Extraction des utilisations typiques à partir de données hétérogènes en vue d'optimiser la maintenance d'une flotte de véhicules

Asma Ben Zakour

N^o d'ordre :4539

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

par Asma Ben Zakour

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Informatique

Extraction des utilisations typiques à partir de données hétérogènes historisées en vue d'optimiser la maintenance d'une flotte de véhicules

---

Soutenue le : 06/07/2012

Après avis de :

M.	Omar Boucelma ...	Professeur des universités	**Rapporteur**
M^{me}	Magelonne Teisseire	Directrice de recherche ..	**Rapporteur**

Devant la Commission d'Examen composée de :

M.	Nicolas Hanusse ..	Directeur de recherche	**Examinateur**
M.	Sofian Maabout .	Maître de conférences	**Co-directeur**
M.	Mohamed Mosbah	Professeur	**Directeur**
M.	Marc Sistiaga	Ingénieur	**Examinateur**

- 2012 -

Extraction des utilisations typiques à partir de données hétérogènes historisées en vue d'optimiser la maintenance d'une flotte de véhicules

Asma Ben Zakour

Remercient

Extraction des utilisations typiques à partir de données hétérogènes historisées en vue d'optimiser la maintenance d'une flotte de véhicules

Résumé : Le travail produit s'inscrit dans un cadre industriel piloté par la société 2MoRO Solutions. La réalisation présentée dans cette thèse doit servir à l'élaboration d'un service à haute valeur, permettant aux exploitants aéronautiques d'optimiser leurs actions de maintenance.

Étant donnée le grand volume de données disponibles autour de l'exploitation des aéronefs, ce travail vise à analyser les historiques des évènements associés aux aéronefs afin d'en extraire des prévisions de maintenance. Les résultats obtenus permettent d'intégrer et de regrouper les tâches de maintenance en vue de minimiser la durée d'immobilisation des aéronefs et d'en réduire les risques de panne.

La méthode que nous proposons comporte trois étapes : (i) une étape de rationalisation des informations afin de pouvoir les combiner, (ii) une étape d'organisation de ces données pour en faciliter l'analyse (iii) une étape d'extraction des connaissances utiles sous formes de *séquences intéressantes*.

Nous introduisons un nouveau type de séquences baptisées les « Séquences Temporelles par Intervalles d'incertitude (*STI*) ». Elles représentent les comportements chronologiques ordonnés en intégrant une souplesse temporelle locale aux évènements qui la composent. Pour extraire de telles séquences, nous définissons l'algorithme *STI-PS*.

Mots-clefs :Extraction, séquences fréquentes, contraintes temporelles, intervalles d'incertitude, fenêtre glissante, maintenance aéronautique, prévision de la maintenance.

Discipline : Informatique.

Critical usages extraction from historical and heterogenious data in order to optimize fleet maintenance

Abstract: The present work is part of an industrial project driven by 2MoRO Solutions company. It aims to develop a high value service enabling aircraft operators to optimize their maintenance actions.

Given the large amount of data available around aircraft exploitation, we aim to analyse the historical events recorded with each aircraft in order to extract maintenance forecasting. The results are used to integrate and consolidate maintenance tasks in order to minimize aircraft downtime and risk of failure.

The proposed method involves three steps : (i) streamlining information in order to combine them, (ii) organizing this data for easy analysis and (iii) an extraction step of useful knowledge in the form of *interesting* sequences.

We introduce a new type of sequences : "Temporal Sequences by ranges of uncertainty (*ITS*)". The events of these sequences are timestamped with temporal intervals expressing a time flexibility of their occurrences. We define the algorithm *STI-PS* to extract these patterns from historical data.

Keywords: Frequent sequences extraction, temporal constraints, uncertainty intervals, sliding window, aeronautic maintenance, maintenance prognostic.
Field: Computer Science.

