- la chronique $s\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ de taille 3 avec $\mathcal{E}_2 = \{e_6 = d, e_7 = d, e_8 = f\}$ et $\mathcal{T}_2 = \{\tau_{(6,8)} = e_6[0,1]e_8, \tau_{(7,8)} = e_7[3,4]e_8\}$.

Définition 2.7. Une **opération** $\omega_{(i,j)}$ sur une chronique $\mathcal{C} = (\mathcal{E} = \{e_1, \ldots, e_n\}, \mathcal{T})$, où $i \neq j$, $i = \nu_{\mathcal{C}}(e_i)$, $j = \nu_{\mathcal{C}}(e_j)$, $\nexists \tau_{(i,j)} \in \mathcal{T}$ et $e_i = e_j$, est la fusion des deux nœuds i et j. C'est une fonction injective $\omega : \mathcal{C} \mapsto \mathcal{C}'$ telle que $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ où $|\mathcal{E}'| = |\mathcal{E}| - 1$ et $|\mathcal{T}'| = |\mathcal{T}|$ avec :

$$\mathcal{E}' = \{e_k \in \mathcal{E} | \nu_{\mathcal{C}}(e_k) \in \{1, \ldots, n\} \setminus \{\nu_{\mathcal{C}}(e_j)\}\}, \tag{2.18}$$

$$\begin{aligned} \mathcal{T}' = \{\tau_{(k,l)} \in \mathcal{T} \mid k \neq j \wedge l \neq j\} \cup \\ \{\tau_{(i,l)} = (e_i, e_l, t^-, t^+) \mid \exists \tau_{(j,l)} = (e_j, e_l, t^-, t^+) \in \mathcal{T}\} \cup \\ \{\tau_{(k,i)} = (e_k, e_i, t^-, t^+) \mid \exists \tau_{(k,j)} = (e_k, e_j, t^-, t^+) \in \mathcal{T}\}. \end{aligned} \tag{2.19}$$

Notons qu'une opération change la taille d'une chronique : la chronique $\mathcal{C}$ de taille n devient une chronique $\mathcal{C}'$ de taille $n - 1$.

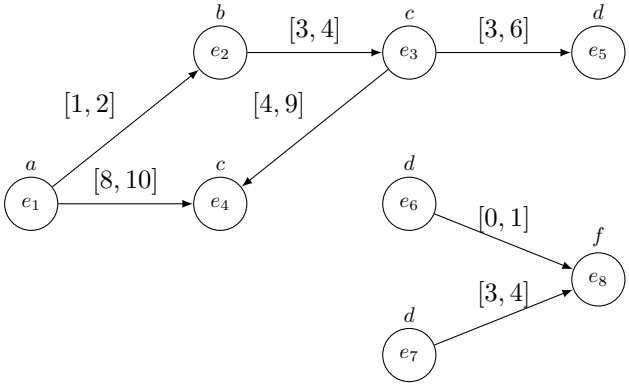
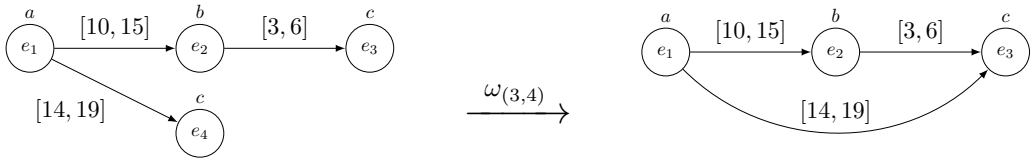


FIGURE 2.1 – Exemple d'une chronique $\mathcal{C} = (\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2\}, \mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2\})$ qui contient deux sous-chroniques indépendantes $s\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ et $s\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ avec $\mathcal{E}_1 = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c, e_5 = d\}$, $\mathcal{E}_2 = \{e_6 = d, e_7 = d, e_8 = f\}$, $\mathcal{T}_1 = \{\tau_{12} = e_1[1, 2]e_2, \tau_{(1,4)} = e_1[8, 10]e_4, \tau_{(2,3)} = e_2[3, 4]e_3, \tau_{(3,4)} = e_3[4, 9]e_4, \tau_{(3,5)} = e_3[3, 6]e_5\}$ et $\mathcal{T}_2 = \{\tau_{(6,8)} = e_6[0, 1]e_8, \tau_{(7,8)} = e_7[3, 4]e_8\}$.

Définition 2.8. Une **opération cohérente** sur une chronique $\mathcal{C}$ bien formée [1] et cohérente est une opération telle qu'après son application, la chronique $\mathcal{C}'$ est bien formée et cohérente. Lorsqu'une chronique n'est pas bien formée ou cohérente après l'application d'une opération, cette opération est dite **incohérente**.

Exemple 2.4. Soit $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ une chronique de taille 4 où $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(2,3)} = e_2[3, 6]e_3, \tau_{(1,4)} = e_1[14, 19]e_4\}$. $\mathcal{C}$ est une chronique cohérente pour une séquence temporelle $\mathcal{S}$ donnée. Après l'application de l'opération $\omega_{(3,4)}$, $\mathcal{C}'$ est une chronique de taille 3 où $\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(2,3)} = e_2[3, 6]e_3, \tau_{(1,3)} = e_1[14, 19]e_3\}$. Si $\mathcal{C}'$ est toujours cohérente, alors cette opération est cohérente. $\mathcal{C}$ et $\mathcal{C}'$ sont représentées respectivement sur les figures 2.2a et 2.2b.



(a) Avant l'opération $\omega_{(3,4)}$.

(b) Après l'opération $\omega_{(3,4)}$.

FIGURE 2.2 – Exemple d'application d'une opération. L'opération $\omega_{(3,4)}$ sur la chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ de taille 4 avec $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c, e_4 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(2,3)} = e_2[3, 6]e_3, \tau_{(1,4)} = e_1[14, 19]e_4\}$ donne la chronique $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$ de taille 3 avec $\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T}' = \{\tau_{(1,2)} = e_1[10, 15]e_2, \tau_{(2,3)} = e_2[3, 6]e_3, \tau_{(1,3)} = e_1[14, 19]e_3\}$.

1. Une chronique bien formée est une chronique qui respecte toutes les conditions d'une chronique (cf. définition 1.6, page 11).

2.1.2 Formulation de la problématique

Dans cette section, la problématique traitée dans ce document est formulée. Cette problématique diffère des autres problématiques traitées par d'autres algorithmes de découverte de chroniques tels que FACE [Dousson 1999] ou encore HCDA [Cram 2012] par plusieurs points. La première grande différence est qu'il n'y a pas de critère de fréquence pour évaluer les chroniques générées. Cela permet de générer des chroniques qui peuvent être aussi bien peu ou très fréquentes dans les données d'entrée. Une deuxième différence est de générer seulement un ensemble de chroniques descriptives des phénomènes présents dans les données d'entrée et non toutes les chroniques ayant des occurrences dans les données d'entrée.

La problématique traitée par l'algorithme appelé CDIRe fournit dans ce document est la suivante :

Problématique. *Soit $\mathcal{S}$ une séquence temporelle composée de plusieurs occurrences d'un ou de plusieurs phénomènes temporels, le problème de la découverte de chroniques par identification et reconstitution est de générer une ou des chroniques descriptives du ou des phénomènes sous-jacents à $\mathcal{S}$.*

Vu la problématique formulée dans la section précédente, il est clair qu'une approche de type "générer & compter" n'est pas pertinente. En effet, la fréquence objective des chroniques apprises n'est pas connue et ne peut donc pas être utilisée comme critère d'arrêt. Seule une approche innovante peut être à même de répondre au mieux à cette problématique. De plus, un critère de choix doit être offert pour déterminer quelles sont les chroniques descriptives des phénomènes sous-jacents aux données d'entrée.

2.1.3 Solution proposée : vue d'ensemble de CDIRe

L'organisation de CDIRe se découpe en une identification de chroniques élémentaires et une reconstitution de chroniques plus complexes. Cette approche permet de trouver les motifs temporels les plus intéressants tout en limitant le nombre de chroniques générées. Dans un premier temps, CDIRe identifie les chroniques élémentaires présentes dans la séquence temporelle d'intérêt $\mathcal{S}$ grâce à un algorithme de partitionnement des données. Le processus d'identification de chroniques élémentaires est décrit dans la section 2.2. Dans un second temps, les chroniques élémentaires identifiées vont être agglomérées suivant un ordre spécifique afin de reconstituer une chronique modélisant les phénomènes complexes sous-jacents aux données d'entrée. Le paramètre $seuil_{sim}$ permettant de distinguer les opérations intéressantes des opérations qui ne le sont pas. La méthodologie adoptée pour la reconstitution de chroniques est fournie dans la section 2.3. Enfin, le processus complet de CDIRe est résumé par l'algorithme 1.

Algorithme 1 CDIRe : *Chronicle Discovery by Identification and Reconstitution*

Entrées : $\mathcal{S}$
Sorties : $\{\mathcal{C}\}_{apprise}$
1: $\{\mathcal{C}\}_{élémentaire} \leftarrow identification\_de\_chroniques\_élémentaires(\mathcal{S})$
2: $\{\mathcal{C}\}_{apprise} \leftarrow reconstitution\_de\_chroniques(\{\mathcal{C}\}_{élémentaire}, seuil_{sim}, \mathcal{S})$
3: **retourner** $\{\mathcal{C}\}_{apprise}$

2.2 Identification de chroniques élémentaires

L'objectif de cette section est de découvrir des motifs temporels simples qui se produisent dans les données d'entrée. La plupart des algorithmes de découverte de chroniques parviennent à ce résultat par le biais d'un critère de fréquence : si un motif temporel se répète plus d'un nombre de fois déterminé à l'avance dans les données d'entrée, alors c'est un motif pertinent. L'approche par identification de chroniques élémentaires présenté dans cette section se base sur des méthodes de regroupement de données pour obtenir des chroniques élémentaires intéressantes. Cette approche permet d'éviter de définir un critère de fréquence qui peut être ardu à choisir pour l'utilisateur.

2.2.1 Algorithme d'identification de chroniques élémentaires

L'algorithme d'identification de chroniques élémentaires est présenté sur l'algorithme 2. Cet algorithme permet de générer un ensemble de chroniques élémentaires à partir d'une séquence temporelle $\mathcal{S}$ donnée en entrée. Tout d'abord, soit $\mathbb{E}_{\mathcal{S}}$ l'ensemble des événements présents dans $\mathcal{S}$, l'instruction de la ligne 3 calcule l'ensemble des distances temporelles $\mathcal{D}_{ab}$ d'un couple d'événement $(a, b) \in \mathbb{E}_{\mathcal{S}}^2$ et ce, pour chaque couple possible de $\mathbb{E}_{\mathcal{S}}$. Puis, la ligne 4 permet de regrouper les distances temporelles suivant un critère de densité des données. Comme détaillé dans la section 2.2.2, l'algorithme utilisé pour cette étape est DBSCAN [Ester 1996]. Enfin, les lignes 5 à 7 génèrent une chronique élémentaire à partir de chaque groupement de distances temporelles trouvées dans l'étape précédente. La section 2.2.3 offre une explication plus approfondie de cette procédure. La base de chroniques élémentaires ainsi générée est ensuite utilisée lors du processus de reconstitution de chroniques détaillé dans la section 2.3.

Le calcul des distances temporelles d'un couple d'événement (a, b) se fait aisément grâce aux définitions 1.17 (page 26) et 1.18 (page 26). Elles permettent de calculer consécutivement les occurrences $\mathcal{O}_{ab}(\mathcal{S})$ et les distances temporelles $\mathcal{O}_{ab}(\mathcal{S})$ avec les formules suivantes :

$$\mathcal{O}_{ab}(\mathcal{S}) = \left\{ \langle (e_i, t_i), (e_j, t_j) \rangle \mid (e_i, t_i), (e_j, t_j) \in \mathcal{S}, \begin{cases} e_i = a,\ e_j = b, & \text{si } a <_{\mathbb{E}} b \\ i < j,\ e_i = a,\ e_j = b, & \text{si } a =_{\mathbb{E}} b \end{cases} \right\}, \tag{2.20}$$

$$\mathcal{D}_{ab}(\mathcal{S}) = \{ d(t_i, t_j) \mid \langle (e_i, t_i), (e_j, t_j) \rangle \in \mathcal{O}_{ab}(\mathcal{S}) \}. \tag{2.21}$$

Algorithme 2 Identification de chroniques élémentaires

Entrées : $\mathcal{S}$
Sorties : $\{\mathcal{C}\}_{élémentaire}$

```
{C}_élémentaire ← ∅
pour chaque couple(a, b) ∈ E²_S faire
    D_ab ← calculer_distances(a, b, S)
    {D}_ab ← grouper_suivant_la_densité(D_ab)
    pour chaque D ∈ {D}_ab faire
        {C}_élémentaire ← {C}_élémentaire ∪ générer_chronique(D)
    fin pour
fin pour
retourner {C}_élémentaire
```

2.2.2 Regroupement de distances temporelles suivant un critère de densité

Une approche différente des algorithmes classiques de découvertes de chroniques [Dousson 1999, Cram 2012] est adoptée pour le traitement des distances temporelles dans CDIRe. L'approche proposée est de regrouper les distances temporelles proches les unes des autres. En d'autres termes, la méthode exploitée construit des ensembles de distances temporelles similaires entre elles tout en différenciant au mieux les divers ensembles. Les distances temporelles peuvent être regroupées suivant un critère de densité. Plus deux distances sont proches, plus il y a de chances qu'elles fassent partie du même ensemble. Cette approche de regroupement par un critère de densité permettrait de générer un ou plusieurs ensembles de distances temporelles qui contiendraient au mieux les informations intrinsèques à l'ensemble de toutes les distances temporelles. Un critère de fréquence n'est donc pas utile dans cette approche, ce qui donne lieu à une identification de chroniques élémentaires étant pourvues d'une fréquence faible aussi bien qu'importante dans une même séquence temporelle. De plus, ce regroupement permet d'éviter de générer une quantité importante de chroniques qui ne contiendraient pas ou peu d'informations pertinentes pour l'utilisateur.

La problématique de regroupement de données soulevée précédemment peut être résolue à l'aide de méthodes de classification non supervisée. Ce type de méthode orientée sur les données est un domaine de recherche particulièrement actif et bien étudié. De nombreux algorithmes de classification non supervisée, et plus particulièrement de partitionnement de données (aussi appelé *clustering*), répondent au problème de regroupement des données. Ces algorithmes visent à regrouper les données en ensembles suivant un critère de similarité. Un partitionnement est jugé bon lorsque les différents ensembles sont homogènes, dans le sens où les différents individus de ces ensembles sont similaires, et que chaque ensemble est bien différencié des autres. Parmi la multitude d'algorithmes proposés, plusieurs méthodes émergent :

— les méthodes basées sur la distance,

— les méthodes basées sur la densité,

— les méthodes basées sur la hiérarchie des données.

Méthodes basées sur la distance Dans les méthodes de regroupement des données basées sur la distance, une mesure de similarité entre les individus et le centre d'un ensemble est calculée pour déterminer s'ils appartiennent ou non à cet ensemble. La mesure de similarité la plus communément utilisée est la distance euclidienne. Une fois les ensembles créés, une mesure objective permet de quantifier la qualité de ces ensembles. La problématique du regroupement de données peut être alors formulée comme un problème d'optimisation : c'est-à-dire regrouper les données afin de minimiser ou maximiser une mesure objective. Parmi les principaux algorithmes basés sur la distance dans la littérature, on peut citer l'algorithme des k-moyennes [Dunn 1973, Lloyd 1982] ainsi que ses variantes prédominantes : les k-médoïdes [Kaufman 1987] et les k-médianes [Bradley 1997]. Un inconvénient significatif de ces algorithmes est qu'ils ne sont pas déterministes, le résultat du partitionnement dépend fortement de l'initialisation des centres des différents ensembles. De plus, dans le contexte de CDIRe, le choix du nombre de classes (le paramètre k) est un désavantage. En effet, ce paramètre est directement proportionnel au nombre de chroniques élémentaires identifiées. Cette information n'est pas connue au préalable et fixer ce paramètre peut significativement altérer les résultats obtenus.

Méthodes basées sur la densité Une autre approche au problème de regroupement des données est basée sur la densité. La densité des données est déterminée par le calcul d'une mesure de similarité entre chaque individu. Ces différentes mesures sont regroupées dans une structure de données. Une matrice de similarité est une solution naïve du type de structure utilisée. Bien souvent, un arbre k-d [Bentley 1975] est une structure de données bien plus adaptée car la recherche dans ce type de structure est plus rapide que dans une matrice de similarité. Cette structure de données est explorée pour extraire les différents ensembles présents dans les données. DBSCAN [Ester 1996] ou encore OPTICS [Ankerst 1999] sont des algorithmes de regroupement des données qui sont basés sur la densité. Ces algorithmes possèdent deux paramètres : la distance ε et le nombre de points minimum $MinPts$ dans le rayon ε. Avec ces paramètres, l'utilisateur donne une estimation de la densité des différents ensembles présents dans les données. Malheureusement, le choix de ces paramètres n'est pas chose aisée car il dépend intrinsèquement des données d'entrée et une mauvaise estimation peut donner des résultats aberrants.

Méthodes basées sur la hiérarchie des données Les méthodes de regroupement basées sur la hiérarchie des données reposent sur un algorithme itératif. Chaque étape de cet algorithme regroupe deux partitions du résultat de l'étape précédente. Ce type d'algorithme commence avec chaque individu seul dans une partition et termine avec une partition unique. Ainsi, les données sont organisées

selon une hiérarchie qui suit l'indice d'agrégation. Les partitions finales sont alors les partitions données au seuil de l'indice d'agrégation voulu. L'algorithme CAH (Classification Ascendante Hiérarchique) [Ward 1963] est un algorithme communément utilisé de regroupement des données basé sur la hiérarchie des données. D'autres méthodes plus récentes (comme HDBSCAN [Campello 2013] ou [Obry 2018]) combinent des aspects de méthodes basées sur la densité et sur la hiérarchie des données.

Parmi les différents algorithmes de partitionnement des données cités précédemment, certains semblent être plus adaptés dans le contexte de regroupement des distances temporelles, notamment DBSCAN. En effet, celui-ci possède plusieurs avantages désirables pour l'identification de chroniques élémentaires. Tout d'abord, DBSCAN est rapide. Il possède une complexité algorithmique de $O(n \log(n))$ où n est le nombre d'individus dans l'ensemble traité. Ensuite, les paramètres d'entrée de DBSCAN (ε et $MinPts$) permettent d'estimer la densité des ensembles recherchés. Ce qui signifie que le nombre d'ensembles obtenus ainsi que leur taille ne sont pas connus a priori. Ces deux particularités sont clairement des points forts pour l'identification de chroniques élémentaires. En effet, les ensembles de distances temporelles ainsi extraits pourront être directement utilisés pour générer des chroniques élémentaires. Il est intéressant d'obtenir ce type d'information directement à partir des données plutôt que de les restreindre à une valeur fixée par l'utilisateur qui n'est pas nécessairement un expert des données traitées.

DBSCAN est donc un candidat intéressant pour l'identification de chroniques élémentaires. Les deux paramètres de cet algorithme, ε et $MinPts$, vont fortement influencer les résultats de CDIRe. En effet, ces paramètres permettent d'estimer la densité des ensembles de distances temporelles. Une densité estimée trop faible signifie qu'aucun regroupement ne sera fait, aucune chronique élémentaire ne sera identifiée et aucune chronique ne sera générée. En revanche, une densité estimée trop élevée signifie que des regroupements plus importants seront faits, par conséquent moins de chroniques élémentaires seront identifiées et la qualité des chroniques obtenues sera amoindrie.

Ainsi, pour les différentes raisons citées précédemment, DBSCAN est l'algorithme de choix exploité pour l'identification de chroniques élémentaires [2]. Le processus de génération de chroniques élémentaires à partir d'un ensemble de distances temporelles est développé dans la section suivante.

Exemple 2.5. Soit $\mathcal{D}_{ab}$ un ensemble de distances temporelles générées à partir d'une séquence temporelle avec :

$$\mathcal{D}_{ab} = \{10, 303, -2, 156, 298, 302, 152, 161, 305, -3, 2, 155, 4, 306, 152, 163,$$
$$155, 310, 303, -3, 0, 5, 8, 305, 153, 155, 152, -1, 150, 169, -2, 303,$$

2. Notons que d'autres algorithmes de partitionnement des données peuvent être utilisés à la place de DBSCAN. Néanmoins, les différences dans les résultats qui peuvent apparaître lors de l'utilisation d'autres algorithmes ne seront pas étudiées dans ce document et feront l'objet de futurs travaux.

$$-3, -4, 164, 158, 5, 8, -3, -2, 150, -2, 302, 152, 155, -5, 2, 297, 4,$$
$$160, 4, -6, 149, 144, 5, -2, 149, 158, 158, 5, 2, 6, 158, 155, 157\}$$

Avec les paramètres $\varepsilon = 10$ et $MinPts = 3$ de DBSCAN, trois ensembles de distances temporelles sont obtenus :

$$\{\mathcal{D}_1\}_{ab} = \{156, 152, 161, 155, 152, 163, 155, 153, 155, 152, 150, 169, 164, 158,$$
$$150, 152, 155, 160, 149, 144, 149, 158, 158, 158, 155, 157\},$$
$$\{\mathcal{D}_2\}_{ab} = \{10, -2, -3, 2, 4, -3, 0, 5, 8, -1, -2, -3, -4, 5, 8, -3, -2, -2, -5, 2,$$
$$4, 4, -6, 5, -2, 5, 2, 6\},$$
$$\{\mathcal{D}_3\}_{ab} = \{303, 298, 302, 305, 306, 310, 303, 305, 303, 302, 297\}.$$

2.2.3 Génération de chroniques élémentaires à partir d'un ensemble de distances temporelles

Dans beaucoup d'algorithmes de découverte de chroniques, la génération de chroniques élémentaires à partir d'un ensemble de distances temporelles est une étape importante. En effet, les chroniques obtenues par ce premier processus sont bien souvent les premiers résultats fournis par de tels algorithmes. Cette génération de chroniques est rendue possible par la propriété suivante.

Propriété 2.9. Soit $\mathcal{D}_{ab}$ un ensemble de distances temporelles pour un couple d'événement (a, b). Une chronique élémentaire $\mathcal{C}^\alpha = (\mathcal{E}^\alpha, \mathcal{T}^\alpha)$ peut être obtenue à partir de $\mathcal{D}_{ab}$. Les événements $\mathcal{E}^\alpha = \{e_1 = a, e_2 = b\}$ sont donnés par les éléments du couple (a, b) et la contrainte temporelle $\mathcal{T}^\alpha = \{\tau_{(1,2)} = e_1[t^-, t^+]e_2\}$ est donnée par l'ensemble de distances temporelles $\mathcal{D}_{ab}$.

La propriété 2.9 est exploitée dans plusieurs algorithmes de découverte de chroniques [Dousson 1999, Cram 2012, Vasquez Capacho 2017, Sahuguède 2018b]. Ces algorithmes prennent comme hypothèse que le temps écoulé entre les occurrences de deux événements datés pertinents respectent une loi de distribution uniforme. Les bornes t^- et t^+ de la contrainte temporelle sont alors définies par le minimum et le maximum de l'ensemble de distances temporelles $\mathcal{D}_{ab}$: $t^- = \min(\mathcal{D}_{ab})$ et $t^+ = \max(\mathcal{D}_{ab})$.

Cependant, dans beaucoup d'applications modernes, le temps écoulé entre les occurrences de deux événements datés pertinents ne respectent pas une loi de distribution uniforme mais une loi normale. En prenant en compte cette nouvelle hypothèse, la méthode de calcul des bornes de la contrainte temporelle proposée précédemment n'est pas pertinente. En effet, cette méthode est très sensible à la présence d'incertitude ou de bruit qui peuvent se manifester couramment dans les données provenant de systèmes dynamiques complexes. Afin de considérer l'hypothèse d'une loi normale, deux méthodes de calcul des bornes t^- et t^+ de la contrainte temporelle sont proposées.

Dans la première solution, les bornes sont définies par plus ou moins deux écarts types σ autour de la moyenne μ de l'ensemble des distances temporelles $\mathcal{D}_{ab}$:

$t^- = \mu - 2\sigma$ et $t^+ = \mu + 2\sigma$.

Alors que dans la seconde solution, les bornes sont définies par plus ou moins trois écarts types σ autour de la moyenne μ de l'ensemble $\mathcal{D}_{ab}$: $t^- = \mu - 3\sigma$ et $t^+ = \mu + 3\sigma$. Les intervalles de confiance, c'est-à-dire la probabilité qu'une variable aléatoire se trouve bien dans l'intervalle défini, sont bien connus pour la loi normale. Ainsi, la première solution possède un intervalle de confiance de 95% et la seconde solution possède un intervalle de confiance de 99.7%.

Définition 2.10. Soit la chronique élémentaire $\mathcal{C}^\alpha = (\mathcal{E}^\alpha, \mathcal{T}^\alpha)$ avec $\mathcal{E}^\alpha = \{e_1 = a, e_2 = b\}$ et $\mathcal{T}^\alpha = \{\tau_{(1,2)} = e_1[t^-, t^+]e_2\}$ générée à partir d'un ensemble $\mathcal{D}_{ab}$ tel que proposé dans la propriété 2.9. Trois **méthodes de calcul des bornes** t^- et t^+ de la contrainte temporelle $\tau_{(1,2)}$ sont définies :

— méthode `minmax` : les bornes sont définies par le minimum et le maximum de l'ensemble des distances temporelles $\mathcal{D}_{ab}$:

$$t^- = \min(\mathcal{D}_{ab}) \text{ et } t^+ = \max(\mathcal{D}_{ab}); \tag{2.22}$$

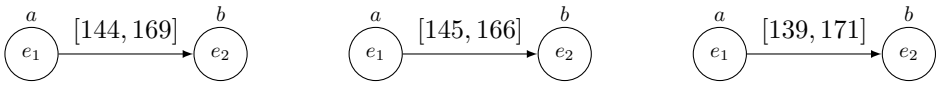
— méthode `2sigma` : les bornes sont définies par plus ou moins deux écarts types σ autour de la moyenne μ de l'ensemble des distances temporelles $\mathcal{D}_{ab}$:

$$t^- = \mu - 2\sigma \text{ et } t^+ = \mu + 2\sigma; \tag{2.23}$$

— méthode `3sigma` : les bornes sont définies par plus ou moins trois écarts types σ autour de la moyenne μ de l'ensemble des distances temporelles $\mathcal{D}_{ab}$:

$$t^- = \mu - 3\sigma \text{ et } t^+ = \mu + 3\sigma. \tag{2.24}$$

Exemple 2.6. Soit $\{\mathcal{D}_1\}_{ab} = \{156, 152, 161, 155, 152, 163, 155, 153, 155, 152, 150, 169, 164, 158, 150, 152, 155, 160, 149, 144, 149, 158, 158, 158, 155, 157\}$ un des ensembles de distances temporelles obtenu dans l'exemple 2.5. Trois chroniques élémentaires différentes peuvent être générées à partir de l'ensemble $\mathcal{D}_{ab}$ suivant la méthode de calcul des bornes de contrainte temporelle utilisée. Ces trois chroniques élémentaires sont représentées sur la figure 2.3.



(a) `minmax` (b) `2sigma` (c) `3sigma`

FIGURE 2.3 – Chroniques élémentaires générées à partir de l'ensemble de distances temporelles $\mathcal{D}_{ab}$ donné dans l'exemple 2.6 suivant les trois méthodes de calcul des bornes de la contrainte temporelle : `minmax`, `2sigma`, et `3sigma`.

2.3 Reconstitution de chroniques

Dans la section précédente, des méthodes de regroupement de données ont été utilisées afin d'identifier des chroniques élémentaires dans une séquence temporelle. Ces chroniques élémentaires représentent les phénomènes événementiels et temporels simples les plus prédominants dans les données d'entrée. Néanmoins, ces chroniques élémentaires ne peuvent pas en elles-mêmes représenter un phénomène complexe sous-jacent à la séquence temporelle. En effet, cette information est fragmentée parmi les différentes chroniques élémentaires identifiées. La reconstitution de chroniques est le processus qui permet de systématiquement rassembler les chroniques élémentaires qui représentent les informations fragmentées du même phénomène d'intérêt.

La reconstitution de chroniques est une méthode où l'on transforme de nombreuses chroniques élémentaires en quelques chroniques complexes. Pour reconstituer dans les meilleures conditions une chronique, deux éléments sont nécessaires. Le premier est un outil permettant d'agréger les différents composants d'une chronique alors que le second est un critère de choix pour cette agrégation. Ici, ces deux éléments sont les opérations (cf. définition 2.7) et l'indice de Jaccard (cf. définition 2.2).

2.3.1 Méthodologie de la reconstitution de chroniques

Dans cette section, une méthodologie pour le processus de reconstitution est développée. Un diagramme de la solution proposée est représenté sur la figure 2.4. Elle est composée de quatre étapes. Tout d'abord, dans la première étape, la chronique solution est initialisée à partir des différentes chroniques élémentaires compatibles (cf. définition 2.4). Dans cet exemple, la chronique solution $\mathcal{C}_{\{1,2,3\}}$ est composée des trois chroniques $\mathcal{C}_1$, $\mathcal{C}_2$ et $\mathcal{C}_3$. Puis, dans la deuxième étape, les opérations Ω qui seront appliquées sont définies à l'aide d'un ensemble d'indices de Jaccard noté $\mathcal{J}$. $\mathcal{J}$ est calculé à partir des sous-chroniques de la chronique solution $\mathcal{C}_{\{1,2,3\}}$. Ensuite, la troisième étape permet d'appliquer séquentiellement et avec un ordonnancement défini les opérations regroupées dans l'ensemble Ω. La chronique solution $\mathcal{C}_{\{1,2,3\}}$ est transformée en $\mathcal{C}_{\{1\cup 2,3\}}$. Enfin, la quatrième étape isole les différentes sous-chroniques indépendantes qui vont enrichir l'ensemble des chroniques apprises. Deux sous-chroniques indépendantes $\mathcal{C}_{1\cup 2}$ et $\mathcal{C}_3$ sont isolées et ajoutées à l'ensemble des chroniques apprises $\{\mathcal{C}\}_{apprise}$ De manière générale, une reconstitution de chroniques réussie génère un nombre restreint de chroniques. La reconstitution de chroniques est itérée avec différents ensembles de chroniques élémentaires compatibles. Des sous-ensembles de chroniques élémentaires peuvent être en commun entre deux itérations.

2.3.2 Algorithme de reconstitution de chroniques

Avec CDIRe, l'ensemble des chroniques élémentaires qui sont traitées proviennent de l'étape précédente d'identification de chroniques élémentaires. Ainsi,

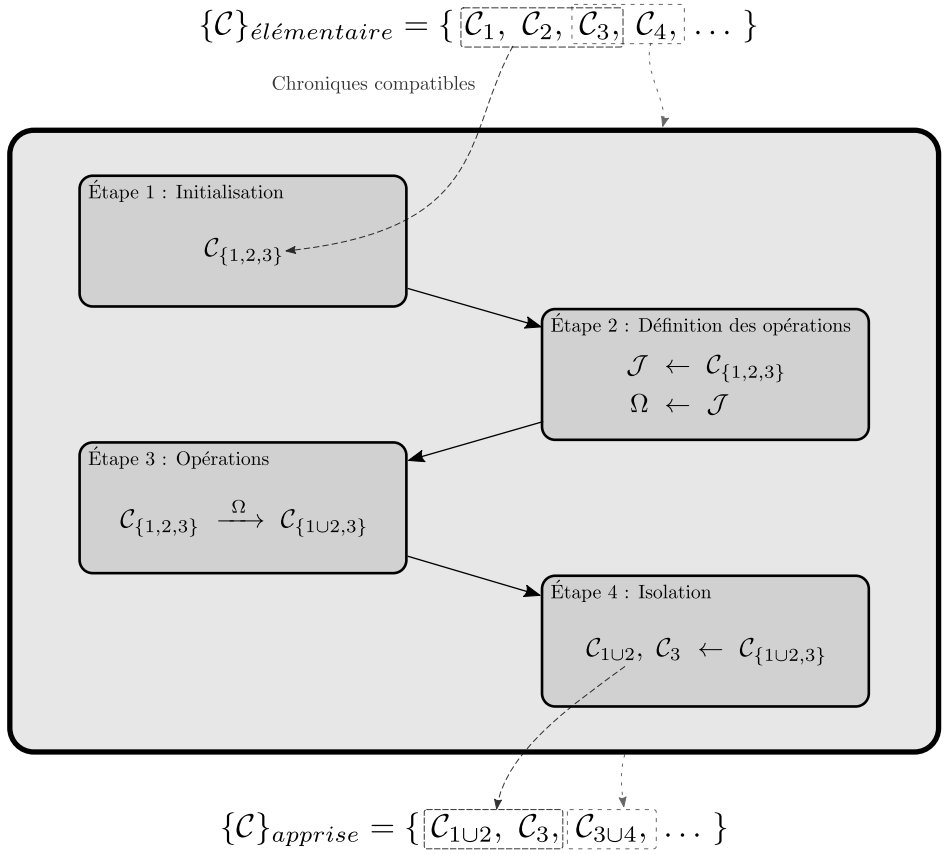


FIGURE 2.4 – La méthodologie proposée pour le processus de reconstitution de chroniques. Un ensemble de chroniques élémentaires compatibles sont sélectionnées à partir de la base de chroniques élémentaires générée par l'algorithme d'identification de chroniques élémentaires. Les chroniques reconstituées sont ajoutées dans l'ensemble des chroniques apprises. Plusieurs itérations sont effectuées avec différents ensembles de chroniques élémentaires compatibles.

un certain nombre d'informations sont à notre disposition et sont nécessaires au bon fonctionnement de l'approche proposée. Parmi ces informations figurent les occurrences des chroniques dans la séquence temporelle $\mathcal{S}$ traitée. La fréquence et l'ensemble des instants d'occurrence des nœuds sont déduits à partir de ces occurrences.

Les fréquences des différentes chroniques élémentaires sont répertoriées dans l'ensemble $\mathcal{F}$. Cet ensemble définit les différentes fréquences admissibles pour la reconstitution. En d'autres termes, aucune chronique ne sera reconstituée en dehors de ces fréquences. En effet, chaque fréquence admissible définit un ensemble de chroniques élémentaires compatibles sélectionnées dans les chroniques élémentaires à disposition. Les ensembles de chroniques élémentaires compatibles aux fréquences qui ne font pas parties de $\mathcal{F}$ sont nécessairement, soit des ensembles vides, soit des sous-ensembles de chroniques élémentaires compatibles à une fréquence admis-

sible. Après le processus de reconstitution de chroniques, ces ensembles seront des chroniques redondantes aux chroniques reconstituées à partir des ensembles de chroniques élémentaires compatibles à une fréquence admissible.

Une fois que la fréquence traitée est sélectionnée et que les chroniques élémentaires compatibles sont répertoriées dans une base de chroniques, une chronique solution $\mathcal{C}_{solution}$ est initialisée. L'initialisation consiste à regrouper toutes les chroniques élémentaires compatibles dans une seule chronique. Chaque chronique élémentaire appartenant à la base de chroniques considérée sera alors une sous-chronique indépendante de la chronique solution initialisée.

L'algorithme de reconstitution de chroniques est présenté sur l'Algorithme 3. Cet algorithme permet de reconstituer des chroniques à partir d'une base de chroniques élémentaires identifiées dans l'étape précédente. Les instructions des lignes 2 à 22 appliquent le processus de reconstitution de chroniques. Ce processus est appliqué pour chaque fréquence admissible dans l'ensemble $\mathcal{F}$. Les lignes 4 à 8 permettent de sélectionner toutes les chroniques élémentaires compatibles avec la fréquence en cours. Le $seuil_{sim}$ est le seuil minimum pour que les indices de Jaccard soient pris en compte. Les instructions des lignes 9 à 11 calculent tous les indices de Jaccard entre les chroniques élémentaires compatibles et les stockent dans une structure de données appelée $\mathcal{J}(\mathcal{S})$. Les différentes opérations sont définies par l'instruction ligne 12 en exploitant cette structure de données $\mathcal{J}(\mathcal{S})$. La section 2.4 détaillera le calcul des indices de Jaccard ainsi que la définition des opérations. Une fois les opérations définies, la chronique solution est initialisée avec l'ensemble des chroniques élémentaires compatibles (ligne 13). Puis l'instruction ligne 14 trie les opérations suivant des critères définis par l'utilisateur et les instructions lignes 15 à 20 appliquent les opérations cohérentes sur la chronique solution. Ce processus est décrit dans la section 2.5. Enfin, la ligne 21 permet d'isoler les différentes sous-chroniques indépendantes présentes dans la chronique solution et de les ajouter à l'ensemble des chroniques apprises.

2.4 Reconstitution de chroniques : opérations

Après initialisation de la chronique solution $\mathcal{C}_{solution}$, la prochaine étape est la reconstitution. En d'autres termes, ceci consiste à réduire la taille de la chronique solution tout en évitant la perte d'information. Les opérations, décrites dans la définition 2.7, jouent un rôle majeur dans ce processus. En effet, c'est grâce à cette union des nœuds qu'il est possible de regrouper les divers phénomènes représentés par les différentes sous-chroniques indépendantes élémentaires.

2.4.1 Définition des opérations à partir des indices de Jaccard entre les événements

Les opérations permettent d'agréger les différentes composantes d'une chronique et, plus précisément, réunir les événements. Néanmoins, les opérations en elles-mêmes ne sont pas des critères de choix pour la reconstitution de chroniques.

Algorithme 3 Reconstitution de chroniques

Entrées : $\{\mathcal{C}\}_{élémentaire}$, $seuil_{sim}$, $\mathcal{S}$
Sorties : $\{\mathcal{C}\}_{apprises}$

$\{\mathcal{C}\}_{apprises} \leftarrow \emptyset$
pour chaque $fréquence \in \mathcal{F}$ **faire**
  $\{\mathcal{C}\}_{compatible} \leftarrow \emptyset$
  pour chaque $\mathcal{C}^{\alpha} \in \{\mathcal{C}\}_{élémentaire}$ **faire**
    si $f(\mathcal{C}^{\alpha}, \mathcal{S}) \geq fréquence \times seuil_{sim}$ **alors**
      $\{\mathcal{C}\}_{compatible} \leftarrow \mathcal{C}^{\alpha}$
    fin si
  fin pour
  pour chaque $couple(\mathcal{C}_1^{\alpha}, \mathcal{C}_2^{\alpha}) \in \{\mathcal{C}\}_{compatible}$ **faire**
    $\mathcal{J}(\mathcal{S}) \leftarrow calculer\_similarité(\mathcal{C}_1^{\alpha}, \mathcal{C}_2^{\alpha}, \mathcal{S})$
  fin pour
  $\Omega(\mathcal{S}) \leftarrow définir\_opérations(\mathcal{J}(\mathcal{S}), seuil_{sim})$
  $\mathcal{C}_{solution} \leftarrow initialiser\_solution(\{\mathcal{C}\}_{compatible})$
  $trier\_opérations(\Omega(\mathcal{S}), \{\mathcal{C}\}_{compatible}, \mathcal{J}(\mathcal{S}))$
  tant que $\Omega(\mathcal{S}) \neq \emptyset$ **faire**
    $opération \leftarrow prendre\_première(\Omega(\mathcal{S}))$
    si $est\_cohérente(opération)$ **alors**
      $opération(\mathcal{C}_{solution})$
    fin si
  fin tant que
  $\{\mathcal{C}\}_{apprises} \leftarrow \{\mathcal{C}\}_{apprises} \cup sous\text{-}chroniques\_indépendantes(\mathcal{C}_{solution})$
fin pour
retourner $\{\mathcal{C}\}_{apprises}$

Appliquer un maximum d'opérations n'est pas automatiquement l'heuristique la plus désirable car le résultat final peut être une chronique non cohérente. C'est un très mauvais résultat pour l'utilisateur car une chronique incohérente est une chronique inutile. Un critère de choix pour distinguer les opérations avantageuses des autres doit être élaboré.

Une méthode pertinente pour définir les opérations intéressantes est de mettre à profit les instants d'occurrence de chaque nœud de la chronique solution $\mathcal{C}_{solution}$ dans la séquence temporelle $\mathcal{S}$. Ces informations sont connues grâce à l'étape précédente d'identification de chroniques élémentaires. Si les instants d'occurrence de deux nœuds différents sont similaires, alors ces deux nœuds sont eux-mêmes identiques. Avec deux nœuds identiques, l'union entre ces nœuds est possible, cela permet l'opération entre ces nœuds. Ainsi, lorsque les instants d'occurrence de deux nœuds sont jugés semblables, une nouvelle opération est ajoutée à l'ensemble des opérations qui seront appliquées à la chronique solution $\mathcal{C}_{solution}$. Afin de mesurer la similarité des nœuds, l'indice de Jaccard est utilisé (cf. définition 2.2). Cette mesure permet de définir sur une échelle de 0 à 1 la ressemblance entre deux ensembles.

Afin d'optimiser au mieux le temps de calcul des différents indices de Jaccard,

les indices calculés sont stockés dans une structure de données appelée $\mathcal{J}(\mathcal{S})$. Cette structure est construite à partir de la chronique solution et avant de définir les opérations qui seront appliquées sur celle-ci. La façon dont est calculée $\mathcal{J}(\mathcal{S})$ présente d'autres avantages. Notamment, un sous-ensemble de chroniques élémentaires peut être commun à deux ensembles de chroniques élémentaires compatibles de deux itérations du processus de reconstitution de chroniques. Il n'est alors pas nécessaire de calculer à nouveau les différents indices de Jaccard entre les nœuds de ces chroniques.

Définition 2.11. L'**ensemble des indices de Jaccard** $\mathcal{J}(\mathcal{S})$ est une structure de données qui répertorie tous les indices de Jaccard calculés entre les différents nœuds de la chronique solution $\mathcal{C}_{solution} = (\mathcal{E}_{solution}, \mathcal{T}_{solution})$ dans la séquence temporelle $\mathcal{S}$:

$$\mathcal{J}(\mathcal{S}) = \{J(\nu_{\mathcal{C}_{solution}}(e_i), \nu_{\mathcal{C}_{solution}}(e_j), \mathcal{S}) | e_i, e_j \in \mathcal{E}_{solution}\} \tag{2.25}$$

Définition 2.12. L'**ensemble des opérations** $\Omega(\mathcal{S})$ à appliquer sur une chronique solution $\mathcal{C}_{solution}$ correspond à toutes les opérations dont l'indice de Jaccard est supérieur au paramètre $seuil_{sim}$ dans la séquence temporelle $\mathcal{S}$:

$$\Omega(\mathcal{S}) = \{\omega_{(i,j)} \mid J(\nu_{\mathcal{C}_{solution}}(e_i), \nu_{\mathcal{C}_{solution}}(e_j), \mathcal{S}) \in \mathcal{J}(\mathcal{S}), \\ seuil_{sim} \leq J(\nu_{\mathcal{C}_{solution}}(e_i), \nu_{\mathcal{C}_{solution}}(e_j), \mathcal{S})\}. \tag{2.26}$$

Outre l'indice de Jaccard, il existe de nombreuses méthodes pour calculer la similarité entre deux ensembles : le coefficient de Dice [Dice 1945, Sørensen 1948], l'indice de Tversky [Tversky 1977], ou encore la similarité cosinus [Baeza-Yates 2011]. Ces mesures peuvent être utilisées en remplacement de l'indice de Jaccard car elles permettent de la même manière de définir sur une échelle de 0 à 1 la ressemblance entre deux groupes. Deux ensembles identiques obtiennent une mesure de similarité de 1 alors que deux ensembles complètement différents ont un indice de similarité de 0. Néanmoins, l'utilisation de ces différents indices et leur impact sur le résultat final de l'étape de reconstitution de chronique ne sont pas étudiés dans ce document. L'indice de Jaccard est utilisé pour définir les opérations car c'est l'indice le plus direct à mettre en place en raison de sa simplicité.

2.4.2 Résultats des opérations entre deux chroniques élémentaires

Afin de mieux cerner les conséquences des opérations sur une chronique, étudions maintenant les différentes opérations possibles entre deux chroniques élémentaires [3]. En effet, deux opérations peuvent avoir des résultats différents. Par exemple, une opération entre les deux nœuds source ne va pas avoir les mêmes conséquences sur la structure d'une chronique qu'une opération entre les deux nœuds destination. Quatre opérations peuvent être définies : une entre les deux nœuds source, une entre les deux nœuds destination, et deux entre chaque nœud source et chaque nœud

3. Notons que dans ce cas les deux chroniques élémentaires sont considérées comme deux sous-chroniques indépendantes d'une même chronique.

destination. Ainsi, les opérations peuvent être catégorisées suivant trois différents types en fonction des nœuds qu'elles affectent.

Définition 2.13. Le **type d'une opération** est défini par les nœuds qu'elle affecte. C'est une opération entre deux sous-chroniques indépendantes élémentaires $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ d'une chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$. Trois types d'opération sont définis :

— Une opération $\omega^s_{(i,j)}$ est dite **source** si les deux nœuds $\nu_{s\mathcal{C}_1^\alpha}(e_i)$ et $\nu_{s\mathcal{C}_2^\alpha}(e_j)$ sont des nœuds sources :

$$\omega^s_{(i,j)} \text{ si } \exists k, l \text{ tel que } \tau_{(i,k)}, \tau_{(j,l)} \in \mathcal{T}. \tag{2.27}$$

Une telle opération engendre une chronique où deux événements sont contraints après un événement.

— Une opération $\omega^d_{(i,j)}$ est dite **destination** si les deux nœuds $\nu_{s\mathcal{C}_1^\alpha}(e_i)$ et $\nu_{s\mathcal{C}_2^\alpha}(e_j)$ sont des nœuds destinations :

$$\omega^d_{(i,j)} \text{ si } \exists k, l \text{ tel que } \tau_{(k,i)}, \tau_{(l,j)} \in \mathcal{T}. \tag{2.28}$$

Une telle opération engendre une chronique où deux événements sont contraints avant un événement.

— Une opération $\omega^m_{(i,j)}$ est dite **mixte** si un des nœuds $\nu_{s\mathcal{C}_1^\alpha}(e_i)$ ou $\nu_{s\mathcal{C}_2^\alpha}(e_j)$ est un nœud source alors que l'autre est un nœud destination :

$$\omega^m_{(i,j)} \text{ si } \exists k, l \text{ tel que } \begin{cases} \tau_{(i,k)}, \tau_{(l,j)} \in \mathcal{T} \\ \quad \text{ou} \\ \tau_{(k,i)}, \tau_{(j,l)} \in \mathcal{T}. \end{cases} \tag{2.29}$$

Une telle opération engendre une chronique où trois événements sont contraints de manière séquentielle.

Exemple 2.7. Soit $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ avec $\mathcal{E} = \{e_1 = e, e_2 = e, e_3 = e, e_4 = e\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(3,4)} = e_3[t^-_{(3,4)}, t^+_{(3,4)}]e_4\}$ la chronique de taille 4 représentée sur la figure 2.5. Quatre opérations sont possibles entre les deux sous-chroniques indépendantes $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$:

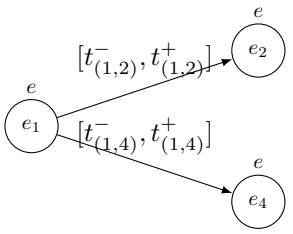
— une opération de type *source*, $\omega^s_{(1,3)}$, qui produit la chronique $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ avec $\mathcal{E}_1 = \{e_1 = e, e_2 = e, e_4 = e\}$ et $\mathcal{T}_1 = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(1,4)} = e_1[t^-_{(1,4)}, t^+_{(1,4)}]e_4\}$ représentée figure 2.6a ;

— une opération de type *destination*, $\omega^d_{(2,4)}$, qui produit la chronique $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ avec $\mathcal{E}_2 = \{e_1 = e, e_2 = e, e_3 = e\}$ et $\mathcal{T}_2 = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(3,2)} = e_3[t^-_{(3,2)}, t^+_{(3,2)}]e_2\}$ représentée figure 2.6b ;

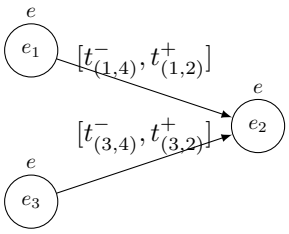
— deux opérations de type *mixte*, $\omega^m_{(1,4)}$ et $\omega^m_{(2,3)}$, qui produisent les chroniques $\mathcal{C}_3 = (\mathcal{E}_3, \mathcal{T}_3)$ avec $\mathcal{E}_3 = \{e_1 = e, e_2 = e, e_3 = e\}$ et $\mathcal{T}_3 = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(3,1)} = e_3[t^-_{(3,1)}, t^+_{(3,1)}]e_1\}$, et $\mathcal{C}_4 = (\mathcal{E}_4, \mathcal{T}_4)$ avec $\mathcal{E}_4 = \{e_1 =$

$e, e_2 = e, e_4 = e\}$ et $\mathcal{T}_4 = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(2,4)} = e_2[t^-_{(2,4)}, t^+_{(2,4)}]e_4\}$, représentées figure 2.6c et figure 2.6d.

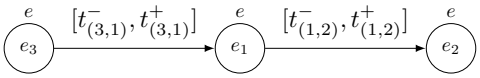
FIGURE 2.5 – Représentation graphique de la chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ avec $\mathcal{E} = \{e_1 = e, e_2 = e, e_3 = e, e_4 = e\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(3,4)} = e_3[t^-_{(3,4)}, t^+_{(3,4)}]e_4\}$. Cette chronique contient deux sous-chroniques indépendantes élémentaires $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$. Quatre opérations sont possibles entre les nœuds de ces chroniques.



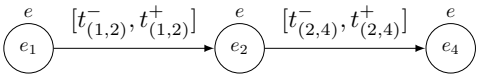
(a) Opération *source* : $\omega^s_{(1,3)}$



(b) Opération *destination* : $\omega^d_{(2,4)}$



(c) Opération *mixte* : $\omega^m_{(1,4)}$



(d) Opération *mixte* : $\omega^m_{(2,3)}$

FIGURE 2.6 – Résultat des différentes opérations possibles sur la chronique représentée figure 2.5. Trois types d'opération existent dans le cas de deux chroniques élémentaires : une opération de type *source*, une opération de type *destination* et deux opérations de type *mixte*.

Notons qu'entre deux chroniques élémentaires, les différentes opérations possibles sont exclusives. En effet, l'application d'une opération rend les autres opérations incohérentes. Cette remarque est élaborée dans la proposition suivante.

Proposition 2.1. *Soient $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ deux sous-chroniques indépendantes élémentaires. Si une des opérations est traitée parmi les quatre opérations possibles entre $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$, les trois autres opérations possibles ne sont plus cohérentes.*

Démonstration. Cf. annexe A.1. □

Cette proposition est particulièrement intéressante car elle permet de limiter les opérations qui seront traitées lors de la reconstitution de chroniques. En effet, une seule opération est possible pour chaque couple de chroniques élémentaires. Ainsi, le nombre d'opérations qui peuvent être définies à partir d'une base de chroniques élémentaires peut être limité.

2.4.3 Résultats des opérations dans le cas d'une base de chroniques

Dans le cas d'une base de chroniques, les différents résultats possibles des opérations augmentent avec la taille de la base de chroniques traitée. À cause de cela, il est difficile de généraliser les résultats obtenus par les opérations. Néanmoins, dans le contexte de la reconstitution de chroniques, les chroniques qui sont traitées sont regroupées dans une base de chroniques qui contient uniquement des chroniques élémentaires. Effectivement, rappelons que cette base de chroniques est générée par l'étape d'identification de chroniques élémentaires vue dans la section 2.2.

Toutes les opérations qui sont traitées dans cette base de chroniques sont réduites à une opération entre deux chroniques élémentaires. Les quatre opérations possibles entre chaque couple de chroniques élémentaires correspondent à l'ensemble des opérations qui seront traitées. Cette méthodologie permet d'exploiter les résultats démontrés dans la section précédente. Notamment, la catégorisation des opérations décrite dans la définition 2.13 peut être conservée pour les opérations ainsi définies. De plus, la propriété des opérations exclusives démontrées par la proposition 2.1 est utilisable. Ce qui permet de réduire le nombre d'opérations qui seront appliquées sur la chronique solution.

Exemple 2.8. Soit $\{\mathcal{C}\}_{compatibles} = \{\mathcal{C}_1^\alpha, \mathcal{C}_2^\alpha, \mathcal{C}_3^\alpha, \mathcal{C}_4^\alpha, \mathcal{C}_5^\alpha, \mathcal{C}_6^\alpha\}$ un ensemble de chroniques compatibles. La chronique solution $\mathcal{C}_{solution}$, représentée sur la figure 2.7, est initialisée à partir des chroniques compatibles $\{\mathcal{C}\}_{compatibles}$ où chaque chronique compatible est associée à une sous-chronique indépendante de $\mathcal{C}_{solution}$. On peut alors noter $\mathcal{C}_{solution} = (\mathcal{E}_{solution}, \mathcal{T}_{solution})$ avec $\mathcal{E}_{solution} = \{\mathcal{E}_1^\alpha, \mathcal{E}_2^\alpha, \mathcal{E}_3^\alpha, \mathcal{E}_4^\alpha, \mathcal{E}_5^\alpha, \mathcal{E}_6^\alpha\}$ et $\mathcal{T}_{solution} = \{\mathcal{T}_1^\alpha, \mathcal{T}_2^\alpha, \mathcal{T}_3^\alpha, \mathcal{T}_4^\alpha, \mathcal{T}_5^\alpha, \mathcal{T}_6^\alpha\}$. Une fois cette initialisation faite, les nœuds sont mis à jour afin de respecter l'ordre lexicographique des événements donné par $\leq_{\mathbb{E}}$. Les différentes sous-chroniques indépendantes élémentaires sont définies comme suit :

- $s\mathcal{C}_1^\alpha = (\mathcal{E}_1^\alpha, \mathcal{T}_1^\alpha)$ avec $\mathcal{E}_1^\alpha = \{e_1 = a, e_8 = b\}$ et $\mathcal{T}_1^\alpha = \{\tau_{(1,8)} = e_1[-176, -161]e_8\}$,
- $s\mathcal{C}_2^\alpha = (\mathcal{E}_2^\alpha, \mathcal{T}_2^\alpha)$ avec $\mathcal{E}_2^\alpha = \{e_2 = a, e_9 = b\}$ et $\mathcal{T}_2^\alpha = \{\tau_{(2,9)} = e_2[-62, -52]e_9\}$,
- $s\mathcal{C}_3^\alpha = (\mathcal{E}_3^\alpha, \mathcal{T}_3^\alpha)$ avec $\mathcal{E}_3^\alpha = \{e_3 = a, e_{10} = b\}$ et $\mathcal{T}_3^\alpha = \{\tau_{(3,10)} = e_3[48, 69]e_{10}\}$,
- $s\mathcal{C}_4^\alpha = (\mathcal{E}_4^\alpha, \mathcal{T}_4^\alpha)$ avec $\mathcal{E}_4^\alpha = \{e_{11} = b, e_{12} = b\}$ et $\mathcal{T}_4^\alpha = \{\tau_{(11,12)} = e_{11}[106, 118]e_{12}\}$,
- $s\mathcal{C}_5^\alpha = (\mathcal{E}_5^\alpha, \mathcal{T}_5^\alpha)$ avec $\mathcal{E}_5^\alpha = \{e_4 = a, e_5 = a\}$ et $\mathcal{T}_5^\alpha = \{\tau_{(4,5)} = e_4[106, 130]e_5\}$,
- $s\mathcal{C}_6^\alpha = (\mathcal{E}_6^\alpha, \mathcal{T}_6^\alpha)$ avec $\mathcal{E}_6^\alpha = \{e_6 = a, e_7 = a\}$ et $\mathcal{T}_6^\alpha = \{\tau_{(6,7)} = e_6[216, 242]e_7\}$.

Il existe 15 couples de sous-chroniques indépendantes élémentaires avec 4 indices de Jaccard à calculer. Un total de 60 indices de Jaccard sont calculés dans cet exemple et sont répertoriés dans l'ensemble $\mathcal{J}(\mathcal{S})$ (cf. annexe A.2). Seuls les 17 indices de Jaccard suivants sont supérieurs au seuil $seuil_{sim}$ lorsque celui-ci est fixé à 0.9 :

$$
\begin{array}{lll}
J(\nu_{\mathcal{C}_1^\alpha}(e_1), \nu_{\mathcal{C}_2^\alpha}(e_2), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_1^\alpha}(e_8), \nu_{\mathcal{C}_2^\alpha}(e_9), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_1^\alpha}(e_8), \nu_{\mathcal{C}_3^\alpha}(e_{10}), \mathcal{S}) = 0.94, \\
J(\nu_{\mathcal{C}_1^\alpha}(e_8), \nu_{\mathcal{C}_4^\alpha}(e_{11}), \mathcal{S}) = 1, & J(\nu_{\mathcal{C}_1^\alpha}(e_1), \nu_{\mathcal{C}_5^\alpha}(e_5), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_1^\alpha}(e_1), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 1, \\
J(\nu_{\mathcal{C}_2^\alpha}(e_9), \nu_{\mathcal{C}_3^\alpha}(e_{10}), \mathcal{S}) = 1, & J(\nu_{\mathcal{C}_2^\alpha}(e_9), \nu_{\mathcal{C}_4^\alpha}(e_{11}), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_2^\alpha}(e_9), \nu_{\mathcal{C}_4^\alpha}(e_{12}), \mathcal{S}) = 0.94, \\
J(\nu_{\mathcal{C}_2^\alpha}(e_2), \nu_{\mathcal{C}_5^\alpha}(e_5), \mathcal{S}) = 1, & J(\nu_{\mathcal{C}_2^\alpha}(e_2), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_3^\alpha}(e_{10}), \nu_{\mathcal{C}_4^\alpha}(e_{11}), \mathcal{S}) = 0.94, \\
J(\nu_{\mathcal{C}_3^\alpha}(e_{10}), \nu_{\mathcal{C}_4^\alpha}(e_{12}), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_3^\alpha}(e_3), \nu_{\mathcal{C}_5^\alpha}(e_4), \mathcal{S}) = 1, & J(\nu_{\mathcal{C}_3^\alpha}(e_3), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = 0.94, \\
J(\nu_{\mathcal{C}_5^\alpha}(e_4), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = 0.94, & J(\nu_{\mathcal{C}_5^\alpha}(e_5), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0.94. &
\end{array}
\tag{2.30}
$$

Suivant l'équation 2.26, les opérations associées à ces indices de Jaccard composent l'ensemble $\Omega(\mathcal{S})$:

$$
\begin{aligned}
\Omega(\mathcal{S}) = \{&\omega_{(1,2)}, \omega_{(8,9)}, \omega_{(8,10)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,10)}, \omega_{(9,11)}, \omega_{(9,12)}, \\
&\omega_{(2,5)}, \omega_{(2,7)}, \omega_{(10,11)}, \omega_{(10,12)}, \omega_{(3,4)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(5,7)}\}.
\end{aligned}
\tag{2.31}
$$

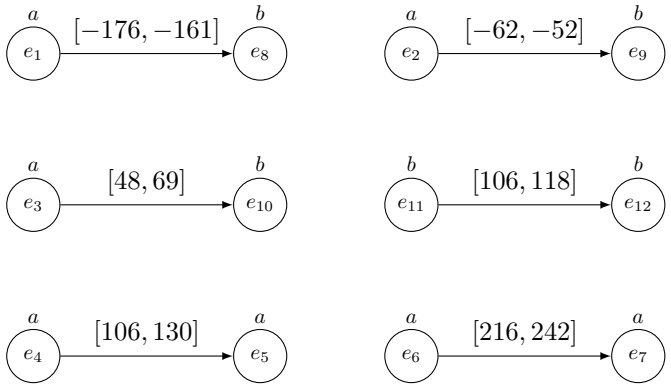


FIGURE 2.7 – La chronique solution $\mathcal{C}_{solution} = (\mathcal{E}_{solution} = \{\mathcal{E}_1^\alpha, \mathcal{E}_2^\alpha, \mathcal{E}_3^\alpha, \mathcal{E}_4^\alpha, \mathcal{E}_5^\alpha, \mathcal{E}_6^\alpha\}, \mathcal{T}_{solution} = \{\mathcal{T}_1^\alpha, \mathcal{T}_2^\alpha, \mathcal{T}_3^\alpha, \mathcal{T}_4^\alpha, \mathcal{T}_5^\alpha, \mathcal{T}_6^\alpha\})$ une fois initialisée où $\mathcal{E}_{solution} = \{\mathcal{E}_1^\alpha = \{e_1 = a, e_8 = b\}, \mathcal{E}_2^\alpha = \{e_2 = a, e_9 = b\}, \mathcal{E}_3^\alpha = \{e_3 = a, e_{10} = b\}, \mathcal{E}_4^\alpha = \{e_{11} = b, e_{12} = b\}, \mathcal{E}_5^\alpha = \{e_4 = a, e_5 = a\}, \mathcal{E}_6^\alpha = \{e_6 = a, e_7 = a\}\}$ et $\mathcal{T}_{solution} = \{\mathcal{T}_1^\alpha = \{\tau_{(1,8)} = e_1[-176, -161]e_8\}, \mathcal{T}_2^\alpha = \{\tau_{(2,9)} = e_2[-62, -52]e_9\}, \mathcal{T}_3^\alpha = \{\tau_{(3,10)} = e_3[48, 69]e_{10}\}, \mathcal{T}_4^\alpha = \{\tau_{(11,12)} = e_{11}[106, 118]e_{12}\}, \mathcal{T}_5^\alpha = \{\tau_{(4,5)} = e_4[106, 130]e_5\}, \mathcal{T}_6^\alpha = \{\tau_{(6,7)} = e_6[216, 242]e_7\}\}$. Chaque sous-chronique indépendante élémentaire correspond à une chronique de $\{\mathcal{C}\}_{compatibles}$.

2.5 Reconstitution de chroniques : ordonnancement des opérations

Dans la section précédente, un ensemble d'opérations $\Omega(\mathcal{S})$ est défini à partir des indices de similarité entre les différents nœuds de la chronique solution $\mathcal{C}_{solution}$. Cet ensemble représente l'intégralité des fusions des nœuds jouant un rôle dans la reconstitution de chroniques. Pour le moment, l'ensemble $\Omega(\mathcal{S})$ ne possède pas de

relation d'ordre définie. En d'autres termes, aucune opération n'est prioritaire sur une autre. Dans cette section, l'intérêt d'avoir une relation d'ordre sur les opérations est étudié. De plus, plusieurs critères sont pris en considération dans la définition d'une relation d'ordre adaptée.

2.5.1 Ordonnancement des opérations : point crucial pour la qualité des chroniques

Comme étudié dans les sections précédentes, une opération change la structure d'une chronique. En effet, celle-ci passe de taille n à taille $n-1$ (cf. définition 2.7) mais conserve le même nombre de contraintes temporelles. Ce changement structurel a plusieurs conséquences : la première est qu'un ensemble d'opérations sur une chronique solution $\mathcal{C}_{solution}$ ne peuvent être appliquées simultanément, elles doivent être appliquées d'une manière séquentielle ; la seconde est que l'application d'une opération peut enlever la propriété de cohérence à une seconde opération. Ces conséquences ont une importance significative sur la méthodologie proposée dans ce document. En effet, elles impliquent que l'ordonnancement des opérations appliquées sur une chronique doit être pris en compte. L'intérêt d'un tel ordonnancement des opérations est mis en avant dans l'exemple suivant.

Exemple 2.9. Soit $\mathcal{C}$ la chronique vue dans l'exemple 2.7 et représentée figure 2.5. Prenons l'hypothèse que les deux opérations $\omega_{(1,4)}$ et $\omega_{(2,3)}$ ont été jugées pertinentes et doivent être appliquées sur $\mathcal{C}$. En appliquant en premier l'opération $\omega_{(1,4)}$, la chronique représentée sur la figure 2.6c est obtenue. Dans ce cas, l'opération $\omega_{(2,3)}$ devient incohérente et ne peut être appliquée. Au contraire, en appliquant en premier l'opération $\omega_{(2,3)}$, c'est la chronique représentée sur la figure 2.6d qui est obtenue. C'est alors l'opération $\omega_{(1,4)}$ qui devient incohérente. Deux chroniques différentes sont obtenues suivant l'ordonnancement des opérations.

2.5.2 Heuristiques pour établir une relation d'ordre sur les opérations pertinentes

Un ordonnancement des opérations ad hoc est impératif au bon déroulement de la reconstitution de chroniques. L'ensemble des opérations $\Omega(\mathcal{S})$ vu dans la définition 2.12 doit être un ensemble *ordonné*. Pour cela, une relation d'ordre doit être définie sur les opérations. *Comment peut-on déterminer les opérations qui sont les plus intéressantes à appliquer en priorité ?* Afin de répondre à cette question, plusieurs critères sont considérés :

— l'*indice de similarité* associé à l'opération,

— le *type d'opération*,

— l'*intervalle des contraintes temporelles* des chroniques élémentaires étudiées.

Chacun de ces critères donne une heuristique intéressante dans la perspective de reconstituer au mieux les chroniques. Ces heuristiques sont décrites plus en détail dans les paragraphes suivants.

Dans le reste de ce document, $\Omega(\mathcal{S})$ est un ensemble ordonné où les opérations sont triées dans l'ordre décroissant, c'est-à-dire des opérations les plus grandes aux opérations les plus petites suivant la relation d'ordre choisie.

Indice de similarité L'indice de similarité entre deux nœuds donne une certaine confiance dans l'opération entre ces nœuds. En effet, une similarité de 1 signifie que les nœuds sont strictement identiques et, en conséquence, l'opération associée est nécessaire. L'heuristique qui en découle naturellement est d'imposer une relation d'ordre allant des opérations ayant l'indice de similarité le plus élevé aux opérations ayant l'indice de similarité le plus faible.

Définition 2.14. Soient $\omega_{(i,j)}$ et $\omega_{(k,l)}$ deux opérations appartenant à $\Omega(\mathcal{S})$, la **relation d'ordre basée sur l'indice de similarité** notée $\leq_s$ est telle que :

$$\omega_{(i,j)} \leq_s \omega_{(k,l)} \Leftrightarrow J(\nu_{\mathcal{C}}(e_i), \nu_{\mathcal{C}}(e_j), \mathcal{S}) \leq J(\nu_{\mathcal{C}}(e_k), \nu_{\mathcal{C}}(e_l), \mathcal{S}). \tag{2.32}$$

Exemple 2.10. Soient $\Omega(\mathcal{S})$ l'ensemble des opérations étudié dans l'exemple 2.8 et l'ensemble $\mathcal{J}(\mathcal{S})$ représenté dans l'annexe A.2. Avec la relation d'ordre basée sur l'indice de similarité $\leq_s$, l'ensemble $\Omega(\mathcal{S})$ devient :

$$\Omega_s(\mathcal{S}) = \{\omega_{(8,11)}, \omega_{(1,7)}, \omega_{(9,10)}, \omega_{(2,5)}, \omega_{(3,4)}, \omega_{(1,2)}, \omega_{(8,9)}, \omega_{(8,10)}, \omega_{(1,5)}, \omega_{(9,11)}, \omega_{(9,12)}, \omega_{(2,7)}, \omega_{(10,11)}, \omega_{(10,12)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(5,7)}\}. \tag{2.33}$$

Avec les valeurs des indices de Jaccard de cet exemple, plusieurs opérations sont égales suivant cette relation d'ordre. Ces opérations sont mises en évidence avec deux nuances de gris : un gris clair et un gris plus clair.

Type d'opération Comme vu dans la définition 2.13, chacun des trois types d'opération change la structure de la chronique synthétisée de manière différente. Ainsi, le type d'une opération est d'une importance fondamentale dans le processus de synthèse de chroniques. Il est donc tout naturel d'exploiter cette notion de type d'opération pour définir une heuristique intéressante pour les relations d'ordre sur les opérations. Par exemple, il peut être avantageux d'appliquer toutes les opérations de type mixte en priorité afin de synthétiser la chronique qui sera la plus séquentielle possible. Suivant ce que souhaite obtenir l'utilisateur, d'autres types d'opération peuvent être priorisés.

Définition 2.15. Soient $\Omega^m(\mathcal{S})$, $\Omega^d(\mathcal{S})$ et $\Omega^s(\mathcal{S})$ les ensembles des opérations, respectivement, mixte ω^m, destination ω^d et source ω^s dans $\Omega(\mathcal{S})$. Soient ω_1 et ω_2 deux opérations appartenant à $\Omega(\mathcal{S})$, la **relation d'ordre basée sur le type d'opération** notée $\leq_t$ est telle que :

$$\omega_1 \leq_t \omega_2 \Leftrightarrow \omega_1 \in \Omega^s(\mathcal{S}) \vee \Big(\omega_1 \in \Omega^d(\mathcal{S}) \wedge \omega_2 \in \Omega^d(\mathcal{S}) \cup \Omega^m(\mathcal{S})\Big) \vee (\omega_1 \in \Omega^m(\mathcal{S}) \wedge \omega_2 \in \Omega^m(\mathcal{S}))\,. \tag{2.34}$$

Exemple 2.11. Soit $\Omega(\mathcal{S})$ l'ensemble des opérations étudié dans l'exemple 2.8. Avec la relation d'ordre basée sur le type d'opération $\leq_t$, l'ensemble $\Omega(\mathcal{S})$ devient :

$$\Omega_t(\mathcal{S}) = \{\omega_{(1,2)}, \omega_{(3,4)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(8,9)}, \omega_{(8,10)}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(10,12)}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(2,5)}, \omega_{(2,7)}, \omega_{(10,11)}\}. \quad (2.35)$$

Dans cet exemple, plusieurs opérations sont du même type et sont donc égales suivant cette relation d'ordre. Ces opérations sont mises en évidence par différentes nuances de gris suivant leur type : opérations de type source, opérations de type destination et opérations de type mixte.

Intervalle des contraintes temporelles Les intervalles des contraintes temporelles sont définis à partir d'un critère de densité. En effet, l'étape d'identification de chroniques élémentaires utilise des méthodes de partitionnement des données dans le processus de calcul des bornes des contraintes temporelles des différentes chroniques (cf. section 2.2). Comme la reconstitution de chroniques consiste en l'agrégation des différentes chroniques élémentaires identifiées par ces méthodes, construire une chronique depuis diverses chroniques possédant ces critères de densités proches est intéressant. Une priorité sur les opérations entre deux chroniques générées à partir d'un regroupement ayant une densité proche peut être établie.

Exemple 2.12. Soient $s\mathcal{C}_1^\alpha$, $s\mathcal{C}_2^\alpha$ et $s\mathcal{C}_3^\alpha$ les trois sous-chroniques indépendantes élémentaires d'une chronique $\mathcal{C}$ représentées sur la figure 2.8. Les intervalles des contraintes temporelles $\tau_{(1,2)}$ de $s\mathcal{C}_1^\alpha$ et $\tau_{(3,4)}$ de $s\mathcal{C}_2^\alpha$ sont plus proches, autant de leur distance temporelle que de la durée maximale, l'une de l'autre qu'elles ne sont de l'intervalle de la contrainte temporelle $\tau_{(5,6)}$ de $s\mathcal{C}_3^\alpha$. Il est plus pertinent d'appliquer des opérations entre les chroniques $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ qu'avec $s\mathcal{C}_3^\alpha$ en raison de cette proximité. En effet, une chronique avec une même granularité sur toutes ses contraintes temporelles est plus appropriée qu'une chronique avec une granularité différente sur les diverses contraintes temporelles.

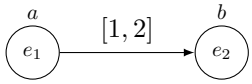
Deux mesures sur les intervalles des contraintes temporelles permettent de retrouver cette information :

— la première est la distance temporelle de l'intervalle : $d(t^-, t^+)$;

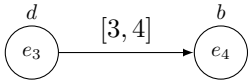
— la seconde mesure est la durée maximale de l'intervalle : $\max(|t^-|, |t^+|)$.

Une relation d'ordre sur les sous-chroniques indépendantes élémentaires d'une chronique $\mathcal{C}$ peut donc être définie à partir de ces deux mesures. Soient $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ deux sous-chroniques indépendantes élémentaires, la relation d'ordre notée $\leq_{ct}$ suivante privilégie la première mesure par rapport à la seconde :

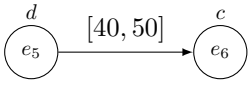
$$s\mathcal{C}_1^\alpha \leq_{ct} s\mathcal{C}_2^\alpha \Leftrightarrow d(t_1^-, t_1^+) < d(t_2^-, t_2^+) \vee \Big(d(t_1^-, t_1^+) = d(t_2^-, t_2^+) \wedge \max(|t_1^-|, |t_1^+|) \leq \max(|t_2^-|, |t_2^+|)\Big). \quad (2.36)$$



(a) La sous-chronique indépendante élémentaire $s\mathcal{C}_1^\alpha = (\mathcal{E}_1^\alpha, \mathcal{T}_1^\alpha)$ avec $\mathcal{E}_1^\alpha = \{e_1 = a, e_2 = b\}$ et $\mathcal{T}_1^\alpha = \{\tau_{(1,2)} = e_1[1,2]e_2\}$.



(b) La sous-chronique indépendante élémentaire $s\mathcal{C}_2^\alpha = (\mathcal{E}_2^\alpha, \mathcal{T}_2^\alpha)$ avec $\mathcal{E}_2^\alpha = \{e_3 = d, e_4 = b\}$ et $\mathcal{T}_2^\alpha = \{\tau_{(3,4)} = e_3[3,4]e_4\}$.



(c) La sous-chronique indépendante élémentaire $s\mathcal{C}_3^\alpha = (\mathcal{E}_3^\alpha, \mathcal{T}_3^\alpha)$ avec $\mathcal{E}_3^\alpha = \{e_5 = d, e_6 = c\}$ et $\mathcal{T}_3^\alpha = \{\tau_{(5,6)} = e_5[40,50]e_6\}$.

FIGURE 2.8 – Un exemple de trois sous-chroniques indépendantes élémentaires.

Avec cette relation d'ordre sur les sous-chroniques indépendantes élémentaires, il est possible de définir une heuristique sur la relation d'ordre des opérations. Ainsi, cette heuristique va directement dépendre des intervalles des contraintes temporelles des différentes sous-chroniques indépendantes élémentaires.

Définition 2.16. Soient quatre sous-chroniques indépendantes élémentaires $s\mathcal{C}_1^\alpha = (\mathcal{E}_1^\alpha, \mathcal{T}_1^\alpha)$, $s\mathcal{C}_2^\alpha = (\mathcal{E}_2^\alpha, \mathcal{T}_2^\alpha)$, $s\mathcal{C}_3^\alpha = (\mathcal{E}_3^\alpha, \mathcal{T}_3^\alpha)$ et $s\mathcal{C}_4^\alpha = (\mathcal{E}_4^\alpha, \mathcal{T}_4^\alpha)$ de $\mathcal{C}$ où $e_1 \in \mathcal{E}_1^\alpha$, $e_2 \in \mathcal{E}_2^\alpha$, $e_3 \in \mathcal{E}_3^\alpha$ et $e_4 \in \mathcal{E}_4^\alpha$. Soient $\omega_{(1,2)}$ et $\omega_{(3,4)}$ deux opérations appartenant à $\Omega(\mathcal{S})$, la **relation d'ordre basée sur l'intervalle des contraintes temporelles** notée $\leq_i$ est telle que :

$$\omega_{(1,2)} \leq_i \omega_{(3,4)} \Leftrightarrow s\mathcal{C}_3^\alpha \leq_{ct} s\mathcal{C}_4^\alpha \wedge (s\mathcal{C}_1^\alpha \leq_{ct} s\mathcal{C}_4^\alpha \vee s\mathcal{C}_2^\alpha \leq_{ct} s\mathcal{C}_4^\alpha) \vee s\mathcal{C}_4^\alpha <_{ct} s\mathcal{C}_3^\alpha \wedge (s\mathcal{C}_1^\alpha \leq_{ct} s\mathcal{C}_3^\alpha \vee s\mathcal{C}_2^\alpha \leq_{ct} s\mathcal{C}_3^\alpha)\,. \quad (2.37)$$

Exemple 2.13. Soient $\Omega(\mathcal{S})$ l'ensemble des opérations étudiées dans l'exemple 2.8 et $s\mathcal{C}_1^\alpha$, $s\mathcal{C}_2^\alpha$, $s\mathcal{C}_3^\alpha$, $s\mathcal{C}_4^\alpha$, $s\mathcal{C}_5^\alpha$ et $s\mathcal{C}_6^\alpha$ les sous-chroniques indépendantes élémentaires détaillées dans le même exemple. Les mesures sur les intervalles des contraintes temporelles de ces chroniques sont les suivantes :

$$s\mathcal{C}_1^\alpha : \begin{cases} d(t^-, t^+) = 15, \\ \max(|t^-|, |t^+|) = 176, \end{cases} \qquad s\mathcal{C}_2^\alpha : \begin{cases} d(t^-, t^+) = 10, \\ \max(|t^-|, |t^+|) = 62, \end{cases}$$

$$s\mathcal{C}_3^\alpha : \begin{cases} d(t^-, t^+) = 21, \\ \max(|t^-|, |t^+|) = 69, \end{cases} \qquad s\mathcal{C}_4^\alpha : \begin{cases} d(t^-, t^+) = 12, \\ \max(|t^-|, |t^+|) = 118, \end{cases}$$

$$s\mathcal{C}_5^\alpha : \begin{cases} d(t^-, t^+) = 24, \\ \max(|t^-|, |t^+|) = 130, \end{cases} \qquad s\mathcal{C}_6^\alpha : \begin{cases} d(t^-, t^+) = 26, \\ \max(|t^-|, |t^+|) = 242. \end{cases}$$

Avec ces mesures, les chroniques sont ordonnées comme suit :

$$s\mathcal{C}_2^\alpha \leq_{ct} s\mathcal{C}_4^\alpha \leq_{ct} s\mathcal{C}_1^\alpha \leq_{ct} s\mathcal{C}_3^\alpha \leq_{ct} s\mathcal{C}_5^\alpha \leq_{ct} s\mathcal{C}_6^\alpha$$

et la relation d'ordre basée sur les intervalles des contraintes temporelles $\leq_i$ impose l'ordonnancement suivant sur l'ensemble $\Omega(\mathcal{S})$:

$$\Omega_i(\mathcal{S}) = \{\omega_{(9,11)}, \omega_{(9,12)}, \omega_{(1,2)}, \omega_{(8,9)}, \omega_{(8,11)}, \omega_{(9,10)}, \omega_{(10,11)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(2,5)}, \omega_{(1,5)}, \omega_{(3,4)}, \omega_{(2,7)}, \omega_{(1,7)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(5,7)}\}. \quad (2.38)$$

Plusieurs opérations sont égales car ce sont des opérations entre les mêmes sous-chroniques indépendantes élémentaires.

2.5.3 Composition des relations d'ordre sur les opérations

Plusieurs heuristiques de relation d'ordre des opérations sont définies d'après des critères qui sont intéressants dans la perspective de reconstituer des chroniques pertinentes. Chacune de ces relations d'ordre peut être utilisée dans la reconstitution de chroniques. Néanmoins, il peut être opportun de composer ces relations d'ordre. En effet, en raison du nombre d'opérations traitées ainsi que la probabilité importante d'égalités (beaucoup d'indices de Jaccard ont la même valeur, il y a plusieurs opérations de chaque type, *etc.*), un nombre non négligeable d'opérations n'ont pas de relation d'ordre stricte. La composition de ces relations d'ordre peut prendre différentes formes, par exemple, il peut être plus pertinent de traiter en priorité un certain type d'opérations avant de considérer les indices de Jaccard ou encore traiter les opérations dans l'ordre défini par les distances temporelles des intervalles des contraintes temporelles puis par les types d'opérations. Plusieurs choix sont possibles, c'est alors à l'utilisateur de choisir quel critère il souhaite privilégier pour la reconstitution de chronique.

Définition 2.17. Soient $\omega_{(i,j)}$ et $\omega_{(k,l)}$ deux opérations appartenant à $\Omega(\mathcal{S})$. Notons $\leq_{abc}$ la **composition des relations d'ordre** $\leq_s$, $\leq_t$ et $\leq_i$ où $a, b, c \in \{s, t, i\}$. La relation d'ordre $\leq_{abc}$ est telle que :

$$\omega_{(i,j)} \leq_{abc} \omega_{(k,l)} \Leftrightarrow \omega_{(i,j)} <_a \omega_{(k,l)} \vee \Big(\omega_{(i,j)} =_a \omega_{(k,l)} \wedge \Big(\omega_{(i,j)} <_b \omega_{(k,l)} \vee \Big(\omega_{(i,j)} =_b \omega_{(k,l)} \wedge \omega_{(i,j)} \leq_c \omega_{(k,l)}\Big)\Big)\Big). \quad (2.39)$$

Exemple 2.14. Soit $\mathcal{C}_{solution}$ la chronique solution représentée sur la figure 2.7 et initialisée à partir de la base de chroniques élémentaires $\mathcal{C}_{compatibles}$. Soit $\Omega(\mathcal{S})$ l'ensemble des opérations définies à partir de l'ensemble $\mathcal{J}(\mathcal{S})$ et étudié dans l'exemple 2.8. Dans cet exemple, l'utilisateur choisit de privilégier le critère du type d'opération avant le critère de l'indice de similarité et d'appliquer en dernier le critère de l'intervalle des contraintes temporelles. Avec cette relation d'ordre $\leq_{tsi}$, l'ensemble des opérations $\Omega(\mathcal{S})$ défini dans l'équation (2.31) devient :

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(2,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(10,11)}, \omega_{(1,5)}, \omega_{(2,7)}\}. \quad (2.40)$$

Une fois que l'ensemble $\Omega_{tsi}(\mathcal{S})$ est obtenu, chaque opération est appliquée séquentiellement sur la chronique solution initialisée représentée sur la figure 2.7. Le résultat final des opérations $\Omega_{tsi}(\mathcal{S})$ est représentée sur la figure 2.9. Le détail de l'application des opérations pas à pas peut être retrouvé dans l'annexe A.3.

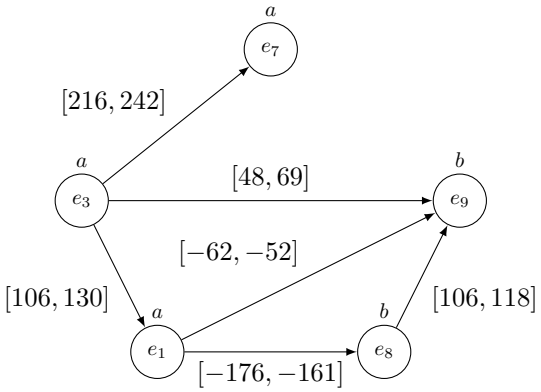


FIGURE 2.9 – La chronique solution $\mathcal{C}_{solution}$ une fois que toutes les opérations définies dans l'ensemble des opérations $\Omega_{tsi}(\mathcal{S})$ de l'exemple 2.14 ont été appliquées. C'est une chronique de taille 5 avec 6 contraintes temporelles.

Dans l'exemple précédent, une relation d'ordre spécifique pour les opérations a été choisie. Cet ordonnancement a donné lieu à une chronique solution possible. Néanmoins, une chronique qui peut être plus pertinente que celle-ci peut être découverte avec une séquence d'opérations différente. Pour illustrer ce propos, l'exemple suivant traite les même opérations vues précédemment avec un ordonnancement différent :

Exemple 2.15. Soient $\mathcal{C}_{solution}$ la chronique solution représentée sur la figure 2.7 et $\Omega(\mathcal{S})$ l'ensemble des opérations étudié dans l'exemple 2.8. Contrairement à l'exemple 2.14, l'utilisateur choisit de privilégier le critère de l'indice de similarité, puis le critère de l'intervalle des contraintes temporelles et enfin le critère du type d'opération. Avec la relation d'ordre $\leq_{sit}$ ainsi définie, l'ensemble ordonné d'opération $\Omega_{sit}(\mathcal{S})$ est le suivant :

$$\Omega_{sit}(\mathcal{S}) = \{\omega_{(8,11)}, \omega_{(9,10)}, \omega_{(2,5)}, \omega_{(3,4)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(9,12)}, \omega_{(1,2)}, \omega_{(8,9)}, \omega_{(10,11)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(1,5)}, \omega_{(2,7)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(5,7)}\}. \quad (2.41)$$

Après application de ces différentes opérations, la chronique solution représentée sur la figure 2.10 est obtenue. Il est intéressant de noter qu'une chronique différente de celle décrite dans l'exemple 2.14 est obtenue à partir du même ensemble d'opérations.

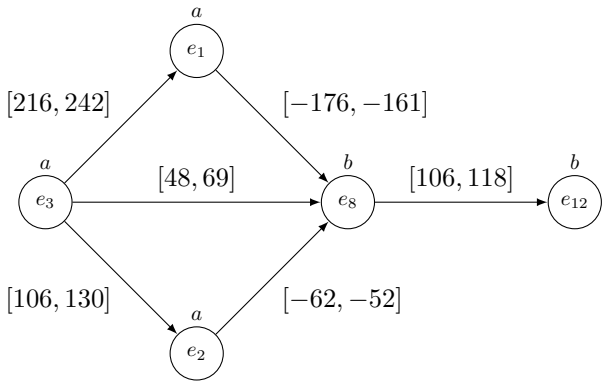


FIGURE 2.10 – La chronique solution $\mathcal{C}'_{solution}$ une fois que toutes les opérations définies dans l'ensemble des opérations $\Omega_{sit}(\mathcal{S})$ de l'exemple 2.15 ont été appliquées.

Ainsi, comme vu dans les Exemples 2.14 et 2.15, deux relations d'ordre sur les opérations différentes peuvent générer des chroniques différentes. En conséquence, il peut être intéressant d'explorer les différents résultats possibles avec des relations d'ordre sur les opérations différentes. Un autre sujet d'étude intéressant serait de trouver un ordonnancement d'opérations optimal pour une mesure objective donnée. Ce dernier point n'est pas exploré dans ce document et est une perspective de futurs travaux intéressante.

2.6 Analyse de la complexité algorithmique

Dans cette section, une analyse autour de la complexité algorithmique de CDIRe est fournie. Cette analyse se concentre dans un premier temps sur les deux étapes d'identification de chroniques élémentaires et de reconstitution de chroniques, puis, dans un second temps, sur la complexité globale de CDIRe. L'interconnectivité forte des notions d'*identification* et de *reconstitution* est mise en avant dans cette analyse.

La complexité algorithmique de l'étape d'identification de chroniques élémentaires fournie dans l'algorithme 2 est calculée en fonction de la taille de la séquence temporelle d'entrée $|\mathcal{S}|$. Le calcul des distances temporelles est polynomial $O\left(|\mathcal{S}|^2\right)$. L'algorithme utilisé pour le partitionnement des données est DBSCAN, celui-ci possède une complexité algorithmique de $O\left(n\log(n)\right)$ [Ester 1996] où n est la taille des données d'entrée. Dans le pire des cas, un seul événement est défini dans le domaine spatial, l'étape de partitionnement des données est donc égal à $O\left(|\mathcal{D}_{ab}|\log(|\mathcal{D}_{ab}|)\right)$. Enfin, la génération de chroniques élémentaires à partir des distances temporelles est linéaire en fonction du nombre de groupements effectués, ce qui dans le pire des cas est l'ensemble $\mathcal{D}_{ab}$, sa complexité est $O\left(|\mathcal{D}_{ab}|\right)$. La complexité algorithmique de l'identification des chroniques élémentaires est $O\left(|\mathcal{S}|^2 + |\mathcal{D}_{ab}|\log(|\mathcal{D}_{ab}|)\right)$. Or, quelle que soit la taille du domaine spatial $\mathbb{E}_{\mathcal{S}}$, le nombre de distances temporelles traité est le même et est égal à $|\mathcal{D}_{ab}| = \frac{|\mathcal{S}|(|\mathcal{S}|-1)}{2}$. Ainsi, la complexité algorithmique de l'étape d'identification de chroniques élémentaires en fonction de la taille de la

séquence temporelle d'entrée est :

$$O\left(|\mathcal{S}|^2 \log(|\mathcal{S}|)\right). \tag{2.42}$$

De plus, le nombre maximal de chroniques élémentaires identifiées est égal au nombre de distances temporelles calculées $|\{\mathcal{C}\}_{élémentaire}| = |\mathcal{D}_{ab}| = \frac{|\mathcal{S}|(|\mathcal{S}|-1)}{2}$.

La complexité algorithmique de l'étape de reconstitution de chroniques représentée dans l'algorithme 3 est calculée en fonction du nombre de chroniques élémentaires $|\{\mathcal{C}\}_{élémentaire}|$. Dans le pire des cas, lorsque le $seuil_{sim}$ est égal à 0, toutes les chroniques élémentaires sont des chroniques compatibles $\{\mathcal{C}\}_{compatible} = \{\mathcal{C}\}_{élémentaire}$. La définition des chroniques compatibles est linéaire $O(|\{\mathcal{C}\}_{élémentaire}|)$. Le calcul des indices de Jaccard est polynomiale en fonction du nombre de chroniques compatibles $O(|\{\mathcal{C}\}_{élémentaire}|^2)$, en effet, $\frac{|\{\mathcal{C}\}_{élémentaire}|(|\{\mathcal{C}\}_{élémentaire}|-1)}{2}$ couples sont possibles. La définition des opérations est linéaire en fonction de la taille de la structure $|\mathcal{J}(\mathcal{S})|$, dans le cas où $seuil_{sim}$, le nombre d'opérations est égal au nombre d'indices de Jaccard calculés $|\Omega(\mathcal{S})| = |\mathcal{J}(\mathcal{S})|$. Puis, le tri des opérations est égal à $O(|\Omega(\mathcal{S})| \log_2(|\Omega(\mathcal{S})|))$. Enfin, la complexité algorithmique du processus d'application des opérations suit : la procédure est faite pour toutes les opérations, la vérification de la cohérence d'une opération est similaire à une recherche du plus court voisin $O(|\mathcal{T}_{solution}| \log(|\mathcal{E}_{solution}|))$ et cette condition est respectée dans $\log(|\Omega(\mathcal{S})|)$ des cas. Ainsi, l'application des opérations est égale à $O(|\Omega(\mathcal{S})||\mathcal{T}_{solution}| \log(|\Omega(\mathcal{S})|) \log(|\mathcal{E}_{solution}|))$. Finalement, la procédure d'extraction des sous-chroniques élémentaires est linéaire en fonction de la taille de la chronique solution $O(|\mathcal{E}_{solution}|)$. La complexité de l'étape de reconstitution de chroniques est la suivante :

$$O\left(|\{\mathcal{C}\}_{élémentaire}|^2 + |\Omega(\mathcal{S})||\mathcal{T}_{solution}| \log(|\Omega(\mathcal{S})|) \log(|\mathcal{E}_{solution}|) + |\mathcal{E}_{solution}|\right). \tag{2.43}$$

Or, en considérant le fait que la chronique solution est initialisée à partir des chroniques compatibles élémentaires, la taille des ensembles de la chronique solution sont $|\mathcal{E}_{solution}| = 2|\{\mathcal{C}\}_{élémentaire}|$ et $|\mathcal{T}_{solution}| = |\{\mathcal{C}\}_{élémentaire}|$. La complexité devient :

$$O\left(|\{\mathcal{C}\}_{élémentaire}|^2 + |\Omega(\mathcal{S})||\{\mathcal{C}\}_{élémentaire}| \log(|\Omega(\mathcal{S})|) \log(|\{\mathcal{C}\}_{élémentaire}|)\right). \tag{2.44}$$

De plus, le nombre maximal des opérations dépend du nombre de chroniques compatibles, en effet, il y a 4 opérations par couple de chroniques et il y a un total de $\frac{|\{\mathcal{C}\}_{élémentaire}|(|\{\mathcal{C}\}_{élémentaire}|-1)}{2}$ couples. Le nombre maximal d'opérations est donc $2|\{\mathcal{C}\}_{élémentaire}|(|\{\mathcal{C}\}_{élémentaire}|-1)$. Avec cette information, la complexité devient :

$$O\left(|\{\mathcal{C}\}_{élémentaire}|^3 \log(|\{\mathcal{C}\}_{élémentaire}|^2) \log(|\{\mathcal{C}\}_{élémentaire}|)\right). \tag{2.45}$$

Ce qui en simplifiant, donne la complexité algorithmique de l'étape de reconstitution

de chroniques en fonction du nombre de chroniques élémentaires en entrée :

$$O\left(|\{\mathcal{C}\}_{élémentaire}|^3 \log^2(|\{\mathcal{C}\}_{élémentaire}|)\right). \tag{2.46}$$

Ainsi, la complexité de cette étape est largement dominée par le processus d'application des opérations. En particulier, la vérification de la cohérence des opérations est très coûteuse. Une étude sur les opérations pourrait permettre de grandement améliorer cette complexité. Cette étude est une perspective intéressante pour de futurs travaux.