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Université de Bordeaux 1
351 cours de la libération
33405 Talence FRANCE

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Introduction

Au vu des récentes avancées technologiques en termes de surveillance des systèmes complexes ainsi que de leurs équipements et en termes de gestion et de stockage des données, la plupart de ces systèmes se voient associer une masse importante d'informations au cours de leurs vies. Ces données correspondent à des relevés de paramètres d'utilisation, de mode de fonctionnement et d'information de maintenance et de réparation.

Il y a lieu de compiler, d'organiser, et de rationaliser cette masse d'informations afin d'en tirer profit pour améliorer les conditions de mise en service, de sécurité et de maintenance des systèmes concernés.

Spécifiquement en aéronautique, à chaque avion mis en service correspond une grande masse d'informations hétérogènes manipulées par plusieurs intervenants qui ne communiquent pas nécessairement entre eux et qui n'utilisent pas forcément les mêmes outils informatiques pour gérer et stocker ces données. Dans ce domaine, la maintenance des appareils s'avère rapidement onéreuse du fait de la complexité des équipements sollicités, des matériaux utilisés, des ressources, qualifications et moyens mis à contribution pour la réaliser. Par voie de conséquence, la disponibilité optimale de la flotte d'aéronefs est indispensable afin de compenser ces coûts. L'objectif de minimisation la durée de l'indisponibilité des équipements aéronautiques s'impose aujourd'hui comme étant une méthode de rationalisation de la gestion de la maintenance. Même si le risque zéro n'existe pas, la sécurité des passagers est une priorité fondamentale qui s'inscrit dans un plan de minimisation des risques et du nombre de pannes.

Ainsi donc, il y a lieu de mettre à profit la masse d'informations disponible autour de la vie d'une flotte d'aéronefs similaires, afin d'en tirer profit pour assurer le maximum de sécurité, à moindre coût (coût d'exploitation et de maintenance). Il s'agit donc d'atteindre le double objectif : Sécurité - coût.
Dans ce contexte, la société *2MoRO Solutions* a développé une plateforme collaborative de services destinée aux acteurs du monde aéronautique. Cette plateforme est capable de stocker et de manipuler un volume de données très important (plusieurs Téraoctets mensuels pour une flotte de 50 hélicoptères). Ces données se caractérisent par leur forte hétérogénéité : boitiers embarqués (trajectoire, courbe de températures, pression, régime moteur, couple, vibrations, ...), conditions environnementales d'opération de l'aéronef (météorologie, type d'atmosphère, survol de territoires particuliers, ...) et d'informations humaines (comptes-rendus pilote, comptes-rendus technicien, ...), le tout étant précisément daté.

Un des services à valeur ajoutée que doit proposer la plateforme collaborative est de fournir au client un moyen d'exploiter ces données afin d'optimiser la maintenance et l'exploitation de

sa flotte de véhicules. L'analyse de l'historique des données de ces vols et les prévisions de ceux à venir permettront la mise en correspondance de caractéristiques significatives afin de dégager des tendances et d'en déduire un programme de maintenance adapté à la flotte ainsi que des recommandations d'utilisation à l'usage des pilotes.

Les travaux présentés dans ce mémoire sont nés suite à cette initiative industrielle. L'objectif étant de proposer une méthode qui permette de fournir la valeur ajoutée énoncée plus haut. Notre démarche consiste à s'appuyer sur l'analyse d'un historique opérationnel et de maintenance compilant des données de plusieurs véhicules pour dégager des tendances de comportements et proposer un service de planification de maintenance intelligent et optimisé, par opposition aux abaques statiques fournis par les constructeurs (OEM pour Original Equipment Manufacturer). Ce service apporte une réelle plus-value basée sur l'utilisation propre des aéronefs par l'opérateur pour proposer une amélioration singulière de la gestion de la maintenance des véhicules de la flotte, une exploitation poussée des données historisées des aéronefs pour tendre à une réduction significative des coûts liés à cette maintenance. L'objectif de cette thèse est donc de proposer un service d'aide à la maintenance.