Comme abordé précédemment, CDIRe est la séquence d'une phase d'identification de chroniques élémentaires et d'une phase de reconstitution de chroniques. En considérant les complexités algorithmiques de ces étapes offertes respectivement par les équations (2.42) et (2.46), la complexité algorithmique de CDIRe suit :

$$O\left(|\mathcal{S}|^2 \log(|\mathcal{S}|) + |\{\mathcal{C}\}_{élémentaire}|^3 \log^2(|\{\mathcal{C}\}_{élémentaire}|)\right). \tag{2.47}$$

Or, comme fourni précédemment, le nombre maximal de chroniques élémentaires identifiées par l'étape d'identification de chroniques élémentaires est polynomial en fonction de la taille de la séquence temporelle d'entrée $\mathcal{S}$. Ainsi, la complexité algorithmique de CDIRe dans le pire des cas est la suivante :

$$O\left(|\mathcal{S}|^6 \log^2(|\mathcal{S}|)\right). \tag{2.48}$$

La complexité algorithmique de CDIRe est largement dominée par l'étape de reconstitution de chroniques. Elle montre qu'un soin particulier aux résultats de l'identification de chroniques élémentaires doit être fait. En effet, la complexité devient catastrophique lorsque cette étape fournit des résultats en trop grand nombre. L'interconnexion des étapes d'*identification* et de *reconstitution* de CDIRe est évidente. Ces deux étapes sont toutes les deux d'une importance équivalente et ne devraient pas être négligées pour obtenir un excellent résultat.

2.7 Conclusion

CDIRe, le cœur de notre contribution, est le principal thème de ce chapitre. La problématique considérée, légèrement différente de celles étudiées par d'autres algorithmes de découverte de chroniques, est posée. Les éléments algorithmiques clefs de CDIRe y sont décrits en détails afin de faciliter sa compréhension. En effet, les deux étapes d'*identification* de chroniques élémentaires et de *reconstitution* ainsi que leur rôle dans le processus de CDIRe sont élaborés. De plus, une analyse de la complexité algorithmique montre que dans le pire des cas, CDIRe est polynomial en fonction de la taille de la séquence temporelle d'entrée $\mathcal{S}$: $O\left(|\mathcal{S}|^6 \log^2(|\mathcal{S}|)\right)$. Une analyse des performances et des résultats de CDIRe est fournie dans le chapitre suivant.

Chapitre 3

Analyse des performances

Sommaire

CDIRe, tel qu'il est détaillé dans le chapitre précédent, est implémenté en `C++`. Dans ce chapitre, une analyse des performances et des résultats de CDIRe sur un jeu de données provenant d'une application réelle est fournie. L'ensemble des mesures de performances de l'algorithme et des mesures de qualité des chroniques qui sont considérées pour cette analyse est posé.

Des mesures de performances provenant de l'analyse de modèles de classification binaires sont exploitées, les chroniques générées par CDIRe sont considérées comme des modèles de classification. Les mesures de performances provenant de ce domaine sont donc adaptées aux chroniques. De plus, le jeu de données exploité est découpé de manière à exploiter ces mesures de performances. Un ensemble d'entraînement, les données d'entrée de CDIRe, ainsi qu'un ensemble de test, les données utilisées pour calculer les mesures de performances, sont définis.

L'organisation du reste de ce chapitre est comme suit. Dans la section 3.1, les différentes mesures de performances et de qualité ainsi que le jeu de données exploité sont posés. Puis, une analyse des paramètres de CDIRe, que ce soit les paramètres de DBSCAN ou le seuil sur l'indice de Jaccard, est donnée dans la section 3.2. Dans la section 3.3, une analyse plus approfondie des résultats de CDIRe et en particulier

des chroniques générées par celui-ci est fournie. Enfin, la section 3.4 offre un bilan des résultats de CDIRe ainsi qu'une étude synthétique des influences des paramètres sur les résultats.

3.1 Préliminaires

Dans l'objectif d'analyser les performances de CDIRe, plusieurs mesures doivent être mises au point. Une méthodologie de mesures des performances de l'algorithme en général et de mesures de la qualité des chroniques reconstituées en particulier est détaillée dans le reste de cette section. De plus, le détail des données des applications réelles utilisées pour cette analyse est fourni.

3.1.1 Méthodologie de mesure des performances

Cette section s'intéresse aux mesures de performances utilisées pour analyser CDIRe : la *précision* et le *rappel*. Ces mesures de performances sont bien connues et largement utilisées dans le domaine du traitement de l'information [Manning 2008, Baeza-Yates 2011]. Elles sont notamment utiles pour l'analyse des modèles de classification binaire. Ces modèles classent les éléments d'un ensemble en deux groupes, généralement un groupe *positif* et un groupe *négatif*. La précision et le rappel sont calculés à partir de mesures de performances spécifiques à la classification binaire que sont les *vrais positifs*, *faux positifs*, *faux négatifs* et *vrais négatifs*. Bien souvent, ce type de modèle requiert un premier ensemble d'éléments appelé ensemble d'*entraînement* qui permet de définir l'architecture, déterminer les paramètres du modèle de classification ; et un deuxième ensemble d'éléments appelé ensemble de *test* permettant de calculer ces mesures. Ces deux ensembles ne peuvent pas avoir d'éléments en commun afin de ne pas biaiser les résultats.

Avec une approche similaire, ces mesures de performances peuvent être utiles à l'évaluation de la qualité des résultats des algorithmes de découverte de chroniques. En effet, chaque chronique peut être assimilée à un modèle de classification. Pour cela, les données utilisées sont divisées en deux ensembles de séquences temporelles, où chaque séquence est composée d'une occurrence d'un phénomène. L'ensemble des séquences *positives* correspond à l'ensemble des séquences contenant une occurrence du phénomène d'intérêt, alors que l'ensemble des séquences *négatives* correspond à l'ensemble des séquences ne contenant pas d'occurrence du phénomène d'intérêt. L'ensemble d'entraînement est pris à partir d'un ensemble de séquences positives. Le reste des séquences positives ainsi qu'un nombre équivalent de séquences négatives correspond à l'ensemble de test.

Afin de calculer les mesures de performances décrites précédemment, un algorithme de reconnaissance de chroniques (comme CRS [Dousson 1993]) est utilisé. Cet algorithme calcule les différentes occurrences dans chacune des séquences temporelles dans l'ensemble de test et permet ainsi de calculer les mesures de *vrais positifs*, *faux positifs*, *faux négatifs* et *vrais négatifs*. Ces mesures, adaptées ici au

contexte des chroniques, sont décrites dans les définitions suivantes et sont récapitulées dans la matrice d'erreur représentée sur le Tableau 3.1.

Définition 3.1. L'ensemble des **vrais positifs** d'une chronique $\mathcal{C}$, noté $VP(\mathcal{C})$, est l'ensemble des séquences positives $\mathcal{S}^+$ tel qu'il existe au moins une occurrence de $\mathcal{C}$ dans $\mathcal{S}^+$:

$$VP(\mathcal{C}) = \{\mathcal{S}^+ \mid \mathcal{O}_{\mathcal{C}}(\mathcal{S}^+) \neq \emptyset\}. \tag{3.1}$$

Définition 3.2. L'ensemble des **faux positifs** d'une chronique $\mathcal{C}$, noté $FP(\mathcal{C})$, est l'ensemble des séquences négatives $\mathcal{S}^-$ tel qu'il existe au moins une occurrence de $\mathcal{C}$ dans $\mathcal{S}^-$:

$$FP(\mathcal{C}) = \{\mathcal{S}^- \mid \mathcal{O}_{\mathcal{C}}(\mathcal{S}^-) \neq \emptyset\}. \tag{3.2}$$

Définition 3.3. L'ensemble des **faux négatifs** d'une chronique $\mathcal{C}$, noté $FN(\mathcal{C})$, est l'ensemble des séquences positives $\mathcal{S}^+$ tel qu'il n'existe pas d'occurrences de $\mathcal{C}$ dans $\mathcal{S}^+$:

$$FN(\mathcal{C}) = \{\mathcal{S}^+ \mid \mathcal{O}_{\mathcal{C}}(\mathcal{S}^+) = \emptyset\}. \tag{3.3}$$

Définition 3.4. L'ensemble des **vrais négatifs** d'une chronique $\mathcal{C}$, noté $VN(\mathcal{C})$, est l'ensemble des séquences négatives $\mathcal{S}^-$ tel qu'il n'existe pas d'occurrences de $\mathcal{C}$ dans $\mathcal{S}^-$:

$$VN(\mathcal{C}) = \{\mathcal{S}^- \mid \mathcal{O}_{\mathcal{C}}(\mathcal{S}^-) = \emptyset\}. \tag{3.4}$$

TABLE 3.1 – Tableau récapitulatif des mesures de *vrais positifs*, *faux positifs*, *faux négatifs* et *vrais négatifs* adaptées au contexte des chroniques.

		ensemble de tests	
		séquences positives	séquences négatives
reconnaissance	occurrences	*vrais positifs*	*faux positifs*
	pas d'occurrence	*faux négatifs*	*vrais négatifs*

Avec ces mesures de performances, la précision et le rappel d'une chronique peuvent être aisément calculés. La précision peut être interprétée comme une mesure de l'exactitude d'une chronique. Plus la précision est élevée, plus la chronique est descriptive du phénomène sous-jacent à l'ensemble d'entraînement. Le rappel peut, quant à lui, être interprété comme une mesure de l'exhaustivité d'une chronique. Plus le rappel est élevé, plus la chronique est susceptible de ne pas être reconnue dans une séquence qui ne correspond pas au phénomène décrit par les données d'entrée.

Définition 3.5. La **précision** d'une chronique $\mathcal{C}$ est le ratio entre les vrais positifs $VP(\mathcal{C})$ et l'ensemble des positifs détectés défini par les vrais positifs $VP(\mathcal{C})$ et les faux positifs $FP(\mathcal{C})$:

$$\textit{précision}(\mathcal{C}) = \frac{|VP(\mathcal{C})|}{|VP(\mathcal{C})| + |FP(\mathcal{C})|}. \tag{3.5}$$

Définition 3.6. Le **rappel** d'une chronique $\mathcal{C}$ est le ratio entre les vrais positifs $VP(\mathcal{C})$ et l'ensemble des séquences positives $\mathcal{S}^+$ défini par les vrais positifs $VP(\mathcal{C})$ et les faux négatifs $FN(\mathcal{C})$:

$$rappel(\mathcal{C}) = \frac{|VP(\mathcal{C})|}{|VP(\mathcal{C})| + |FN(\mathcal{C})|}. \tag{3.6}$$

Exemple 3.1. Soit la chronique $\mathcal{C} = \{\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}, \mathcal{T} = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(2,3)} = e_2[3,4]e_3\}\}$ de taille 3 représentées sur la figure 3.1 et extraite d'un ensemble de séquences positives. Soit l'ensemble de tests composé de quatre séquences positives et quatre séquences négatives et représenté dans la table 3.2. Dans cet exemple, un algorithme de reconnaissance de chroniques trouve une occur-

TABLE 3.2 – Exemple de séquences temporelles positives et négatives.

séquences positives	séquences négatives
$s_1^+ = \{(a,3),(b,5),(c,8)\}$	$s_1^- = \{(a,3),(b,5),(c,9)\}$
$s_2^+ = \{(b,2),(a,4),(b,5),(c,8)\}$	$s_2^- = \{(a,5),(d,6),(b,7)\}$
$s_3^+ = \{(a,1),(b,2),(c,6)\}$	$s_3^- = \{(a,2),(b,5),(d,7)\}$
$s_4^+ = \{(a,0),(b,2),(c,4)\}$	$s_4^- = \{(a,6),(b,6),(d,7),(c,9)\}$

rence de $\mathcal{C}$ dans les séquences temporelles suivantes : s_1^+, s_2^+, s_3^+ et s_1^-. Les ensembles de vrais positifs $VP(\mathcal{C})$ et faux positifs $FP(\mathcal{C})$ sont donc les suivants :

$$VP(\mathcal{C}) = \{s_1^+, s_2^+, s_3^+\}, \qquad FP(\mathcal{C}) = \{s_1^-\}.$$

Par contre, il n'y pas d'occurrence de $\mathcal{S}$ dans les séquences temporelles restantes : s_4^+, s_2^-, s_3^- et s_4^-. Ainsi, les ensembles de faux négatifs $FN(\mathcal{C})$ et de vrais négatifs $VN(\mathcal{C})$ sont les suivants :

$$FN(\mathcal{C}) = \{s_4^+\}, \qquad VN(\mathcal{C}) = \{s_2^-, s_3^-, s_4^-\}.$$

À partir de ces mesures, la précision et le rappel de $\mathcal{C}$ dans l'ensemble de test sont aisément calculés :

$$pr\acute{e}cision(\mathcal{C}) = \frac{|VP(\mathcal{C})|}{|VP(\mathcal{C})| + |FP(\mathcal{C})|} = \frac{3}{3+1} = 0.75,$$
$$rappel(\mathcal{C}) = \frac{|VP(\mathcal{C})|}{|VP(\mathcal{C})| + |FN(\mathcal{C})|} = \frac{3}{3+1} = 0.75.$$

La précision et le rappel quantifient la qualité d'une chronique générée par un algorithme de découverte de chroniques. Afin d'analyser les performances de ces algorithmes, l'ensemble des chroniques découvertes est représenté sur un tracé avec le rappel en fonction de la précision. Chaque chronique correspond alors à un point sur ce tracé. Ce type de courbe, appelée courbe Précision-Rappel (PR), est fréquemment utilisé dans la recherche d'informations [Raghavan 1989]. Le calcul de

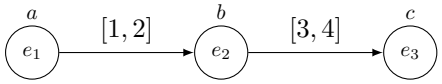


FIGURE 3.1 – Une chronique $\mathcal{C} = \{\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}, \mathcal{T} = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(2,3)} = e_2[3,4]e_3\}\}$ de taille 3 générée par un algorithme de découverte de chroniques.

l'aire sous la courbe, appelé *AUC* (*Area Under Curve*), est comprise entre 0 et 1. L'*AUC* permet de quantifier les performances des résultats des algorithmes de découverte de chroniques. Plus l'*AUC* est proche de 1, plus les résultats sont bons et l'algorithme performant. L'aire sous la courbe est obtenue de manière analogue à la formule suivante :

$$AUC = \int_{-\infty}^{\infty} précision(\mathcal{C}) rappel(\mathcal{C}) d\mathcal{C}. \tag{3.7}$$

Exemple 3.2. Soient les chroniques reconstituées par CDIRe et tracées en fonction de leur précision et de leur rappel sur la figure 3.2. Sur les 25 chroniques représentées sur cette courbe, 21 ont leur précision et rappel égal à 0. Les précision et rappel des chroniques restantes sont :

$$\begin{aligned} &précision(\mathcal{C}_{24}) = 1, && rappel(\mathcal{C}_{24}) = 0.33, \\ &précision(\mathcal{C}_{25}) = 1, && rappel(\mathcal{C}_{25}) = 0.33, \\ &précision(\mathcal{C}_{27}) = 1, && rappel(\mathcal{C}_{27}) = 0.5, \\ &précision(\mathcal{C}_{31}) = 0.5, && rappel(\mathcal{C}_{31}) = 1. \end{aligned}$$

Le résultat de l'*AUC* pour ces chroniques reconstituées est donné par :

$$AUC = 0.75.$$

Le résultat de l'*AUC* correspond à l'aire grisée sur la courbe *PR* représentée sur la figure 3.2.

3.1.2 Méthodologie de mesure de la qualité des chroniques

Un algorithme de découverte de chroniques générant bien souvent une quantité non négligeable de chroniques, des critères de choix doivent être définis par l'utilisateur de tels algorithmes. Dans cet objectif, un ensemble de mesures de la qualité des chroniques générées sont utilisées. Ces différentes mesures sont basées sur l'aspect structurel d'une chronique (taille, compacité), sur l'information temporelle (durée minimale, durée maximale), ou encore sur la fréquence dans la séquence temporelle d'entrée. Dans cette section, ces mesures de la qualité d'une chronique sont définies et analysées.

La *taille* d'une chronique donne un aperçu de la complexité du phénomène sous-jacent. Par exemple, une chronique de taille 2 est très simple et communé-

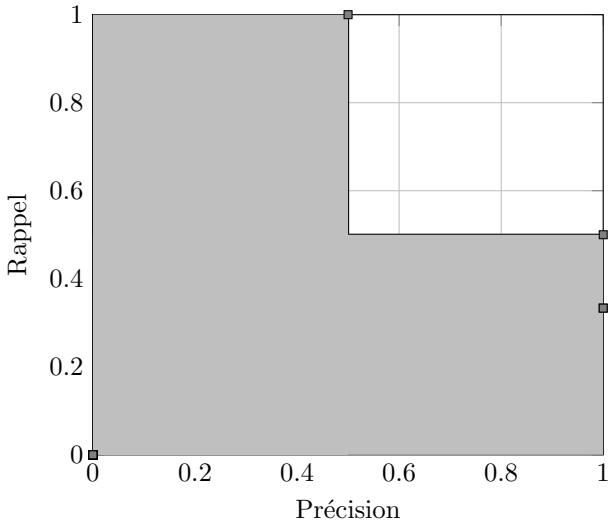


FIGURE 3.2 – Exemple d'une courbe *PR*. Chaque chronique est représentée par un carré gris. L'*AUC*, correspondant à l'aire de la zone grisée, est égal à 0.75.

ment retrouvée dans un algorithme de découverte de chroniques. Plus sa taille sera importante, plus elle sera complexe et rarement extraite par ces algorithmes. Une autre mesure utile pour l'analyse des résultats de CDIRe est la *compacité*.

Définition 3.7. La **compacité** $c(\mathcal{C})$ d'une chronique $\mathcal{C}$ est le produit entre la taille n de la chronique et le ratio entre le nombre m de contraintes temporelles de la chronique et le nombre $(\frac{n(n-1)}{2})$ de contraintes temporelles maximal qui est :

$$c(\mathcal{C}) = n \times m \times \frac{2}{n(n-1)} = \frac{2m}{n-1}. \tag{3.8}$$

Exemple 3.3. La chronique $\mathcal{C} = \{\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}, \mathcal{T} = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(2,3)} = e_2[3,4]e_3\}\}$ de taille $n = 3$ composée de $m = 2$ contraintes temporelles, représentée sur la figure 3.1, possède la compacité suivante :

$$c(\mathcal{C}) = \frac{2m}{n-1} = \frac{2 \times 2}{3-1} = 2.$$

Relativement au fonctionnement de CDIRe, la compacité est une mesure particulièrement opportune. En effet, elle permet de mesurer la redondance de l'information dans les données d'entrée. Par exemple, une chronique qui possède trois événements et trois contraintes temporelles comme celle représentée sur la figure 3.3 possède une bonne compacité. En effet, l'information donnée par la contrainte temporelle $\tau_{(1,3)}$ est similaire à l'information donnée par les contraintes temporelles $\tau_{(1,2)}$ et $\tau_{(2,3)}$. Or, ces contraintes temporelles ont été générées de manière indépendante lors de l'étape d'identification de chroniques élémentaires (cf. section 2.2.1, page 44). Ainsi, l'information qu'un événement c doit se produire entre 4 et 6 unités de temps après l'occurrence d'un événement a a été trouvée deux fois dans les données d'entrée, ce

qui rend la chronique plus remarquable. C'est pourquoi, une chronique qui possède une compacité importante réclame une étude plus approfondie par l'utilisateur.

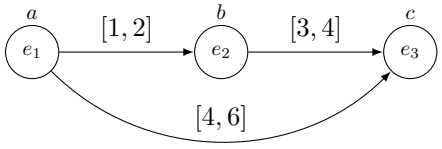


FIGURE 3.3 – Une chronique $\mathcal{C}' = \{\mathcal{E}' = \{e_1 = a, e_2 = b, e_3 = c\}, \mathcal{T}' = \{\tau_{(1,2)} = e_1[1,2]e_2, \tau_{(1,3)} = e_1[4,6]e_3, \tau_{(2,3)} = e_2[3,4]e_3\}\}$ de taille 3 générée par CDIRe avec la compacité suivante : $c(\mathcal{C}) = 3$.

Les informations sur la durée de reconnaissance d'une chronique peuvent être utiles pour l'utilisateur. En effet, la *durée minimale* et la *durée maximale* permet de donner une approximation de la granularité du temps de reconnaissance d'une chronique. Plus ces mesures seront petites, plus la chronique sera reconnue rapidement. Au contraire, une chronique avec des durées minimale et maximale importantes sera reconnue plus tardivement. Enfin, une chronique avec des valeurs de durée minimale et maximale sur des granularités de temps différentes peut être synonyme de bruit et être supprimée.

Définition 3.8. La **durée minimale** d'une chronique $\mathcal{C}$ est le temps minimum écoulé entre le début et la fin d'une occurrence de la chronique $\mathcal{C}$.

Définition 3.9. La **durée maximale** d'une chronique $\mathcal{C}$ est le temps maximum écoulé entre le début et la fin d'une occurrence de la chronique $\mathcal{C}$.

Enfin, la *fréquence* d'une chronique peut être une mesure très pertinente pour l'utilisateur. En effet, l'utilisateur peut être intéressé par des phénomènes qui sont très fréquents dans les données d'entrée, ou au contraire, par privilégier les chroniques peu fréquentes, souvent synonymes d'une anomalie, qu'il s'agisse de bruit ou d'un phénomène fautif.

3.1.3 Descriptif des jeux de données exploités

CDIRe a été évalué sur un jeu de données provenant d'une application réelle. Ce jeu de données provient d'une collection, définie par [Mörchen 2010b, Fradkin 2015], qui est composée de 9 ensembles de séquences temporelles illustrant divers phénomènes dynamiques d'intérêt. Chaque séquence temporelle contient uniquement une occurrence d'un phénomène, deux phénomènes différents n'apparaissent pas dans une même séquence temporelle. Cette collection de jeux de données peut être retrouvée sur le site internet de l'auteur [Mörchen 2010a]. Le jeu de données est décrit dans la suite de cette section et est récapitulé dans la table 3.3. Dans cette table sont récapitulés les différents *phénomènes* présents, le *nombre de séquences temporelles* contenant chaque phénomène et le *nombre d'événements* datés ainsi que le *temps*

TABLE 3.3 – Tableau récapitulatif du jeu de données exploité. Les différentes catégories de ce tableau sont : *Jeu de données* est le nom des données, *Phénomène* correspond au phénomène sous-jacent aux séquences temporelles, *# séquences temporelles* est le nombre de séquences temporelles de ce scénario spécifique, *# événements* est le nombre d'événements datés contenus dans ces séquences temporelles où la borne inférieure est le nombre minimal alors que la borne supérieure est le nombre maximal d'événements et *Temps moyen* est le temps moyen écoulé entre le premier et le dernier événement des différentes séquences temporelles.

Jeu de données	Phénomène	# séquences temporelles	# événements	Temps moyen
`blocks`	*pick-up*	30	$[6, 10]$	31.53
	put-down	30	$[6, 10]$	33.9
	stack	30	$[12, 14]$	31.93
	unstack	30	$[12, 14]$	29.27
	move-left	15	10	57.2
	move-right	15	$[10, 14]$	49.4
	assemble	30	$[12, 24]$	84.1
	disassemble	30	$[16, 18]$	107.9

moyen écoulé entre le premier et le dernier événement de ces différentes séquences temporelles.

Ce jeu de données est à propos pour l'analyse des performances de CDIRe. En effet, les données sont sous forme de séquences temporelles étiquetées, où chaque étiquette correspond à un phénomène. En d'autres termes, cela signifie que le phénomène sous-jacent à chacune de ces séquences temporelles est connu. De plus, les phénomènes sous-jacents à chaque séquence temporelle représentent des actions opposées. Ces étiquettes permettent de calculer les mesures de performances décrites dans la section précédente et ainsi quantifier les résultats de notre algorithme.

Les séquences temporelles appartenant au jeu de données intitulé `blocks` représentent différents phénomènes qui prennent leur source dans des vidéos de mains empilant des blocs de couleurs. Un total de 8 phénomènes de complexité croissante constitue ce jeu de données :

— *pick-up* : une main retire un bloc rouge posé sur un bloc vert.

— *put-down* : une main pose un bloc rouge sur un bloc vert.

— *stack* : une main pose un bloc rouge sur une pile composée d'un bloc bleu et d'un bloc vert.

— *unstack* : une main retire un bloc rouge posé sur une pile composée d'un bloc bleu et d'un bloc vert.

— *move-left* : une main déplace un bloc rouge posé sur un bloc bleu et le dépose sur un bloc vert.

— *move-right* : une main déplace un bloc rouge posé sur un bloc vert et le dépose sur un bloc bleu.

— *assemble* : une main assemble trois blocs de couleur en une pile composée d'un bloc vert, d'un bloc bleu et d'un bloc rouge.

— *disassemble* : une main désassemble une pile composée d'un bloc vert, d'un bloc bleu et d'un bloc rouge en trois blocs de couleur différente.

Les différents événements de ces séquences temporelles correspondent aux contacts entre les différents blocs de couleur et les actions de la main. Les unités de temps associées correspondent aux numéros de l'image de la vidéo.

Par exemple, les séquences temporelles associées au phénomène étiqueté *pick-up* sont composées de six types d'événements différents qui sont interprétés comme suit :

— CONTACTS__GREEN__REDd : début d'un contact entre un bloc rouge et un bloc vert,

— CONTACTS__GREEN__REDe : fin d'un contact entre un bloc rouge et un bloc vert,

— ATTACHED__GREEN__REDd : début d'un contact partiel entre un bloc rouge et un bloc vert,

— ATTACHED__HAND__REDd : fin d'un contact partiel entre un bloc rouge et un bloc vert,

— ATTACHED__GREEN__REDe : début d'un contact entre la main et un bloc rouge,

— ATTACHED__HAND__REDe : fin d'un contact entre la main et un bloc rouge.

Des extraits d'une vidéo à l'origine du phénomène étiqueté *pick-up* sont représentés sur la figure 3.4. Un exemple d'une séquence temporelle représentant un phénomène *pick-up* est donné ci-dessous :

$$\{(\text{CONTACTS\_\_GREEN\_\_RED}d, 199), (\text{CONTACTS\_\_GREEN\_\_RED}e, 208), (\text{ATTACHED\_\_GREEN\_\_RED}d, 209), (\text{ATTACHED\_\_HAND\_\_RED}d, 209), (\text{ATTACHED\_\_GREEN\_\_RED}e, 221), (\text{ATTACHED\_\_HAND\_\_RED}e, 233)\} \quad (3.9)$$

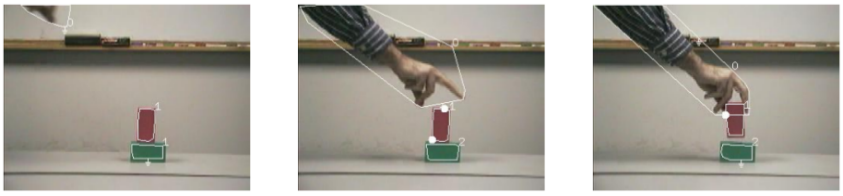


FIGURE 3.4 – Extraits de vidéo à l'origine d'un phénomène étiqueté *pick-up*. Images issues de [Fern 2002].

Les actions des différents phénomènes sous-jacents aux séquences temporelles sont opposées. Par exemple le phénomène *pick-up* correspond au retrait d'un bloc alors que le phénomène *put-down* correspond à la pose d'un bloc. Ainsi, les différentes séquences temporelles associées aux différents phénomènes sont considérées comme des séquences *positives* ou *négatives* afin de calculer les mesures de performances décrites dans la section précédente. Dans ce contexte, les séquences *positives* sont les séquences dont le phénomène sous-jacent est le phénomène d'intérêt

et les séquences *négatives* sont les séquences dont le phénomène sous-jacent est un phénomène opposé au phénomène d'intérêt. La table 3.4 récapitule les différentes séquences positives et négatives.

TABLE 3.4 – Tableau récapitulatif de la classification en séquences *positive* et *négative* en fonction du phénomène sous-jacent. Une séquence *positive* est une séquence dont le phénomène sous-jacent est le phénomène d'intérêt, alors qu'une séquence *négative* est une séquence dont le phénomène sous-jacent est un phénomène opposé au phénomène d'intérêt.

Jeu de données	Phénomène	Séquences *positives*	Séquences *négatives*
`blocks`	*pick-up*	*pick-up*	*put-down*
	put-down	*put-down*	*pick-up*
	stack	*stack*	*unstack*
	unstack	*unstack*	*stack*
	move-left	*move-left*	*move-right*
	move-right	*move-right*	*move-left*
	assemble	*assemble*	*disassemble*
	disassemble	*disassemble*	*assemble*

Afin d'évaluer CDIRe, les différentes séquences temporelles sont divisées en deux ensembles. Le premier est l'ensemble d'entraînement qui comprend 80% des séquences temporelles du phénomène étudié, c'est cet ensemble qui est considéré comme données d'entrée pour CDIRe. Le second ensemble est l'ensemble de test permettant de calculer les différentes mesures de performances définies dans la section précédente. Il comprend les 20% restant des séquences temporelles du phénomène étudié, qui sont considérées comme les séquences *positives*, ainsi que 20% des séquences temporelles du phénomène opposé au phénomène étudié, qui sont alors considérées comme les séquences *négatives*. La répartition du nombre de séquences temporelles dans chaque ensemble est fournie dans la table 3.5.

TABLE 3.5 – Tableau récapitulatif du nombre de séquences temporelles dans les ensembles d'entraînement et de test pour chaque phénomène.

Jeu de données	Phénomène	Ensemble d'entraînement	Ensemble de test	
			positif	*négatif*
`blocks`	*pick-up*	24	6	6
	put-down	24	6	6
	stack	24	6	6
	unstack	24	6	6
	move-left	12	3	3
	move-right	12	3	3
	assemble	24	6	6
	disassemble	24	6	6

Le jeu de données utilisé dans les expérimentations décrites dans ce chapitre est formé de *plusieurs séquences temporelles* dont chacune contient une occurrence d'un phénomène d'intérêt. Néanmoins, CDIRe ne considère qu'*une seule séquence temporelle* en entrée. Ainsi, une étape préliminaire de traitement des données doit être appliquée afin de pouvoir utiliser notre algorithme sur ces données. Cette étape

est effectuée simplement en concaténant les différentes séquences temporelles sélectionnées dans l'ensemble d'entraînement afin de construire une seule séquence temporelle. En effet, les différentes séquences temporelles contenant le même phénomène d'intérêt ne se recouvrent pas temporellement. De plus, la granularité de temps entre deux séquences temporelles est suffisamment différente pour qu'il n'y ait pas de chroniques générées pouvant reposer sur plusieurs séquences temporelles. Dans la table 3.6, le résultat de cette concaténation préliminaire est donné pour chacune des bases de test. Dans cette table, la concaténation est le résultat lorsque l'ensemble des séquences temporelles correspondant à un scénario est considéré. Néanmoins, d'autres jeux de données peuvent requérir une étape de décalage temporel avant une telle concaténation.

Exemple 3.4. Soit une séquence temporelle de type *pick-up* définie par l'Équation (3.9). Prenons une deuxième séquence temporelle de type *pick-up* définie ci-dessous :

$$\{(\text{CONTACTS\_GREEN\_RED}d, 736), (\text{CONTACTS\_GREEN\_RED}e, 745), (\text{ATTACHED\_HAND\_RED}d, 746), (\text{ATTACHED\_GREEN\_RED}d, 746), (\text{ATTACHED\_GREEN\_RED}e, 757), (\text{ATTACHED\_HAND\_RED}e, 768)\} \quad (3.10)$$

Ces deux séquences temporelles ne se recouvrent pas temporellement, une étape de décalage temporel n'est pas nécessaire. Leur concaténation est la suivante :

$$\{(\text{CONTACTS\_GREEN\_RED}d, 199), (\text{CONTACTS\_GREEN\_RED}e, 208), (\text{ATTACHED\_GREEN\_RED}d, 209), (\text{ATTACHED\_HAND\_RED}d, 209), (\text{ATTACHED\_GREEN\_RED}e, 221), (\text{ATTACHED\_HAND\_RED}e, 233), (\text{CONTACTS\_GREEN\_RED}d, 736), (\text{CONTACTS\_GREEN\_RED}e, 745), (\text{ATTACHED\_HAND\_RED}d, 746), (\text{ATTACHED\_GREEN\_RED}d, 746), (\text{ATTACHED\_GREEN\_RED}e, 757), (\text{ATTACHED\_HAND\_RED}e, 768)\} \quad (3.11)$$

TABLE 3.6 – Tableau récapitulatif des jeux de données avec les séquences temporelles concaténées. Les différentes catégories de ce tableau sont : *Jeu de données* est le nom des données, *Phénomène* correspond au phénomène sous-jacent à la séquence temporelle concaténée, $|\mathcal{S}|$ est le nombre d'événements datés contenus dans la séquence temporelle et $|\mathbb{E}_{\mathcal{S}}|$ est le nombre d'événements contenus dans la séquence temporelle.

Jeu de données	Phénomène	$\lvert\mathcal{S}\rvert$	$\lvert\mathbb{E}_{\mathcal{S}}\rvert$
`blocks`	*pick-up*	148	6
	put-down	146	6
	stack	290	10
	unstack	292	10
	move-left	120	10
	move-right	120	10
	assemble	424	12
	disassemble	388	12

3.2 Analyse des paramètres

Comme vu dans le chapitre précédent, CDIRe possède plusieurs paramètres affectant la qualité et la quantité des chroniques découvertes ainsi que le temps de calcul. Cette section a pour objectif d'analyser ces paramètres grâce aux résultats obtenus sur les jeux de données décrits dans la section précédente. Tout d'abord, les paramètres de l'algorithme de partitionnement des données utilisé sont étudiés et les valeurs de ces paramètres offrant les résultats les plus pertinents pour la suite de l'analyse des résultats de CDIRe sont définis. Puis, le seuil de l'indice de Jaccard est considéré.

Dans l'ensemble des résultats présentés dans ce chapitre, l'ordre des opérations choisi correspond à $\leq_{tsi}$ (cf. section 2.5.3, page 63) et, sauf indications contraires, la méthode de calcul des bornes de la contrainte temporelle utilisée est `minmax` (cf. section 2.2.3, page 48).

3.2.1 Paramètres de l'algorithme de partitionnement des données

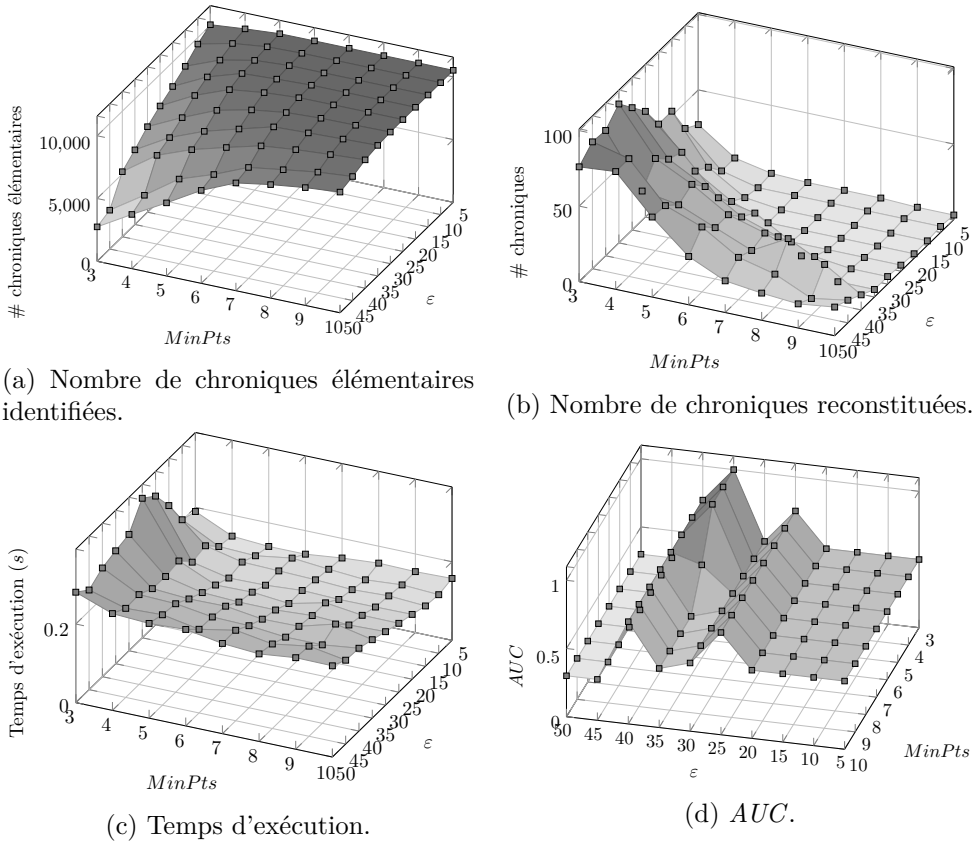
L'algorithme de partitionnement des données utilisé pour la phase d'identification de chroniques élémentaires de CDIRe est DBSCAN. En effet, comme vu dans la section 2.2.2 (page 45), cet algorithme est basé sur la densité des données et cette propriété est utile car elle offre des chroniques élémentaires pertinentes. Cet algorithme fait appel à deux paramètres, ε et $MinPts$, qui affectent directement la quantité et la qualité des chroniques élémentaires identifiées et, par conséquent, la qualité du résultat final. Dans cette section, ces paramètres sont évalués sur le jeu de données décrit dans la section précédente.

Pour chaque séquence temporelle considérée, CDIRe est exécuté avec plusieurs valeurs des paramètres ε et $MinPts$. En vue de limiter l'influence des autres variables, les paramètres du seuil sur l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9. Ces paramètres et leur rôle dans le processus de découverte de chroniques sont définis respectivement dans les sections 2.4.1 (page 52) et 2.2.3 (page 48). Dix valeurs différentes sont considérées pour ε, $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ et huit pour $MinPts$, $\{3, 4, 5, 6, 7, 8, 9, 10\}$. Chaque combinaison de ces valeurs correspond à une exécution de CDIRe pour un total de 80 points de mesures utiles. Quatre mesures sont utilisées pour évaluer les performances, le nombre de chroniques élémentaires identifiées, qui correspond directement au nombre de partitions que l'algorithme DBSCAN fournit, le nombre de chroniques reconstituées, le temps d'exécution de l'algorithme [1] et l'AUC. L'ensemble des points de mesures utilisés pour l'évaluation des paramètres de l'algorithme de partitionnement des données DBSCAN sont fournis dans l'annexe B.1.

Les premiers résultats considérés sont issus de la séquence temporelle correspondant au phénomène *pick-up*. Ce phénomène, comme décrit dans la section précédente, équivaut à une main retirant un bloc rouge posé sur un bloc vert et est un

1. Le temps d'exécution est mesuré du début à la fin du processus de découverte de chronique. Le temps de calcul des différentes mesures de performances n'est pas inclus.

des phénomènes les plus simples du jeu de données `blocks`.



(a) Nombre de chroniques élémentaires identifiées.

(b) Nombre de chroniques reconstituées.

(c) Temps d'exécution.

(d) AUC.

FIGURE 3.5 – Résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *pick-up*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixée à 0.9.

Les points de mesures sur le nombre de chroniques élémentaires identifiées en fonction des paramètres de DBSCAN représentés sur la figure 3.5a confirment les analyses existantes sur ces paramètres [Ester 1996]. En effet, sachant que le nombre de chroniques élémentaires identifiées est directement proportionnel au nombre de groupement fournis par DBSCAN, il est parfaitement compréhensible que plus le paramètre ε est important, moins de groupements seront effectués, et inversement, plus le paramètre $MinPts$ est grand, plus le nombre de groupements fournis sera important. Ici, le nombre maximum de chroniques élémentaires identifiées est de 10520 avec les paramètres $\varepsilon = 5$ et $MinPts = 10$, alors que le nombre minimum de chroniques élémentaires identifiées est de 2742 avec les paramètres $\varepsilon = 50$ et $MinPts = 3$.

Le nombre de chroniques reconstituées, représenté sur la figure 3.5b, est inversement proportionnel au nombre de chroniques élémentaires identifiées. Cet effet s'explique par le fonctionnement de la phase de reconstitution de CDIRe (cf. section 2.3, page 50). En effet, celle-ci dépend des indices de Jaccard pour définir les

opérations entre les différentes chroniques élémentaires. Un groupement avec un peu de distances temporelles générées dans la phase d'identification de chroniques élémentaires entraîne des ensembles des instants d'occurrence d'un nœud de petite taille. Les indices de Jaccard entre ces différents ensembles atteindront avec une probabilité plus importante le seuil requis $seuil_{sim}$. Plus d'opérations seront définies et, en conséquence, les chroniques reconstituées seront de taille plus grande et moins nombreuses. Au contraire, des grands regroupements dans la phase d'identification rendent la phase de reconstitution plus ardue car plus sensible aux variations entre les différents regroupements qui peuvent être importants. Cet effet semble s'accentuer jusqu'au point où aucune opération n'est possible entre les différentes chroniques élémentaires identifiées. Ainsi, l'utilisateur doit faire un choix suivant si son objectif est de découvrir des chroniques plus simples, dans ce cas il utilisera des valeurs des paramètres de DBSCAN offrant des grands regroupements, ou si son objectif est de découvrir des chroniques plus complexes, et dans ces conditions il utilisera des valeurs des paramètres de DBSCAN fournissant peu de regroupements. Ici, le nombre maximum de chroniques reconstituées est de 92 avec les paramètres $\varepsilon = 35$ et $MinPts = 3$, alors que le nombre minimum de chroniques reconstituées est de 3 avec les paramètres $\varepsilon = 5$ et $MinPts = 10$.

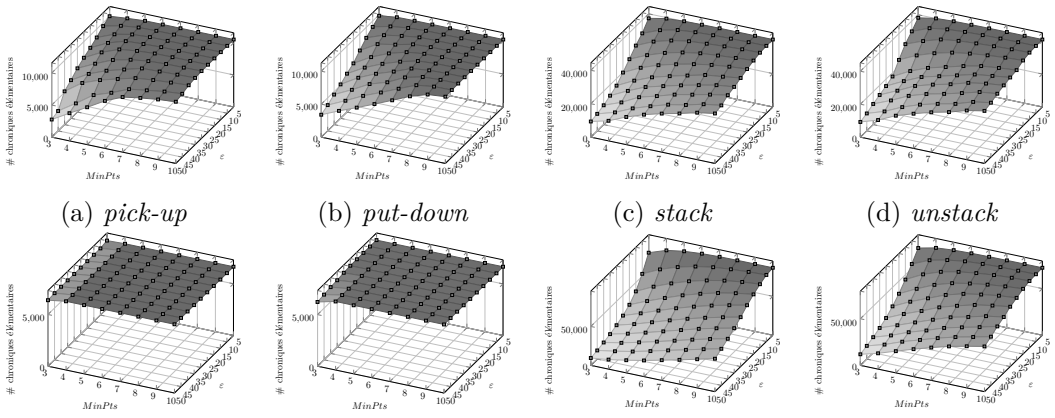
Le temps d'exécution de CDIRe pour les différents résultats de la séquence temporelle *pick-up* est représentée sur la figure 3.5c. Ces mesures montrent que le temps d'exécution reste autour de 0.2 secondes dans cet exemple. Le temps d'exécution est légèrement influencé par le nombre de chroniques élémentaires identifiées.

Enfin, l'AUC est donné sur la figure 3.5d [2]. Il apparaît que cette mesure n'est pas affectée par le paramètre $MinPts$ de DBSCAN. L'AUC semble indiquer que les résultats les plus performants pour la séquence temporelle correspondent aux valeurs des paramètres de ε égal à 35 et de $MinPts$ égal à 3, 4 ou 5. Néanmoins, l'AUC n'est pas nécessairement la mesure de performance la plus pertinente dans cet exemple. En effet, des chroniques élémentaires peuvent fausser cette mesure. Ce phénomène est discuté plus amplement dans la section 3.3.

Les résultats pour la séquence temporelle correspondant au phénomène *pick-up* étant analysés en détail précédemment, les résultats pour les séquences temporelles associées aux autres phénomènes du jeu de données `blocks` sont analysés plus globalement dans le reste de cette section. Les figures 3.6, 3.7, 3.8 et 3.9 présentent respectivement le nombre de chroniques élémentaires identifiées, le nombre de chroniques reconstituées, le temps d'exécution et l'AUC pour chaque séquence temporelle du jeu de données exploité.

La figure 3.6 représentant le nombre de chroniques élémentaires identifiées en fonction des paramètres de DBSCAN par chaque séquence temporelle confirme l'analyse fournie pour la séquence temporelle *pick-up*. De plus, le nombre de chroniques élémentaires identifiées est en relation avec la taille de la séquence temporelle, plus celle-ci est importante, plus le nombre de chroniques élémentaires identifiées

2. Afin de conserver une visibilité satisfaisante, les axes du tracé de la figure 3.5d n'ont pas la même orientation que sur les autres tracés de la figure 3.5.

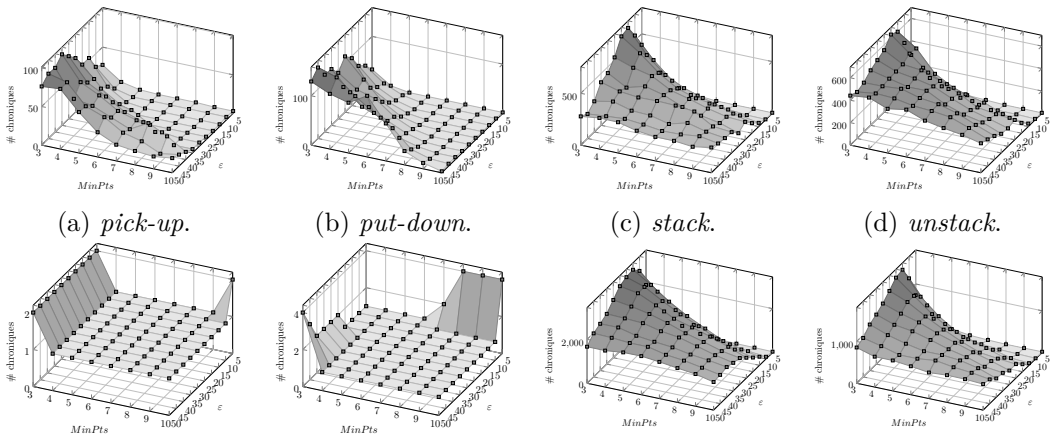


(a) *pick-up* (b) *put-down* (c) *stack* (d) *unstack*

(e) *move-left* (f) *move-right* (g) *assemble* (h) *disassemble*

FIGURE 3.6 – Nombre de chroniques élémentaires fournies par l'étape d'identification de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixée à 0.9.

seront nombreuses. Enfin, dans cet exemple, les séquences temporelles associées aux phénomènes *move-left* et *move-right* donnent le même résultat quels que soient les paramètres de DBSCAN. Ceci peut s'expliquer par la fréquence réduite des phénomènes sous-jacents. En effet, les algorithmes de partitionnement des données offrent typiquement de mauvais résultats lorsque la taille des données est faible.