Fouille de données

Les techniques de fouille de données", basées sur des analyses statistiques, des méthodes d'apprentissage supervisées ou non supervisée (clustering,...), connaissent un développement importants dans plusieurs domaines scientifiques, économiques et médicaux. Ces dernières années, ces techniques de fouilles de données ont connu des avancées significatives et des résultats probants. Cependant, dans le domaine de l'aéronautique les applications concrètes sont restées, jusque là, restreintes et limitées.

Ces différentes techniques se rejoignent autour de la stratégie d'application des différentes méthodes de fouille sur les données brutes. Effectivement, toutes les méthodes proposées de fouille de données appliquent la même stratégie en trois étapes : La première consiste en la préparation et le pré-traitement des données brutes, la seconde met en œuvre la méthode de fouille et la troisième est une étape de validation et de correction des modèles mis en place lors de la deuxième étape.

Le travail présenté dans ce mémoire a pour but de développer un modèle de prédiction et

d'aide à la décision destiné à la maintenance aéronautique en appliquant un algorithme d'extraction de séquences fréquentes.

La prise en compte de la nature des données hétérogènes et datées (initiales) est cruciale pour le développement des modèles. De plus, ces derniers doivent être efficaces, en terme de performances techniques et fonctionnelles et doivent aussi assurer la pertinence et l'intégrité des données ainsi que des résultats retournés.

En partant des données historiques hétérogènes et datées qui sont collectées, notre travail se compose de trois principaux modules :

- Le premier consiste en une rationalisation et consolidation des informations hétérogènes recueillies. Durant cette étape, deux principales phases sont entreprises : (1) La première consiste à unifier les données issues de diverses sources afin d'en considérer la fusion comme un seul bloc d'informations. (2) La seconde organise les données et les découpe de manière à en faciliter l'analyse et l'exploitation par le modèle de prévision.
- Le second module consiste à définir et construire le modèle de prévision. La technique de fouille de données choisie est appliquée sur les données pré-traitées.
- Le troisième module est une validation des résultats retournés par la méthode mise en place. Cette validation prend en compte le double aspect technique et fonctionnel.

Pré-analyse et préparation des données brutes

La préparation des données brutes est une étape cruciale dans tout travail traitant de la fouille de données. Elle permet de formaliser les informations et de les préparer à l'analyse. Différents types de traitements peuvent être appliqués durant cette étape : l'élimination du bruit, la normalisation des données discrètes, [RBP09], la discrétisation des données continues [KT09] ou encore l'organisation des données en séquences [AIS93].

Partant du principe que les données dont nous disposons sont hétérogènes et proviennent de sources diverses, nous fusionnons dans un premier temps cette masse d'information en les alignant sur un axe temporel qui permet de créer un flux d'information pour chaque aéronef. Dans un deuxième temps, un découpage des données permet de créer des séquences d'informations reliant les données relatives à l'utilisation du véhicule, à la maintenance qui peut en être affectée. Dans un troisième temps les séquences sont regroupées selon une structure arborescente représentant la nomenclature du véhicule.

Extraction de comportements fréquents

Les techniques d'extraction permettent d'identifier des comportements typiques qui apparaissent souvent dans les données analysées. A travers ce type de fouille, différents aspects des comportement peuvent être mis en avant : la simple fréquence de succession des évènements [AS95] permet d'extraire des comportements sous forme de motifs séquentiels. La prise en compte du décalage temporel dans les successions des évènements quant à elle, extrait des motifs séquentiels temporels [PHMA+01]. Il est aussi possible d'extraire des relations entre les apparitions ou la durée des apparitions des évènements [WC07]. Afin de répondre aux besoins d'extraction spécifiés par les experts de maintenance aéronautique, nous définissons un nouveau type de comportements « intéressants ». Ces derniers combinent fréquence de succession globale des évènements des séquences extraites et souplesse temporelle locale des apparitions de ces derniers. La souplesse locale est gérée par une incertitude temporelle contrôlée et paramétrable par l'utilisateur de notre méthode.

Validation

Dans la plupart des travaux conduits sur l'extraction de connaissances pour l'apprentissage, la validation des modèles se fait par l'intervention d'un expert humain du domaine. Ce dernier intervient afin de calibrer le modèle construit et en ajuster les résultats [Fau07].