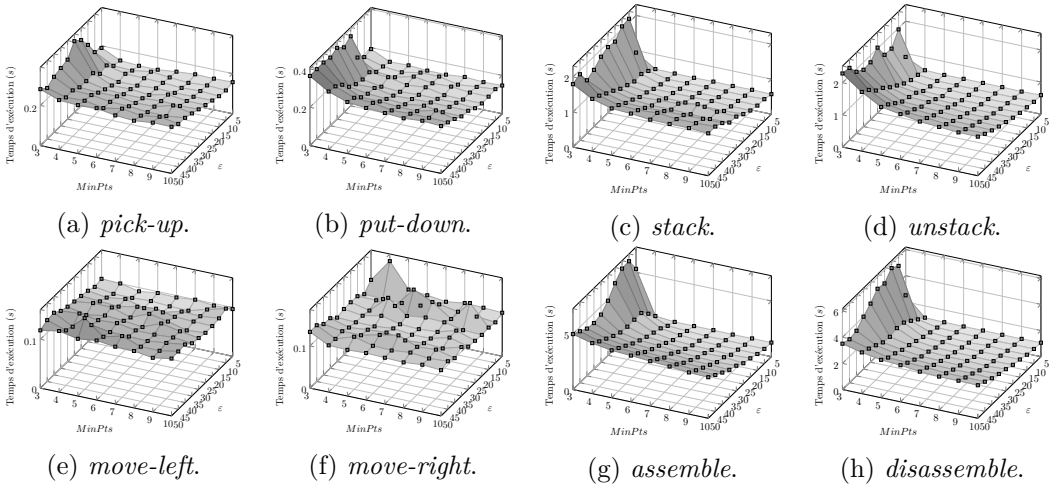
(a) *pick-up.* (b) *put-down.* (c) *stack.* (d) *unstack.*

(e) *move-left.* (f) *move-right.* (g) *assemble.* (h) *disassemble.*

FIGURE 3.7 – Nombre de chroniques reconstituées par CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixée à 0.9.

Sur la figure 3.7 présentant le nombre de chroniques découvertes, les courbes fournissant les résultats des séquences temporelles associées aux phénomènes *stack*, *unstack*, *assemble* et *disassemble* présentent un pic lorsque les deux paramètres ε et $MinPts$ sont faibles. Ce pic identique que la proportionnalité inverse entre le

nombre de chroniques élémentaires identifiées et le nombre de chroniques reconstituées qui est constatée sur les résultats des séquences temporelles associées aux phénomènes *pick-up* et *put-down* n'est pas vérifiée dans tous les résultats. Ainsi, au même nombre de chroniques élémentaires identifiées, il peut y avoir un nombre de chroniques reconstituées différent. Ces courbes indiquent que moins de chroniques sont reconstituées lorsque les paramètres ε et $MinPts$ sont élevés par rapport à lorsque ceux-ci sont faibles. De plus, les courbes présentant les résultats des séquences temporelles associées aux phénomènes *move-left* et *move-right* montrent qu'une chronique reconstituée est prédominante dans les données d'entrée. Ceci peut s'expliquer par la complexité du phénomène sous-jacent par rapport à la fréquence de celui-ci.



(a) *pick-up*. (b) *put-down*. (c) *stack*. (d) *unstack*.

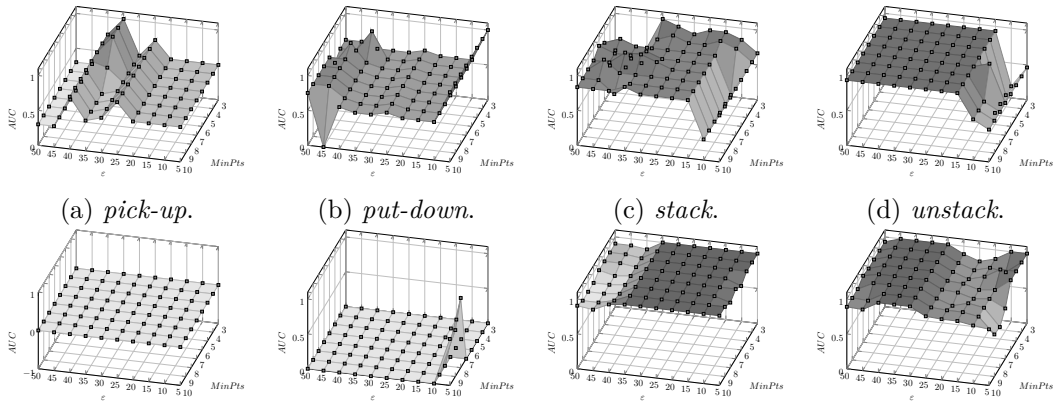
(e) *move-left*. (f) *move-right*. (g) *assemble*. (h) *disassemble*.

FIGURE 3.8 – Temps d'exécution de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixée à 0.9.

Le temps d'exécution de CDIRe en fonction des paramètres ε et $MinPts$ de DBSCAN est présenté sur la figure 3.8. Ces courbes montrent que le temps d'exécution est proportionnel à la taille de la séquence temporelle d'entrée. Un pic dans le temps d'exécution visible sur les courbes des résultats des séquences temporelles associées aux phénomènes *stack*, *unstack*, *assemble* et *disassemble* correspond au pic présent dans le nombre de chroniques reconstituées. Le temps d'exécution est relatif au nombre de chroniques reconstituées suivant un ordre de grandeur polynomial. Une étude plus approfondie est requise pour confirmer ces observations.

Les résultats sur l'AUC représenté sur la figure 3.9 confirment les conclusions précédentes. En effet, l'AUC ne semble pas affecté par le paramètre $MinPts$ de DBSCAN. Suivant cette mesure de performance, CDIRe donne de meilleur résultats avec les séquences temporelles les plus importantes du jeu de données `blocks`.

Le choix des paramètres de l'algorithme de partitionnement des données est critique car ils influencent fortement la quantité et la qualité des chroniques reconstituées par CDIRe. En effet, un mauvais choix de ces paramètres ne permet pas



(a) *pick-up*. (b) *put-down*. (c) *stack*. (d) *unstack*.

(e) *move-left*. (f) *move-right*. (g) *assemble*. (h) *disassemble*.

FIGURE 3.9 – Résultat de l'AUC pour CDIRe avec plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

à CDIRe d'offrir des chroniques utiles, soit parce qu'une part trop importante des chroniques reconstituées sont associées à du bruit, soit parce qu'aucune chronique n'est générée. Avec le jeu de données exploité, les paramètres ε et $MinPts$ sont choisis afin de respecter un compromis entre le nombre de chroniques élémentaires identifiées et le nombre de chroniques reconstituées. Elles sont déterminées lorsque à la fois, la courbe du nombre de chroniques élémentaires identifiées commence à décroître et la courbe du nombre de chroniques reconstituées commence à croître. La table 3.7 résume le choix des paramètres fait pour le reste de l'analyse des performances de CDIRe.

TABLE 3.7 – Tableau récapitulatif du choix des paramètres de DBSCAN pour chaque séquence temporelle.

Jeu de données	Phénomène	ε	$MinPts$
`blocks`	*pick-up*	25	5
	put-down	25	5
	stack	15	6
	unstack	15	6
	move-left	10	4
	move-right	10	4
	assemble	5	5
	disassemble	5	5

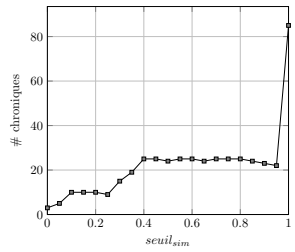
3.2.2 Seuil sur l'indice de Jaccard

Le paramètre $seuil_{sim}$ est la valeur minimale à laquelle l'indice de Jaccard entre deux événements doit se situer afin qu'une opération soit définie entre ces événements (cf. section 2.4.1, page 52). Plus ce paramètre est faible, plus des opérations entre des événements jugés différents seront permises, ce qui augmente le nombre

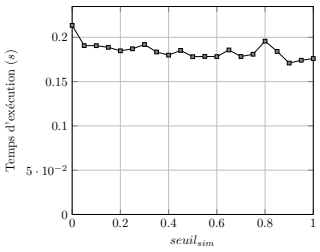
d'opérations à appliquer dans l'étape de reconstitution. Dans le reste de cette section, ce paramètre est évalué sur le jeu de données `blocks`.

Pour chacune des huit séquences temporelles de ce jeu de données, CDIRe est exécuté pour plusieurs valeurs du paramètre $seuil_{sim}$. Les valeurs du paramètre de l'algorithme de partitionnement des données DBSCAN varient suivant la séquence temporelle considérée et sont définies suivant les résultats de leur analyse fournie dans la section précédente. Ces valeurs sont répertoriées dans la table 3.7. Les résultats sont évalués pour des valeurs de $seuil_{sim}$ variant de 0 à 1 par pas de 0.05 pour un total de vingt points de mesure. Trois mesures de performances sont utilisées pour évaluer les performances, le nombre de chroniques reconstituées, le temps d'exécution de l'algorithme et l'AUC. L'ensemble des points de mesure utilisées pour l'évaluation du paramètre $seuil_{sim}$ sont fournis dans l'annexe B.2.1.

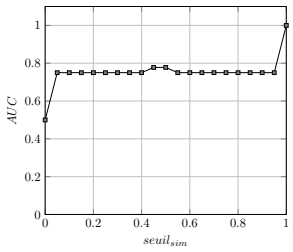
Similairement à l'analyse sur les paramètres de DBSCAN donnée dans la section précédente, les résultats sont détaillés pour la séquence temporelle associée au phénomène *pick-up*, puis abordés globalement sur le reste du jeu de données `blocks`.



(a) Nombre de chroniques reconstituées.



(b) Temps d'exécution.



(c) AUC.

FIGURE 3.10 – Résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *pick-up*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 25$.

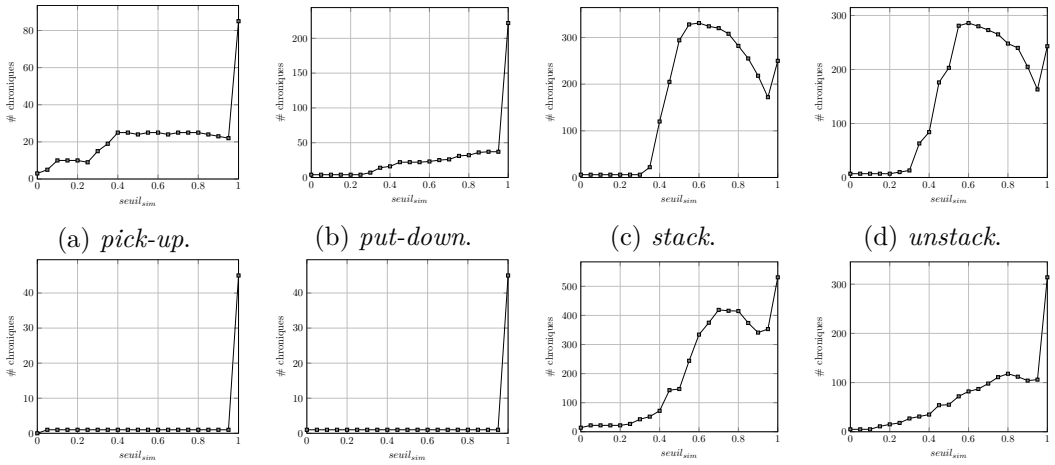
Le nombre de chroniques reconstituées en fonction du paramètre $seuil_{sim}$ est représenté sur la figure 3.10a. Cette courbe révèle une croissance entre le nombre de chroniques reconstituées et le paramètre étudié. Ce résultat est compréhensible car plus le $seuil_{sim}$ est faible, plus il y a d'opérations permises. Ce qui signifie qu'une reconstitution des chroniques plus importante est produite. Néanmoins, ce résultat n'est pas nécessairement désirable. En effet, un $seuil_{sim}$ trop permissif engendre des opérations entre des événements qui peuvent être trop différents pour produire un résultat cohérent. De plus, un pic est présent lorsque le paramètre étudié est égal à 1. Très peu d'opérations sont permises dans ce cas, le nombre de chroniques reconstituées est en conséquence restreint. Considérer le $seuil_{sim}$ à 1 est trop strict en tenant compte de la nature de l'étape d'identification de chroniques élémentaires. En effet, du bruit peut se glisser dans les groupements offerts par l'algorithme de partitionnement des données utilisé. En ne prenant pas en compte ce bruit, l'efficacité de la reconstitution de chroniques est limitée. Le nombre minimal de chroniques reconstituées est de 3 pour $seuil_{sim} = 0$ et le nombre maximal de

chroniques reconstituées est de 85 pour $seuil_{sim} = 1$.

Le temps d'exécution de CDIRe est représenté sur la figure 3.10b, il décroit légèrement en fonction du $seuil_{sim}$. Néanmoins, cette décroissance est faible et peut ne pas être significative. Ce résultat est cohérent avec le fonctionnement de l'étape de reconstitution de chroniques. En effet, plus le paramètre $seuil_{sim}$ est faible, plus le nombre d'opérations devant être appliquées est important, ce qui impacte négativement le temps d'exécution.

L'AUC, représenté sur la figure 3.10c, n'est pas affecté par le paramètre $seuil_{sim}$. Néanmoins, un pic avec la valeur de ce paramètre à 1 est visible. Ce pic confirme les observations constatées lors du pic présent sur la figure 3.10a. En effet, des chroniques élémentaires peuvent se glisser dans les chroniques reconstituées en raison du faible nombre d'opérations admissibles, ce qui peut fausser le calcul de l'AUC.

Le reste de cette section analyse de manière globale les résultats de CDIRe pour chaque séquence temporelle du jeu de données `blocks`. Les figures 3.11, 3.12 et 3.13 présentent respectivement le nombre de chroniques reconstituées, le temps d'exécution et l'AUC pour chacune de ces séquences temporelles.



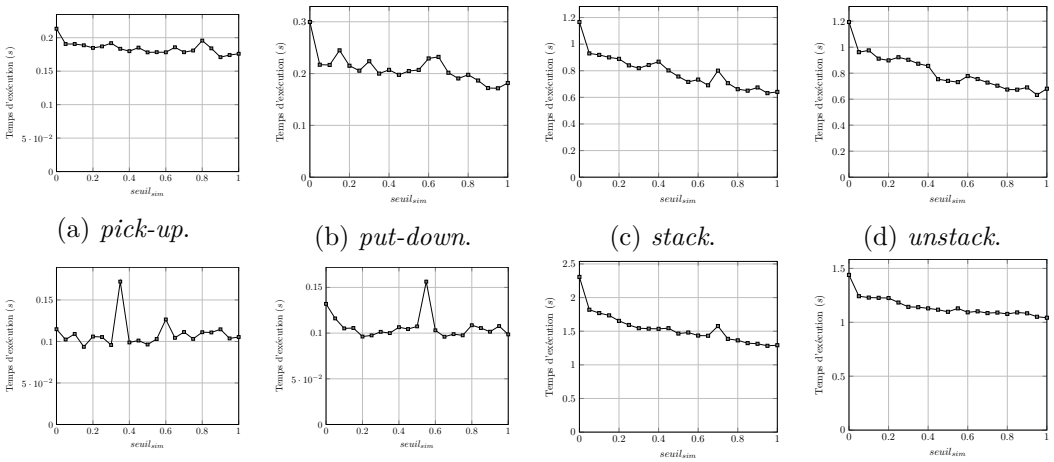
(a) *pick-up.* (b) *put-down.* (c) *stack.* (d) *unstack.*

(e) *move-left.* (f) *move-right.* (g) *assemble.* (h) *disassemble.*

FIGURE 3.11 – Nombre de chroniques reconstituées par CDIRe avec plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7.

Sur la figure 3.11, les nombres de chroniques reconstituées par CDIRe en fonction du paramètre $seuil_{sim}$ par chaque séquence temporelle du jeu de données `blocks` sont présentés. Sur les figures présentant les résultats de *stack* et *unstack*, le nombre de chroniques reconstituées croit brusquement aux alentours du $seuil_{sim}$ égal à 0.4 puis décroît lorsque celui est égal à 0.7. L'ordre des opérations imposé peut être une des explications de ce pic. En effet, plus d'opérations signifie que l'ordre a une plus grande importance dans l'étape de reconstitution de chroniques. Une opération qui était appliquée avec le $seuil_{sim} = 0.95$ peut ne pas être appliquée avec le $seuil_{sim} = 0.9$ car une opération qui n'était pas permise précédemment peut la

rendre incohérente ; ce qui a comme conséquence que bien que le paramètre $seuil_{sim}$ soit plus permissif, moins d'opérations sont appliquées. Le même phénomène peut être observé sur les courbes des figures 3.11g et 3.11h. De plus, les courbes représentant les résultats des séquences temporelles associées aux phénomènes *move-left* et *move-right* confirment l'hypothèse émise dans la section précédente. Le phénomène sous-jacent est suffisamment simple par rapport à la fréquence de celui-ci pour qu'il n'y ait pas d'ambiguïté possible pour CDIRe.



(a) *pick-up.* (b) *put-down.* (c) *stack.* (d) *unstack.*

(e) *move-left.* (f) *move-right.* (g) *assemble.* (h) *disassemble.*

FIGURE 3.12 – Résultat du temps d'exécution pour CDIRe avec plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. Les valeurs des paramètres de DBSCAN sont définis dans la table 3.7.

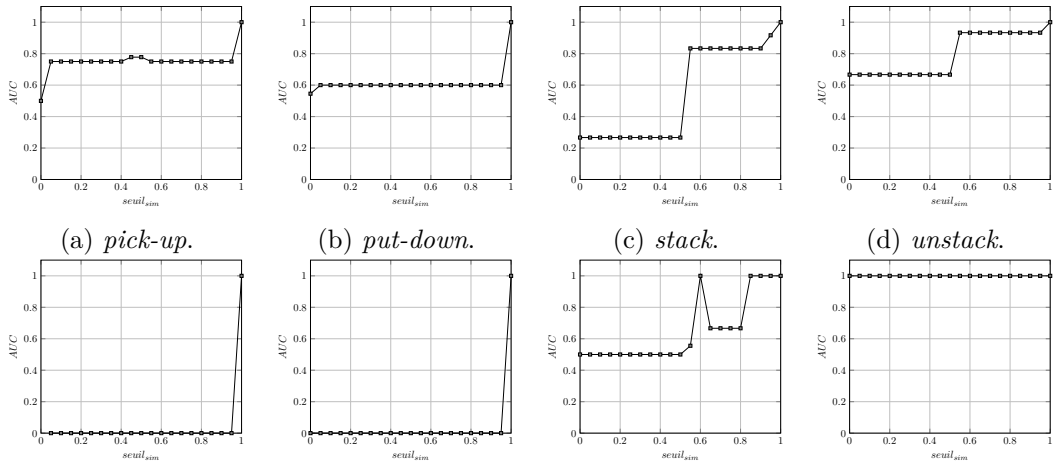
Le temps d'exécution de CDIRe en fonction du paramètre $seuil_{sim}$ est présenté sur la figure 3.12. Ces courbes confirment l'impact négatif du $seuil_{sim}$ sur le temps d'exécution. Ce phénomène est particulièrement visible sur les séquences temporelles les plus complexes (figures 3.12g et 3.12h).

La figure 3.13 qui présente les résultats du calcul de l'AUC pour les différentes valeurs du paramètre $seuil_{sim}$ montre qu'un meilleur résultat est obtenu pour des fortes valeurs de ce paramètre. Néanmoins, des chroniques élémentaires peuvent fausser le calcul de l'AUC.

Une étude plus approfondie doit être effectuée sur les chroniques reconstituées afin de définir quelle valeur du paramètre $seuil_{sim}$ semble la plus pertinente pour chaque séquence temporelle du jeu de données exploité. Cette analyse est faite en détail dans la section 3.3.2.

3.3 Analyse des résultats

Dans la section précédente, des mesures de performances sur les résultats de CDIRe ont été utilisées afin d'établir une analyse des différents paramètres de CDIRe. Malheureusement, ces mesures de performances ne suffisent pas pour dé-



(a) *pick-up.* (b) *put-down.* (c) *stack.* (d) *unstack.*

(e) *move-left.* (f) *move-right.* (g) *assemble.* (h) *disassemble.*

FIGURE 3.13 – Résultat de l'AUC pour CDIRe avec plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7.

terminer l'efficacité de cet algorithme. En effet, l'objectif recherché est de découvrir une ou plusieurs chroniques descriptives du phénomène sous-jacent aux données d'entrée. Ainsi, seules quelques chroniques sont considérées pour chaque résultat et non l'ensemble des chroniques reconstituées.

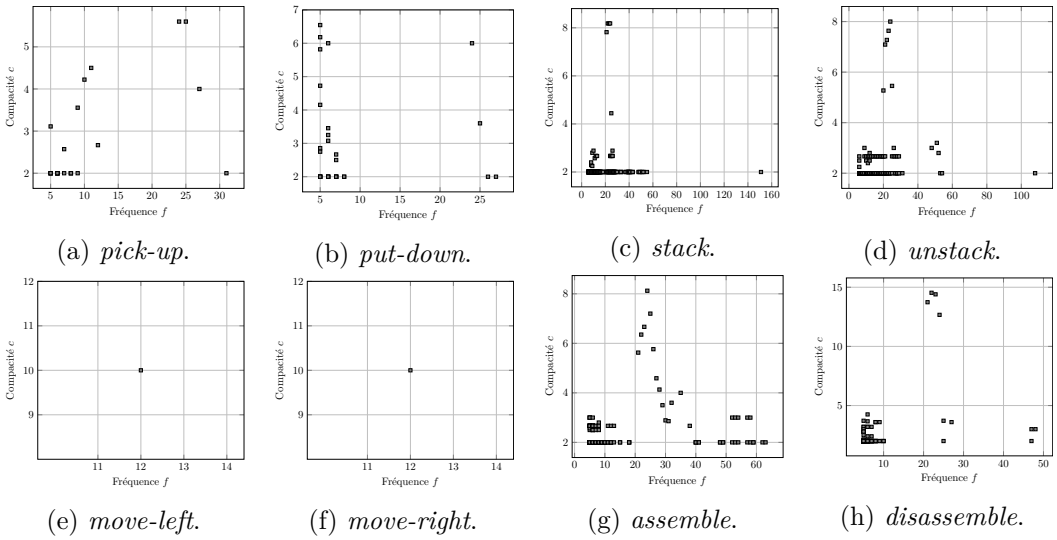
Cette section présente une analyse plus approfondie des chroniques obtenues dans les différentes exécutions de CDIRe effectuées dans l'étude précédente. Une étude sur les mesures de la qualité des chroniques, en particulier la compacité, est donnée et offre quelques directions dans le choix des chroniques pouvant être pertinentes pour l'utilisateur. Puis, une nouvelle analyse de certains paramètres, le seuil sur l'indice de Jaccard et la méthode de calcul des bornes de la contrainte temporelle, est fournie en prenant en compte ces mesures sur les chroniques.

3.3.1 Étude de la compacité

Comme décrit précédemment, la compacité est une mesure de la qualité des chroniques ad hoc pour l'analyse des résultats de CDIRe. Cette mesure est particulièrement bien adaptée pour distinguer les chroniques nécessitant une analyse approfondie par l'utilisateur. Cette section présente une étude approfondie de cette mesure sur les chroniques reconstituées par CDIRe avec le jeu de données `blocks`.

Les chroniques analysées sont fournies par CDIRe pour chacune des huit séquences temporelles de ce jeu de données. Pour obtenir ces résultats, les paramètres de DBSCAN sont répertoriés dans la table 3.7 et le $seuil_{sim}$ est fixé à 0.8. Les mesures de performances, comme le nombre de chroniques reconstituées, de CDIRe pour ces paramètres sont répertoriées dans la section 3.2.2 et dans l'annexe B.2.1. Sur la figure 3.14, les chroniques reconstituées sont tracées suivant leur compacité en fonction de leur fréquence. L'ensemble des mesures calculées pour les chroniques

reconstituées est répertorié dans l'annexe B.3.



(a) *pick-up.* (b) *put-down.* (c) *stack.* (d) *unstack.*

(e) *move-left.* (f) *move-right.* (g) *assemble.* (h) *disassemble.*

FIGURE 3.14 – Résultats de la compacité en fonction de la fréquence des chroniques reconstituées par CDIRe pour chaque séquence temporelle du jeu de données `blocks`. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7 et le $seuil_{sim} = 0.8$.

Sur les tracés de la compacité en fonction de la fréquence des chroniques reconstituées, un ensemble de chroniques avec une compacité importante se détachent lorsque la fréquence est aux environs de 24. En raison de ce détachement, cet ensemble de chroniques mérite une analyse plus complète par l'utilisateur ou un expert du système sous-jacent. Une analyse sur la durée de reconnaissance des chroniques, ou encore les connaissances a priori de l'utilisateur peuvent permettre d'isoler plus précisément les chroniques d'intérêt. Connaissant les différents phénomènes sous-jacents aux données d'entrée ainsi que leur fréquence, ces résultats sont prometteurs. En effet, comme décrit dans la section 3.1.3, le jeu de données exploité contient des séquences temporelles étiquetées et les résultats, et en particulier, leurs fréquences peuvent être vérifiées sur les données d'entrée. Dans le cas du jeu de données `blocks`, les différents pics visibles sur les tracés de la figure 3.14 concordent avec le nombre de séquences temporelles du phénomène d'intérêt de l'ensemble d'entraînement défini dans la table 3.5.

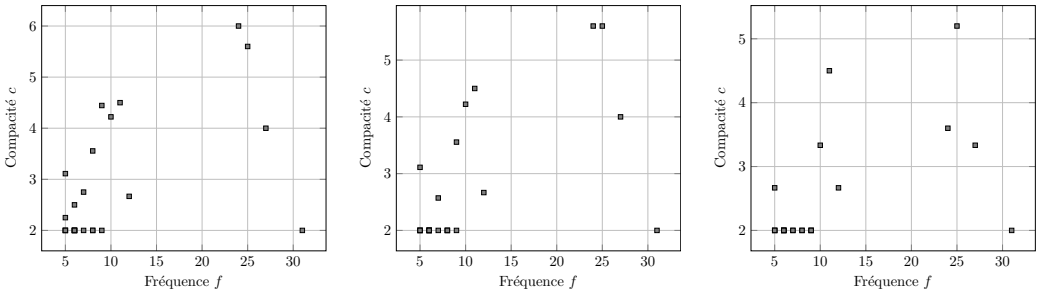
Une analyse des chroniques appartenant aux ensembles de chroniques ayant une bonne compacité et une fréquence voisine de 24 montrent que celles-ci sont proches dans les différents mesures de leurs qualités définies (cf. annexe B.3 pour les détails de ces mesures). Cet ensemble de chroniques peut consister en plusieurs chroniques décrivant une partie ou le tout d'un même phénomène d'intérêt. Ceci est dû au processus de l'étape de reconstitution de chroniques décrit dans la section 2.3 (page 50). En effet, la reconstitution est réalisée pour chaque fréquence présente dans les chroniques élémentaires, et en raison du seuil sur l'indice de Jaccard, les différentes

chroniques élémentaires peuvent fréquemment se retrouver plusieurs fois dans le processus de reconstitution. Par exemple, une chronique élémentaire de fréquence 25 possède une probabilité non négligeable que celle-ci se retrouve dans le processus de reconstitution à la fréquence 25 et à la fréquence 24.

Sur la figure 3.14b représentant les résultats des chroniques reconstituées par CDIRe avec pour données d'entrée la séquence temporelle associée au phénomène *put-down*, un pic de chroniques possédant une compacité importante est visible. Une analyse plus approfondie de ces chroniques montre que leurs durées de reconnaissance ne sont pas sur la même granularité de temps que le phénomène sous-jacent aux données d'entrée. Les résultats sur les séquences temporelles associées aux phénomènes *move-left* et *move-right* (figures 3.14e et 3.14f) montrent qu'il n'y a qu'une seule chronique reconstituée et que celle-ci est pertinente car elle possède une compacité importante. De plus, sa fréquence est compatible au nombre d'occurrences du phénomène sous-jacent.

3.3.2 Influence du paramètre du seuil sur l'indice de Jaccard

Dans la section 3.2.2, une première analyse sur le paramètre $seuil_{sim}$ est donnée. Ce paramètre influe grandement sur le nombre de chroniques reconstituées et requiert une étude plus approfondie de ce paramètre, en particulier de son influence sur les différentes mesures de la qualité des chroniques en elles-mêmes.



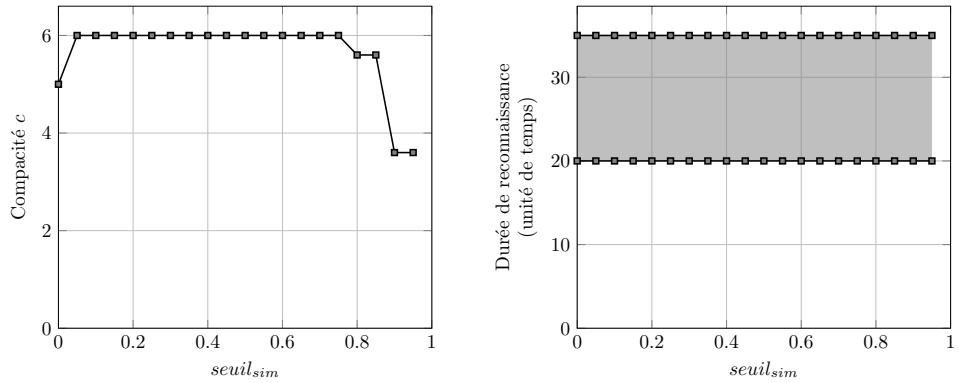
(a) $seuil_{sim} = 0.7$. (b) $seuil_{sim} = 0.8$. (c) $seuil_{sim} = 0.9$.

FIGURE 3.15 – Résultats de la compacité et de la fréquence des chroniques reconstituées par CDIRe pour quelques valeurs du paramètre $seuil_{sim}$ et avec la séquence temporelle associée au phénomène *pick-up*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 25$.

La figure 3.15 montre plusieurs tracés de la compacité en fonction de la fréquence des chroniques reconstituées pour plusieurs valeurs du paramètre $seuil_{sim}$. Outre la conclusion qu'il y a un plus grand nombre de chroniques reconstituées lorsque ce paramètre décroit, il semble que celles-ci ont globalement une compacité plus importante lorsque le $seuil_{sim}$ est plus faible. Ce phénomène est particulièrement visible sur la chronique ayant une fréquence de 24 qui possède une compacité de 6, 5.6 et 3.6 lorsque le $seuil_{sim}$ est fixé à, respectivement, 0.7, 0.8 et 0.9.

Une étude plus approfondie de ce phénomène est donnée dans le reste de cette

section. Pour cela, la chronique la plus descriptive du phénomène sous-jacent aux données d'entrée considéré est identifiée. Et cela pour chacun des vingt points de mesure des huit séquences temporelles étudiées dans la section 3.2.2. Les mesures de la qualité des chroniques sélectionnées sont données dans l'annexe B.2.2.



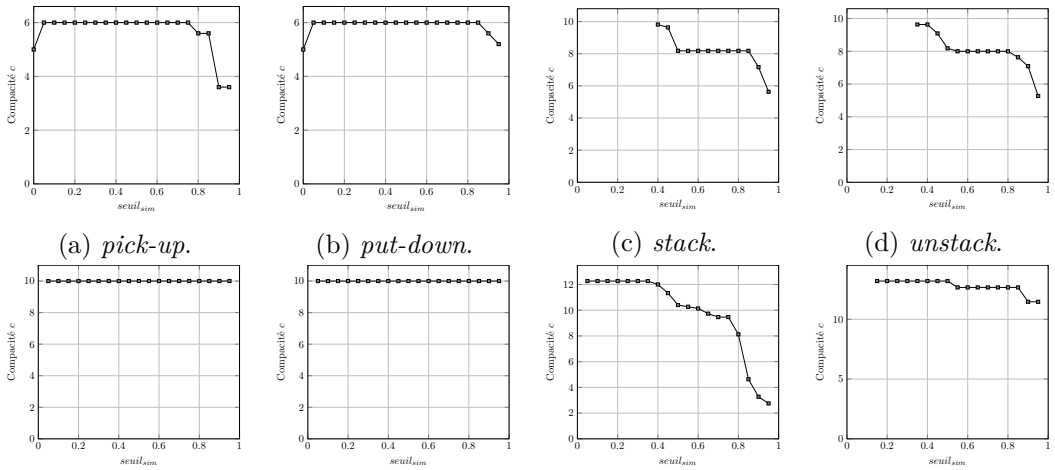
(a) Compacité. (b) Durée de reconnaissance.

FIGURE 3.16 – Résultats des mesures de la qualité de la chronique la plus descriptive du phénomène *pick-up* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 25$.

La figure 3.16 présente la compacité et la durée de reconnaissance en fonction du paramètre $seuil_{sim}$. La compacité est à son maximum dès que $seuil_{sim}$ est en dessous de 0.75 et décroit au-dessus. De plus, la compacité chute lorsque $seuil_{sim}$ est à 0. La durée de reconnaissance ne varie pas en fonction du $seuil_{sim}$, la durée minimale est à 20 unités de temps et la durée maximale est à 35.

Les résultats de la compacité des chroniques les plus descriptives des divers phénomènes d'intérêt du jeu de données `blocks` pour plusieurs valeurs du paramètre $seuil_{sim}$ sont tracés sur la figure 3.17. L'observation que les chroniques reconstituées ont une compacité plus importante lorsque le $seuil_{sim}$ décroit semble se confirmer sur ces courbes. Ce phénomène étant particulièrement visible sur la courbe présentant les chroniques les plus descriptives du phénomène *assemble* (cf. figure 3.17g). Cet effet est parfaitement explicable par le fonctionnement de l'étape de reconstitution des chroniques. En effet, lorsque le $seuil_{sim}$ est plus permissif, un plus grand nombre d'opérations est effectué, les chroniques subissent une reconstitution plus importante, ce qui augmente la compacité de celles-ci. De plus, la compacité atteint un plateau où diminuer le $seuil_{sim}$ n'améliore plus cette mesure.

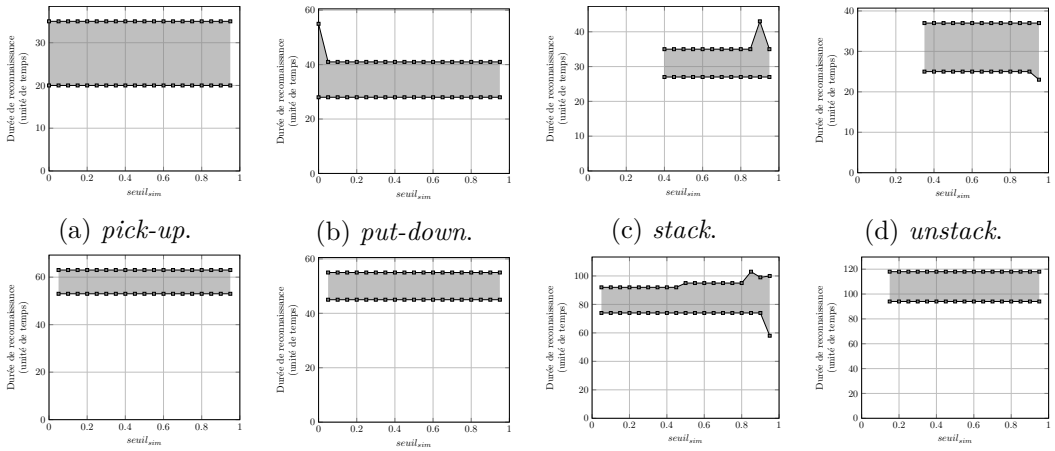
Ainsi, afin d'obtenir les résultats les plus performants, le $seuil_{sim}$ devrait être choisi sur ce plateau où la compacité ne varie pas. Néanmoins, une valeur trop faible ne devrait pas être choisie pour ce paramètre. En effet, rendre le $seuil_{sim}$ trop permissif peut ne pas permettre de générer une chronique descriptive du phénomène sous-jacent aux données d'entrée. Par exemple, aucune chronique descriptive des phénomènes *stack* et *unstack* ne sont générées lorsque le $seuil_{sim}$ est inférieur à 0.35. Le choix du paramètre $seuil_{sim}$ doit faire l'objet d'une attention particulière.



(a) *pick-up*. (b) *put-down*. (c) *stack*. (d) *unstack*.

(e) *move-left*. (f) *move-right*. (g) *assemble*. (h) *disassemble*.

FIGURE 3.17 – Résultats de la compacité des chroniques les plus descriptives des divers phénomènes du jeu de données `blocks` pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7.



(a) *pick-up*. (b) *put-down*. (c) *stack*. (d) *unstack*.

(e) *move-left*. (f) *move-right*. (g) *assemble*. (h) *disassemble*.

FIGURE 3.18 – Résultats de la durée de reconnaissance des chroniques les plus descriptives des divers phénomènes du jeu de données `blocks` pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7.

Les mesures de la durée de reconnaissance, qui répertorient la durée minimale et la durée maximale, pour les chroniques les plus descriptives des phénomènes étudiés sont représentées sur la figure 3.18. La durée de reconnaissance est moins affectée par le $seuil_{sim}$ que la compacité. Néanmoins, dans certain cas, comme sur la figure 3.18g, la durée de reconnaissance est plus stricte lorsque le $seuil_{sim}$ décroit. Elle présente le même comportement que la compacité. Ce résultat est cohérent

car lorsque la compacité est importante, le nombre de contraintes temporelles l'est également, ce qui signifie qu'il y a plus de contraintes sur la durée de reconnaissance.

Ainsi, comme vu dans cette section, le paramètre $seuil_{sim}$ n'influe pas seulement sur les performances de CDIRe mais également sur la qualité des chroniques reconstituées. Afin d'obtenir les résultats les plus pertinents pour l'utilisateur, ce paramètre doit être choisi avec une attention toute particulière. Avec le jeu de données exploité dans cette analyse, les paramètres choisis permettant d'obtenir les résultats les plus pertinents pour décrire les phénomènes d'intérêt sont répertoriés dans la table 3.8.

TABLE 3.8 – Tableau récapitulatif du choix des paramètres de CDIRe pour chaque séquence temporelle permettant de reconstituer la chronique la plus descriptive du phénomène sous-jacent.

Jeu de données	Phénomène	ε	$MinPts$	$seuil_{sim}$
`blocks`	*pick-up*	25	5	0.75
	put-down	25	5	0.85
	stack	15	6	0.4
	unstack	15	6	0.4
	move-left	10	4	0.95
	move-right	10	4	0.95
	assemble	5	5	0.35
	disassemble	5	5	0.5

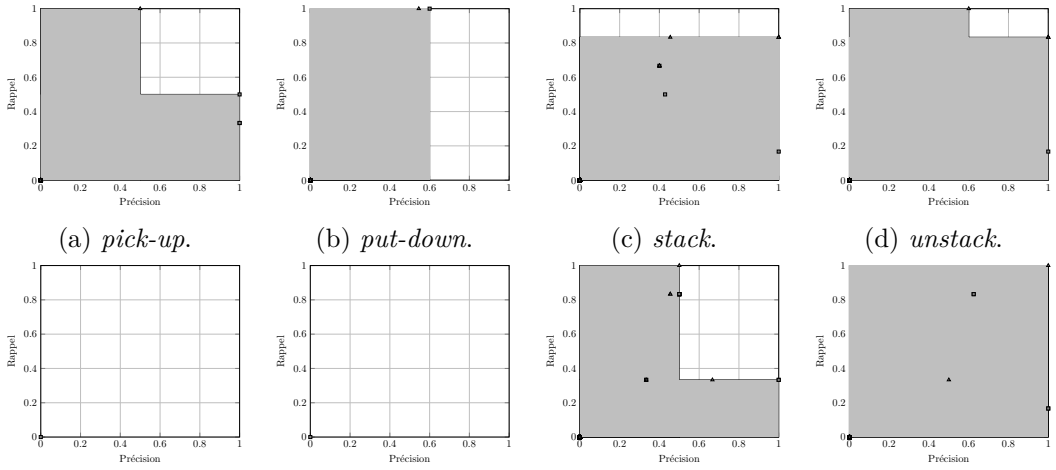
3.3.3 Méthode de calcul des bornes de la contrainte temporelle et l'*AUC* comme mesure de performance

Dans cette section, les différentes méthodes de calcul des bornes de la contrainte temporelle sont abordées. Comme décrit dans la section 2.2.3 (page 48), ce paramètre influe sur la façon dont les intervalles sont calculés sur les différentes contraintes temporelles des chroniques élémentaires. Ce paramètre ne permet pas d'affecter la quantité de chroniques reconstituées par CDIRe mais influe sur la qualité de celles-ci. De plus, l'*AUC* ainsi que sa pertinence comme mesure de performance de CDIRe sont abordées.

TABLE 3.9 – Tableau récapitulatif des résultats du calcul de l'*AUC* pour CDIRe pour chaque méthode de calcul des bornes des contraintes temporelles. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7 et le $seuil_{sim} = 0.8$.

Jeu de données	Phénomène	*AUC*		
		`minmax`	`2sigma`	`3sigma`
`blocks`	*pick-up*	0.75	0.75	0.75
	put-down	0.6	1	1
	stack	0.83	0.83	0.83
	unstack	0.93	0.93	0.93
	move-left	0	0	0
	move-right	0	0	0
	assemble	0.66	0.66	0.66
	disassemble	1	1	1

Malheureusement, sur les résultats répertoriés sur la table 3.9, les différentes méthodes de calcul des bornes de la contrainte temporelle ne donnent pas d'amélioration sur l'AUC, excepté sur les données d'entrée liées au phénomène *put-down*, où les méthodes `2sigma` et `3sigma` améliorent significativement les résultats. Ces résultats peuvent être liés à deux phénomènes abordés dans le reste de cette section.



(a) *pick-up*. (b) *put-down*. (c) *stack*. (d) *unstack*.

(e) *move-left*. (f) *move-right*. (g) *assemble*. (h) *disassemble*.

FIGURE 3.19 – Résultats des courbes PR pour CDIRe pour chaque séquence temporelle du jeu de données `blocks`. Chaque chronique est représentée soit par un triangle si celle-ci est une chronique élémentaire, soit par un carré sinon. Les valeurs des paramètres de DBSCAN sont définies dans la table 3.7, le $seuil_{sim} = 0.8$ et la méthode de calcul des bornes de la contrainte temporelle utilisée est `minmax`.

Le premier phénomène remettant en cause la pertinence de l'AUC comme mesure de performance pour CDIRe est lié aux différentes chroniques élémentaires identifiées. Sur la figure 3.19, les courbes PR des différentes séquences temporelles du jeu de données `blocks` sont représentées lorsque la méthode de calcul des bornes de la contrainte temporelle utilisée est `minmax`. Les différents points représentent les chroniques générées par CDIRe. Les chroniques élémentaires sont représentées par un triangle, alors que les autres sont représentées par un carré. Il semble que sur la plupart des courbes PR les chroniques élémentaires améliorent grandement le résultat du calcul de l'AUC. En effet, plusieurs chroniques élémentaires représentant du bruit présent dans les données d'entrée influencent les résultats de l'AUC, donnant de manière trompeuse de meilleurs résultats que les résultats réels.

Le deuxième phénomène rentrant en jeu dans le contexte de cette étude est le sur-apprentissage. En effet, les chroniques reconstituées peuvent être relativement complexes par rapport à la taille des données. Par exemple, la chronique descriptive du phénomène *disassemble* possède 16 événements et 99 contraintes temporelles pour une séquence temporelle d'entrée de seulement 388 événements datés. La taille des données d'entrée est simplement trop faible pour obtenir 99 intervalles suffisamment généraux pour éviter le phénomène de sur-apprentissage. Ce phénomène de

sur-apprentissage reste un problème à résoudre dans des travaux futurs.

3.4 Bilan sur les résultats de l'analyse des performances

Dans les sections précédentes, une analyse des performances de CDIRe et des chroniques reconstituées est fournie. Cette analyse offre des directions à l'utilisateur pour le choix des paramètres de CDIRe d'une part et le choix des chroniques les plus descriptives d'autre part. En effet, la table 3.8 apporte les valeurs des différents paramètres de CDIRe afin d'obtenir les résultats les plus pertinents pour le jeu de données exploité. Cette section présente les chroniques les plus descriptives du jeu de données `blocks` d'une part et dresse un bilan des influences des paramètres de CDIRe d'autre part.

3.4.1 Chroniques descriptives du jeu de données exploité

Dans cette section, les chroniques les plus descriptives des phénomènes d'intérêts générées par CDIRe sont détaillées pour les phénomènes *pick-up* et *put-down*. De plus, une étude sur les mesures de performances des chroniques les plus descriptives de chaque phénomène du jeu de données `blocks` est fournie.

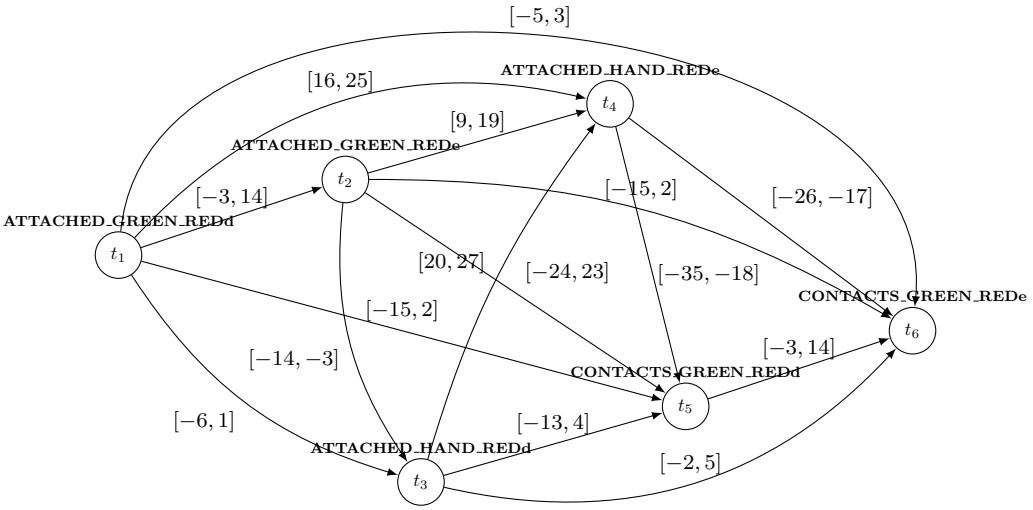


FIGURE 3.20 – La chronique la plus descriptive du phénomène *pick-up*. C'est une chronique de taille 6 notée $\mathcal{C}_{pick\text{-}up} = \{\mathcal{E}_{pick\text{-}up}, \mathcal{T}_{pick\text{-}up}\}$ avec 15 contraintes temporelles.

La chronique la plus descriptive du phénomène *pick-up* est représentée graphiquement sur la figure 3.20. C'est une chronique notée $\mathcal{C}_{pick\text{-}up} = \{\mathcal{E}_{pick\text{-}up}, \mathcal{T}_{pick\text{-}up}\}$. Elle possède six événements :

$$\mathcal{E}_{pick\text{-}up} = \{e_1 = \text{ATTACHED\_\_GREEN\_\_RED}d, e_2 = \text{ATTACHED\_\_GREEN\_\_RED}e,$$
$$e_3 = \text{ATTACHED\_\_HAND\_\_RED}d, e_4 = \text{ATTACHED\_\_HAND\_\_RED}e,$$
$$e_5 = \text{CONTACTS\_\_GREEN\_\_RED}d, e_6 = \text{CONTACTS\_\_GREEN\_\_RED}e\}.$$

De plus, quinze contraintes temporelles sont imposées sur ces événements :

$$\mathcal{T}_{pick\text{-}up} = \{\tau_{(1,2)} = e_1[-3,14]e_2, \tau_{(1,3)} = e_1[-6,1]e_3, \tau_{(1,4)} = e_1[16,25]e_4,$$
$$\tau_{(1,5)} = e_1[-15,2]e_5, \tau_{(1,6)} = e_1[-5,3]e_6, \tau_{(2,3)} = e_2[-14,-3]e_3,$$
$$\tau_{(2,4)} = e_2[9,19]e_4, \tau_{(2,5)} = e_2[-24,23]e_5, \tau_{(2,6)} = e_2[-15,2]e_6,$$
$$\tau_{(3,4)} = e_3[20,27]e_4, \tau_{(3,5)} = e_3[-13,4]e_5, \tau_{(3,6)} = e_3[-2,5]e_6,$$
$$\tau_{(4,5)} = e_4[-35,-18]e_5, \tau_{(4,6)} = e_4[-26,-17]e_6, \tau_{(5,6)} = e_4[-3,14]e_6\}.$$

Au début de cette chronique, un bloc rouge est posé sur un bloc vert (événements e_5 et e_6). Puis, une main attrape (événements e_1 et e_3) et soulève le bloc rouge (événement e_2). Enfin, la main sort du cadre de la vidéo (événement e_4). La durée de reconnaissance de cette chronique se déroule entre 20 et 35 unités de temps. En prenant en compte la durée d'échantillonnage des séquences temporelles de $30Hz$, le temps d'exécution de ce phénomène dure entre 0.66 et 1.16 secondes et est cohérent vis-à-vis du type de phénomène considéré.

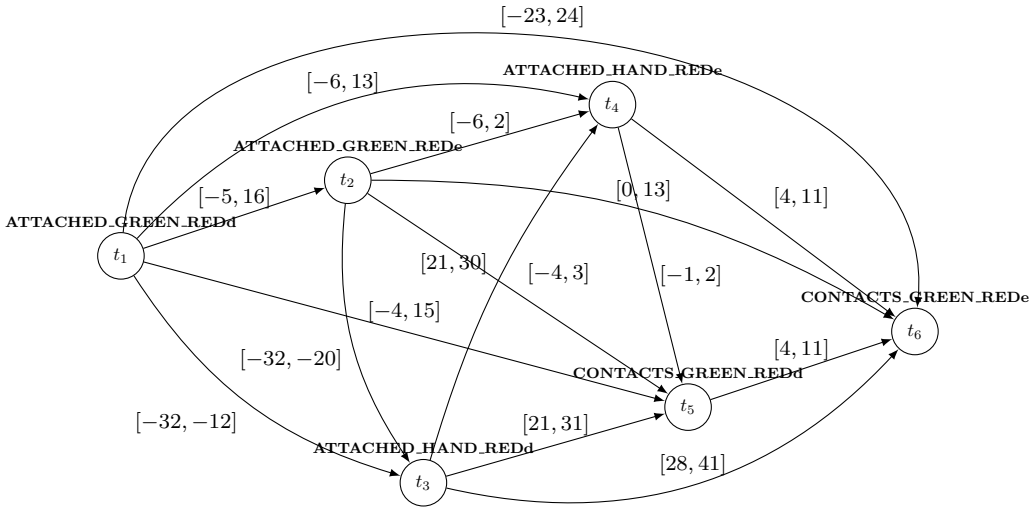


FIGURE 3.21 – La chronique la plus descriptive du phénomène *put-down*. C'est une chronique de taille 6 notée $\mathcal{C}_{put\text{-}down} = \{\mathcal{E}_{put\text{-}down}, \mathcal{T}_{put\text{-}down}\}$ avec 15 contraintes temporelles.

La chronique descriptive du phénomène opposé à *pick-up*, qui est *put-down*, est représentée graphiquement sur la figure 3.21. C'est une chronique notée $\mathcal{C}_{put\text{-}down} = \{\mathcal{E}_{put\text{-}down}, \mathcal{T}_{put\text{-}down}\}$. Similairement à $\mathcal{C}_{pick\text{-}up}$, elle possède le même ensemble d'événements :

$$\mathcal{E}_{put\text{-}down} = \{e_1 = \text{ATTACHED\_\_GREEN\_\_RED}d, e_2 = \text{ATTACHED\_\_GREEN\_\_RED}e,$$
$$e_3 = \text{ATTACHED\_\_HAND\_\_RED}d, e_4 = \text{ATTACHED\_\_HAND\_\_RED}e,$$
$$e_5 = \text{CONTACTS\_\_GREEN\_\_RED}d, e_6 = \text{CONTACTS\_\_GREEN\_\_RED}e\}.$$

Néanmoins, ses quinze contraintes temporelles sont sur des intervalles différents :

$$\mathcal{T}_{put\text{-}down} = \{\tau_{(1,2)} = e_1[-5,16]e_2, \tau_{(1,3)} = e_1[-32,-12]e_3, \tau_{(1,4)} = e_1[-6,13]e_4,$$
$$\tau_{(1,5)} = e_1[-4,15]e_5, \tau_{(1,6)} = e_1[-23,24]e_6, \tau_{(2,3)} = e_2[-32,-20]e_3,$$
$$\tau_{(2,4)} = e_2[-6,2]e_4, \tau_{(2,5)} = e_2[-4,3]e_5, \tau_{(2,6)} = e_2[0,13]e_6,$$
$$\tau_{(3,4)} = e_3[21,30]e_4, \tau_{(3,5)} = e_3[21,31]e_5, \tau_{(3,6)} = e_3[28,41]e_6,$$
$$\tau_{(4,5)} = e_4[-1,2]e_5, \tau_{(4,6)} = e_4[4,11]e_6, \tau_{(5,6)} = e_4[4,11]e_6\}.$$

Contrairement à $\mathcal{C}_{pick\text{-}up}$, $\mathcal{C}_{put\text{-}down}$ commence par une main tenant un bloc rouge (événement e_3). Puis, cette main pose ce bloc rouge sur un bloc vert (événement e_1) et le lâche (événements e_4 et e_2). Enfin, sur le reste de la vidéo, le bloc rouge reste posé sur le bloc vert (événements e_5 et e_6). Quant à la durée de reconnaissance de cette chronique, elle est un peu plus longue que celle de $\mathcal{C}_{pick\text{-}up}$ et dure entre 28 et 41 unités de temps. Ce qui donne un temps d'exécution durant entre 0.93 et 1.36 secondes.

Les phénomènes *pick-up* et *put-down* étant les plus simples du jeu de données `blocks`, il est possible de les représenter mathématiquement et graphiquement de manière claire et lisible. Néanmoins, les phénomènes suivants étant d'une complexité croissante (par exemple, la chronique descriptive du phénomène *disassemble* possède 99 contraintes temporelles), il est difficile de les représenter clairement. C'est pourquoi, seule une analyse sur les mesures de la qualité de ces chroniques est fournie. Ces mesures de qualité sont répertoriées dans la table 3.10.

TABLE 3.10 – Tableau récapitulatif des mesures de la qualité des chroniques les plus descriptives de chaque phénomène sous-jacent au jeu de données `blocks`. Les paramètres de CDIRe choisis pour chaque phénomène sont répertoriés dans la table 3.8.

Jeu de données	Phénomène	taille	compacité	fréquence	durée minimale	durée maximale
`blocks`	*pick-up*	6	6	24	20	35
	put-down	6	6	24	28	41
	stack	12	9.81819	24	27	35
	unstack	12	9.63637	24	25	37
	move-left	10	10	12	53	63
	move-right	10	10	12	45	55
	assemble	16	12.2667	24	74	92
	disassemble	16	13.2001	24	94	118

La compacité croît en fonction de la complexité du phénomène sous-jacent. En effet, plus un phénomène est complexe, plus le nombre d'événements nécessaire à sa description est grand, et en conséquence, le nombre de contraintes temporelles augmente de manière polynomiale. En effet, le nombre m de contraintes temporelles possibles d'une chronique est en fonction de la taille n d'une chronique suivant la formule : $m = \frac{n(n-1)}{2}$. La taille des données d'entrée doit être proportionnelle aux

nombre de contraintes temporelles afin d'éviter les phénomènes de sur-apprentissage abordés dans la section précédente. De plus, une chronique représentant un phénomène plus complexe possède une durée de reconnaissance plus longue. C'est parfaitement compréhensible car les vidéos des phénomènes complexes sont plus longues que les vidéos des phénomènes plus simples. En prenant en compte la fréquence d'échantillonnage des vidéos, la chronique la plus complexe, $\mathcal{C}_{disassemble}$, dure entre 3.13 et 3.93 secondes.

3.4.2 Bilan des influences des paramètres de CDIRe

Cette section dresse un bilan des influences des différents paramètres de CDIRe, que cela soit sur les mesures de performances de cet algorithme ou encore sur les mesures de qualité des chroniques reconstituées. Les paramètres étudiés sont : les paramètres de DBSCAN, ε et $MinPts$, le seuil sur l'indice de Jaccard $seuil_{sim}$ et la méthode de calcul des bornes de la contrainte temporelle (`minmax`, `2sigma` ou `3sigma`). L'influence de l'ordre des opérations sur les résultats ne fait pas partie de cette analyse. Les tables suivantes dressant une synthèse des influences des paramètres permettent d'aider l'utilisateur dans le choix des paramètres de CDIRe vis-à-vis du type de résultat qu'il souhaite obtenir.

Dans la table 3.11, les influences des paramètres de CDIRe sur les mesures de performances détaillées dans ce document sont répertoriées. La méthode de calcul des bornes de la contrainte temporelle influence seulement le résultat de l'AUC, ayant un meilleur résultat lorsque la méthode `3sigma` est utilisée. Le $seuil_{sim}$ n'influence pas le nombre de chroniques élémentaires identifiées, celui-ci n'intervenant pas dans l'étape d'identification de chroniques élémentaires. Enfin, l'influence des paramètres sur l'AUC n'est pas démontrée en raison des chroniques élémentaires parasites et du phénomène de sur-apprentissage constaté dans la section 3.3.3.

TABLE 3.11 – Tableau récapitulatif des influences des paramètres de CDIRe sur les mesures de performances.

	ε	$MinPts$	$seuil_{sim}$	Méthode de calcul des bornes
# chroniques élémentaires	proportionnel	inversement proportionnel	pas d'influence	pas d'influence
# chroniques reconstituées	inversement proportionnel	proportionnel	proportionnel, présente un pic	pas d'influence
Temps d'exécution	inversement proportionnel	proportionnel	inversement proportionnel	pas d'influence
AUC	dépend des données	dépend des données	dépend des données	dépend des données et de la méthode

La table 3.12 récapitule les influences des paramètres de CDIRe sur les mesures de qualité des chroniques reconstituées. La taille, la compacité et la durée

de reconnaissance dépendent des paramètres de DBSCAN. Il serait plus juste de dire que ces mesures sont dépendantes du nombre de chroniques élémentaires identifiées. Un nombre de chroniques élémentaires faible résulte dans des chroniques ayant une taille, une compacité et une durée de reconnaissance plus importante. De plus, le $seuil_{sim}$ doit être considéré avec une attention particulière car lorsqu'il est trop bas, les phénomènes sous-jacents peuvent ne pas être découverts. Enfin, la méthode de calcul des bornes de la contrainte temporelle n'influence que la durée de reconnaissance.

TABLE 3.12 – Tableau récapitulatif des influences des paramètres de CDIRe sur les mesures de qualité des chroniques reconstituées.

	ε	$MinPts$	$seuil_{sim}$	Méthode de calcul des bornes
Taille	proportionnel	inversement proportionnel	dépend des données	pas d'influence
Compacité	proportionnel	inversement proportionnel	inversement proportionnel	pas d'influence
Durée de reconnaissance	proportionnel	inversement proportionnel	plus large lorsque le $seuil_{sim}$ est proche de 1	plus large avec la méthode `3sigma`

3.5 Conclusion

Les performances et les résultats de CDIRe sont analysés dans ce chapitre. Tout d'abord, les mesures de performance et de qualité utilisées pour cette étude sont posées. Puis, une analyse des différents paramètres de CDIRe permet de confirmer les conclusions fournies dans le chapitre précédent quant aux performances de notre algorithme. Une analyse plus approfondie des chroniques générées par CDIRe est également établie. Enfin, un bilan des influences des paramètres sur les performances et les résultats donne un guide de conduite pour le choix des valeurs des paramètres aux utilisateurs de CDIRe suivant les résultats qu'ils souhaitent obtenir.

Malheureusement, aucune comparaison des résultats de CDIRe par rapport à d'autres algorithmes de découverte de chroniques significatifs quant aux inspirations de notre contribution n'est faite. En effet, pour des raisons diverses, ni HCDA ni FACE ne sont considérés dans l'analyse fournie dans ce chapitre. Une implémentation de HCDA est disponible [Cram 2009], mais celle-ci n'est plus maintenue et n'est pas fonctionnelle. Quant à FACE, celui-ci fait partie d'un logiciel propriétaire [Dousson 2008] utilisé par une grande entreprise de télécommunication française.

Chapitre 4

Projection de chroniques

Sommaire

Une des problématiques du processus de découverte de chroniques est la quantité importante de chroniques générées où une grande partie de celles-ci est inutilisable car représentant du bruit. Une étape d'analyse de ces chroniques est donc bien souvent indispensable. Malheureusement, les outils actuels fournis par l'analyse de chroniques sont inexistants ou inadaptés à la tâche de comparaison et de classification de chroniques.

Dans un espace euclidien, ces tâches deviennent aisées. En effet, de nombreux travaux traitent de comparaison entre points ou ensembles dans un espace euclidien. Établir une méthode mathématique permettant d'associer une chronique à un objet dans un espace euclidien ne peut que être bénéfique au processus de découverte de chroniques. Un goulet d'étranglement à un telle méthode est la difficulté de considérer à la fois le domaine spatial et temporel d'une chronique.

Dans ce chapitre, une telle méthode qui repose sur une technique de projection aléatoire communément utilisée dans le domaine de la réduction de dimension est proposée. Cette méthode appliquée aux chroniques y est démontrée mathématiquement. Les premiers résultats de ce travail sont prometteurs et ouvrent de nombreuses perspectives dans le contexte du processus de découverte de chroniques en général.

Ce chapitre est organisé comme suit. La section 4.1 présente un tour d'horizon de la méthode de projection aléatoire dans la littérature. La méthodologie de la projection aléatoire appliquée aux chroniques est posée dans la section 4.2. La section 4.3

présente quelques propriétés de la projection utiles à des fins de comparaison. La section 4.4 étudie le nombre k de dimensions et son influence sur la projection. Enfin, des pistes d'applications de la méthode de projection de chroniques dans un contexte de découverte de chroniques sont offertes dans la section 4.5.

4.1 État de l'art sur la méthode de projection aléatoire

La projection aléatoire [Vempala 2005] est une méthode de réduction de dimension. Le fondement de la réduction de dimension consiste à prendre des données dans un espace de grande dimension et de les remplacer par des données dans un espace de plus petite dimension et ce sans perte, ou avec une perte minimale d'information. Ce type de méthode permet d'accélérer le traitement des données, ce qui est particulièrement utile dans un contexte d'apprentissage automatique. Le théorème de Johnson-Lindenstrauss [Johnson 1984, Dasgupta 2003] est la pierre angulaire de la projection aléatoire. Ce théorème révèle que lorsque l'on projette n points d'un espace de haute dimension dans un hyper-plan aléatoire de $O(\log(n))$ dimensions, la probabilité de conserver toutes les distances entre les différents points, dans une certaine marge d'erreur, est significative.

Dans des domaines d'étude où seule la distance entre les données est importante, la projection aléatoire est particulièrement utile. Un exemple concret concerne la fouille de données, la projection aléatoire est comparable voir plus efficace en termes de complexité que l'analyse en composante principale [Bingham 2001] comme méthode de réduction de dimension. La projection aléatoire est également utilisée dans le domaine de l'extraction de connaissance. Entre autres, [Atkison 2009] exploite cette méthode en conjonction avec l'indice de similarité cosinus pour la détection de logiciels suspects.

La projection aléatoire est une méthode qui est utilisée pour projeter un modèle d'un espace dans lequel la comparaison est difficile en raison de l'absence de distances efficaces vers un espace euclidien où une simple distance euclidienne peut être aisément calculée. Dans [Mannila 2001], la projection aléatoire est appliquée sur des séquences temporelles. Une séquence temporelle est associée à un point dans un espace euclidien par l'utilisation d'une fonction aléatoire. Chaque événement est associé à un vecteur aléatoire propre et la fonction aléatoire est la somme des vecteurs aléatoires associés aux événements pondérés par leurs instants d'occurrence. Cette méthode est appliquée sur des données réelles et les résultats sont prometteurs.

Dans ce travail, l'étude des chroniques nécessite de comparer et d'analyser des chroniques. La projection aléatoire peut être utile pour les propriétés d'un espace euclidien, et en particulier l'aisance avec laquelle deux objets peuvent être comparés dans cet espace. Dans le reste de ce chapitre, cette méthode mathématique de projection aléatoire est développée pour le modèle des chroniques. Une partie de ce travail exploratoire est publiée dans [Sahuguède 2018a].

4.2 Méthodologie de la projection de chroniques dans un espace euclidien

Comme vu dans la section précédente, la projection aléatoire est une méthode communément utilisée pour le problème de réduction de dimension. Nous proposons dans ce chapitre d'adapter cette méthode au formalisme des chroniques. L'objectif est de mettre en place un système de projection de chroniques dans un espace euclidien où le calcul des distances entre points et ensembles est plus aisé que le calcul des distances sur le formalisme de base. Grâce à cette projection aléatoire dans cet espace euclidien k-dimensionnel, les chroniques sont représentées par un volume géométrique représentant toutes les occurrences possibles de celles-ci.

Dans un premier temps (section 4.2.1), seul le domaine temporel est considéré pour la projection de chroniques. Dans ce cas, une simple projection orthogonale suivant le temps possible des instants d'occurrence des nœuds de la chronique est possible.

Dans un second temps (section 4.2.2), à la fois les domaines temporel et spatial d'une chronique sont considérés à l'aide d'une projection aléatoire. Des hypothèses à la fois sur la taille de l'espace euclidien dans lequel les chroniques sont projetées et sur la structure même de celles-ci sont données afin de donner un cadre formel à cette méthode et permettre sa réalisation.

4.2.1 Projection de chroniques suivant le temps

Cette section montre qu'une chronique de taille n peut être représentée par un polytope convexe [1] non borné, un volume géométrique dans un espace euclidien de dimension n. Toute chronique peut être aisément associée à un volume géométrique par une simple projection orthogonale sur le temps. Par exemple, [Guyet 2008, Guyet 2011] exploitent une telle projection orthogonale sur des motifs temporels plus simples que les chroniques et où seule l'information temporelle est d'intérêt.

Quelques hypothèses sont différentes dans cette contribution. En effet, le domaine temporel $\mathcal{T}$ considéré ici est l'ensemble des réels $\mathcal{R}$. De plus, les occurrences d'une chronique ne sont pas liées à une séquence temporelle $\mathcal{S}$ particulière. La notation $\mathcal{S}$ est omise de la notation des occurrences $\mathcal{O}_\mathcal{C}$ d'une chronique $\mathcal{C}$.

Définition 4.1. Un **polytope** dans un espace euclidien $\mathbb{R}^n$ est l'intersection d'un nombre fini de demi-espaces délimités par des hyper-plans affines. Avec $A \in \mathbb{R}^{m\times n}$ et $b \in \mathbb{R}^m$, le polytope $\mathcal{P}$ est défini par :

$$\mathcal{P} = \{x \in \mathbb{R}^n | Ax \geq b\}. \tag{4.1}$$

Définition 4.2. La **projection d'une chronique** $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ de taille n dans l'espace euclidien $\mathbb{R}^n$ avec son ensemble de m contraintes temporelles $\mathcal{T}$ est définie par le polytope $\mathcal{P}$. Chaque dimension de $\mathbb{R}^n$ est associée à un instant d'occurrence

1. Dans le reste de ce chapitre, un polytope convexe est appelé polytope.

possible d'un nœud de $\mathcal{C}$. Le polytope $\mathcal{P}$ est défini par la formule suivante :

$$\mathcal{P} = \left\{ -t_i + t_j \geq t^-_{(i,j)},\ t_i - t_j \geq -t^+_{(i,j)} \mid \tau_{(i,j)} = e_i[t^-_{(i,j)}, t^+_{(i,j)}]e_j \in \mathcal{T} \right\}. \tag{4.2}$$

Soient $b \in \mathbb{R}^{2m}$ le vecteur contenant les bornes de toutes les contraintes temporelles, $A \in \mathbb{R}^{2m \times n}$ la matrice définissant les nœuds sur lesquels les contraintes temporelles prennent place et $t \in \mathbb{R}^n$ un vecteur d'instants d'occurrence des nœuds de $\mathcal{C}$. $\mathcal{P}$ peut être alors défini comme la contrainte linéaire suivante :

$$At \geq b. \tag{4.3}$$

Les solutions t de l'équation (4.3) correspondent aux instants d'occurrence des nœuds de toutes les occurrences possibles de la chronique $\mathcal{C}$ tels qu'il existe au moins une séquence temporelle $\mathcal{S}$. Comme le nombre de séquences temporelles est infini, il existe un nombre infini d'occurrences de $\mathcal{C}$. Ainsi, le polytope $\mathcal{P}$ sera toujours non borné [Toth 2017].

Exemple 4.1. Soit $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ une chronique de taille 3 avec $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 20]e_2, \tau_{(1,3)} = e_1[25, 35]e_3, \tau_{(2,3)} = e_2[10, 20]e_3\}$ représentée sur la figure 4.1. Cette chronique $\mathcal{C}$ définit le polytope $\mathcal{P}$ représenté sur la figure 4.2. Elle caractérise la contrainte linaire suivante où $A \in \mathbb{R}^{6\times3}$, $t \in \mathbb{R}^3$ et $b \in \mathbb{R}^6$:

$$At \geq b,$$
$$\begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} t \geq \begin{bmatrix} 10 \\ -20 \\ 25 \\ -35 \\ 10 \\ -20 \end{bmatrix}.$$

Une solution possible de cette contrainte linéaire et représentée par un carré noir sur la figure 4.2 est la suivante :

$$t = \begin{bmatrix} 8 & 21 & 33 \end{bmatrix}^\top.$$

Cette solution correspond à la projection de $o_{\mathcal{C}} = \{(a, 8), (b, 21), (c, 33)\}$.

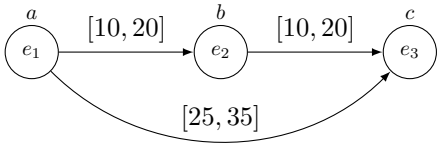


FIGURE 4.1 – La chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ de taille 3 avec $\mathcal{E} = \{e_1 = a, e_2 = b, e_3 = c\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[10, 20]e_2, \tau_{(1,3)} = e_1[25, 35]e_3, \tau_{(2,3)} = e_2[10, 20]e_3\}$.

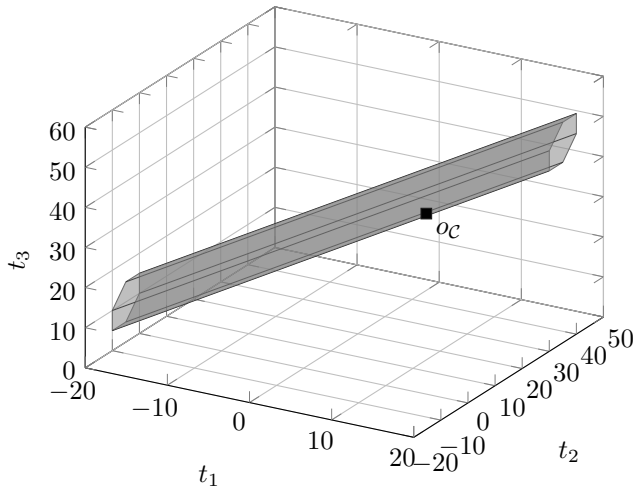


FIGURE 4.2 – Le polytope $\mathcal{P}$ de l'exemple 4.1 qui est une projection de la chronique $\mathcal{C}$. La projection de $o_\mathcal{C} = \{(a, 8), (b, 21), (c, 33)\}$ est mise en avant par un carré noir.

La représentation d'une chronique par un polytope dans $\mathbb{R}^n$ est sujette à plusieurs inconvénients limitant son utilité. En effet, la dimension de l'espace euclidien est la taille n de la chronique et en conséquence la comparaison de chroniques de tailles différentes dans un même espace n'est pas triviale. De plus, la projection n'est faite que sur les instants d'occurrence des nœuds de la chronique et ses événements ne sont pas pris en compte. Ainsi, cette projection n'est utile que lorsque le domaine temporel est d'intérêt. Le domaine spatial doit être considéré pour qu'une telle projection soit utile.

4.2.2 Projection de chroniques suivant les événements et le temps

Les chroniques peuvent être facilement projetées dans un espace euclidien par une simple projection orthogonale suivant le temps. Néanmoins, dans une telle projection, seul le domaine temporel d'une chronique est considéré alors que le domaine spatial n'est pas utilisé. Les événements d'une chronique apportent une information non négligeable dans l'objectif de comparaison et de classification des chroniques. Cette section présente une méthode de projection aléatoire permettant de considérer à la fois le domaine temporel $\mathbb{T}$ et spatial $\mathbb{E}$ (cf. définition 1.2, page 10) lors de la projection d'une chronique dans un espace euclidien.

Définition 4.3. Soit e un événement du domaine spatial $\mathbb{E}$. Soit $\phi(e)$ un vecteur k-dimensionnel composé de k tirages de variables aléatoires normalement distribuées avec une espérance $\mu = 0$ et une variance $\sigma^2 = 1$. Le vecteur aléatoire $\phi(e)$ k-dimensionnel est l'**empreinte d'un événement** e dans l'espace euclidien k-dimensionnel.

Définition 4.4. La **matrice d'empreintes** $\Phi \in \mathbb{R}^{n \times k}$ d'une chronique $\mathcal{C} = (\mathcal{E} = \{e_1, e_2, \ldots, e_n\}, \mathcal{T})$ est l'ensemble des empreintes des événements de $\mathcal{C}$ dans un es-

pace euclidien k-dimensionnel :

$$\Phi = \begin{bmatrix} \phi(e_1) & \phi(e_2) & \dots & \phi(e_n) \end{bmatrix}. \tag{4.4}$$

Grâce à l'utilisation de la matrice d'empreintes d'une chronique en lien avec la projection orthogonale vue dans la section précédente, l'information sur le domaine spatial et temporel contenue dans l'occurrence d'une chronique $o_{\mathcal{C}}$ est considérée dans la projection proposée ci-dessous. Les instants d'occurrences des nœuds sont aisément pris en compte par la projection orthogonale alors que les événements sont considérés par leurs empreintes.

Définition 4.5. La **projection aléatoire d'une occurrence d'une chronique** $o_{\mathcal{C}}$ dans un espace euclidien k-dimensionnel notée $x \in \mathbb{R}^k$ est le produit entre la matrice d'empreintes $\Phi \in \mathbb{R}^{n \times k}$ de $\mathcal{C}$ et les instants d'occurrences des nœuds de $\mathcal{C}$ et noté $t \in \mathbb{R}^n$ dans $o_{\mathcal{C}}$:

$$x = \Phi t. \tag{4.5}$$

Exemple 4.2. Soit $o_{\mathcal{C}} = \{(a, 8), (b, 21), (c, 33)\}$ une occurrence d'une chronique $\mathcal{C}$ et $\Phi = \begin{bmatrix} \phi(a) & \phi(b) & \phi(c) \end{bmatrix}$ la matrice d'empreintes de cette chronique générée aléatoirement telle que :

$$\begin{aligned} \phi(a) &= \begin{bmatrix} -0.1220 & -1.0868 & 0.6843 \end{bmatrix}^\top, \\ \phi(b) &= \begin{bmatrix} -1.0752 & 0.0333 & 0.7448 \end{bmatrix}^\top, \\ \phi(c) &= \begin{bmatrix} 0.0336 & -0.5266 & 0.4625 \end{bmatrix}^\top. \end{aligned}$$

Dans un souci de simplicité et de visualisation, k est ici fixé à 3 même si des problèmes apparaissent lorsque celui-ci est défini trop petit comme détaillé dans la section 4.4. La projection aléatoire x de $o_{\mathcal{C}}$ dans cet espace de dimension 3 est représentée dans la figure 4.3 par un carré noir. Elle est obtenue par la projection linéaire vue dans l'équation (4.5) :

$$x = \begin{bmatrix} -22.4457 & -25.3749 & 36.3794 \end{bmatrix}^\top.$$

Afin de pouvoir comparer, mesurer, ou encore analyser plusieurs occurrences de différentes chroniques dans cet espace euclidien, la projection aléatoire de celles-ci doit être dans le même espace. Les matrices d'empreintes des différentes chroniques doivent être cohérentes. En effet, si des chroniques partagent des événements, les empreintes de ceux-ci vont également être retrouvées dans leurs matrices d'empreintes respectives. Ainsi, un dictionnaire d'empreintes d'événements correspondant au domaine spatial des chroniques considérées doit être créé et maintenu afin d'appliquer cette méthode de projection aléatoire de manière cohérente.

Grâce à la matrice d'empreintes d'une chronique Φ, x est la représentation dans l'espace euclidien d'une occurrence $o_{\mathcal{C}}$. L'utilisation de la matrice d'empreintes d'une chronique directement sur le polytope $\mathcal{P}$ de cette chronique permet la projection de

l'ensemble de toutes les occurrences possibles de celle-ci. Cette projection mène à un nouveau polytope $\mathcal{P}'$ dans $\mathbb{R}^k$. Néanmoins, la taille k de l'espace euclidien doit être supérieure à la taille n de la chronique. En effet, la matrice d'empreintes Φ doit nécessairement être inversible par la gauche. Les solutions de la contrainte linéaire définie par le polytope $\mathcal{P}'$ sont toutes les occurrences possibles de la chronique $\mathcal{C}$ projetées dans l'espace euclidien.

Proposition 4.1. *Soit $At \geq b$ une contrainte linéaire définie par une chronique $\mathcal{C}$ et $x = \Phi t$ la projection d'une occurrence $o_{\mathcal{C}}$ dans l'espace euclidien k-dimensionnel. Il existe une matrice $A' \in \mathbb{R}^{2m \times k}$, si et seulement si $k \geq n$, telle que :*

$$A'x \geq b. \tag{4.6}$$

Démonstration. Comme Φ est remplie de par des tirages de variables aléatoires indépendantes et identiquement distribuées, chaque ligne et colonne sont linéairement indépendantes. La pseudo-inverse $\Phi^+ \in \mathbb{R}^{k \times n}$ peut être calculée [Penrose 1955]. Quand $k \geq n$, Φ^+ est inversable par la gauche et est donnée par :

$$\Phi^+ = (\Phi^\top \Phi)^{-1} \Phi^\top.$$

L'équation (4.4) devient :

$$\Phi^+ x = t.$$

Puis, avec l'équation (4.3) :

$$A\Phi^+ x \geq b.$$

Enfin, avec $A' = A\Phi^+$,

$$A'x \geq b.$$

□

Définition 4.6. La **projection aléatoire d'une chronique** $\mathcal{C}$ dans un espace euclidien k-dimensionnel est définie par le polytope $\mathcal{P}'$ dans $\mathbb{R}^k$ tel que celui-ci respecte la contrainte linéaire suivante :

$$A'x \geq b.$$

Exemple 4.3. Soit $At \geq b$ la projection dans $\mathbb{R}^3$ ($n = 3$) de la chronique $\mathcal{C}$ de taille 3 détaillée dans l'exemple 4.1 et représentée sur la figure 4.1. Soit $x = \Phi t$ la projection de l'occurrence $o_{\mathcal{C}}$ dans l'espace k-dimensionnel où $k = 3$ et détaillée dans l'exemple 4.2. Tout d'abord, la pseudo-inverse $\Phi^+ \in \mathbb{R}^{3 \times 3}$ est calculée :

$$\Phi^+ = \begin{bmatrix} -1.7670 & -2.2641 & -2.4495 \\ -0.6169 & 0.3442 & 0.4367 \\ 3.6075 & 2.7953 & 5.0827 \end{bmatrix}.$$

Puis, la matrice $A' \in \mathbb{R}^{6\times 3}$ est donnée par $A' = A\Phi^+$:

$$A' = \begin{bmatrix} 1.1501 & 2.6083 & 2.8863 \\ -1.1501 & -2.6083 & -2.8863 \\ 5.3745 & 5.0594 & 7.5322 \\ -5.3745 & -5.0594 & -7.5322 \\ 4.2244 & 2.4511 & 4.6459 \\ -4.2244 & -2.4511 & -4.6459 \end{bmatrix}.$$

Le polytope $\mathcal{P}'$ défini par $A'x \geq b$ est représenté sur la figure 4.3. La projection de l'occurrence $o_\mathcal{C}$ détaillée dans l'exemple 4.2 est une solution de $A'x \geq b$:

$$\begin{bmatrix} 1.1501 & 2.6083 & 2.8863 \\ -1.1501 & -2.6083 & -2.8863 \\ 5.3745 & 5.0594 & 7.5322 \\ -5.3745 & -5.0594 & -7.5322 \\ 4.2244 & 2.4511 & 4.6459 \\ -4.2244 & -2.4511 & -4.6459 \end{bmatrix} \begin{bmatrix} -22.4457 \\ -25.3749 \\ 36.3794 \end{bmatrix} \geq \begin{bmatrix} 10 \\ -20 \\ 25 \\ -35 \\ 10 \\ -20 \end{bmatrix}.$$

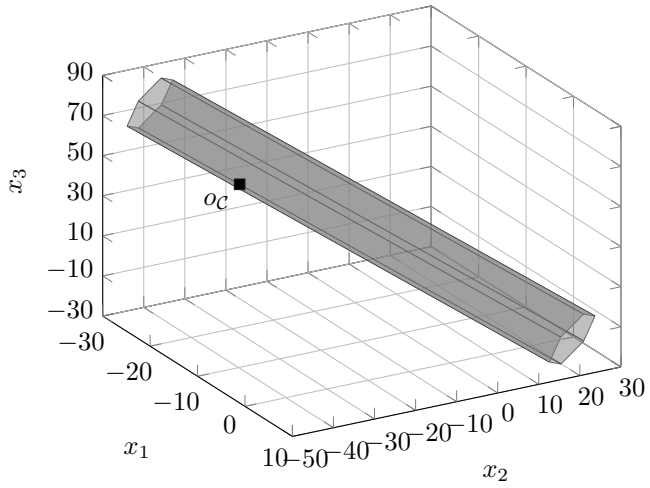


FIGURE 4.3 – Le polytope $\mathcal{P}'$ est la projection d'une chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ dans l'espace euclidien de dimension 3. La projection de l'occurrence $o_\mathcal{C} = \{(a, 8), (b, 21), (c, 33)\}$ est mise en avant par un carré noir.

La proposition 4.1 définit un moyen de calculer la projection d'une chronique seulement dans le cas où chaque nœud de la chronique est associé à un événement unique (autrement dit $\mathcal{E}$ est un ensemble classique). Une difficulté se pose dans le cas général où $\mathcal{E}$ est un multi-ensemble. C'est-à-dire qu'il peut contenir plusieurs fois le même événement. Plusieurs utilisations de la même empreinte dans la matrice d'empreintes Φ rend celle-ci non-inversible car, dans ce cas, toutes les colonnes ne sont pas linéairement indépendantes.

Une solution possible à ce problème est de différencier les événements identiques

dans ce multi-ensemble. Par exemple, la chronique représentée sur la figure 4.4a possède deux événements b. L'empreinte $\phi(b_1)$ du premier événement b et l'empreinte $\phi(b_2)$ du deuxième événement b sont différenciées. Ainsi, la matrice d'empreintes de cette chronique correspond à :

$$\begin{aligned}\Phi &= \begin{bmatrix}\phi(a) & \phi(b_1) & \phi(b_2)\end{bmatrix}, \\ &= \begin{bmatrix}-0.1220 & -1.0752 & 0.5377 \\ 1.0868 & 0.0333 & 1.8339 \\ 0.6843 & 0.7448 & -2.2588\end{bmatrix}.\end{aligned}$$

Néanmoins, d'autres problèmes émergent avec cette solution. En effet, la projection des chroniques dépend de la relation d'ordre arbitraire $<_{\mathbb{E}}$ définie sur le domaine spatial. Ainsi, des chroniques équivalentes où la seule différence et le choix arbitraire de la numérotation des nœuds peuvent avoir des projections différentes. Par exemple, les deux chroniques ont une projection différente suivant si les nœuds sont numérotés comme sur la figure 4.4a ou comme sur la figure 4.4b. Avec la solution proposée ici, leur matrices d'empreintes sont identiques mais les différences dans les contraintes temporelles donnent un résultat différent sur la projection. Ce travail doit être approfondi afin de résoudre cette problématique.

(a) Première configuration. (b) Seconde configuration.

FIGURE 4.4 – Deux configurations de la numération des nœuds de la même chroniques qui sont projetées différemment.

4.3 Propriétés de la projection d'une chronique

Ainsi, il est possible de projeter une chronique sur un polytope non-borné dans un espace euclidien k-dimensionnel. En raison de cette projection, ce volume géométrique possède des propriétés qui sont étroitement liées à la chronique sous-jacente. Ces propriétés peuvent être utiles à l'analyse de ces volumes, des chroniques sous-jacentes, ou encore à des fins de comparaison. Dans cette section, quelques-unes de ces propriétés sont décrites. Dans un premier temps, les relations entre une chronique et sa projection sont détaillées. Puis, dans un second temps, quelques indices de comparaison entre deux projections de chroniques sont fournis.

4.3.1 Relations entre une chronique et sa projection

Cette section offre une analyse sur le polytope formé par la projection d'une chronique. Le polytope $\mathcal{P}'$ est un tube infini dont toutes les arêtes sont parallèles les unes par rapport aux autres. L'orientation de ce tube est définie par le domaine spatial de la chronique sous-jacente $\mathcal{C}$, alors que le volume de ce tube est déterminé par le domaine temporel de $\mathcal{C}$. Ces propriétés sont démontrées dans la suite de cette section.

Proposition 4.2. *Une chronique $\mathcal{C}$ qui ne possède pas de sous-chronique indépendante autre qu'elle-même projetée dans un espace euclidien k-dimensionnel est un polytope $\mathcal{P}'$ où toutes ses arêtes sont parallèles et possèdent le même vecteur euclidien $\vec{v'}$.*

Démonstration. Le vecteur euclidien $\vec{v}$ est aisément calculé pour le polytope $\mathcal{P}$ défini dans l'équation (4.3). Pour tout $n \in \mathbb{N}$, considérons la chronique $\mathcal{C}$ de taille n définissant $At \geq b$. Le noyau de A est caractérisé par l'ensemble d'équations $At = 0$:

$$\ker(A) = \{t \in K^n \mid At = 0\}.$$

Comme $\mathcal{C}$ ne possède pas de sous-chronique indépendante autre qu'elle-même, tous les instants d'occurrences t_i d'un nœud sont contraints par au moins un instant d'occurrence t_j d'un autre nœud, $At = 0$ peut être réécrit comme suit :

$$\{-t_i + t_j = 0 \mid \forall \nu_{\mathcal{C}}(e_i), \exists \nu_{\mathcal{C}}(e_j)\}.$$

Donc, le rang du noyau de A est toujours 1. Pour tout $n \in \mathbb{N}$, le vecteur euclidien $\vec{v} \in \mathbb{R}^n$ de $\mathcal{C}$ est unique et défini par :

$$\vec{v} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^\top. \tag{4.7}$$

Dans l'espace euclidien k-dimensionnel, le vecteur euclidien $\vec{v'}$ du polytope $\mathcal{P}'$ est déduit de $\vec{v}$ grâce à la matrice d'empreintes Φ de la chronique $\mathcal{C}$:

$$\vec{v'} = \Phi\vec{v}. \tag{4.8}$$

Ainsi, le vecteur euclidien $\vec{v'}$ est une représentation de l'ensemble des événements $\mathcal{E}$ présent dans la chronique $\mathcal{C}$. □

Le tube représenté par le polytope $\mathcal{P}'$ projection d'une chronique $\mathcal{C}$ est la répétition à l'infini de sa section $\mathcal{V}$ suivant l'orientation du vecteur euclidien $\vec{v'}$. Cette section peut être aisément calculée par l'intersection entre un hyper-plan orthogonal au vecteur euclidien $\vec{v'}$ et le polytope $\mathcal{P}'$.

La section $\mathcal{V}$ la plus proche de l'origine est caractérisée par l'intersection entre l'hyper-plan orthogonal au vecteur euclidien $\vec{v'}$ et passant par l'origine, et le polytope $\mathcal{P}'$. Dans le reste de ce chapitre, la section $\mathcal{V}$ d'un polytope $\mathcal{P}'$ est caractérisée par la section la plus proche de l'origine.

Exemple 4.4. Soient $\mathcal{P}'$ le polytope détaillé dans l'exemple 4.3 et la projection de la chronique $\mathcal{C}$ détaillée dans l'exemple 4.1 et représentée sur la figure 4.1. La matrice d'empreintes Φ de la chronique $\mathcal{C}$ est la suivante :

$$\Phi = \begin{bmatrix} -0.1220 & -1.0752 & 0.0336 \\ 1.0868 & 0.0333 & -0.5266 \\ 0.6843 & 0.7448 & 0.4625 \end{bmatrix}. \tag{4.9}$$

Le vecteur euclidien $\vec{v'}$ du polytope $\mathcal{P}'$ est défini par :

$$\vec{v'} = \Phi \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top = \begin{bmatrix} -1.1635 & -1.5802 & 1.8917 \end{bmatrix}^\top. \tag{4.10}$$

$\mathcal{P}'$ est représenté sur la figure 4.5 avec le vecteur euclidien $\vec{v'}$ ainsi que la section $\mathcal{V}$. $\mathcal{V}$ est l'intersection entre l'hyper-plan orthogonal à $\vec{v'}$ et passant par l'origine et $\mathcal{P}'$ et est mise en avant par un contour noir.

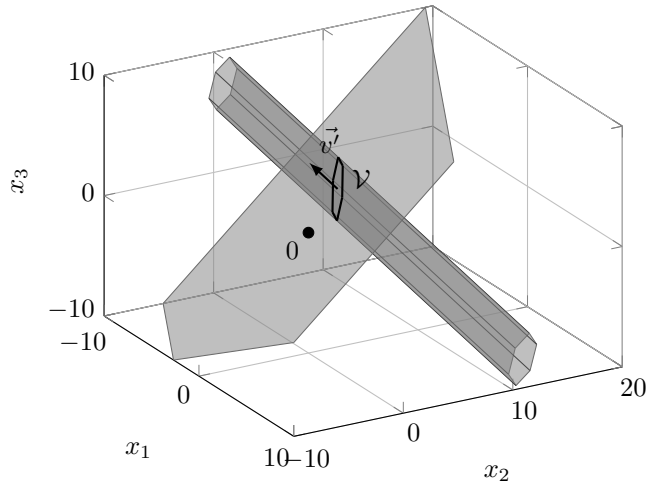


FIGURE 4.5 – Le polytope $\mathcal{P}'$ qui est la projection de la chronique $\mathcal{C}$ dans l'espace euclidien de dimension 3. La section $\mathcal{V}$ est représentée par le volume à l'intérieur du contour noir. Le vecteur euclidien $\vec{v'}$ ainsi que l'origine sont représentés sur ce tracé.

Avec les différentes distances temporelles ($d(t^-, t^+)$) des intervalles des contraintes temporelles faibles, un ensemble plus petit d'instants d'occurrence acceptables mène à une plus petite section $\mathcal{V}$. Alors qu'avec ce type de mesure plus élevé, plus d'instants d'occurrence sont possibles et une plus grande section $\mathcal{V}$ est possible.

La durée de reconnaissance d'une chronique influe sur la distance de l'origine de la section $\mathcal{V}$. En effet, avec une durée de reconnaissance rapide, $\mathcal{V}$ est proche du 0, alors qu'avec une durée de reconnaissance plus lente, $\mathcal{V}$ est plus éloignée.

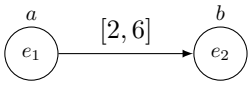
Ainsi, un vecteur euclidien $\vec{v'}$ et une section $\mathcal{V}$ suffissent pour caractériser la

projection d'une chronique $\mathcal{C}$. $\vec{v'}$ représente le domaine spatial de $\mathcal{C}$, alors que $\mathcal{V}$ représente le domaine temporel de $\mathcal{C}$.

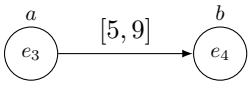
4.3.2 Indices de comparaison entre deux projections de chroniques

Une méthode communément utilisée pour comparer des chroniques est à travers leurs occurrences. En effet, la propriété d'équivalence de deux chroniques (cf. définition 1.14, page 14) signifie qu'elles possèdent exactement le même ensemble d'occurrences. Dans l'espace euclidien, les occurrences qui sont communes à deux chroniques sont projetées dans l'intersection des deux polytopes associés. Ainsi, soit $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ et $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ deux chroniques où $\mathcal{E}_1 = \mathcal{E}_2$. Leurs vecteurs euclidiens $\vec{v_1'}$ et $\vec{v_2'}$ sont identiques $\vec{v_1'} = \vec{v_2'}$ car elles possèdent la même matrice d'empreintes.