Dans ce mémoire, nous validons notre méthode en découpant les données en deux jeux de données ; le premier pour la construction du modèle et la seconde pour la validation de ce dernier.

Plan du mémoire

Ce mémoire est organisé en deux parties. La première présente un état de l'art des deux domaines concernés par notre travail : la maintenance aéronautique et l'extraction de séquences fréquentes. La seconde partie présente en trois chapitres la contribution que nous avons apportée à ces domaines à travers les travaux menés durant l'étude.

Première partie Les deux premiers chapitres concernent l'état de l'art : Le premier chapitre présente les fondements et les différentes politiques utilisées dans la maintenance aéronautique en spécifiant les différents intervenants qui s'y mêlent. Il détaille aussi les données qui apparaissent autour de la vie d'un aéronef et les travaux précédemment mis en place dans le monde de la recherche et de l'industrie pour les mettre à profit pour la maintenance et le diagnostic. Bien que

nous n'ayons aucune contribution dans ce domaine (la maintenance ne fait pas partie de notre champs d'étude) nous avons jugé utile d'inclure ce chapitre afin de situer le contexte de notre travail et motiver certains des choix faits lors de la mise en œuvre de notre solution

Un deuxième chapitre concerne l'extraction de séquence fréquente. Il présente dans un premier temps les définitions et concepts généraux de l'extraction de

séquence fréquente. Dans un deuxième temps, il présente les techniques d'extractions utilisées et les différents types de temporalités et de contraintes utilisées pour adapter et spécifié la formulation des séquences fréquentes extraites.

Deuxième partie La deuxième partie de ce mémoire présente la contribution apportée par les travaux réalisés en trois chapitres. Le premier présente la préparation et le prétraitement des données brutes. Il détaille la mise en relation et l'organisation de ces dernières en spécifiant les caractéristiques des structures utilisées. Le deuxième chapitre présente une nouvelle définition de séquences « intéressantes » pour la prévision des tâches de maintenance à partir de données historiques hétérogènes et datées et détaille l'algorithme mis en place pour les extraire. Le troisième chapitre évalue les performances techniques et fonctionnelles de la méthode que nous avons mise en place. Finalement, nous concluons et présentons les perspectives envisagées pour ce travail.

Première partie

État de l'art

Introduction

Pour l'extraction de connaissance à partir d'un volume important de données, toutes les techniques de fouille de données nécessitent une étude préalable du domaine d'application afin de spécifier l'utilisation et la nature des informations à extraire. De plus, une bonne connaissance de la représentation et de la forme des données à analyser permet de déterminer et de motiver le choix de la méthode d'extraction à utiliser.

Le but du travail présenté dans ce mémoire est de mettre à profit des données aéronautique hétérogènes et datées pour apporter une valeur ajoutée aux intervenants du domaine visant à améliorer la gestion de la maintenance d'une flotte d'aéronefs et d'en réduire le coût.

Les données à étudier englobent des informations fortement hétérogènes et portant sur différents niveaux de connaissance. Elles contiennent des données de bas niveau tels que les relevés de capteurs mais aussi des données de haut niveau tel que les rapports de vols ou de maintenance.

Nous proposons dans ce mémoire une méthode qui permet de consolider ces informations et de les analyser afin de fournir des prévisions de tâches de maintenance permettant un ordonnancement plus « économique » des interruptions pour la maintenance des avions. Cette partie présente un état de l'art sur la maintenance aéronautique dans un premier chapitre et sur les techniques d'extraction de séquences fréquentes dans un deuxième chapitre. Notre contribution sera présentée dans la deuxième partie du mémoire.