Exemple 4.5. Soit $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ avec $\mathcal{E}_1 = \{e_1 = a, e_2 = b\}$, $\mathcal{T}_1 = \{\tau_{(1,2)} = e_1[2,6]e_2\}$ et $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ avec $\mathcal{E}_2 = \{e_3 = a, e_4 = b\}$, $\mathcal{T}_2 = \{\tau_{(3,4)} = e_3[5,9]e_4\}$ deux chroniques élémentaires représentées sur la figure 4.6. Ces chroniques possèdent le même vecteur euclidien $\vec{v_1'} = \vec{v_2'}$ car leurs matrices d'empreintes sont identiques. Des occurrences de ces chroniques sont identiques, $o_{\mathcal{C}_1} = o_{\mathcal{C}_2} = \{(a,3),(b,8)\}$ est une occurrence à la fois de $\mathcal{C}_1$ et $\mathcal{C}_2$. Dans l'espace euclidien k-dimensionnel où $k = 2$, les projections aléatoires de $o_{\mathcal{C}_1}$ et $o_{\mathcal{C}_2}$ sont égales et les deux polytopes $\mathcal{P}_1'$ et $\mathcal{P}_2'$ qui sont les projections aléatoires des chroniques, respectivement, $\mathcal{C}_1$ et $\mathcal{C}_2$ contiennent ces deux occurrences. Ces deux polytopes sont représentés graphiquement sur la figure 4.7. L'intersection de ces deux polytopes correspond à l'ensemble des occurrences possibles qui sont communes à $\mathcal{C}_1$ et $\mathcal{C}_2$.



(a) La chronique élémentaire $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ avec $\mathcal{E}_1 = \{e_1 = a, e_2 = b\}$, $\mathcal{T}_1 = \{\tau_{(1,2)} = e_1[2,6]e_2\}$.



(b) La chronique élémentaire $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ avec $\mathcal{E}_2 = \{e_3 = a, e_4 = b\}$, $\mathcal{T}_2 = \{\tau_{(3,4)} = e_3[5,9]e_4\}$.

FIGURE 4.6 – Les chroniques élémentaires $\mathcal{C}_1$ et $\mathcal{C}_2$.

De plus, le volume des sections $\mathcal{V}_1$ et $\mathcal{V}_2$ des polytopes $\mathcal{P}_1'$ et $\mathcal{P}_2'$ sont similaires car les distances temporelles de l'intervalle des contraintes temporelles $\tau_{(1,2)}$ de $\mathcal{C}_1$ et $\tau_{(3,4)}$ de $\mathcal{C}_2$ sont identiques, $d(t^-_{(1,2)}, t^+_{(1,2)}) = 4$ et $d(t^-_{(3,4)}, t^+_{(3,4)}) = 4$. Enfin, $\mathcal{P}_2'$ est plus éloigné de l'origine que $\mathcal{P}_1'$ car la durée minimale de $\mathcal{C}_1$ est plus petite que la durée minimale de $\mathcal{C}_2$.

Proposition 4.3. *Soient $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ et $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ deux chroniques où $\mathcal{E}_1 = \mathcal{E}_2$. Soient $\mathcal{P}_1'$ et $\mathcal{P}_2'$ les projections des chroniques $\mathcal{C}_1$ et $\mathcal{C}_2$ dans un espace euclidien k-dimensionnel. Si $\mathcal{C}_1$ couvre $\mathcal{C}_2$, alors $\mathcal{P}_2'$ est inclus dans $\mathcal{P}_1'$.*

Démonstration. D'après la définition de la couverture (cf. définition 1.13, page 14), $\mathcal{C}_1$ couvre $\mathcal{C}_2$ si toutes les occurrences $o_{\mathcal{C}_2}$ sont incluses dans $o_{\mathcal{C}_1}$ quelle que soit la séquence temporelle $\mathcal{S}$. Or, $\mathcal{P}_1'$ est composé de toutes les occurrences possibles de

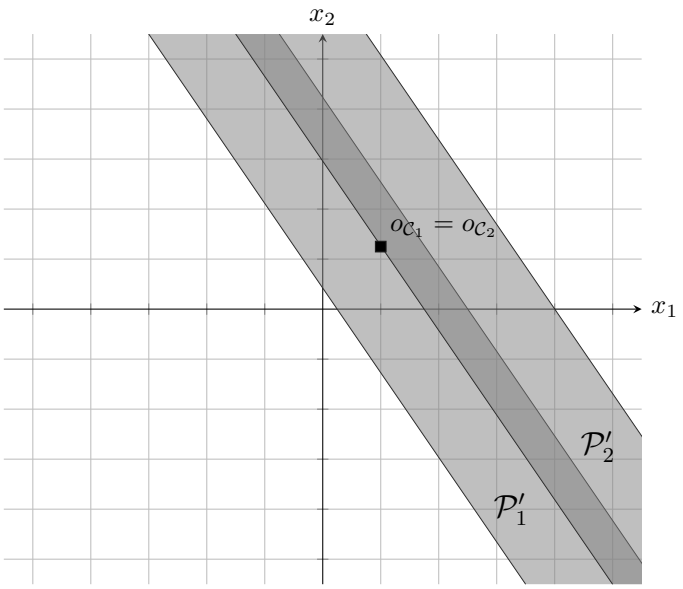


FIGURE 4.7 – Les polytopes $\mathcal{P}'_1$ et $\mathcal{P}'_2$ qui sont les projections aléatoires des deux chroniques élémentaires $\mathcal{C}_1$ et $\mathcal{C}_2$ représentées sur la figure 4.6 dans l'espace euclidien $\mathbb{R}^k$ avec $k = 2$. La projection des occurrences $o_{\mathcal{C}_1}$ et $o_{\mathcal{C}_2}$ est représentée par un carré noir.

$\mathcal{C}_1$ et $\mathcal{P}'_2$ est composé de toutes les occurrences possibles de $\mathcal{C}_2$. Donc :

$$\mathcal{P}'_2 \subseteq \mathcal{P}'_1. \tag{4.11}$$

□

Proposition 4.4. *Soient $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ et $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$ deux chroniques où $\mathcal{E}_1 = \mathcal{E}_2$. Soient $\mathcal{P}'_1$ et $\mathcal{P}'_2$ les projections des chroniques $\mathcal{C}_1$ et $\mathcal{C}_2$ dans un espace euclidien k-dimensionnel. Si $\mathcal{C}_1$ et $\mathcal{C}_2$ sont équivalentes, alors $\mathcal{P}'_1$ et $\mathcal{P}'_2$ sont égaux.*

Démonstration. D'après la définition de l'équivalence (cf. définition 1.14, page 14), $\mathcal{C}_1$ et $\mathcal{C}_2$ sont équivalentes si l'ensemble des occurrences $\mathcal{O}_{\mathcal{C}_1}(\mathcal{S})$ et $\mathcal{O}_{\mathcal{C}_2}(\mathcal{S})$ sont identiques quelle que soit la séquence temporelle $\mathcal{S}$. Or, $\mathcal{P}'_1$ est composé de toutes les occurrences possibles de $\mathcal{C}_1$ et $\mathcal{P}'_2$ est composé de toutes les occurrences possibles de $\mathcal{C}_2$. Donc :

$$\mathcal{P}'_1 = \mathcal{P}'_2. \tag{4.12}$$

□

4.4 Analyse du nombre de dimensions de l'espace euclidien

Cette section propose une analyse autour du paramètre k associé au nombre de dimensions de l'espace euclidien considéré. Ce paramètre influence considérable-

ment la projection. En effet, comme vu dans la section 4.2.2, k doit être supérieur à la taille n de la chronique projetée pour que la projection soit possible (cf. proposition 4.1). De plus, des phénomènes indésirables peuvent être constatés lorsque ce paramètre est trop faible et qui sont liés au facteur aléatoire de la projection. Néanmoins, augmenter de manière exagérée le nombre de dimensions peut complexifier les calculs de distances ou toute autre méthode de comparaison.

Dans un premier temps, un exemple d'un phénomène indésirable est fourni. Puis, dans un second temps, une analyse statistique sur la norme de la projection d'une occurrence est détaillée.

4.4.1 Impact du nombre de dimensions

Les événements d'une chronique sont considérés via une projection aléatoire (cf. section 4.2.2). En raison de cette part d'aléatoire, une approximation de l'information est inévitable. Le nombre k de dimensions permet de tempérer plus ou moins cette approximation. En effet, le théorème de Johnson-Lindenstrauss [Johnson 1984], qui est un des piliers de la méthode de projection aléatoire, montre que cette approximation peut être quantifiée par la valeur ϵ dans l'intervalle $[0, 1]$. Une valeur élevée de ϵ, obtenue avec un nombre k de dimensions important, permet de réduire l'approximation obtenue alors qu'un k fixé trop petit va augmenter l'approximation. Ainsi, le choix du nombre de dimensions doit être l'objet d'une attention et d'optimisation pour éviter le phénomène visible dans l'exemple suivant.

Exemple 4.6. Soit $\mathcal{C}_1$ et $\mathcal{C}_2$ deux chroniques élémentaires où $\mathcal{E}_1 = \{e_1 = a, e_2 = b\}$ et $\mathcal{E}_2 = \{e_1 = c, e_2 = d\}$ qui sont projetées dans $\mathbb{R}^k$, avec $k = 2$. Cette projection définit les deux polytopes $\mathcal{P}'_1$ et $\mathcal{P}'_2$ représentés sur la figure 4.8. Sur ce tracé, $\mathcal{P}'_1$ et $\mathcal{P}'_2$ se croisent. D'après la section 4.3.2, cela signifie qu'au moins une des occurrences $o_{\mathcal{C}_1}$ de $\mathcal{C}_1$ est égale à une des occurrences $o_{\mathcal{C}_2}$ de $\mathcal{C}_2$. Néanmoins, $\mathcal{C}_1$ et $\mathcal{C}_2$ ne peuvent pas avoir d'instances en commun car leurs événements sont différents $E_{\mathcal{C}_3} \cap E_{\mathcal{C}_4} = \emptyset$. Cette intersection est le résultat de l'approximation ϵ liée à la projection aléatoire. La probabilité de ce phénomène indésirable augmente plus le nombre k de dimensions diminue.

4.4.2 Analyse statistique de la norme d'une occurrence projetée

Grâce à une analyse statistique, il est possible de calculer l'espérance et la variance de la norme au carré d'une occurrence projetée dans un espace euclidien k-dimensionnel. Cette analyse peut se faire grâce à l'utilisation de variables aléatoires pour représenter l'information événementielle d'une chronique. Cette norme, notée $\|x\|$, dépend du nombre k de dimensions, de la taille n de la chronique ainsi que des différents instants d'occurrence des nœuds de cette chronique. Cette section propose une idée de démonstration pour le calcul de l'espérance et de la variance de $\|x\|$. Une analyse plus complète peut être retrouvée dans l'annexe C.1.

Considérons la norme au carré $\|x\|^2$ du vecteur x obtenu par la projection d'une occurrence d'une chronique de taille n dans l'espace euclidien k-dimensionnel. $x(j)$

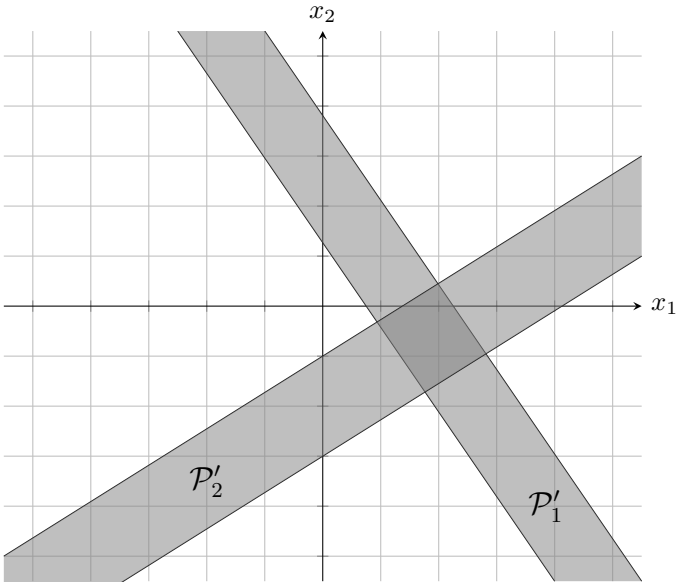


FIGURE 4.8 – Un phénomène indésirable obtenu lorsque le nombre k de dimensions est trop faible.

est le j-ième élément de x et $\|x\|^2$ est défini par :

$$\|x\|^2 = \sum_{j=1}^{k} x(j)^2. \tag{4.13}$$

Comme x est défini par l'équation (4.5), $x(j)^2$ peut être récrit comme suit :

$$x(j)^2 = \left(\sum_{i=1}^{n} \phi(e_i, j) t_i \right)^2. \tag{4.14}$$

De plus, x est un vecteur rempli de variables aléatoires indépendantes avec une espérance $\mu = 0$ et une variance $\sigma^2 = 1$. Donc, l'espérance de $\phi(e_i, j)^2$ est $E\left(\phi(e_i, j)^2\right) = 1$ et l'espérance de $\phi(e_i, j)$ est $E\left(\phi(e_i, j)\right) = 0$. Ainsi, l'espérance de $x(j)^2$ est :

$$E\left(x(j)^2\right) = E\left(\left(\sum_{i=1}^{n} \phi(e_i, j) t_i \right)^2 \right) = \sum_{i=1}^{n} t_i^2. \tag{4.15}$$

Donc, l'espérance de $\|x\|^2$ est donnée par :

$$E\left(\|x\|^2\right) = \sum_{j=1}^{k} E\left(x(j)^2\right) = k \sum_{i=1}^{n} t_i^2. \tag{4.16}$$

Rappelons que la variance d'une variable aléatoire X est donnée par la formule

suivante :

$$Var(X) = E\left(X^2\right) - E\left(X\right)^2. \tag{4.17}$$

Ainsi, la variance de $x(j)^2$ est :

$$Var(x(j)^2) = E\left(\left(x(j)^2\right)^2\right) - E\left(x(j)^2\right)^2, \tag{4.18}$$

$$= 3\left(\sum_{i=1}^{n} t_i^2\right)^2 - \left(\sum_{i=1}^{n} t_i^2\right)^2, \tag{4.19}$$

$$= 2\left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{4.20}$$

Donc, la variance de $\|x\|^2$ est donnée par :

$$Var\left(\|x\|^2\right) = \sum_{j=1}^{k} Var\left(x(j)^2\right) = 2k\left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{4.21}$$

L'espérance et la variance de la norme au carré d'une occurrence x sont connues et sont fonction du nombre k de dimension. Un nombre de dimensions important va influencer fortement la dispersion des chroniques projetées dans cet espace euclidien. En d'autres termes, une chronique de taille n est plus éloignée de l'origine dans un espace de dimension élevé que dans un espace de dimension restreint. De plus, la taille n de la chronique projetée va influencer la variance de la projection. Les chroniques de taille importante et complexes sont plus aisément différenciées que des chroniques plus simples.

4.5 Distance entre chroniques et autres perspectives de la projection de chroniques

La projection de chroniques offre des remarquables perspectives pour l'analyse des chroniques. En effet, la représentation de chroniques dans un espace euclidien où il est aisé de comparer et d'analyser des objets est avantageuse. Elle permet de répondre à des questions auxquelles il n'est pas aisé de répondre avec les outils offerts par l'analyse des chroniques. Ces problématiques peuvent être les suivantes :

1. *Parmi un ensemble de chroniques, quelles sont les chroniques qui sont les plus différentes ?*
2. *Quel est le résultat d'un partitionnement sur un ensemble de chroniques ?*
3. *Quelle est la chronique la plus représentative d'un ensemble de chroniques ?*

Pour répondre à ces différentes problématiques, une distance entre chroniques doit être fournie. Une telle distance peut être équivalente à la distance entre les projections de celles-ci. Le calcul de distances entre polytopes est communément exploitée dans les applications géo-spatiales [Atallah 1991]. La distance de Hausdorff

[Henrikson 1999] est un exemple de distance entre polytopes utilisé à cette fin et en particulier dans le domaine du traitement d'images. Néanmoins, ces distances sont calculées entre deux polytopes bornés. Or, les projections des chroniques sont des polytopes non bornés. Plusieurs solutions sont prospectées pour résoudre cette difficulté :

— La distance entre les duals des polytopes peut être utilisée. En effet, quel que soit le polytope, le dual d'un polytope existe et est un polytope borné [Coxeter 1973]. Néanmoins, le calcul du dual nécessite une inversion de matrice qui est coûteuse en temps de calcul car polynomiale en fonction du nombre de dimensions.

— L'intersection entre le polytope $\mathcal{P}'$ résultat de la projection d'une chronique et l'hyper-plan orthogonal au vecteur euclidien $\vec{v'}$ forme un polytope borné. Contrairement au dual, ce polytope peut être aisément calculé. En effet, $\vec{v'}$ est le simple produit entre la matrice d'empreintes Φ d'une chronique et un vecteur rempli de 1. Cette solution répond élégamment au problème de la distance entre chroniques.

— Une autre solution possible consiste à ne pas considérer le polytope dans son ensemble mais seulement un ensemble de points. Dans ce cas, une nouvelle question se pose, comment déterminer les points qui sont pertinents pour représenter le polytope $\mathcal{P}'$.

Ainsi, avec une distance entre chroniques définie, les problématiques soulevées plus tôt dans cette section peuvent être résolues :

1. *Parmi un ensemble de chroniques, quelles sont les chroniques qui sont les plus différentes ?*

Cette question est résolue par un calcul des plus proches voisins de chaque chronique considérée. Les chroniques les plus différentes sont celles dont le plus proche voisin est le plus éloigné.

2. *Quel est le résultat d'un partitionnement sur un ensemble de chroniques ?*

Une matrice de distance ou toute autre structure permettant de calculer efficacement la distance entre deux objets peut être calculée dans l'espace euclidien. Avec une telle structure, n'importe quel algorithme de partitionnement de données peut être exploité pour regrouper les différentes chroniques par similarité. Par exemple, DBSCAN, qui est utilisé dans l'étape d'identification de chroniques élémentaires de CDIRe, permet un tel partitionnement. Quelques algorithmes de partitionnement de données sont référencés dans la section 2.2.2 (page 45).

3. *Quelle est la chronique la plus représentative d'un ensemble de chroniques ?*

Pour répondre à cette question, le centre dans l'espace euclidien de l'ensemble des projections des chroniques considéré doit être calculé. Puis, la chronique considérée la plus représentative est celle qui est la plus proche de ce centre.

Ces différentes problématiques sont des questions qui peuvent être soulevées lors d'un traitement des chroniques générées par un algorithme de découverte de

chroniques. Avoir un tel outil supplémentaire permettant de répondre à ces questions de manière automatique et rapide permet d'améliorer significativement le processus de découverte de chroniques. En effet, comme vu dans le chapitre 1, un inconvénient majeur de ce type d'algorithme est la quantité phénoménale de chroniques générées.

De nombreuses questions théoriques sur la projection aléatoire et en particulier sur le nombre k de dimensions restent en suspens. En effet, suivant le théorème de Johnson-Lindenstrauss, k doit être très grand pour conserver les informations inhérentes au modèle des chroniques. Néanmoins, les premiers résultats pratiques montrent qu'un nombre de dimensions très inférieur à ceux préconisés par ce théorème ne dégrade pas significativement les informations sous-jacentes aux chroniques. Les différentes applications de la projection aléatoire, telle que [Bingham 2001], ont une observation similaire quant au nombre de dimensions. Ceci peut être dû au fait que le théorème de Johnson-Lindenstrauss est le pire cas possible. De plus, les variables aléatoires exploitées peuvent respecter d'autres lois de probabilité. Par exemple, une loi uniforme discrète dont le support est l'ensemble $\{-1, 0, 1\}$ respecte également le théorème de Johnson-Lindenstrauss.

Les propriétés de la projection aléatoire dans le contexte de la projection de chroniques restent largement à explorer. Néanmoins, les premiers résultats théoriques de cette approche innovante sont prometteurs. La projection aléatoire de chroniques peut être un outil efficace et puissant de plus à rajouter à la panoplie de l'analyse de chroniques.

4.6 Conclusion

Ce chapitre se penche sur un travail introductif qui est une nouvelle représentation des chroniques dans un espace euclidien k-dimensionnel. Cette représentation repose sur une méthode mathématique appelée projection aléatoire. Un rapide tour d'horizon de l'utilisation de cette méthode dans la littérature est fournie. La méthodologie de la projection aléatoire appliquée aux chroniques ainsi que quelques propriétés sont posées. Puis, une étude du nombre k de dimensions de l'espace euclidien considéré est offerte. Enfin, quelques perspectives de cette méthode concernant son utilisation dans un contexte de découverte de chroniques sont établies.

Conclusion et perspectives

Le sujet de ce mémoire de thèse porte sur les chroniques en général et en particulier le processus de découverte de chroniques. Une chronique est un modèle temporel pertinent pour le diagnostic en raison de sa capacité d'abstraction des événements. La découverte de chroniques est le processus de construction automatique de chroniques à partir d'un ensemble de données prenant la forme de séquences temporelles. Les contributions de ce mémoire dont le thème principal est la découverte de chroniques sont les suivantes : une nouvelle approche au problème de la découverte de chroniques, un algorithme adoptant cette approche intitulée CDIRe (*Chronicle Discovery by Identification and Reconstitution*), une implémentation de CDIRe en `C++`, une analyse de cet algorithme sur un jeu de données provenant d'une application réelle, et une représentation innovante d'une chronique dans un espace euclidien.

La première contribution importante de ce mémoire vient d'une nouvelle approche au problème de la découverte de chroniques, une approche par *identification* de chroniques élémentaires et *reconstitution* de chroniques plus complexes à partir de ces chroniques élémentaires. Une chronique élémentaire est une chronique avec deux événements et une contrainte temporelle entre ces événements. Cette nouvelle approche au problème de la découverte de chroniques soulève une problématique différente de celles proposées dans la littérature :

Problématique. *Soit $\mathcal{S}$ une séquence temporelle composée de plusieurs occurrences d'un ou de plusieurs phénomènes temporels, le problème de la découverte de chroniques par identification et reconstitution est de générer une ou des chroniques descriptives du ou des phénomènes sous-jacents à $\mathcal{S}$.*

Un algorithme intitulé CDIRe qui a pour objectif de répondre à cette problématique grâce à cette approche innovante est présenté dans le chapitre 2. CDIRe dépend de deux étapes définies par ces notions d'identification et de reconstitution et fonctionnent de pair pour générer des chroniques qui sont les plus descriptives du ou des phénomènes sous-jacents à une séquence temporelle d'entrée. De plus, une analyse de la complexité algorithmique de CDIRe montre que le pire des cas est polynomial en fonction de la taille de la séquence temporelle $\mathcal{S}$ d'entrée : $O\left(|\mathcal{S}|^6 \log^2(|\mathcal{S}|)\right)$. Cette analyse met également en évidence une interconnexion forte entre les deux étapes d'identification et de reconstitution de CDIRe. Ainsi, deux axes de travail pour le perfectionnement de CDIRe sont identifiés.

L'étape d'identification de chroniques élémentaires repose sur des algorithmes de partitionnement de données pour extraire des chroniques simples se démarquant dans la séquence temporelle d'entrée. Toutes les distances temporelles, le temps

écoulé entre deux événements, sont calculées pour chaque couple d'événements présent dans la séquence temporelle d'entrée. Puis, pour chaque couple d'événements, un algorithme de partitionnement de données, ici, CDIRe exploite DBSCAN et permet de partitionner ces distances temporelles. Puis, une chronique élémentaire est extraite de chaque groupement généré, où les événements correspondent aux événements du couple et la contrainte temporelle correspond aux distances temporelles. Plusieurs perspectives pour des futurs travaux naissent de cette étape d'identification.

Une analyse plus poussée des paramètres ε et $MinPts$ de DBSCAN peut être accomplie. En effet, ces paramètres ne sont définis qu'une seule fois pour tous les ensembles de distances temporelles, quel que soit le couple d'événements considéré. Il est possible d'obtenir de meilleurs résultats si ces paramètres sont optimisés pour chaque ensemble de distances temporelles indépendamment. Une telle approche permet de générer des chroniques dont les durées de reconnaissance des occurrences dans la séquence temporelle d'entrée sont sur des granularités différentes.

Perspective 1. *Est-il possible d'optimiser le choix des paramètres de l'algorithme de partitionnement de données utilisé pour chaque ensemble de distances temporelles des différents couples d'événements ?*

Seul DBSCAN est exploité pour cette étape d'identification de chroniques élémentaires. Néanmoins, n'importe quel algorithme de partitionnement de données peut être utilisé à la place de celui-ci. Cet algorithme est utilisé car il est avantageux dans le contexte de CDIRe. Aucune analyse sur les différents résultats pouvant être générés par l'utilisation d'algorithmes de partitionnement de données différents n'est proposé dans ce mémoire. Une telle analyse est pertinente pour, soit confirmer l'utilisation de DBSCAN, soit l'utilisation d'un autre algorithme plus adapté.

Perspective 2. *Est-ce qu'un algorithme de partitionnement de données différent de DBSCAN est plus adapté pour la tâche d'identification de chroniques élémentaires ?*

L'étape de reconstitution de chroniques génère des chroniques par un assemblage des chroniques élémentaires identifiées précédemment. Cet assemblage de chroniques élémentaires se fait grâce aux opérations, c'est-à-dire à une fusion des nœuds suivant un critère de similarité. L'indice de Jaccard est calculé entre chaque nœud de la chronique en cours de reconstitution et les indices au-dessus d'un certain seuil, appelé $seuil_{sim}$, génère une opération à appliquer sur la chronique. Les opérations devant être appliquées sur la chronique solution sont stockées dans une liste ordonnée et sont appliquées séquentiellement. L'ordre des opérations a son importance dans le résultat de CDIRe car l'application d'une opération peut rendre une autre opération précédemment sélectionnée obsolète. Plusieurs heuristiques pour définir un ordre sur ces opérations sont proposées.

Le stockage des opérations dans une liste ordonnée rend cette étape de reconstitution de chroniques similaire à un algorithme glouton. Ce qui contribue à l'efficacité de la complexité algorithmique de CDIRe. En contrepartie d'un coût sur l'efficacité de l'étape de reconstitution, permettre un retour en arrière sur l'application

des opérations peut significativement améliorer la qualité des chroniques générées. Pour cela, une structure de données plus élaborée qu'une simple liste ordonnée pour le stockage des opérations est nécessaire. Une telle structure peut prendre la forme d'un arbre. Avec ce type de structure, une heuristique différente d'application des opérations peut être adoptée. Par exemple, *appliquer le plus d'opérations possibles* est une heuristique particulièrement pertinente. De plus, une analyse a priori de cette structure de données peut donner des indices quant aux opérations les plus pertinentes à appliquer.

Perspective 3. *Est-ce qu'une liste ordonnée est la structure de données la plus pertinente pour stocker les opérations ? Comment une structure de données plus complexe peut améliorer la qualité des chroniques reconstituées ?*

Une analyse des performances ainsi que des résultats de CDIRe sur un jeu de données provenant d'une application réelle est fournie dans le chapitre 3. Cette observation du comportement de CDIRe dans des conditions réelles nous permet de conclure que CDIRe répond à la problématique soulevée par ce mémoire. Enfin, un bilan des influences des différents paramètres de CDIRe est fournie et offre à l'utilisateur de CDIRe des directions quant aux choix de ces paramètres afin d'obtenir les résultats désirés.

Un problème inattendu, le problème de sur-apprentissage des chroniques, est mis en avant par l'analyse des résultats. En effet, les chroniques générées sont bien souvent beaucoup plus complexes que la taille des données ne le permet. Une première approche pour atténuer ce phénomène est proposée avec les différentes méthodes de calcul des bornes des contraintes temporelles des chroniques élémentaires. Néanmoins, cette approche ne résout pas de manière significative le problème de sur-apprentissage et d'autres solutions doivent être apportées pour résoudre cette problématique.

Perspective 4. *Comment résoudre le problème de sur-apprentissage de chroniques ?*

La deuxième contribution de ce mémoire vient d'une représentation innovante d'une chronique. En effet, il est possible, avec une méthode mathématique provenant du domaine de la réduction de dimension appelée projection aléatoire [Vempala 2005], d'associer une chronique à un polytope dans un espace euclidien [Sahuguède 2018a]. Ce polytope prend en compte à la fois le domaine spatial et temporel d'une chronique. Cette contribution provient d'une recherche exploratoire et de nombreuses questions théoriques restent ouvertes. Néanmoins, les résultats présentés dans le chapitre 4 sont prometteurs. La représentation des chroniques dans un espace euclidien dans lequel il est aisé de comparer des points et des ensembles offre de remarquables perspectives pour l'analyse des chroniques. En effet, de nombreuses questions auxquelles il est difficile de répondre avec les outils offerts par l'analyse des chroniques deviennent des problèmes triviaux en considérant des ensembles dans un espace euclidien. L'intégration de cette nouvelle représentation

de chroniques dans le processus de découverte de chroniques est un axe de travail riche ouvert par cette contribution.

Perspective 5. *Suivant quelles modalités la projection de chroniques dans un espace euclidien k-dimensionnel peut aider à améliorer les résultats obtenus par le processus de découverte de chroniques ?*

Un axe de travail qui n'est pas abordé dans ce mémoire mais qui présente néanmoins des perspectives intéressantes pour le processus de découverte de chroniques concerne la phase de pré-traitement des données d'entrée. CDIRe ainsi que l'ensemble des algorithmes de découverte de chroniques à notre connaissance exploitent des données discrètes qui peuvent s'écrire sous la forme de séquences temporelles. Néanmoins, de nombreux systèmes dynamiques modernes génèrent des données hétérogènes où les informations d'intérêt sont représentées par des séries temporelles numériques en complément de séquences temporelles. Une étape de pré-traitement des données est nécessaire pour uniformiser ces données et les transformer dans un format utilisable par les algorithmes de découverte de chroniques. La discrétisation de séries temporelles, qu'elles soient uni-variables ou multi-variables, est un domaine de recherche actif ces dernières années [Mörchen 2005, Lin 2007, Moskovitch 2015].

En plus d'uniformiser les données d'entrée, le modèle de discrétisation choisi doit permettre d'améliorer les résultats du processus de découverte de chroniques. De plus, dans le cadre formel des chroniques, l'étape de reconnaissance de chroniques doit être prise en compte dans le choix du modèle de discrétisation. En effet, la même discrétisation utilisée pour la découverte de chroniques doit nécessairement être appliquée sur le flux de données en ligne provenant du système supervisé par l'algorithme de reconnaissance de chroniques. Cette caractéristique associée aux chroniques doit être prise en compte pour le modèle de discrétisation, celui-ci doit être efficace en ligne afin de ne pas retarder significativement la reconnaissance de chroniques. À notre connaissance, personne ne s'est intéressé au problème de la discrétisation appliquée à la découverte de chroniques.

Perspective 6. *Comment construire un modèle de discrétisation permettant d'améliorer les résultats obtenus par le processus de découverte de chroniques tout en fournissant une discrétisation en ligne efficace ?*

Annexe A

Compléments à la reconstitution de chroniques

A.1 Preuve de la proposition 2.1

Dans cette section la preuve complète de la proposition 2.1 (page 56) est fournie. Pour rappel, la proposition 2.1 est la suivante :

Proposition 2.1. *Soient $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ deux sous-chroniques indépendantes élémentaires. Si une des opérations est traitée parmi les quatre opérations possibles entre $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$, les trois autres opérations possibles ne sont plus cohérentes.*

Démonstration. Soit $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ deux chroniques élémentaires. Soit $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ la chronique constituée de $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ où $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$ sont des sous-chroniques indépendantes de $\mathcal{C}$. Les opérations n'impliquant que des nœuds avec les mêmes événements, il suffit de montrer le résultat dans l'hypothèse où $\mathcal{E} = \{e_1 = e, e_2 = e, e_3 = e, e_4 = e\}$, les autres cas découlent toujours d'un sous-ensemble de scénarios listés dans ce cas. La chronique $\mathcal{C}$ est composé de $\mathcal{E} = \{e_1 = e, e_2 = e, e_3 = e, e_4 = e\}$ et $\mathcal{T} = \{\tau_{(1,2)} = e_1[t^-_{(1,2)}, t^+_{(1,2)}]e_2, \tau_{(3,4)} = e_3[t^-_{(3,4)}, t^+_{(3,4)}]e_4\}$. Quatre opérations possibles entre les deux sous-chroniques indépendantes $s\mathcal{C}_1^\alpha$ et $s\mathcal{C}_2^\alpha$: $\omega_{(1,3)}$, $\omega_{(2,4)}$, $\omega_{(1,4)}$ et $\omega_{(2,3)}$. En considérant toutes les combinaisons, il y a 4×3 scénarios possibles. Chacun de ces scénarios est détaillé dans le reste de cette preuve. Dans les trois premiers scénarios, l'opération $\omega_{(1,3)}$ est appliquée en premier et la chronique représentée sur la figure A.1 est obtenue.

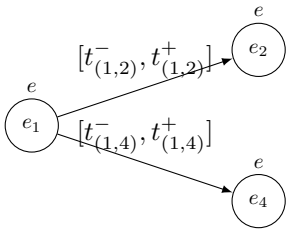


Figure A.1 – Chronique obtenue après application de l'opération $\omega_{(1,3)}$

— Scénario 1, $\omega_{(1,3)}$ puis $\omega_{(2,4)}$: $\omega_{(2,4)}$ n'est plus cohérente car les deux nœuds $\nu_{\mathcal{C}}(e_2)$ et $\nu_{\mathcal{C}}(e_4)$ ont tous les deux une contrainte temporelle en provenance du même nœud $\nu_{\mathcal{C}}(e_1)$.

— Scénario 2, $\omega_{(1,3)}$ puis $\omega_{(1,4)}$: $\omega_{(1,4)}$ n'est plus cohérente car il existe une contrainte temporelle entre les deux nœuds $\nu_{\mathcal{C}}(e_1)$ et $\nu_{\mathcal{C}}(e_4)$.

— Scénario 3, $\omega_{(1,3)}$ puis $\omega_{(2,3)}$: ici, $\omega_{(2,3)}$ devient $\omega_{(2,1)}$ après traitement de $\omega_{(1,3)}$. $\omega_{(2,1)}$ n'est plus cohérente car il existe une contrainte temporelle entre les deux nœuds $\nu_{\mathcal{C}}(e_2)$ et $\nu_{\mathcal{C}}(e_1)$.

Dans les trois scénarios suivants, l'opération $\omega_{(2,4)}$ est traitée en premier et la chronique représentée sur la figure A.2 est obtenue.

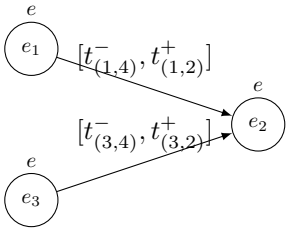


FIGURE A.2 – Chronique obtenue après application de l'opération $\omega_{(2,4)}$

— Scénario 4, $\omega_{(2,4)}$ puis $\omega_{(1,3)}$: même situation que le scénario 1, $\omega_{(1,3)}$ n'est plus cohérente.

— Scénario 5, $\omega_{(2,4)}$ puis $\omega_{(1,4)}$: ici, $\omega_{(1,4)}$ devient $\omega_{(1,2)}$ après traitement de $\omega_{(2,4)}$. $\omega_{(1,2)}$ n'est plus cohérente car il existe une contrainte temporelle entre les deux nœuds $\nu_{\mathcal{C}}(e_1)$ et $\nu_{\mathcal{C}}(e_2)$.

— Scénario 6, $\omega_{(2,4)}$ puis $\omega_{(2,3)}$: $\omega_{(2,3)}$ n'est plus cohérente car il existe une contrainte temporelle entre les deux nœuds $\nu_{\mathcal{C}}(e_2)$ et $\nu_{\mathcal{C}}(e_3)$.

Dans les trois scénarios qui suivent, l'opération $\omega_{(1,4)}$ est traitée en premier et la chronique représentée sur la figure A.3 est obtenue.

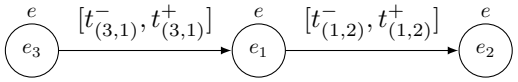


FIGURE A.3 – Chronique obtenue après application de l'opération $\omega_{(1,4)}$

— Scénario 7, $\omega_{(1,4)}$ puis $\omega_{(1,3)}$: même situation que le scénario 2, $\omega_{(1,3)}$ n'est plus cohérente.

— Scénario 8, $\omega_{(1,4)}$ puis $\omega_{(2,4)}$: même situation que le scénario 5, $\omega_{(2,4)}$ n'est plus cohérente.

— Scénario 9, $\omega_{(1,4)}$ puis $\omega_{(2,3)}$: $\omega_{(2,3)}$ n'est plus cohérente car les deux nœuds $\nu_{\mathcal{C}}(e_2)$ et $\nu_{\mathcal{C}}(e_3)$ ont tous les deux une contrainte temporelle avec un même nœud $\nu_{\mathcal{C}}(e_1)$.

Dans les trois derniers scénarios, l'opération $\omega_{(2,3)}$ est traitée en premier et la chronique représentée sur la figure A.4 est obtenue.

— Scénario 10, $\omega_{(2,3)}$ puis $\omega_{(1,3)}$: même situation que le scénario 3, $\omega_{(1,3)}$ n'est plus cohérente.

— Scénario 11, $\omega_{(2,3)}$ puis $\omega_{(2,4)}$: même situation que le scénario 6, $\omega_{(2,4)}$ n'est plus cohérente.

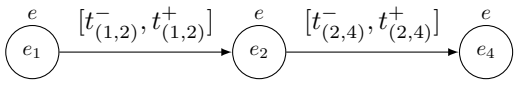


FIGURE A.4 – Chronique obtenue après application de l'opération $\omega_{(2,3)}$

— Scénario 12, $\omega_{(2,3)}$ puis $\omega_{(1,4)}$: même situation que le scénario 9, $\omega_{(1,4)}$ n'est plus cohérente.

Ainsi, si une opération est traitée parmi les quatre opérations possibles entre deux sous-chroniques indépendantes élémentaires , les trois autres opérations possibles deviennent incohérentes. □

A.2 Résultat du calcul des indices de Jaccard dans une base de six chroniques élémentaires

La structure de donnée $\mathcal{J}(\mathcal{S})$ est composée des 60 indices de Jaccard calculés entre les nœuds des différents couples des sous-chroniques indépendantes élémentaires représentée figure 2.7 (page 58). Les indices supérieurs au seuil $seuil_{sim} = 0.9$ sont mis en évidence en gras.

$$(\mathcal{C}^\alpha_1, \mathcal{C}^\alpha_2) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_2}(e_2), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_2}(e_9), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_2}(e_2), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_2}(e_9), \mathcal{S}) = \mathbf{0.94}, \end{cases} \quad (\mathcal{C}^\alpha_1, \mathcal{C}^\alpha_3) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_3}(e_3), \mathcal{S}) = 0.83, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_3}(e_{10}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_3}(e_3), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_3}(e_{10}), \mathcal{S}) = \mathbf{0.94}, \end{cases}$$

$$(\mathcal{C}^\alpha_1, \mathcal{C}^\alpha_4) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_4}(e_{11}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_4}(e_{12}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_4}(e_{11}), \mathcal{S}) = \mathbf{1}, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_4}(e_{12}), \mathcal{S}) = 0.88, \end{cases} \quad (\mathcal{C}^\alpha_1, \mathcal{C}^\alpha_5) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_5}(e_4), \mathcal{S}) = 0.83, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_5}(e_5), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_5}(e_4), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_5}(e_5), \mathcal{S}) = 0, \end{cases}$$

$$(\mathcal{C}^\alpha_1, \mathcal{C}^\alpha_6) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_6}(e_6), \mathcal{S}) = 0.77, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_1), \nu_{\mathcal{C}^\alpha_6}(e_7), \mathcal{S}) = \mathbf{1}, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_6}(e_6), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_1}(e_8), \nu_{\mathcal{C}^\alpha_6}(e_7), \mathcal{S}) = 0, \end{cases} \quad (\mathcal{C}^\alpha_2, \mathcal{C}^\alpha_3) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_3}(e_3), \mathcal{S}) = 0.88, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_3}(e_{10}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_3}(e_3), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_3}(e_{10}), \mathcal{S}) = \mathbf{1}, \end{cases}$$

$$(\mathcal{C}^\alpha_2, \mathcal{C}^\alpha_4) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_4}(e_{11}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_4}(e_{12}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_4}(e_{11}), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_4}(e_{12}), \mathcal{S}) = \mathbf{0.94}, \end{cases} \quad (\mathcal{C}^\alpha_2, \mathcal{C}^\alpha_5) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_5}(e_4), \mathcal{S}) = 0.88, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_5}(e_5), \mathcal{S}) = \mathbf{1}, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_5}(e_4), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_5}(e_5), \mathcal{S}) = 0, \end{cases}$$

$$(\mathcal{C}^\alpha_2, \mathcal{C}^\alpha_6) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_6}(e_6), \mathcal{S}) = 0.83, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_2), \nu_{\mathcal{C}^\alpha_6}(e_7), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_6}(e_6), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_2}(e_9), \nu_{\mathcal{C}^\alpha_6}(e_7), \mathcal{S}) = 0, \end{cases} \quad (\mathcal{C}^\alpha_3, \mathcal{C}^\alpha_4) : \begin{cases} J(\nu_{\mathcal{C}^\alpha_3}(e_3), \nu_{\mathcal{C}^\alpha_4}(e_{11}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_3}(e_3), \nu_{\mathcal{C}^\alpha_4}(e_{12}), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}^\alpha_3}(e_{10}), \nu_{\mathcal{C}^\alpha_4}(e_{11}), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}^\alpha_3}(e_{10}), \nu_{\mathcal{C}^\alpha_4}(e_{12}), \mathcal{S}) = \mathbf{0.94}, \end{cases}$$

$$(\mathcal{C}_3^\alpha, \mathcal{C}_5^\alpha) : \begin{cases} J(\nu_{\mathcal{C}_3^\alpha}(e_3), \nu_{\mathcal{C}_5^\alpha}(e_4), \mathcal{S}) = \mathbf{1}, \\ J(\nu_{\mathcal{C}_3^\alpha}(e_3), \nu_{\mathcal{C}_5^\alpha}(e_5), \mathcal{S}) = 0.88, \\ J(\nu_{\mathcal{C}_3^\alpha}(e_{10}), \nu_{\mathcal{C}_5^\alpha}(e_4), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_3^\alpha}(e_{10}), \nu_{\mathcal{C}_5^\alpha}(e_5), \mathcal{S}) = 0, \end{cases} \quad (\mathcal{C}_3^\alpha, \mathcal{C}_6^\alpha) : \begin{cases} J(\nu_{\mathcal{C}_3^\alpha}(e_3), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}_3^\alpha}(e_3), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0.83, \\ J(\nu_{\mathcal{C}_3^\alpha}(e_{10}), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_3^\alpha}(e_{10}), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0, \end{cases}$$

$$(\mathcal{C}_4^\alpha, \mathcal{C}_5^\alpha) : \begin{cases} J(\nu_{\mathcal{C}_4^\alpha}(e_{11}), \nu_{\mathcal{C}_5^\alpha}(e_4), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_4^\alpha}(e_{11}), \nu_{\mathcal{C}_5^\alpha}(e_5), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_4^\alpha}(e_{12}), \nu_{\mathcal{C}_5^\alpha}(e_4), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_4^\alpha}(e_{12}), \nu_{\mathcal{C}_5^\alpha}(e_5), \mathcal{S}) = 0, \end{cases} \quad (\mathcal{C}_4^\alpha, \mathcal{C}_6^\alpha) : \begin{cases} J(\nu_{\mathcal{C}_4^\alpha}(e_{11}), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_4^\alpha}(e_{11}), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_4^\alpha}(e_{12}), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = 0, \\ J(\nu_{\mathcal{C}_4^\alpha}(e_{12}), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0, \end{cases}$$

$$(\mathcal{C}_5^\alpha, \mathcal{C}_6^\alpha) : \begin{cases} J(\nu_{\mathcal{C}_5^\alpha}(e_4), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = \mathbf{0.94}, \\ J(\nu_{\mathcal{C}_5^\alpha}(e_4), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = 0.83, \\ J(\nu_{\mathcal{C}_5^\alpha}(e_5), \nu_{\mathcal{C}_6^\alpha}(e_6), \mathcal{S}) = 0.83, \\ J(\nu_{\mathcal{C}_5^\alpha}(e_5), \nu_{\mathcal{C}_6^\alpha}(e_7), \mathcal{S}) = \mathbf{0.94}. \end{cases}$$

A.3 Exemple détaillé d'application d'un ensemble d'opérations

Soit la $\mathcal{C}_{solution}$ la chronique solution initialisée représentée sur la figure A.5. La chronique initiale $\mathcal{C}_{solution}$ est une chronique de taille 12 où $\mathcal{E}_{solution} = \{e_1 = a, e_2 = a, e_3 = a, e_4 = a, e_5 = a, e_6 = a, e_7 = a, e_8 = b, e_9 = b, e_{10} = b, e_{11} = b, e_{12} = b\}$ et $\mathcal{T}_{solution} = \{\tau_{(1,8)} = e_1[-176, -161]e_8, \tau_{(2,9)} = e_2[-62, -52]e_9, \tau_{(3,10)} = e_3[48, 69]e_{10}, \tau_{(4,5)} = e_4[106, 130]e_5, \tau_{(6,7)} = e_6[216, 242]e_7, \tau_{(11,12)} = e_{11}[106, 118]e_{12}\}$. L'ensemble ordonnés d'opérations $\Omega_{tsi}(\mathcal{S})$ défini dans l'exemple 2.14 doit être appliquée :

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \omega_{(4,6)}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(2,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(10,11)}, \omega_{(1,5)}, \omega_{(2,7)}\}. \quad (A.1)$$

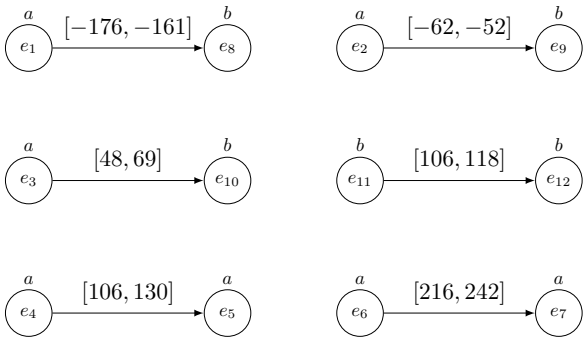


FIGURE A.5 – Chronique initiale.

La première opération traitée est $\omega_{(3,4)}$. Celle-ci est cohérente et est donc appliquée sur $\mathcal{C}_{solution}$. Après application, la chronique est représentée sur la figure A.6. Le nœud $\nu_{\mathcal{C}_{solution}}(e_4)$ n'existe plus et toutes les opérations sur ce nœud ont mainte-

nant lieu sur $\nu_{\mathcal{C}_{solution}}(e_3)$. $\omega_{(4,6)}$ devient $\omega_{(3,6)}$, or cette opération est déjà présente dans $\Omega_{tsi}(\mathcal{S})$. La deuxième opération $\omega_{(3,6)}$ n'est pas traitée est peut être retirée de l'ensemble $\Omega_{tsi}(\mathcal{S})$:

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \cancel{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(2,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(10,11)}, \omega_{(1,5)}, \omega_{(2,7)}\}. \quad (A.2)$$

Dans cet exemple, les opérations qui ont été traitées sont représentées en gris, les changements sur les nœuds sont également représentés en gris, et les opérations qui ne sont pas appliquées sont ~~barrées~~.

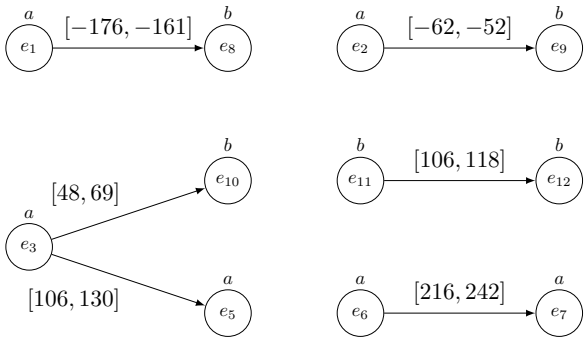


FIGURE A.6 – Chronique après application de l'opération $\omega_{(3,4)}$.

La seconde opération est $\omega_{(1,2)}$. C'est une opération cohérente qui est appliquée sur $\mathcal{C}_{solution}$ qui devient la chronique représentée sur la figure A.7. Les opérations $\omega_{(2,5)}$ et $\omega_{(2,7)}$ deviennent $\omega_{(1,5)}$ et $\omega_{(1,7)}$. Ces opérations sont déjà représentées dans l'ensemble des opérations $\Omega_{tsi}(\mathcal{S})$ et les opérations en double sont retirées :

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \cancel{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(10,11)}, \cancel{\omega_{(1,5)}}, \cancel{\omega_{(1,7)}}\}. \quad (A.3)$$

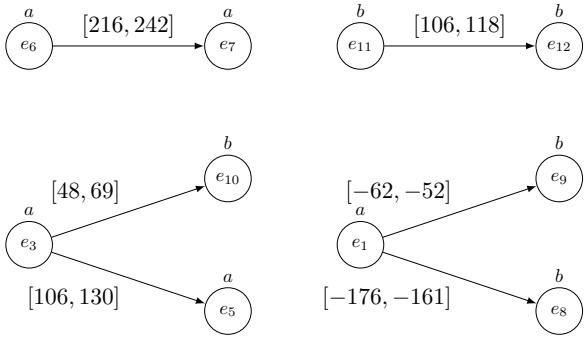


FIGURE A.7 – Chronique après application de l'opération $\omega_{(1,2)}$.

L'opération suivante et $\omega_{(3,6)}$. C'est aussi une opération cohérente et après application de celle-ci, la chronique représentée sur la figure A.8 est obtenue. Il n'y

pas de changement sur les opérations de l'ensemble $\Omega_{tsi}(\mathcal{S})$:

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \cancel{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \omega_{(10,12)}, \omega_{(8,10)}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \omega_{(10,11)}, \cancel{\omega_{(1,5)}}, \cancel{\omega_{(1,7)}}\}. \quad (A.4)$$

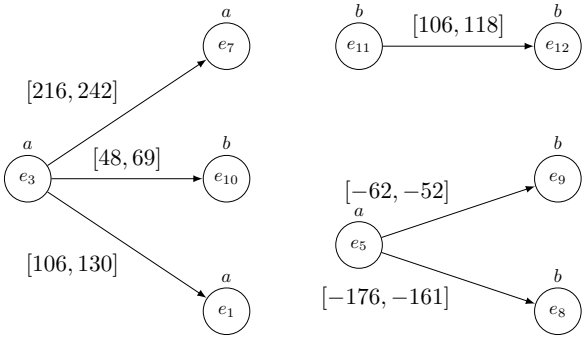


FIGURE A.8 – Chronique après application de l'opération $\omega_{(3,6)}$.

L'opération en cours est maintenant $\omega_{(9,10)}$ qui est également cohérente. Son application produit la chronique représentée sur la figure A.9. Les opérations $\omega_{(10,12)}$, $\omega_{(8,10)}$ et $\omega_{(10,11)}$ deviennent $\omega_{(9,12)}$, $\omega_{(8,9)}$ et $\omega_{(9,11)}$ et les doublons sont retirés :

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \cancel{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \cancel{\omega_{(9,12)}}, \cancel{\omega_{(8,9)}}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \cancel{\omega_{(9,11)}}, \cancel{\omega_{(1,5)}}, \cancel{\omega_{(1,7)}}\}. \quad (A.5)$$

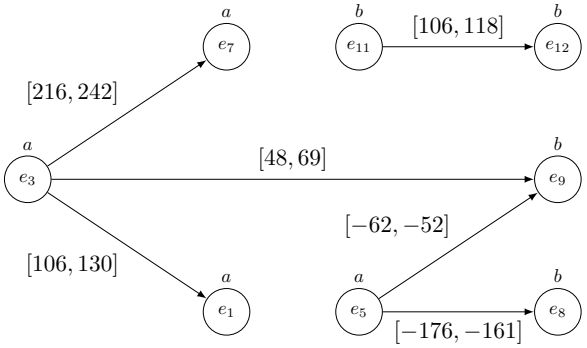


FIGURE A.9 – Chronique après application de l'opération $\omega_{(9,10)}$.

L'opération qui suit est $\omega_{(9,12)}$ qui est cohérente. $\mathcal{C}_{solution}$ devient la chronique représentée sur la figure A.10. Il n'y a pas de changement sur $\Omega_{tsi}(\mathcal{S})$:

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \cancel{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,9)}, \cancel{\omega_{(9,12)}}, \cancel{\omega_{(8,9)}}, \omega_{(5,7)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \cancel{\omega_{(9,11)}}, \cancel{\omega_{(1,5)}}, \cancel{\omega_{(1,7)}}\}. \quad (A.6)$$

L'opération suivante, $\omega_{(8,9)}$, n'est pas cohérente car les deux nœuds $\nu_{\mathcal{C}_{solution}}(e_8)$

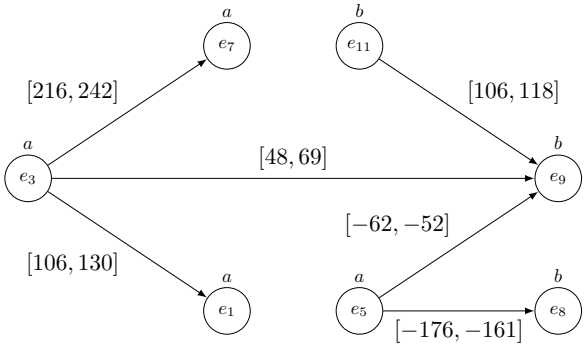


FIGURE A.10 – Chronique après application de l'opération $\omega_{(9,12)}$.

et $\nu_{\mathcal{C}_{solution}}(e_9)$ ont tous les deux une contrainte temporelle en provenance d'un même nœud $\nu_{\mathcal{C}_{solution}}(e_5)$ et n'est donc pas appliquée.

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \underline{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \underline{\omega_{(8,9)}}, \underline{\omega_{(9,12)}}, \underline{\omega_{(8,9)}},$$
$$\omega_{(5,7)}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \underline{\omega_{(9,11)}}, \underline{\omega_{(1,5)}}, \underline{\omega_{(1,7)}}\}. \quad \text{(A.7)}$$

$\omega_{(5,7)}$ n'est pas cohérente. En effet, si $\omega_{(5,7)}$ est appliquée, le chemin $\nu_{\mathcal{C}_{solution}}(e_3) \rightarrow \nu_{\mathcal{C}_{solution}}(e_9)$ ($[48, 69]$) est très différent du chemin $\nu_{\mathcal{C}_{solution}}(e_3) \rightarrow \nu_{\mathcal{C}_{solution}}(e_5) \rightarrow \nu_{\mathcal{C}_{solution}}(e_9)$ ($[216, 242] + [-62, -52] = [154, 190]$). $\omega_{(5,7)}$ n'est pas appliquée.

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \underline{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \underline{\omega_{(8,9)}}, \underline{\omega_{(9,12)}}, \underline{\omega_{(8,9)}},$$
$$\underline{\omega_{(5,7)}}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,11)}, \underline{\omega_{(9,11)}}, \underline{\omega_{(1,5)}}, \underline{\omega_{(1,7)}}\}. \quad \text{(A.8)}$$

L'opération en cours est maintenant $\omega_{(8,11)}$ et est cohérente. Son application sur la chronique $\mathcal{C}_{sol}$ donne la chronique représentée sur la figure A.11. L'opération $\omega_{(9,11)}$ devient $\omega_{(9,8)}$:

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \underline{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \underline{\omega_{(8,9)}}, \underline{\omega_{(9,12)}}, \underline{\omega_{(8,9)}},$$
$$\underline{\omega_{(5,7)}}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,8)}, \underline{\omega_{(9,11)}}, \underline{\omega_{(1,5)}}, \underline{\omega_{(1,7)}}\}. \quad \text{(A.9)}$$

L'opération suivante est $\omega_{(1,5)}$. C'est une opération cohérente et produit la chronique représentée sur la figure A.12.

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \underline{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \underline{\omega_{(8,9)}}, \underline{\omega_{(9,12)}}, \underline{\omega_{(8,9)}},$$
$$\underline{\omega_{(5,7)}}, \omega_{(8,11)}, \omega_{(1,5)}, \omega_{(1,7)}, \omega_{(9,8)}, \underline{\omega_{(9,11)}}, \underline{\omega_{(1,5)}}, \underline{\omega_{(1,7)}}\}. \quad \text{(A.10)}$$

L'opération qui suit, $\omega_{(1,7)}$, n'est pas cohérente car les nœuds $\nu_{\mathcal{C}_{solution}}(e_1)$ et $\nu_{\mathcal{C}_{solution}}(e_7)$ ont tous les deux une contrainte temporelle provenant d'un nœud en commun $\nu_{\mathcal{C}_{solution}}(e_3)$.

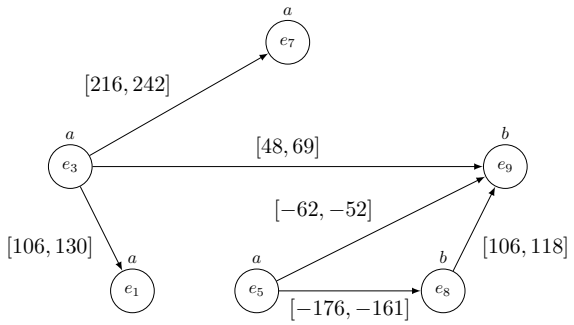


FIGURE A.11 – Chronique après application de l'opération $\omega_{(8,11)}$.

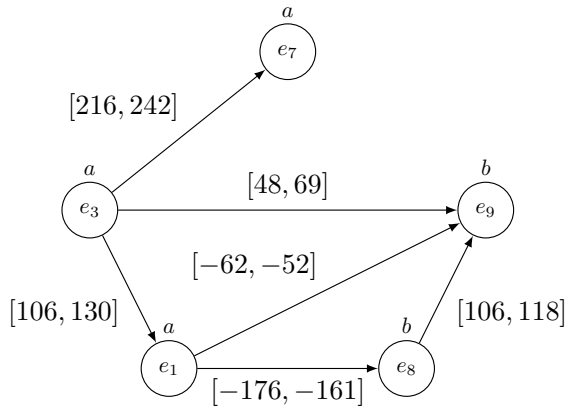


FIGURE A.12 – Chronique après application de l'opération $\omega_{(1,5)}$.

$$\Omega_{tsi}(\mathcal{S}) = \{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \cancel{\omega_{(3,6)}}, \omega_{(9,10)}, \omega_{(9,12)}, \cancel{\omega_{(8,9)}}, \cancel{\omega_{(9,12)}}, \cancel{\omega_{(8,9)}}, \cancel{\omega_{(5,7)}}, \omega_{(8,11)}, \omega_{(1,5)}, \cancel{\omega_{(1,7)}}, \omega_{(9,8)}, \cancel{\omega_{(9,11)}}, \cancel{\omega_{(1,5)}}, \cancel{\omega_{(1,7)}}\}. \quad \text{(A.11)}$$

Enfin, la dernière opération $\omega_{(9,8)}$ n'est pas cohérente car il existe déjà une contrainte temporelle entre les deux nœuds $\nu_{\mathcal{C}_{solution}}(e_9)$ et $\nu_{\mathcal{C}_{solution}}(e_8)$.

Le résultat final de l'ensemble des opérations $\Omega_{tsi}(\mathcal{S})$ sur la chronique $\mathcal{C}_{solution}$ est représentée sur la figure A.12. C'est une chronique de taille 5 où $\mathcal{E}_{solution} = \{e_1 = a, e_3 = a, e_7 = a, e_8 = b, e_9 = b\}$ et $\mathcal{T}_{solution} = \{\tau_{(1,8)} = e_1[-176, -161]e_8, \tau_{(1,9)} = e_1[-62, -52]e_9, \tau_{(3,9)} = e_3[48, 69]e_9, \tau_{(3,1)} = e_3[106, 130]e_1, \tau_{(3,7)} = e_3[216, 242]e_7, \tau_{(8,9)} = e_8[106, 118]e_9\}$. Sur les 17 opérations présentes initialement dans l'ensemble $\Omega_{tsi}(\mathcal{S})$, seul 7 opérations ont été réellement appliquées. Ces opérations sont :

$$\{\omega_{(3,4)}, \omega_{(1,2)}, \omega_{(3,6)}, \omega_{(9,10)}, \omega_{(9,12)}, \omega_{(8,11)}, \omega_{(2,5)}\}. \quad \text{(A.12)}$$

Au final, la chronique a été réduite de taille 12 à taille 5.

Annexe B

Compléments à l'analyse des performances

B.1 Résultats des mesures de performances pour l'analyse des paramètres de DBSCAN

Table B.1 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *pick-up*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		$MinPts$							
		3	4	5	6	7	8	9	10
ε	5	10051	10305	10506	10518	10518	10518	10520	10520
	10	9487	10119	10441	10510	10510	10510	10510	10510
	15	8689	9770	10336	10507	10507	10507	10507	10507
	20	8059	9406	10267	10405	10507	10507	10507	10507
	25	7485	8909	9960	10309	10491	10503	10503	10503
	30	6814	8121	9288	9907	10285	10415	10444	10486
	35	5945	7508	8592	9365	10119	10227	10321	10403
	40	5183	7070	8110	8868	9584	9870	10066	10230
	45	3093	4867	6746	7694	8843	9248	9643	9879
	50	2742	4318	5827	7407	8588	9041	9388	9616

(b) Nombre de chroniques reconstituées.

		$MinPts$							
		3	4	5	6	7	8	9	10
ε	5	27	10	5	4	4	4	3	3
	10	33	10	4	4	4	4	4	4
	15	53	21	6	3	3	3	3	3
	20	53	27	12	3	3	3	3	3
	25	70	48	23	18	4	4	4	4
	30	81	52	39	24	16	8	5	5
	35	92	61	49	34	27	17	5	5
	40	83	48	44	34	24	37	25	10
	45	84	78	52	43	28	22	11	15
	50	76	78	53	32	21	18	16	19

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	10	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	15	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	20	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	25	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	30	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	35	1.0	1.0	1.0	0.68	0.43	0.43	0.43	0.43
	40	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	45	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
	50	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3

TABLE B.2 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *put-down*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	9851	10194	10238	10238	10238	10238	10238	10238
	10	9111	9755	10083	10231	10233	10233	10233	10233
	15	8402	9379	9840	10187	10229	10229	10229	10229
	20	7433	8968	9462	10081	10227	10229	10229	10229
	25	6433	7949	9083	9843	10164	10224	10228	10228
	30	5818	7310	8455	9352	10025	10204	10226	10226
	35	5144	6722	7896	8800	9833	10129	10223	10223
	40	4551	5844	7286	8127	9436	10075	10208	10215
	45	3905	5136	6449	7530	8671	9778	10136	10178
	50	3396	4552	5616	6725	7866	9042	9886	10127

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	33	8	2	2	2	2	2	2
	10	54	26	10	3	3	3	3	3
	15	67	32	13	3	3	3	3	3
	20	96	44	22	8	3	3	3	3
	25	113	74	37	15	6	4	4	4
	30	93	67	56	28	11	4	4	4
	35	114	75	53	33	16	6	4	4
	40	116	99	67	53	20	10	5	5
	45	144	132	102	76	53	18	7	5
	50	129	112	102	124	76	28	12	3

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	10	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
	15	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
	20	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
	25	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
	30	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
	35	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
	40	0.54	0.92	0.6	0.6	0.6	0.6	0.6	0.6
	45	0.62	0.62	0.85	0.75	0.75	0.75	0.75	0
	50	0.62	0.62	0.54	0.75	0.75	0.75	0.75	0.75

TABLE B.3 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *stack*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	37311	39407	40059	40262	40367	40376	40379	40379
	10	32067	36224	38565	39498	39938	40151	40273	40342
	15	28493	33144	36031	37852	38615	39289	39792	39988
	20	24103	28774	32379	34884	36641	37557	38340	39035
	25	19863	24982	28780	31576	33894	35696	36884	37703
	30	15779	21034	25268	28204	30841	33415	34644	36125
	35	13539	18660	23217	26005	28723	31099	33314	34561
	40	11546	15512	20872	23799	26936	29153	31429	33335
	45	10168	13293	17944	21479	24677	26725	29432	30968
	50	9347	12013	16555	20183	23072	25393	28129	29890

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	395	164	53	19	5	5	6	6
	10	548	349	188	89	44	25	9	6
	15	685	461	307	218	162	113	69	36
	20	667	542	391	279	222	201	151	97
	25	528	466	424	325	279	225	184	173
	30	440	338	388	342	266	220	187	127
	35	415	365	345	349	343	324	218	201
	40	295	331	281	273	229	277	231	207
	45	237	253	266	249	225	230	251	219
	50	277	300	297	253	232	232	213	244

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
	10	0.83	0.83	0.29	0.29	0.29	0.29	0.29	0.29
	15	0.91	0.91	0.83	0.83	0.83	0.83	0.83	0.83
	20	0.91	0.83	0.83	0.83	0.83	0.83	0.83	0.83
	25	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
	30	0.91	0.91	0.83	0.83	0.91	0.83	0.83	0.83
	35	1.0	0.83	0.83	0.91	1.0	0.83	0.83	0.91
	40	0.56	0.46	0.83	0.83	1.0	0.83	0.65	0.83
	45	0.37	0.37	0.91	0.83	0.83	1.0	0.83	0.83
	50	0.46	0.37	0.83	0.91	0.83	0.83	0.83	0.83

TABLE B.4 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *unstack*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	38169	40072	40630	40799	40900	40939	40961	40961
	10	33734	37941	39514	40266	40584	40793	40867	40879
	15	29776	33934	37187	38631	39365	39936	40362	40614
	20	24040	29031	32642	35298	36982	38223	39076	39649
	25	17756	23327	27529	30348	33214	35508	36651	37598
	30	15762	21031	25813	28587	31389	34206	35632	36936
	35	14170	19040	24214	27293	29814	32938	34778	35954
	40	12516	16918	21999	25382	27774	30531	33442	34697
	45	10561	14069	18963	22753	25515	27790	31265	33050
	50	9202	12634	17568	21049	24199	26610	29834	31901

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	274	104	41	21	13	6	5	5
	10	440	254	124	57	29	15	15	10
	15	599	485	294	205	122	77	43	27
	20	633	498	364	296	241	202	135	92
	25	552	461	443	383	308	218	193	177
	30	522	456	429	394	319	253	209	161
	35	490	451	412	414	327	272	237	171
	40	439	442	455	408	388	342	274	211
	45	415	437	428	397	386	366	305	239
	50	441	459	394	422	359	328	303	263

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
	10	0.16	0.16	0.16	0.16	1.0	0.58	0.58	0.58
	15	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	20	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	25	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	30	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	35	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	40	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	45	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
	50	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93

TABLE B.5 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *move-left*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	6581	6645	6645	6645	6645	6645	6645	6650
	10	6445	6645	6645	6645	6645	6645	6645	6645
	15	6445	6645	6645	6645	6645	6645	6645	6645
	20	6445	6645	6645	6645	6645	6645	6645	6645
	25	6445	6645	6645	6645	6645	6645	6645	6645
	30	6445	6645	6645	6645	6645	6645	6645	6645
	35	6445	6645	6645	6645	6645	6645	6645	6645
	40	6445	6645	6645	6645	6645	6645	6645	6645
	45	6442	6645	6645	6645	6645	6645	6645	6645
	50	6380	6645	6645	6645	6645	6645	6645	6645

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	2	1	1	1	1	1	1	2
	10	2	1	1	1	1	1	1	1
	15	2	1	1	1	1	1	1	1
	20	2	1	1	1	1	1	1	1
	25	2	1	1	1	1	1	1	1
	30	2	1	1	1	1	1	1	1
	35	2	1	1	1	1	1	1	1
	40	2	1	1	1	1	1	1	1
	45	2	1	1	1	1	1	1	1
	50	2	1	1	1	1	1	1	1

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0
	20	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0
	30	0	0	0	0	0	0	0	0
	35	0	0	0	0	0	0	0	0
	40	0	0	0	0	0	0	0	0
	45	0	0	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0

Table B.6 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *move-right*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	6645	6645	6645	6645	6649	6653	6657	6667
	10	6645	6645	6645	6645	6645	6645	6645	6645
	15	6645	6645	6645	6645	6645	6645	6645	6645
	20	6645	6645	6645	6645	6645	6645	6645	6645
	25	6641	6645	6645	6645	6645	6645	6645	6645
	30	6575	6645	6645	6645	6645	6645	6645	6645
	35	6473	6645	6645	6645	6645	6645	6645	6645
	40	6426	6645	6645	6645	6645	6645	6645	6645
	45	6317	6645	6645	6645	6645	6645	6645	6645
	50	6227	6645	6645	6645	6645	6645	6645	6645

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	1	1	1	1	2	4	4	4
	10	1	1	1	1	1	1	1	1
	15	1	1	1	1	1	1	1	1
	20	1	1	1	1	1	1	1	1
	25	1	1	1	1	1	1	1	1
	30	2	1	1	1	1	1	1	1
	35	2	1	1	1	1	1	1	1
	40	2	1	1	1	1	1	1	1
	45	2	1	1	1	1	1	1	1
	50	3	1	1	1	1	1	1	1

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	0	0	0	0	0	1.0	0.66	0.66
	10	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0
	20	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0
	30	0	0	0	0	0	0	0	0
	35	0	0	0	0	0	0	0	0
	40	0	0	0	0	0	0	0	0
	45	0	0	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0

TABLE B.7 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *assemble*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	72210	80052	83688	85469	86315	86567	86635	86679
	10	53095	63662	71448	76998	80936	83117	84607	85506
	15	38603	49514	58526	66121	72000	76621	79722	81634
	20	29473	39048	47790	56122	62937	68774	73323	76668
	25	23600	31372	39710	47556	54447	60673	65974	70769
	30	19329	25583	32795	40066	46955	53054	58528	63481
	35	15925	21366	27530	34004	40481	46894	52424	57383
	40	13265	17646	23123	28893	34905	41061	46749	51847
	45	11108	14885	19371	24723	30505	36011	41688	47011
	50	9431	12225	16244	20768	25966	31252	36828	42310

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	1857	826	341	147	76	55	49	39
	10	2790	2017	1352	845	500	317	180	100
	15	3135	2617	2029	1548	1152	757	548	391
	20	3087	2817	2345	1914	1550	1164	835	645
	25	2966	2819	2442	2008	1786	1544	1252	928
	30	2630	2505	2340	2103	1811	1589	1347	1206
	35	2288	2256	2213	2020	1911	1681	1453	1278
	40	2127	2134	1929	1831	1751	1599	1508	1377
	45	1982	1955	1891	1894	1704	1613	1465	1312
	50	1665	1662	1755	1760	1673	1565	1467	1279

(c) *AUC*.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	1.0	1.0	1.0	0.91	1.0	0.93	1.0	1
	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	15	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	20	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	25	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	30	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	35	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	40	0.87	0.87	0.87	0.87	0.87	0.87	0.9	0.97
	45	0.9	0.9	0.9	0.85	0.85	0.85	0.85	0.87
	50	0.9	0.9	0.9	0.9	0.85	0.85	0.85	0.90

TABLE B.8 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs des paramètres ε et $MinPts$ de DBSCAN. La séquence temporelle considérée est *disassemble*. La valeur du seuil de l'indice de Jaccard $seuil_{sim}$ est fixé à 0.9.

(a) Nombre de chroniques élémentaires identifiées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	65303	69500	71088	71753	72070	72223	72305	72372
	10	54901	62968	67175	69862	71137	71722	71915	72130
	15	44934	54539	61314	65306	67977	69924	71082	71588
	20	36450	45766	54036	59865	63532	66363	68540	69949
	25	29641	38662	46329	53983	59203	62509	65279	66971
	30	23868	32207	39477	47408	53094	58073	61408	64214
	35	19950	26868	33912	40852	47490	51909	56792	59868
	40	16756	22646	30136	36050	42808	47478	51763	56544
	45	14187	19582	26729	32631	38606	43762	47900	52240
	50	11867	16950	23287	29449	35223	40480	44677	48528

(b) Nombre de chroniques reconstituées.

		MinPts							
		3	4	5	6	7	8	9	10
ε	5	730	259	104	46	25	14	14	19
	10	1288	731	378	157	77	41	28	15
	15	1647	1092	718	434	271	189	82	52
	20	1610	1265	920	651	503	383	243	179
	25	1445	1270	1017	794	594	428	375	335
	30	1296	1146	1028	858	724	588	469	334
	35	1185	1032	968	851	722	624	486	432
	40	1100	938	823	736	737	699	605	451
	45	931	827	708	710	758	689	648	582
	50	856	728	729	682	624	652	648	622

(c) AUC.

		$MinPts$							
		3	4	5	6	7	8	9	10
ε	5	1.0	1.0	1.0	1.0	0.72	0.58	0.6	0.8
	10	1.0	0.83	1.0	0.83	0.83	0.83	0.83	0.83
	15	0.87	0.87	0.87	0.8	0.8	0.87	0.87	0.87
	20	0.8	0.8	0.8	0.87	0.87	0.8	0.87	0.87
	25	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
	30	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	35	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	40	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
	45	1.0	1.0	1.0	1.0	1.0	1.0	0.87	0.87
	50	0.93	0.93	0.93	0.92	0.95	0.88	0.88	0.88

B.2 Résultats des mesures pour l'analyse du seuil sur l'indice de Jaccard

B.2.1 Résultats des mesures de performances

TABLE B.9 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *pick-up*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 25$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0.00	3	0.213442	0.5
0.05	5	0.190607	0.75
0.1	10	0.190556	0.75
0.15	10	0.188674	0.75
0.2	10	0.184747	0.75
0.25	9	0.187017	0.75
0.3	15	0.191847	0.75
0.35	19	0.183428	0.75
0.4	25	0.179862	0.75
0.45	25	0.18521	0.77
0.5	24	0.178123	0.77
0.55	25	0.178368	0.75
0.6	25	0.178189	0.75
0.65	24	0.185703	0.75
0.7	25	0.178322	0.75
0.75	25	0.180862	0.75
0.8	25	0.195676	0.75
0.85	24	0.184089	0.75
0.9	23	0.170906	0.75
0.95	22	0.174103	0.75
1	85	0.176015	1

TABLE B.10 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *put-down*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 25$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	4	0.299544	0.54
0.05	4	0.217106	0.6
0.1	4	0.216645	0.6
0.15	4	0.244833	0.6
0.2	4	0.21511	0.6
0.25	4	0.2056	0.6
0.3	7	0.223827	0.6
0.35	14	0.199967	0.6
0.4	16	0.207158	0.6
0.45	22	0.197582	0.6
0.5	22	0.204897	0.6
0.55	22	0.207035	0.6
0.6	23	0.229217	0.6
0.65	25	0.232213	0.6
0.7	26	0.201632	0.6
0.75	31	0.190553	0.6
0.8	32	0.197597	0.6
0.85	36	0.186595	0.6
0.9	37	0.172121	0.6
0.95	37	0.171623	0.6
1	222	0.181777	1

TABLE B.11 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *stack*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 6$ et $MinPts = 15$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	6	1.16641	0.26
0.05	6	0.930298	0.26
0.1	6	0.919212	0.26
0.15	6	0.900474	0.26
0.2	6	0.88958	0.26
0.25	6	0.840469	0.26
0.3	6	0.819244	0.26
0.35	22	0.843153	0.26
0.4	120	0.868195	0.26
0.45	205	0.802466	0.26
0.5	294	0.756842	0.26
0.55	328	0.716088	0.83
0.6	331	0.733778	0.83
0.65	324	0.690729	0.83
0.7	320	0.800732	0.83

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0.75	308	0.706628	0.83
0.8	282	0.66131	0.83
0.85	255	0.650393	0.83
0.9	218	0.674334	0.83
0.95	172	0.631603	0.91
1	250	0.641093	1

TABLE B.12 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *unstack*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 6$ et $MinPts = 15$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	7	1.19391	0.66
0.05	7	0.961678	0.66
0.1	7	0.975544	0.66
0.15	7	0.911948	0.66
0.2	7	0.898441	0.66
0.25	10	0.922771	0.66
0.3	13	0.903382	0.66
0.35	63	0.872995	0.66
0.4	84	0.856107	0.66
0.45	176	0.753896	0.66
0.5	203	0.741007	0.66
0.55	281	0.731412	0.93
0.6	286	0.778168	0.93
0.65	280	0.755282	0.93
0.7	273	0.727966	0.93
0.75	265	0.70398	0.93
0.8	248	0.674344	0.93
0.85	240	0.673259	0.93
0.9	205	0.690548	0.93
0.95	163	0.632916	0.93
1	243	0.68053	1

TABLE B.13 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *move-left*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 4$ et $MinPts = 10$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	0	0.114765	NaN
0.05	1	0.102329	0
0.1	1	0.109096	0
0.15	1	0.0935241	0
0.2	1	0.105928	0

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0.25	1	0.105331	0
0.3	1	0.0956951	0
0.35	1	0.172068	0
0.4	1	0.0988111	0
0.45	1	0.101104	0
0.5	1	0.0963481	0
0.55	1	0.102949	0
0.6	1	0.126484	0
0.65	1	0.104395	0
0.7	1	0.111352	0
0.75	1	0.103112	0
0.8	1	0.111138	0
0.85	1	0.110852	0
0.9	1	0.11464	0
0.95	1	0.103664	0
1	45	0.105379	1

TABLE B.14 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *move-right*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 4$ et $MinPts = 10$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	1	0.132009	0
0.05	1	0.116094	0
0.1	1	0.105157	0
0.15	1	0.105596	0
0.2	1	0.0962471	0
0.25	1	0.0975131	0
0.3	1	0.101422	0
0.35	1	0.100009	0
0.4	1	0.106483	0
0.45	1	0.104476	0
0.5	1	0.107356	0
0.55	1	0.15636	0
0.6	1	0.103066	0
0.65	1	0.0960471	0
0.7	1	0.0988801	0
0.75	1	0.0975791	0
0.8	1	0.108632	0
0.85	1	0.105517	0
0.9	1	0.101542	0
0.95	1	0.107708	0
1	45	0.0985791	1

TABLE B.15 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *assemble*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 5$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	14	2.30689	0.5
0.05	22	1.81952	0.5
0.1	22	1.76967	0.5
0.15	22	1.73536	0.5
0.2	22	1.65361	0.5
0.25	27	1.59272	0.5
0.3	43	1.54465	0.5
0.35	52	1.53832	0.5
0.4	72	1.53542	0.5
0.45	143	1.54468	0.5
0.5	147	1.46516	0.5
0.55	244	1.48061	0.55
0.6	334	1.43538	1
0.65	375	1.43249	0.66
0.7	419	1.57659	0.66
0.75	416	1.38617	0.66
0.8	415	1.3642	0.66
0.85	374	1.32288	1
0.9	341	1.31329	1
0.95	353	1.28317	1
1	531	1.29193	1

TABLE B.16 – Table récapitulant les résultats de CDIRe pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$. La séquence temporelle considérée est *disassemble*. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$ et $MinPts = 5$.

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0	5	1.43958	1
0.05	5	1.24222	1
0.1	5	1.22941	1
0.15	11	1.22718	1
0.2	15	1.22517	1
0.25	18	1.18361	1
0.3	27	1.14328	1
0.35	31	1.14038	1
0.4	35	1.12943	1
0.45	54	1.11708	1
0.5	55	1.09737	1
0.55	72	1.1292	1
0.6	82	1.09261	1
0.65	87	1.1015	1
0.7	98	1.08533	1

$seuil_{sim}$	# chroniques reconstituées	temps d'exécution (s)	AUC
0.75	111	1.09021	1
0.8	118	1.07784	1
0.85	112	1.09158	1
0.9	104	1.08357	1
0.95	106	1.05151	1
1	314	1.04245	1

B.2.2 Résultats des mesures de qualité pour les chroniques reconstituées les plus descriptives

TABLE B.17 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *pick-up* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$, $MinPts = 25$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	7	5	24	20	35	0	0	0
0.05	6	6	24	20	35	0.33	0.33	0.5
0.1	6	6	24	20	35	0.33	0.33	0.5
0.15	6	6	24	20	35	0.33	0.33	0.5
0.2	6	6	24	20	35	0.33	0.33	0.5
0.25	6	6	24	20	35	0.33	0.33	0.5
0.3	6	6	24	20	35	0.33	0.33	0.5
0.35	6	6	24	20	35	0.33	0.33	0.5
0.4	6	6	24	20	35	0.33	0.33	0.5
0.45	6	6	24	20	35	0.33	0.33	0.5
0.5	6	6	24	20	35	0.33	0.33	0.5
0.55	6	6	24	20	35	0.33	0.33	0.5
0.6	6	6	24	20	35	0.33	0.33	0.5
0.65	6	6	24	20	35	0.33	0.33	0.5
0.7	6	6	24	20	35	0.33	0.33	0.5
0.75	6	6	24	20	35	0.33	0.33	0.5
0.8	6	5.60001	24	20	35	0.33	0.5	0.5
0.85	6	5.60001	24	20	35	0.33	0.33	0.5
0.9	6	3.60001	24	20	35	0.33	0.33	0.5
0.95	6	3.60001	24	20	35	0.33	0.33	0.5
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.18 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *put-down* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$, $MinPts = 25$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	7	5	24	28	55	0	0	0
0.05	6	6	24	28	41	0	0	0
0.1	6	6	24	28	41	0	0	0
0.15	6	6	24	28	41	0	0	0
0.2	6	6	24	28	41	0	0	0
0.25	6	6	24	28	41	0	0	0
0.3	6	6	24	28	41	0	0	0
0.35	6	6	24	28	41	0	0	0
0.4	6	6	24	28	41	0	0	0
0.45	6	6	24	28	41	0	0	0
0.5	6	6	24	28	41	0	0	0
0.55	6	6	24	28	41	0	0	0
0.6	6	6	24	28	41	0	0	0
0.65	6	6	24	28	41	0	0	0
0.7	6	6	24	28	41	0	0	0
0.75	6	6	24	28	41	0	0	0
0.8	6	6	24	28	41	0	0	0
0.85	6	6	24	28	41	0	0	0
0.9	6	5.60001	24	28	41	0	0	0
0.95	6	5.20001	24	28	41	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.19 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *stack* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 6$, $MinPts = 15$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.05	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.15	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.25	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.3	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.35	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.4	12	9.81819	24	27	35	0	0	0

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0.45	12	9.63637	24	27	35	0	0	0
0.5	12	8.18182	23	27	35	0	0	0
0.55	12	8.18182	24	27	35	0	0	0
0.6	12	8.18182	24	27	35	0	0	0
0.65	12	8.18182	24	27	35	0	0	0
0.7	12	8.18182	24	27	35	0	0	0
0.75	12	8.18182	23	27	35	0	0	0
0.8	12	8.18182	24	27	35	0	0	0
0.85	12	8.18182	24	27	35	0	0	0
0.9	13	7.16667	24	27	43	0	0	0
0.95	12	5.63637	24	27	35	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.20 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *unstack* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 6$, $MinPts = 15$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.05	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.15	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.25	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.3	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.35	12	9.63637	24	25	37	0	0	0
0.4	12	9.63637	24	25	37	0	0	0
0.45	12	9.09091	24	25	37	0	0	0
0.5	12	8.18182	24	25	37	0	0	0
0.55	12	8	24	25	37	0	0	0
0.6	12	8	24	25	37	0	0	0
0.65	12	8	24	25	37	0	0	0
0.7	12	8	24	25	37	0	0	0
0.75	12	8	24	25	37	0	0	0
0.8	12	8	24	25	37	0	0	0
0.85	12	7.63637	24	25	37	0	0	0
0.9	12	7.09091	24	25	37	0	0	0
0.95	12	5.27273	24	23	37	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.21 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *move-left* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 4$, $MinPts = 10$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.05	10	10	12	53	63	0	0	0
0.1	10	10	12	53	63	0	0	0
0.15	10	10	12	53	63	0	0	0
0.2	10	10	12	53	63	0	0	0
0.25	10	10	12	53	63	0	0	0
0.3	10	10	12	53	63	0	0	0
0.35	10	10	12	53	63	0	0	0
0.4	10	10	12	53	63	0	0	0
0.45	10	10	12	53	63	0	0	0
0.5	10	10	12	53	63	0	0	0
0.55	10	10	12	53	63	0	0	0
0.6	10	10	12	53	63	0	0	0
0.65	10	10	12	53	63	0	0	0
0.7	10	10	12	53	63	0	0	0
0.75	10	10	12	53	63	0	0	0
0.8	10	10	12	53	63	0	0	0
0.85	10	10	12	53	63	0	0	0
0.9	10	10	12	53	63	0	0	0
0.95	10	10	12	53	63	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.22 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *move-right* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 4$, $MinPts = 10$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.05	10	10	12	45	55	0	0	0
0.1	10	10	12	45	55	0	0	0
0.15	10	10	12	45	55	0	0	0
0.2	10	10	12	45	55	0	0	0
0.25	10	10	12	45	55	0	0	0
0.3	10	10	12	45	55	0	0	0
0.35	10	10	12	45	55	0	0	0
0.4	10	10	12	45	55	0	0	0

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0.45	10	10	12	45	55	0	0	0
0.5	10	10	12	45	55	0	0	0
0.55	10	10	12	45	55	0	0	0
0.6	10	10	12	45	55	0	0	0
0.65	10	10	12	45	55	0	0	0
0.7	10	10	12	45	55	0	0	0
0.75	10	10	12	45	55	0	0	0
0.8	10	10	12	45	55	0	0	0
0.85	10	10	12	45	55	0	0	0
0.9	10	10	12	45	55	0	0	0
0.95	10	10	12	45	55	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.23 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *assemble* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$, $MinPts = 5$.

$seuil_{sim}$	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	AUC (`minmax`)	AUC (`2sigma`)	AUC (`3sigma`)
0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.05	16	12.2667	24	74	92	0	0	0
0.1	16	12.2667	24	74	92	0	0	0
0.15	16	12.2667	24	74	92	0	0	0
0.2	16	12.2667	24	74	92	0	0	0
0.25	16	12.2667	24	74	92	0	0	0
0.3	16	12.2667	24	74	92	0	0	0
0.35	16	12.2667	24	74	92	0	0	0
0.4	16	12	24	74	92	0	0	0
0.45	16	11.3334	24	74	92	0	0	0
0.5	16	10.4001	24	74	95	0	0	0
0.55	16	10.2667	24	74	95	0	0	0
0.6	16	10.1334	24	74	95	0	0	0
0.65	16	9.73334	24	74	95	0	0	0
0.7	16	9.46667	24	74	95	0	0	0
0.75	16	9.46667	24	74	95	0	0	0
0.8	17	8.125	24	74	95	0	0	0
0.85	23	4.63637	24	74	103	0	0	0
0.9	12	3.27273	24	74	99	0	0	0
0.95	9	2.75	24	58	100	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

TABLE B.24 – Tableau récapitulatif des chroniques les plus descriptives reconstituées par CDIRe avec la séquence temporelle associée au phénomène *disassemble* pour plusieurs valeurs du seuil de l'indice de Jaccard $seuil_{sim}$ et pour chaque méthode de calcul des bornes de la contrainte temporelle. Les valeurs des paramètres de DBSCAN sont $\varepsilon = 5$, $MinPts = 5$.

nom	taille	compacité	fréquence	durée minimale (`minmax`)	durée maximale (`minmax`)	*AUC* (`minmax`)	*AUC* (`2sigma`)	*AUC* (`3sigma`)
0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.05	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.15	16	13.2001	24	94	118	0	0	0
0.2	16	13.2001	24	94	118	0	0	0
0.25	16	13.2001	24	94	118	0	0	0
0.3	16	13.2001	24	94	118	0	0	0
0.35	16	13.2001	24	94	118	0	0	0
0.4	16	13.2001	24	94	118	0	0	0
0.45	16	13.2001	24	94	118	0	0	0
0.5	16	13.2001	24	94	118	0	0	0
0.55	16	12.6667	24	94	118	0	0	0
0.6	16	12.6667	24	94	118	0	0	0
0.65	16	12.6667	24	94	118	0	0	0
0.7	16	12.6667	24	94	118	0	0	0
0.75	16	12.6667	24	94	118	0	0	0
0.8	16	12.6667	24	94	118	0	0	0
0.85	16	12.6667	24	94	118	0	0	0
0.9	16	11.4667	24	94	118	0	0	0
0.95	16	11.4667	24	94	118	0	0	0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

B.3 Résultats des mesures de qualité des chroniques reconstituées pour l'étude de la compacité et du calcul de l'*AUC*

TABLE B.25 – Tableau récapitulatif des chroniques reconstituées par CDIRe avec la séquence temporelle associée au phénomène *pick-up*. Les valeurs des paramètres sont $\varepsilon = 5$, $MinPts = 25$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	*AUC*
C5_sub_0	7	2	5	658	706	0	0	0
C5_sub_1	6	2	5	10446	10495	0	0	0
C5_sub_2	7	2	5	94	107	0	0	0
C5_sub_3	4	2	5	135	206	0	0	0
C5_sub_4	10	3.11112	5	1104	1181	0	0	0
C6_sub_0	7	2	6	658	706	0	0	0
C6_sub_1	2	2	6	113	165	0	0	0

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C6_sub_2	6	2	6	10446	10495	0	0	0
C6_sub_3	2	2	6	1043	1109	0	0	0
C6_sub_4	2	2	6	1083	1150	0	0	0
C6_sub_5	7	2	6	133	258	0	0	0
C7_sub_0	2	2	7	511	593	0	0	0
C7_sub_1	8	2.57143	7	133	221	0	0	0
C8_sub_0	4	2	8	126	179	0	0	0
C8_sub_1	8	2	8	517	703	0	0	0
C8_sub_2	3	2	8	91	177	0	0	0
C9_sub_0	7	2	9	517	603	0	0	0
C9_sub_1	10	3.55556	9	91	229	0	0	0
C10_sub_0	10	4.22223	10	89	238	0	0	0
C11_sub_0	9	4.5	11	89	156	0	0	0
C12_sub_0	4	2.66667	12	68	156	0	0	0
C24_sub_0	6	5.60001	24	20	35	1	0.33	0.33
C25_sub_0	6	5.60001	25	18	35	1	0.33	0.33
C27_sub_0	4	4	27	0	24	1	0.5	0.5
C31_sub_0	2	2	31	0	24	0.5	1	0.5

TABLE B.26 – Tableau récapitulatif des chroniques reconstituées par CDIRe avec la séquence temporelle associée au phénomène *put-down*. Les valeurs des paramètres sont $\varepsilon = 5$, $MinPts = 25$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C5_sub_0	4	2	5	4060	4128	0	0.0	0
C5_sub_1	12	4.72728	5	862	926	0	0.0	0
C5_sub_10	12	6.54546	5	2572	2587	0	0.0	0
C5_sub_11	2	2	5	548	583	0	0.0	0
C5_sub_2	12	6.18182	5	1734	1778	0	0.0	0
C5_sub_3	12	5.81819	5	1520	1591	0	0.0	0
C5_sub_4	2	2	5	522	549	0	0.0	0
C5_sub_5	4	2	5	4039	4158	0	0.0	0
C5_sub_6	14	4.15385	5	3250	3278	0	0.0	0
C5_sub_7	9	2.75	5	2428	2500	0	0.0	0
C5_sub_8	4	2	5	4018	4056	0	0.0	0
C5_sub_9	8	2.85715	5	272	306	0	0.0	0
C6_sub_0	2	2	6	237	285	0	0.0	0
C6_sub_1	12	6	6	1502	1620	0	0.0	0
C6_sub_2	9	3.25	6	1720	1786	0	0.0	0
C6_sub_3	12	3.45455	6	862	930	0	0.0	0
C6_sub_4	2	2	6	237	284	0	0.0	0
C6_sub_5	4	2	6	4039	4158	0	0.0	0
C6_sub_6	14	3.07693	6	3244	3348	0	0.0	0
C6_sub_7	5	2	6	4060	4150	0	0.0	0
C7_sub_0	2	2	7	4043	4108	0	0.0	0
C7_sub_1	7	2.66667	7	3209	3289	0	0.0	0

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	*AUC*
C7_sub_2	5	2.5	7	848	911	0	0.0	0
C7_sub_3	2	2	7	3244	3291	0	0.0	0
C7_sub_4	2	2	7	1698	1747	0	0.0	0
C7_sub_5	5	2	7	1495	1613	0	0.0	0
C8_sub_0	2	2	8	826	880	0	0.0	0
C8_sub_1	2	2	8	3184	3241	0	0.0	0
C24_sub_0	6	6	24	28	41	0	0.0	0
C25_sub_0	6	3.60001	25	20	45	0	0.0	0
C26_sub_0	3	2	26	0	39	0.6	1.0	0.6
C27_sub_0	2	2	27	0	16	0.54	1.0	0.54

TABLE B.27 – Tableau récapitulatif d'une partie des 282 chroniques reconstituées par CDIRe avec la séquence temporelle associée au phénomène *stack*. Les valeurs des paramètres sont $\varepsilon = 6$, $MinPts = 15$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	*AUC*
C21_sub_0	2	2	21	1214	1294	0	0	0
C21_sub_1	2	2	21	4965	5065	0	0	0
C21_sub_2	12	7.81819	21	27	35	0	0	0
C21_sub_3	2	2	21	481	548	0	0	0
C21_sub_4	3	2	21	2213	2293	0	0	0
C21_sub_5	2	2	21	56	119	0	0	0
C21_sub_6	2	2	21	4243	4351	0	0	0
C22_sub_0	4	2	22	489	629	0	0	0
C22_sub_1	2	2	22	1214	1294	0	0	0
C22_sub_2	3	2	22	4967	5071	0	0	0
C22_sub_3	12	8.18182	22	27	35	0	0	0
C22_sub_4	4	2	22	2213	2387	0	0	0
C22_sub_5	2	2	22	4243	4351	0	0	0
C23_sub_0	4	2	23	4967	5165	0	0	0
C23_sub_1	2	2	23	896	966	0	0	0
C23_sub_2	3	2	23	2206	2300	0	0	0
C23_sub_3	2	2	23	1214	1294	0	0	0
C23_sub_4	12	8.18182	23	27	35	0	0	0
C23_sub_5	4	2	23	489	629	0	0	0
C24_sub_0	12	8.18182	24	27	35	0	0	0
C24_sub_1	2	2	24	896	966	0	0	0
C24_sub_2	4	2.66667	24	489	571	0	0	0
C24_sub_3	2	2	24	2190	2300	0	0	0
C24_sub_4	2	2	24	1214	1294	0	0	0
C24_sub_5	4	2.66667	24	4967	5071	0	0	0
C24_sub_6	3	2	24	264	380	0	0	0
C25_sub_0	4	2.66667	25	4967	5071	0	0	0
C25_sub_1	2	2	25	1214	1294	0	0	0
C25_sub_2	2	2	25	15	38	1	0.83	0.83
C25_sub_3	3	2	25	264	380	0	0	0