Tout d'abord, dans le premier chapitre de cette première partie, nous faisons une description du domaine d'application de nos travaux. Pour cela, nous parlons d'abord des politiques adoptées en maintenance aéronautique et des différents intervenants qui y interagissent. Par la suite, nous énumérons les données disponibles autour d'une flotte d'aéronefs. Nous finirons le chapitre par une synthèse des travaux rencontrés dans le monde de l'industrie et de la recherche et qui traitent

de l'exploitation des données pour la prévision de la maintenance et l'aide au diagnostic pour la maintenance aéronautique.

Ensuite, nous présentons dans un deuxième chapitre, un état de l'art succinct sur les techniques d'extraction de séquences fréquentes. Il s'agit de la technique de fouille de données que nous avons choisi de mettre en œuvre pour la prévision des tâches de maintenance. Effectivement, les besoins d'extraction et les constats de complexité et d'hétérogénéité des données nous ont permis de conclure qu'il n'est pas possible d'utiliser des techniques d'apprentissage supervisé car les données ne présentent pas de classe d'apprentissage.

D'une part, une étude préalable de l'application de méthodes de clustering sur les données s'est avérée inefficace à cause de la complexité de la distance entre des données aussi disparates. Aussi, le besoin de minimiser l'intervention d'experts humains nous a amené à écarter l'utilisation des réseaux bayésiens qui nécessitent de telles interventions à différents niveaux de l'extraction.

D'autre part la souplesse des techniques d'extractions de séquences fréquentes par rapport aux données à analyser et leurs applicabilités à des domaines variés [FVNN08] [AS94] [HY06] nous ont permis de porter notre choix sur cette méthode.

Nous présentons donc dans le deuxième chapitre de cette partie un état de l'art sur l'extraction des séquences fréquentes. D'abord, nous en spécifions les définitions et les concepts généraux. Par la suite, nous présentons les différentes techniques rencontrées dans la littératures, les améliorations et les optimisations d'extraction.

Chapitre 1

Extraction de connaissances pour la maintenance

Sommaire

1 Introduction

Dans le domaine aéronautique, la sécurité des avions et des passagers représente l'enjeu majeur de tous les intervenants : constructeurs , exploitants , intervenants de la maintenance .... Afin de garantir la sécurité des trajets aériens, la maintenance aéronautique se doit de respecter des procédures strictes. Elle applique une double stratégie préventive et curative.

La partie préventive est une application directe des différents documents de références émis par les OEM et les autorités de certifications (AMM, IPC, ...). La partie curative consiste à réparer ou remplacer les équipements identifiés comme défaillants entre deux vols et pendant les interruptions non prévues des véhicules suite à des dommages importants.

Cependant, malgré une stratégie de maintenance aussi stricte, les problèmes de fonctionnement et de diagnostic restent persistants à cause de la complexité des véhicules. Ces problèmes génèrent des coûts supplémentaires pour le maintien en condition opérationnelle d'une flotte dûs aux interruptions imprévues et aux délais de diagnostics important lors de la maintenance en ligne.

Pour pallier ces problèmes, un grand volume de données, issues de trois sources différentes et contenant des données embarquées, des données de vie et des données de référence, est utilisé au profit de stratégies d'amélioration de la gestion de la maintenance.

Ces stratégies diffèrent selon les besoins, et peuvent agir à deux niveaux différents : au niveau du diagnostic et ou au niveau du pronostic. Le premier agit pour une correspondance symptômes-fautes et permet l'isolation des pannes. Le second estime un niveau d'endommagement du système en général et de ses parties en particulier.
Ce chapitre présente dans une première section la définition industrielle de la maintenance et les différentes politiques qui peuvent êtres appliquées aux systèmes complexes. Dans une deuxième section, il décrit la maintenance dans le domaine aéronautique, ses différents intervenants et les protocoles à suivre ainsi que la politique adoptée. La troisième section du chapitre énumère les données disponibles autour de la mise en service et la maintenance d'un avion. La quatrième section présente un résumé de quelques techniques utilisées dans les domaines de la recherche et de l'industrie pour l'amélioration de la gestion de maintenance aéronautique.