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C25_sub_4	2	2	25	2190	2300	0	0	0
C25_sub_5	4	2.66667	25	489	571	0	0	0
C25_sub_6	2	2	25	896	966	0	0	0
C25_sub_7	10	4.44445	25	27	48	0	0	0
C26_sub_0	2	2	26	15	38	1	0.83	0.83
C26_sub_1	2	2	26	1214	1294	0	0	0
C26_sub_2	2	2	26	2190	2300	0	0	0
C26_sub_3	2	2	26	896	966	0	0	0
C26_sub_4	4	2.66667	26	489	571	0	0	0
C26_sub_5	4	2.66667	26	4967	5071	0	0	0
C26_sub_6	3	2	26	264	380	0	0	0
C26_sub_7	10	2.88889	26	27	75	0	0	0
C27_sub_0	2	2	27	0	21	0.45	0.83	0.37
C27_sub_1	4	2	27	489	629	0	0	0
C27_sub_2	4	2	27	4967	5167	0	0	0
C27_sub_3	2	2	27	896	966	0	0	0
C27_sub_4	4	2	27	272	515	0	0	0
C27_sub_5	2	2	27	15	38	1	0.83	0.83
C27_sub_6	2	2	27	2190	2300	0	0	0

TABLE B.28 – Tableau récapitulatif d'une partie des 248 chroniques reconstituées par CDIRe avec la séquence temporelle associée au phénomène *unstack*. Les valeurs des paramètres sont $\varepsilon = 6$, $MinPts = 15$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C20_sub_0	4	2.66667	20	382	460	0	0	0
C20_sub_1	2	2	20	3183	3262	0	0	0
C20_sub_2	12	5.27273	20	23	37	0	0	0
C20_sub_3	2	2	20	6195	6274	0	0	0
C20_sub_4	2	2	20	3213	3260	0	0	0
C20_sub_5	4	2	20	1877	2051	0	0	0
C20_sub_6	4	2.66667	20	3006	3062	0	0	0
C20_sub_7	3	2	20	2234	2305	0	0	0
C21_sub_0	4	2.66667	21	382	460	0	0	0
C21_sub_1	3	2	21	2976	3040	0	0	0
C21_sub_2	2	2	21	3183	3262	0	0	0
C21_sub_3	12	7.09091	21	25	37	0	0	0
C21_sub_4	3	2	21	1877	1969	0	0	0
C21_sub_5	2	2	21	2227	2305	0	0	0
C21_sub_6	2	2	21	3213	3260	0	0	0
C22_sub_0	3	2	22	2976	3040	0	0	0
C22_sub_1	2	2	22	1856	1961	0	0	0
C22_sub_2	4	2	22	375	538	0	0	0
C22_sub_3	2	2	22	3183	3262	0	0	0
C22_sub_4	12	7.27273	22	25	37	0	0	0
C23_sub_0	12	7.63637	23	25	37	0	0	0

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	*AUC*
C23_sub_1	3	2	23	2976	3040	0	0	0
C23_sub_2	2	2	23	368	455	0	0	0
C23_sub_3	2	2	23	3183	3262	0	0	0
C24_sub_0	3	2	24	2976	3040	0	0	0
C24_sub_1	4	2	24	738	872	0	0	0
C24_sub_2	12	8	24	25	37	0	0	0
C24_sub_3	2	2	24	3183	3262	0	0	0
C25_sub_0	4	2.66667	25	738	811	0	0	0
C25_sub_1	12	5.45455	25	25	52	0	0	0
C25_sub_2	3	2	25	2976	3040	0	0	0
C26_sub_0	4	2.66667	26	738	811	0	0	0
C26_sub_1	11	3	26	25	83	0	0	0
C27_sub_0	4	2.66667	27	738	811	0	0	0
C27_sub_1	2	2	27	0	17	0	0	0
C27_sub_2	2	2	27	17	42	1	0.83	0.83
C27_sub_3	4	2	27	16	66	0	0	0
C28_sub_0	4	2.66667	28	738	811	0	0	0
C28_sub_1	4	2	28	16	66	0	0	0
C28_sub_2	2	2	28	17	42	1	0.83	0.83
C29_sub_0	2	2	29	16	43	0	0	0
C29_sub_1	2	2	29	17	42	1.0	0.83	0.83
C29_sub_2	4	2.66667	29	738	811	0	0	0
C31_sub_0	2	2	31	716	805	0	0	0

TABLE B.29 – Tableau récapitulatif de la chronique reconstituée par CDIRe avec la séquence temporelle associée au phénomène *move-left*. Les valeurs des paramètres sont $\varepsilon = 4$, $MinPts = 10$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	*AUC*
C12_sub_0	10	10	12	53	63	0	0	0

TABLE B.30 – Tableau récapitulatif de la chronique reconstituée par CDIRe avec la séquence temporelle associée au phénomène *move-right*. Les valeurs des paramètres sont $\varepsilon = 4$, $MinPts = 10$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	*AUC*
C12_sub_0	10	10	12	45	55	0	0	0

TABLE B.31 – Tableau récapitulatif d'une partie des 415 chroniques reconstituées par CDIRe avec la séquence temporelle associée au phénomène *assemble*. Les valeurs des paramètres sont $\varepsilon = 5$, $MinPts = 5$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C21_sub_0	17	5.625	21	74	96	0	0	0
C22_sub_0	18	6.35295	22	74	96	0	0	0
C23_sub_0	19	6.66667	23	74	105	0	0	0
C24_sub_0	17	8.125	24	74	95	0	0	0
C25_sub_0	16	7.20001	25	73	99	0	0	0
C26_sub_0	18	5.76471	26	73	103	0	0	0
C27_sub_0	18	4.58824	27	73	118	0	0	0
C28_sub_0	16	4.13334	28	71	123	0	0	0
C29_sub_0	13	3.5	29	49	94	0	0	0
C30_sub_0	10	2.88889	30	49	116	0	0	0
C31_sub_0	8	2.85715	31	49	96	0	0	0
C32_sub_0	6	3.60001	32	40	67	0	0	0
C35_sub_0	4	4	35	0	21	0	0	0
C38_sub_0	4	2.66667	38	0	27	0	0	0
C40_sub_0	2	2	40	0	18	0	0	0
C40_sub_1	2	2	40	0	17	0.66	0.33	0.22
C40_sub_2	4	2	40	57	107	1	0.33	0.33
C41_sub_0	4	2	41	57	107	1	0.33	0.33
C41_sub_1	2	2	41	0	18	0	0	0
C48_sub_0	2	2	48	0	4	0.45	0.83	0.37
C48_sub_1	13	2	48	57	112	0	0	0
C49_sub_0	2	2	49	0	4	0.45	0.83	0.37
C49_sub_1	12	2	49	34	93	0	0	0
C52_sub_0	3	3	52	0	17	0.5	0.83	0.41
C52_sub_1	10	2	52	34	93	0	0	0
C53_sub_0	3	3	53	0	17	0.5	0.83	0.41
C53_sub_1	6	2	53	0	45	0	0	0
C54_sub_0	3	3	54	0	17	0.5	0.83	0.41
C54_sub_1	5	2	54	0	40	0	0	0

TABLE B.32 – Tableau récapitulatif d'une partie des 118 chroniques reconstituées par CDIRe avec la séquence temporelle associée au phénomène *disassemble*. Les valeurs des paramètres sont $\varepsilon = 5$, $MinPts = 5$ et $seuil_{sim} = 0.8$.

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C9_sub_0	6	2	9	2222	2238	0	0	0
C9_sub_1	6	3.60001	9	3854	3861	0	0	0
C9_sub_2	3	2	9	2263	2279	0	0	0
C10_sub_0	2	2	10	3839	3853	0	0	0
C10_sub_1	3	2	10	2263	2279	0	0	0
C10_sub_2	4	2	10	2208	2217	0	0	0
C21_sub_0	16	13.7334	21	94	118	0	0	0

nom	taille	compacité	fréquence	durée minimale	durée maximale	précision	rappel	AUC
C22_sub_0	16	14.5334	22	94	118	0	0	0
C23_sub_0	16	14.4001	23	94	118	0	0	0
C24_sub_0	16	12.6667	24	94	118	0	0	0
C25_sub_0	8	3.71429	25	66	93	1	0.16	0.16
C25_sub_1	2	2	25	7	20	0.5	0.33	0.16
C27_sub_0	6	3.60001	27	63	77	1	0.16	0.16
C47_sub_0	2	2	47	70	118	1	1	1
C47_sub_1	3	3	47	9	21	0.625	0.83	0.52
C48_sub_0	3	3	48	9	21	0.625	0.83	0.52

Annexe C

Compléments à la projection de chroniques

C.1 Compléments à l'analyse statistique de la norme d'une occurrence projetée

C.1.1 Démonstration complète du calcul de l'espérance

Considérons la norme au carré $\|x\|^2$ d'un vecteur obtenu par la projection d'une occurrence d'une chronique de taille n dans l'espace euclidien k-dimensionnel. $x(j)$ est le j-ième élément de x et $\|x\|^2$ est défini par :

$$\|x\|^2 = \sum_{j=1}^{k} x(j)^2. \tag{C.1}$$

Tous les éléments $x(j)$ du vecteur x est la somme de n variables aléatoires pondérées par un instant d'occurrence :

$$x(j) = \sum_{i=1}^{n} \phi(e_i, j) t_i. \tag{C.2}$$

L'élément au carré $x(j)^2$ du vecteur x est donné par :

$$x(j)^2 = \left(\sum_{i=1}^{n} \phi(e_i, j) t_i \right)^2. \tag{C.3}$$

En développant :

$$x(j)^2 = \sum_{i=1}^{n} \phi(e_i, j)^2 t_i^2 + 2 \sum_{l<i} \phi(e_i, j) \phi(e_l, j) t_i t_l. \tag{C.4}$$

Donc, l'espérance de chaque élément $E\left(x(j)^2\right)$ du vecteur x correspond à :

$$E\left(x(j)^2\right) = E\left(\sum_{i=1}^{n} \phi(e_i, j)^2 t_i^2 + 2 \sum_{l<i} \phi(e_i, j) \phi(e_l, j) t_i t_l \right). \tag{C.5}$$

Comme l'espérance de la somme est égale à la somme des espérances :

$$E\left(x(j)^2\right) = \sum_{i=1}^{n} E\left(\phi(e_i,j)^2\right) t_i^2 + 2\sum_{l<i} E\left(\phi(e_i,j)\right) E\left(\phi(e_l,j)\right) t_i t_l. \tag{C.6}$$

De plus, x est un vecteur rempli de variables aléatoires indépendantes avec une espérance $\mu = 0$ et une variance $\sigma^2 = 1$. Donc, l'espérance de $\phi(e_i,j)^2$ est $E\left(\phi(e_i,j)^2\right) = 1$ et l'espérance de $\phi(e_i,j)$ est $E\left(\phi(e_i,j)\right) = 0$. Ainsi, l'espérance de $x(j)^2$ est :

$$E\left(x(j)^2\right) = \sum_{i=1}^{n} t_i^2. \tag{C.7}$$

Donc, l'espérance de $\|x\|^2$ est donnée par :

$$E\left(\|x\|^2\right) = E\left(\sum_{j=1}^{k} x(j)^2\right) = \sum_{j=1}^{k} E\left(x(j)^2\right). \tag{C.8}$$

Ainsi :

$$E\left(\|x\|^2\right) = k\sum_{i=1}^{n} t_i^2. \tag{C.9}$$

C.1.2 Démonstration complète du calcul de la variance

Rappelons que la variance d'une variable aléatoire X est donnée par la formule suivante :

$$Var(X) = E\left(X^2\right) - E\left(X\right)^2. \tag{C.10}$$

Ainsi, la variance de $x(j)^2$ est :

$$Var\left(x(j)^2\right) = E\left(\left(x(j)^2\right)^2\right) - E\left(x(j)^2\right)^2. \tag{C.11}$$

La valeur de l'espérance $E\left(x(j)^2\right)$ est déjà connue grâce à l'équation (C.7). Calculons la valeur de l'espérance $E\left(\left(x(j)^2\right)^2\right)$:

$$x(j)^4 = \left(x(j)^2\right)^2. \tag{C.12}$$

En développant $x(j)^2$:

$$x(j)^4 = \left(\sum_{i=1}^{n} \phi(e_i,j)^2 t_i^2 + 2\sum_{l<i} \phi(e_i,j)\phi(e_l,j) t_i t_l\right)^2. \tag{C.13}$$

En continuant de développer :

$$x(j)^4 = \left(\sum_{i=1}^{n} \phi(e_i,j)^2 t_i^2\right)^2 + 4\left(\sum_{l<i} \phi(e_i,j)\phi(e_l,j) t_i t_l\right)^2 + 4\left(\sum_{i=1}^{n} \phi(e_i,j)^2 t_i^2\right)\left(\sum_{l<i} \phi(e_i,j)\phi(e_l,j) t_i t_l\right). \quad \text{(C.14)}$$

Enfin :

$$x(j)^4 = \sum_{i=1}^{n} \phi(e_i,j)^4 t_i^4 + 6\sum_{l<i} \phi(e_i,j)^2 \phi(e_l,j)^2 t_i^2 t_l^2 + 8\prod_{i=1}^{n} \phi(e_i,j) t_i \sum_{l=1}^{n} \phi(e_i,j) t_i + 4\left(\sum_{i=1}^{n} \phi(e_i,j)^2 t_i^2\right)\left(\sum_{l<i} \phi(e_i,j)\phi(e_l,j) t_i t_l\right). \quad \text{(C.15)}$$

Donc la variance de chaque élément $E\left(x(j)^4\right)$ du vecteur x correspond à :

$$E\left(x(j)^4\right) = E\left(\sum_{i=1}^{n} \phi(e_i,j)^4 t_i^4 + 6\sum_{l<i} \phi(e_i,j)^2 \phi(e_l,j)^2 t_i^2 t_l^2 + 8\prod_{i=1}^{n} \phi(e_i,j) t_i \sum_{l=1}^{n} \phi(e_i,j) t_i + 4\left(\sum_{i=1}^{n} \phi(e_i,j)^2 t_i^2\right)\left(\sum_{l<i} \phi(e_i,j)\phi(e_l,j) t_i t_l\right)\right) \quad \text{(C.16)}$$

Comme l'espérance de la somme est égale à la somme des espérances :

$$E\left(x(j)^4\right) = \sum_{i=1}^{n} E\left(\phi(e_i,j)^4\right) t_i^4 + 6\sum_{l<i} E\left(\phi(e_i,j)^2\right) E\left(\phi(e_l,j)^2\right) t_i^2 t_l^2 + 8\prod_{i=1}^{n} E\left(\phi(e_i,j)\right) t_i \sum_{l=1}^{n} E\left(\phi(e_i,j)\right) t_i + 4\left(\sum_{i=1}^{n} E\left(\phi(e_i,j)^2\right) t_i^2\right)\left(\sum_{l<i} E\left(\phi(e_i,j)\right) E\left(\phi(e_l,j)\right) t_i t_l\right). \quad \text{(C.17)}$$

De plus, x est un vecteur rempli de variables aléatoires indépendantes avec une espérance $\mu = 0$ et une variance $\sigma^2 = 1$. Donc, l'espérance de $\phi(e_i,j)^4$ est $E\left(\phi(e_i,j)^4\right) = 3$, l'espérance de $\phi(e_i,j)^2$ est $E\left(\phi(e_i,j)^2\right) = 1$ et l'espérance de $\phi(e_i,j)$ est $E\left(\phi(e_i,j)\right) = 0$. Ainsi, l'espérance de $x(j)^4$ est :

$$E\left(x(j)^4\right) = 3\sum_{i=1}^{n} t_i^4 + 6\sum_{l<i} t_i^2 t_l^2. \quad \text{(C.18)}$$

En factorisant :

$$E\left(x(j)^4\right) = 3\left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{C.19}$$

En revanche, l'espérance au carré de $E\left(x(j)^2\right)$ est :

$$E\left(x(j)^2\right)^2 = \left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{C.20}$$

Calculons maintenant la variance $Var\left(x(j)^2\right)$:

$$Var\left(x(j)^2\right) = E\left(x(j)^4\right) - E\left(x(j)^2\right)^2. \tag{C.21}$$

Grâce aux équations (C.20) et (C.19) :

$$Var\left(x(j)^2\right) = 3\left(\sum_{i=1}^{n} t_i^2\right)^2 - \left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{C.22}$$

Donc, la variance de $x(j)^2$ est :

$$Var\left(x(j)^2\right) = 2\left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{C.23}$$

Enfin, la variance de la norme au carré $\|x\|^2$ est donnée par :

$$Var\left(\|x\|^2\right) = Var\left(\sum_{j=1}^{k} x(j)^2\right) = \sum_{j=1}^{k} Var\left(x(j)^2\right). \tag{C.24}$$

Donc :

$$Var\left(\|x\|^2\right) = 2k\left(\sum_{i=1}^{n} t_i^2\right)^2. \tag{C.25}$$

Bibliographie

[Aggarwal 2014] Charu C Aggarwal et Jiawei Han. Frequent pattern mining. Springer, 2014. (Cité en page 21.)

[Agrawal 1993] Rakesh Agrawal, Tomasz Imieliński et Arun Swami. *Mining association rules between sets of items in large databases.* Dans ACM sigmod record, volume 22, pages 207–216, 1993. (Cité en pages 21 et 22.)

[Agrawal 1994] Rakesh Agrawal et Ramakrishnan Srikant. *Fast Algorithms for Mining Association Rules in Large Databases.* Dans VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, pages 487–499, 1994. (Cité en pages 21, 22 et 26.)

[Agrawal 1995] Rakesh Agrawal et Ramakrishnan Srikant. *Mining sequential patterns.* Dans Proceedings of the Eleventh International Conference on Data Engineering, 1995., pages 3–14, 1995. (Cité en pages 21 et 22.)

[Agrawal 1998] Rakesh Agrawal, Dimitrios Gunopulos et Frank Leymann. *Mining process models from workflow logs.* Dans International Conference on Extending Database Technology, pages 467–483, 1998. (Cité en page 23.)

[Allen 1983] James F. Allen. *Maintaining Knowledge about Temporal Intervals.* Communications of the ACM, vol. 26, no. 11, 1983. (Cité en pages 18 et 19.)

[Alur 1994] Rajeev Alur et David L. Dill. *A Theory of Timed Automata.* Theoretical Computer Science, vol. 126, pages 183–235, 1994. (Cité en page 20.)

[Álvarez 2013] Miguel R. Álvarez, Paulo Félix et Purificación Cariñena. *Discovering metric temporal constraint networks on temporal databases.* Artificial Intelligence in Medicine, vol. 58, no. 3, pages 139–154, 2013. (Cité en pages 9, 33 et 34.)

[Ankerst 1999] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel et Jörg Sander. *OPTICS : ordering points to identify the clustering structure.* Dans ACM Sigmod record, volume 28, pages 49–60, 1999. (Cité en page 46.)

[Atallah 1991] Mikhail J. Atallah, Celso C. Ribeiro et Sergio Lifschitz. *Computing some distance functions between polygons.* Pattern Recognition, vol. 24, no. 8, pages 775 – 781, 1991. (Cité en page 116.)

[Atkison 2009] Travis Atkison. *Applying randomized projection to aid prediction algorithms in detecting high-dimensional rogue applications.* Dans Proceedings of the 47th Annual Southeast Regional Conference, page 23, 2009. (Cité en pages 24 et 102.)

[Baccelli 1992] François Baccelli, Guy Cohen, Geert Jan Olsder et Jean-Pierre Quadrat. Synchronization and linearity : an algebra for discrete event systems. John Wiley & Sons Ltd, 1992. (Cité en page 20.)

[Baeza-Yates 2011] Ricardo Baeza-Yates, Berthier de Araújo Neto Ribeiro*et al.* Modern information retrieval. New York : ACM Press ; Harlow, England : Addison-Wesley, 2011. (Cité en pages 54 et 70.)

[Baniardalani 2013] Sobhi Baniardalani et Javad Askari. *Fault diagnosis of timed discrete event systems using dioid algebra.* International Journal of Control, Automation and Systems, vol. 11, no. 6, pages 1095–1105, 2013. (Cité en page 20.)

[Bentley 1975] Jon Louis Bentley. *Multidimensional binary search trees used for associative searching.* Communications of the ACM, vol. 18, no. 9, pages 509–517, 1975. (Cité en page 46.)

[Bettini 1998] Claudio Bettini, Xiaoyang Sean Wang et Sushil Jajodia. *Mining Temporal Relationships with Multiple Granularities in Time Sequences.* IEEE Data Engineering Bulletin, vol. 21, no. 1, pages 32–38, 1998. (Cité en page 25.)

[Bingham 2001] Ella Bingham et Heikki Mannila. *Random projection in dimensionality reduction : applications to image and text data.* Dans Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 245–250, 2001. (Cité en pages 24, 102 et 118.)

[Boufaied 2005] Amine Boufaied, Audine Subias et Michel Combacau. *Distributed time constraints verification modelled with time Petri nets.* Dans 17th IMACS Word Congress on Scientific Computation, Applied Mathematics and Simulation, 2005. (Cité en page 20.)

[Bradley 1997] Paul S. Bradley, Olvi L. Mangasarian et W. Nick Street. *Clustering via concave minimization.* Dans Advances in neural information processing systems, pages 368–374, 1997. (Cité en page 46.)

[Campello 2013] Ricardo J. G. B. Campello, Davoud Moulavi et Jörg Sander. *Density-based clustering based on hierarchical density estimates.* Dans Pacific-Asia conference on knowledge discovery and data mining, pages 160–172, 2013. (Cité en page 47.)

[Carle 2012] Patrice Carle, Christine Choppy, Romain Kervarc et Ariane Piel. *Behavioural Analysis for Distributed Simulations.* Dans 19th Asia-Pacific Software Engineering Conference, APSEC 2012, pages 482–487, 2012. (Cité en page 17.)

[Carrault 1999] Guy Carrault, Marie-Odile Cordier, René Quiniou, Mireille Garreau, Jean-Jacques Bellanger et Alain Bardou. *A Model-Based Approach for Learning to Identify Cardiac Arrhythmias.* Dans Artificial Intelligence in Medicine : Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, AIMDM'99, pages 165–174, 1999. (Cité en page 17.)

[Cordier 2007] Marie-Odile Cordier, Xavier Le Guillou, Sophie Robin, Laurence Rozé et Thierry Vidal. *Distributed chronicles for on-line diagnosis of web*

services. Dans 18th International Workshop on Principles of Diagnosis, pages 37–44, 2007. (Cité en page 17.)

[Coxeter 1973] Harold Scott Macdonald Coxeter. Regular polytopes. Courier Corporation, 1973. (Cité en page 117.)

[Cram 2008] Damien Cram. *Découverte interactive et complète de motifs temporels intéressants à partir de traces d'interactions*. Rapport technique, 2008. (Cité en pages 28, 29 et 31.)

[Cram 2009] Damien Cram. *Scheme Emerger*. `https://sourceforge.net/projects/schemerger/`, 2009. [consulté le 30 septembre 2019]. (Cité en page 100.)

[Cram 2010] Damien Cram. *Découverte interactive et complète de chroniques : application à la co-construction de connaissances à partir de traces*. Thèse de doctorat, Université Claude Bernard Lyon I, 2010. (Cité en page 28.)

[Cram 2012] Damien Cram, Benoît Mathern et Alain Mille. *A complete chronicle discovery approach : application to activity analysis*. Expert Systems, vol. 29, no. 4, pages 321–346, 2012. (Cité en pages 25, 28, 29, 30, 33, 34, 43, 45 et 48.)

[Dasgupta 2003] Sanjoy Dasgupta et Anupam Gupta. *An elementary proof of a theorem of Johnson and Lindenstrauss*. Random Struct. Algorithms, vol. 22, no. 1, pages 60–65, 2003. (Cité en page 102.)

[Dauxais 2017] Yann Dauxais, Thomas Guyet, David Gross-Amblard et André Happe. *Discriminant Chronicles Mining - Application to Care Pathways Analytics*. Dans 16th Conference on Artificial Intelligence in Medicine, AIME 2017, pages 234–244, 2017. (Cité en pages 17, 32 et 33.)

[Dauxais 2018] Yann Dauxais. *Exctraction de chroniques discriminantes*. Thèse de doctorat, Rennes 1, 2018. (Cité en pages 17 et 32.)

[De Smedt 2017] Johannes De Smedt, Galina Deeva et Jochen De Weerdt. *Behavioral Constraint Template-Based Sequence Classification*. Dans Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 20–36, 2017. (Cité en page 21.)

[Dechter 1991] Rina Dechter, Itay Meiri et Judea Pearl. *Temporal Constraint Networks*. Artificial Intelligence, vol. 49, no. 1, pages 61–95, 1991. (Cité en pages 8, 9 et 15.)

[Dice 1945] Lee Raymond Dice. *Measures of the amount of ecologic association between species*. Ecology, vol. 26, no. 3, pages 297–302, 1945. (Cité en page 54.)

[Dousson 1993] Christophe Dousson, Paul Gaborit et Malik Ghallab. *Situation Recognition : Representation and Algorithms*. Dans Proceedings of the 13th International Joint Conference on Artificial Intelligence, pages 166–174, 1993. (Cité en pages 12, 16, 27, 30 et 70.)

[Dousson 1994] Christophe Dousson. *Suivi d'évolutions et reconnaissance de chroniques*. Thèse de doctorat, Université Paul Sabatier Toulouse III, 1994. (Cité en pages 12, 15 et 16.)

[Dousson 1999] Christophe Dousson et Thang Vu Duong. *Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems.* Dans IJCAI 99 : Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pages 620–626, 1999. (Cité en pages 25, 26, 27, 29, 33, 43, 45 et 48.)

[Dousson 2007] Christophe Dousson et Pierre Le Maigat. *Chronicle Recognition Improvement Using Temporal Focusing and Hierarchization.* Dans IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 324–329, 2007. (Cité en page 17.)

[Dousson 2008] Christophe Dousson, Fabrice Clérot et Françoise Fessant. *Method for the machine learning of frequent chronicles in an alarm log for the monitoring of dynamic systems*, 2008. U.S. Patent 7 388 482 B2, 17 juin 2008. (Cité en page 100.)

[Dunn 1973] Joseph C. Dunn. *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.* Journal of Cybernetics, vol. 3, no. 3, pages 32–57, 1973. (Cité en page 46.)

[Ester 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise.* Dans Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pages 226–231, 1996. (Cité en pages 44, 46, 65 et 81.)

[Fern 2002] Alan Fern, Robert Givan et Jeffrey Mark Siskind. *Specific-to-general learning for temporal events with application to learning event definitions from video.* Journal of Artificial Intelligence Research, vol. 17, pages 379–449, 2002. (Cité en page 77.)

[Fessant 2004] Françoise Fessant, Fabrice Clérot et Christophe Dousson. *Mining of an alarm log to improve the discovery of frequent patterns.* Dans Industrial Conference on Data Mining, pages 144–152, 2004. (Cité en page 28.)

[Fessant 2006] Françoise Fessant et Fabrice Clérot. *An efficient SOM-based preprocessing to improve the discovery of frequent patterns in alarm logs.* Dans Conference on Data Mining| DMIN, volume 6, page 277, 2006. (Cité en page 28.)

[Fradkin 2015] Dmitriy Fradkin et Fabian Mörchen. *Mining sequential patterns for classification.* Knowledge and Information Systems, vol. 45, no. 3, pages 731–749, 2015. (Cité en pages 25 et 75.)

[Giannella 2002] Chris Giannella, Jiawei Han, Jian Pei, Xifeng Yan et Philip S. Yu. Mining frequent patterns in data streams at multiple time granularities, pages 191–212. Chapman & Hall/CRC, 2002. (Cité en page 25.)

[Giannotti 2006] Fosca Giannotti, Mirco Nanni, Dino Pedreschi et Fabio Pinelli. *Mining sequences with temporal annotations.* Dans Proceedings of the 2006 ACM symposium on Applied computing, pages 593–597, 2006. (Cité en page 24.)

[Gougam 2012] Houssam-Eddine Gougam, Audine Subias et Yannick Pencolé. *Timed diagnosability analysis based on chronicles.* Dans 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pages 1256–1261, 2012. (Cité en page 20.)

[Gougam 2015] Houssam-Eddine Gougam. *Analyse de l'impact du temps sur la diagnosticabilité des systèmes à événements discrets.* Thèse de doctorat, INSA de Toulouse, 2015. (Cité en page 19.)

[Gougam 2017] Houssam-Eddine Gougam, Yannick Pencolé et Audine Subias. *Diagnosability analysis of patterns on bounded labeled prioritized Petri nets.* Discrete Event Dynamic Systems, vol. 27, no. 1, pages 143–180, 2017. (Cité en page 19.)

[Gunopulos 2003] Dimitrios Gunopulos, Roni Khardon, Heikki Mannila, Sanjeev Saluja, Hannu Toivonen et Ram Sewak Sharma. *Discovering All Most Specific Sentences.* ACM Trans. Database Syst., vol. 28, no. 2, pages 140–174, 2003. (Cité en page 22.)

[Guyet 2008] Thomas Guyet et René Quiniou. *Mining Temporal Patterns with Quantitative Intervals.* Dans 2008 IEEE International Conference on Data Mining Workshops, pages 218–227, 2008. (Cité en pages 25 et 103.)

[Guyet 2011] Thomas Guyet et René Quiniou. *Extracting Temporal Patterns from Interval-Based Sequences.* Dans IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pages 1306–1311, 2011. (Cité en pages 25 et 103.)

[Henrikson 1999] Jeff Henrikson. *Completeness and total boundedness of the Hausdorff metric.* MIT Undergraduate Journal of Mathematics, vol. 1, pages 69–80, 1999. (Cité en page 117.)

[Houssin 2007] Laurent Houssin, Sébastien Lahaye et Jean-Louis Boimond. *Just in time control of constrained (max,+)-linear systems.* Discrete Event Dynamic Systems, vol. 17, no. 2, pages 159–178, 2007. (Cité en page 20.)

[Huang 2012] Zhengxing Huang, Xudong Lu et Huilong Duan. *On mining clinical pathway patterns from medical behaviors.* Artificial intelligence in medicine, vol. 56, no. 1, pages 35–50, 2012. (Cité en pages 17, 32 et 33.)

[Jaccard 1912] Paul Jaccard. *The distribution of the flora in the alpine zone.* New phytologist, vol. 11, no. 2, pages 37–50, 1912. (Cité en page 39.)

[Johnson 1984] William Johnson et Joram Lindenstrauss. *Extensions of Lipschitz maps into a Hilbert space.* Contemporary Mathematics, vol. 26, pages 189–206, 1984. (Cité en pages 102 et 114.)

[Kaufman 1987] Leonard Kaufman et Peter J. Rousseeuw. *Clustering by means of Medoids.* Data Analysis based on the L1-Norm and Related Methods, pages 405–416, 1987. (Cité en page 46.)

[Le Corronc 2017] Euriell Le Corronc, Alexandre Sahuguède et Yannick Pencolé. *Détection et localisation de fautes temporelles dans les systèmes (max,+)-*

linéaires. Dans Modélisation des Systèmes Réactifs (MSR 2017), page 14, 2017. (Cité en page 20.)

[Le Corronc 2018] Euriell Le Corronc, Alexandre Sahuguède, Yannick Pencolé et Claire Paya. *Localization of time shift failures in (max,+)-linear systems*. Dans 14th IFAC Workshop on Discrete Event Systems WODES 2018, volume 51, pages 186–191, 2018. (Cité en page 20.)

[Le Guillou 2008] Xavier Le Guillou, Marie-Odile Cordier, Sophie Robin et Laurence Rozé. *Chronicles for On-line Diagnosis of Distributed Systems*. Dans ECAI, volume 8, pages 194–198, 2008. (Cité en page 17.)

[Lin 2007] Jessica Lin, Eamonn Keogh, Li Wei et Stefano Lonardi. *Experiencing SAX : a novel symbolic representation of time series*. Data Mining and knowledge discovery, vol. 15, no. 2, pages 107–144, 2007. (Cité en page 122.)

[Lloyd 1982] Stuart Lloyd. *Least squares quantization in PCM*. IEEE Transactions on Information Theory, vol. 28, no. 2, pages 129–137, 1982. (Cité en page 46.)

[Ma 2001] Sheng Ma et Joseph L. Hellerstein. *Mining Partially Periodic Event Patterns with Unknown Periods*. Dans Proceedings of the 17th International Conference on Data Engineering, pages 205–214, 2001. (Cité en page 25.)

[Maitre 2014] Ghyslain Maitre. Analyse de chroniques : cadre formel et critères d'évaluation. Mémoire de maîtrise, Université Paul Sabatier Toulouse III, 2014. (Cité en page 13.)

[Maitre 2015] Ghyslain Maitre, Yannick Pencolé, Audine Subias et Houssam Eddine Gougam. *Modélisation et Analyse de chroniques pour le diagnostic*. Dans Modélisation des Systèmes Réactifs (MSR 2015), 2015. (Cité en page 13.)

[Mannila 1995] Heikki Mannila, Hannu Toivonen et A. Inkeri Verkamo. *Discovering Frequent Episodes in Sequences*. Dans Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), pages 210–215, 1995. (Cité en pages 18 et 22.)

[Mannila 1996] Heikki Mannila et Hannu Toivonen. *Discovering Generalized Episodes Using Minimal Occurrences*. Dans Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), volume 96, pages 146–151, 1996. (Cité en pages 18 et 19.)

[Mannila 1997] Heikki Mannila, Hannu Toivonen et A Inkeri Verkamo. *Discovery of frequent episodes in event sequences*. Data mining and knowledge discovery, vol. 1, no. 3, pages 259–289, 1997. (Cité en pages 18 et 22.)

[Mannila 2001] Heikki Mannila et Jouni K. Seppänen. *Finding similar situations in sequences of events via random projections*. Dans Proceedings of the First SIAM International Conference on Data Mining, SDM 2001, pages 1–16, 2001. (Cité en pages 24 et 102.)

[Manning 2008] Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze. Introduction to information retrieval. Cambridge University Press, 2008. (Cité en page 70.)

[Mayer 1998] Emmanuel Mayer. *Inductive Learning of Chronicles.* Dans Proceedings of the 13th European Conference on Artificial Intelligence, pages 471–472, 1998. (Cité en page 33.)

[Meiri 1996] Itay Meiri. *Combining qualitative and quantitative constraints in temporal reasoning.* Artificial Intelligence, vol. 87, no. 1-2, pages 343–385, 1996. (Cité en page 8.)

[Merlin 1976] Philip Merlin et David Farber. *Recoverability of Communication Protocols - Implications of a Theoretical Study.* IEEE Transactions on Communications, vol. 24, no. 9, 1976. (Cité en page 20.)

[Mitsa 2010] Theophano Mitsa. Temporal data mining. Chapman and Hall/CRC, 2010. (Cité en page 21.)

[Mörchen 2005] Fabian Mörchen et Alfred Ultsch. *Optimizing time series discretization for knowledge discovery.* Dans Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 660–665, 2005. (Cité en page 122.)

[Mörchen 2010a] Fabian Mörchen. *SIPO.* `http://www.mybytes.de/sipo.zip`, 2010. [consulté le 1 août 2019]. (Cité en page 75.)

[Mörchen 2010b] Fabian Mörchen et Dmitriy Fradkin. *Robust mining of time intervals with semi-interval partial order patterns.* Dans Proceedings of the 2010 SIAM International Conference on Data Mining, pages 315–326, 2010. (Cité en pages 24 et 75.)

[Morin 2003] Benjamin Morin et Hervé Debar. *Correlation of Intrusion Symptoms : An Application of Chronicles.* Dans Recent Advances in Intrusion Detection : 6th International Symposium, RAID 2003, pages 94–112, 2003. (Cité en page 17.)

[Moskovitch 2015] Robert Moskovitch et Yuval Shahar. *Classification-driven temporal discretization of multivariate time series.* Data Mining and Knowledge Discovery, vol. 29, no. 4, pages 871–913, 2015. (Cité en page 122.)

[Obry 2016] Tom Obry. Acquisition de traces d'activité pour l'apprentissage de démarches métier. Mémoire de maîtrise, Université Paul Sabatier Toulouse III, 2016. (Cité en page 19.)

[Obry 2017] Tom Obry, Audine Subias et Louise Travé-Massuyès. *A Learning Algorithm for Episodes.* Dans 28th International Workshop on Principles of Diagnosis, pages 1–11, 2017. (Cité en page 19.)

[Obry 2018] Tom Obry, Louise Travé-Massuyès et Audine Subias. *Computer-aided Diagnosis via Hierarchical Density Based Clustering.* Dans 29th International Workshop on Principles of Diagnosis, page 8, 2018. (Cité en page 47.)

[Pandalai 2000] Deepa N Pandalai et Larry E Holloway. *Template languages for fault monitoring of timed discrete event processes.* IEEE transactions on automatic control, vol. 45, no. 5, pages 868–882, 2000. (Cité en page 21.)

[Paya 2018] Claire Paya. Formalisation d'une méthode de diagnostic de systèmes (max,+)-linéaires incertains. Mémoire de maîtrise, Université Paul Sabatier Toulouse III, 2018. (Cité en page 20.)

[Pearson 1900] Karl Pearson. *On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 50, no. 302, pages 157–175, 1900. (Cité en page 25.)

[Pencolé 2009] Yannick Pencolé et Audine Subias. *A Chronicle-based Diagnosability Approach for Discrete Timed-event Systems : Application to Web-Services.* Journal of Universal Computer Science, vol. 15, no. 17, pages 3246–3272, 2009. (Cité en pages 17 et 20.)

[Penrose 1955] Roger Penrose. *A generalized inverse for matrices.* Mathematical Proceedings of the Cambridge Philosophical Society, vol. 51, no. 3, pages 406–413, 1955. (Cité en page 107.)

[Peterson 1977] James L Peterson. *Petri nets.* ACM Computing Surveys (CSUR), vol. 9, no. 3, pages 223–252, 1977. (Cité en page 19.)

[Provan 2017] Gregory M. Provan. *An Algebraic Approach for Diagnosing Discrete-Time Hybrid Systems.* Dans 28th International Workshop on Principles of Diagnosis, pages 37–51, 2017. (Cité en page 20.)

[Quinqueton 1997] Joël Quinqueton, Babak Esfandiari et Richard Nock. *Chronicle learning and agent oriented techniques for network management and supervision.* Dans Intelligent Networks and Intelligence in Networks, pages 131–146. Springer, 1997. (Cité en page 33.)

[Raghavan 1989] Vijay Raghavan, Peter Bollmann et Gwang S. Jung. *A critical investigation of recall and precision as measures of retrieval system performance.* ACM Transactions on Information Systems (TOIS), vol. 7, no. 3, pages 205–229, 1989. (Cité en page 72.)

[Sahuguède 2016] Alexandre Sahuguède. Formalisation d'une méthode de diagnostic de systèmes (max,+)-linéaires. Mémoire de maîtrise, Université Paul Sabatier Toulouse III, 2016. (Cité en page 20.)

[Sahuguède 2017] Alexandre Sahuguède, Euriell Le Corronc et Yannick Pencolé. *Design of indicators for the detection of time shift failures in (max, +)-linear systems.* Dans 20th World Congress of The International Federation of Automatic Control, pages 6813–6818, 2017. (Cité en page 20.)

[Sahuguède 2018a] Alexandre Sahuguède, Soheib Fergani, Euriell Le Corronc et Marie-Véronique Le Lann. *Mapping Chronicles to a k-dimensional Euclidean Space via Random Projections.* Dans 14th annual IEEE International Conference on Automation Science and Engineering, IEEE CASE 2018, pages 1177–1182, 2018. (Cité en pages 102 et 121.)

[Sahuguède 2018b] Alexandre Sahuguède, Euriell Le Corronc et Marie-Véronique Le Lann. *Chronicle Discovery for Diagnosis from Raw Data : A Clustering*

Approach. Dans 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS 2018, pages 1–8, 2018. (Cité en pages 34 et 48.)

[Sahuguède 2018c] Alexandre Sahuguède, Euriell Le Corronc et Marie-Véronique Le Lann. *An Ordered Chronicle Discovery Algorithm*. Dans 3nd ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, AALTD'18, page 6, 2018. (Cité en page 34.)

[Sørensen 1948] Thorvald Sørensen. *A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons*. Biologiske Skrifter, vol. 5, pages 1–34, 1948. (Cité en page 54.)

[Subias 2013] Audine Subias. *Diagnosis from chronicles : an overview of related challenges*. Dans Congreso International de Ingenieria Mecatronica y Automatizacion, pages 1–11, 2013. (Cité en page 17.)

[Toth 2017] Csaba D Toth, Joseph O'Rourke et Jacob E. Goodman. Handbook of discrete and computational geometry. CRC Press, 2017. (Cité en page 104.)

[Tversky 1977] Amos Tversky. *Features of similarity*. Psychological review, vol. 84, no. 4, pages 327–352, 1977. (Cité en page 54.)

[van der Aalst 2004] Wil M. P. van der Aalst, Anton J. M. M. Weijters et Laura Maruster. *Workflow mining : Discovering process models from event logs*. IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pages 1128–1142, 2004. (Cité en page 23.)

[van der Aalst 2007] Wil M. P. van der Aalst, Hajo A. Reijers, Anton J. M. M. Weijters, Boudewijn F. van Dongen, A. K. Alves De Medeiros, Minseok Song et H. M. W. Verbeek. *Business process mining : An industrial application*. Information Systems, vol. 32, no. 5, pages 713–732, 2007. (Cité en pages 23 et 24.)

[Vasquez Capacho 2017] John William Vasquez Capacho, Audine Subias, Louise Travé-Massuyès et Fernando Jimenez. *Alarm management via temporal pattern learning*. Engineering Applications of Artificial Intelligence, vol. 65, pages 506–516, 2017. (Cité en pages 17, 30, 33, 34 et 48.)

[Vempala 2005] Santosh S. Vempala. The random projection method. American Mathematical Soc., 2005. (Cité en pages 102 et 121.)

[Vesanto 2000] Juha Vesanto et Esa Alhoniemi. *Clustering of the self-organizing map*. IEEE Transactions on neural networks, vol. 11, no. 3, pages 586–600, 2000. (Cité en page 28.)

[Vila 1994] Lluís Vila et Lluís Godo Lacasa. *On fuzzy temporal constraint networks*. Mathware & soft computing, vol. 1, no. 3, pages 315–334, 1994. (Cité en page 9.)

[Vu Duong 2001] Thang Vu Duong. *Découverte de chroniques à partir de journaux d'alarmes : application à la supervision de réseaux de télécommunications*. Thèse de doctorat, Toulouse, INPT, 2001. (Cité en page 26.)

[Wang 2004] Jianyong Wang et Jiawei Han. *BIDE : Efficient mining of frequent closed sequences.* Dans Proceedings. 20th international conference on data engineering, pages 79–90, 2004. (Cité en page 24.)

[Ward 1963] Joe H. Ward Jr. *Hierarchical grouping to optimize an objective function.* Journal of the American statistical association, vol. 58, no. 301, pages 236–244, 1963. (Cité en page 47.)

[Wongsuphasawat 2012] Krist Wongsuphasawat, Catherine Plaisant, Meirav Taieb-Maimon et Ben Shneiderman. *Querying event sequences by exact match or similarity search : Design and empirical evaluation.* Interacting with Computers, vol. 24, no. 2, pages 55–68, 2012. (Cité en page 24.)

[Yoshida 2000] Mariko Yoshida, Tetsuya Iizuka, Hisako Shiohara et Masanori Ishiguro. *Mining sequential patterns including time intervals.* Dans Data Mining and Knowledge Discovery : Theory, Tools, and Technology II, volume 4057, pages 213–220, 2000. (Cité en page 24.)

[Zaki 2001] Mohammed J. Zaki. *SPADE : An efficient algorithm for mining frequent sequences.* Machine learning, vol. 42, no. 1-2, pages 31–60, 2001. (Cité en page 22.)

Résumé : Les chroniques sont des schémas temporels particulièrement bien adaptés pour une représentation de modèles complexes et dynamiques. Des algorithmes de reconnaissance de chroniques permettent d'identifier des chroniques dans un flux de données en ligne et de prendre des actions adéquates de manière rapide et efficace. Les chroniques peuvent être utilisées dans des domaines d'applications divers, tels que le domaine médical, les réseaux internets, ou encore des applications industrielles. Néanmoins, la construction des chroniques n'est pas chose aisée en raison de la complexification et de l'augmentation des capacités de génération de données des systèmes modernes. Le processus de découverte de chroniques a pour objectif de répondre à cette problématique en construisant de manière automatique des chroniques à partir des données directement générées par le système étudié. Dans ce mémoire de thèse, une approche innovante à la problématique de la découverte de chroniques est abordée. Cette nouvelle approche repose sur une identification de chroniques élémentaires et une reconstitution de chroniques plus complexes à partir de celles-ci. L'algorithme proposé, appelé CDIRe (*Chronicle Discovery by Identification and Reconstitution)*, permet de découvrir des chroniques avec peu de connaissance sur le système sous-jacent.
Mots clés : Apprentissage, chroniques, fouille de données, systèmes à événements discrets.

Abstract: Chronicles are temporal patterns well-suited for an abstract representation of complex dynamic systems. Chronicle recognition algorithms allow the identification of chronicles in an on-line stream of data to be done and take adequate action in an quick and efficient manner. Chronicles are used in a vast array of applications such as medical field, internet networks, or industrial applications. Nevertheless, designing chronicles is not an easy task due to the sophistication and the increase of data generation capacity of modern systems. The chronicle discovery process try and tackle this problem by an automatic design of chronicles from data directly generated by the studied system. In this thesis, an innovative approach to the problem of chronicle discovery is introduced. This new approach lies of the identification of elementary chronicles and a reconstitution of complex chronicles from them. The algorithm introduced, called CDIRe (*Chronicle Discovery by Identification and Reconstitution*), allows the discovery of chronicles with few knowledge from the underlying system to be done.
Keywords: Machine learning, chronicles, data mining, discrete event systems.