2 La maintenance dans l'industrie

2.1 Définition

La maintenance est une composante fondamentale et indispensable des systèmes industriels complexes. Elle intervient à toutes les étapes de leur cycle de vie : de la conception au retrait de service. En effet, au cours de la conception d'un système, la prise en compte de sa maintenance permet de fixer ses paramètres de sécurité, sa résistance et sa limite de fonctionnement, ainsi que sa stratégie de maintenance. En revanche, pendant sa période de service, la maintenance intervient pour garantir son bon fonctionnement et permettre sa remise en service en cas de pannes. Enfin, à la fin de sa vie, la disponibilité des références de documents de maintenance ainsi que l'existence des valeurs de paramètres de sécurité permettront de choisir une des trois décisions suivantes : sa remise en service, son recyclage ou tout simplement sa destruction [BS08].

La section suivante présente les différents types de maintenance appliquées aux systèmes industriels complexes.

2.2 Procédés de maintenance

Il existe deux procédés différents de maintenance : la maintenance curative et la maintenance préventive.

- La maintenance curative : Elle est la forme la plus simple de maintenance ; elle intervient essentiellement après une panne et ne nécessite aucune planification, elle peut être palliative ou curative. Dans le cas où elle est palliative, les actions d'interventions et de maintenance ont un effet plutôt temporaire afin de garantir une reprise provisoire des fonctionnalités du système. Dans le deuxième cas (curative), elle permet une reprise permanente des fonctionnalités du système.

- La maintenance préventive : elle permet de réduire la probabilité de défaillance du système, c'est-à-dire de diminuer les risques de pannes et donc elle augmente les chances de disponibilité du système. Les actions de la maintenance préventive sont appliquées en amont des interruptions de fonctionnement du système.
  La maintenance préventive peut avoir trois formes : systémique, conditionnelle ou prévisionnelle.
  - La maintenance systémique définit des actions d'interventions à des intervalles réguliers calculées sur la base de relevés empiriques sur le système.
  - La maintenance conditionnelle se base sur la surveillance de l'état en fonctionnement du système et le suivi de l'état de dégradation. En général, dans les systèmes complexes la surveillance se fait sur l'enregistrement des données relevées par les capteurs intégrés.
  - La maintenance prévisionnelle se base sur le concept de la maintenance conditionnelle, elle exploite les relevés des équipements surveillés dans une analyse stochastique ou déterministe pour extrapoler les périodicités des applications des tâches de maintenance.

3 La maintenance aéronautique

Pour tout système complexe de type industriel la maintenance est un procédé de première importance dans la vie d'un spécimen.

De manière plus particulière, en aéronautique, la maintenance obéit à des règles très strictes émises et consignées par le constructeur dans une documentation règlementée et spécifique à

chaque produit vendu et livré. Ces documents mettent en place et décrivent de façon très détaillée des interventions et actions cycliques pré-planifiées à des intervalles de temps réguliers.

3.1 Généralités

La maintenance aéronautique appelée en anglais Maintenance Repair and Overhaul (MRO) est gérée par plusieurs intervenants :

- Les exploitants des véhicules qui appliquent la maintenance de haut niveau.
- Les intervenants externes qui appliquent la maintenance de haut et moyen niveau.
- Les constructeurs et équipes spécialisées qui agissent pour la maintenance de bas niveau.

La maintenance aéronautique est aussi sujette à des modifications approuvées par les autorités de certification. Elle obéit à un plan de maintenance émis par le constructeur qui décrit la périodicité et le détail des tâches de maintenance à effectuer. Un véhicule est donc associé à une stratégie de maintenance préventive.

Cependant des actions de maintenance correctives pourraient être appliquées suite à des identifications de défaillance lors des fréquentes inspections ou consignées par les rapports de défaillance. Ces rapports peuvent être émis par les acteurs de la maintenance ou le personnel d'utilisation du véhicule.

Le paragraphe suivant présente une énumération des différents types de maintenance appliquées aux aéronefs selon deux critères : le temps que nécessite la maintenance et la parie de l'avion à laquelle elle est appliquée.

3.2 Les deux critères de maintenance

En aéronautique, une action de maintenance est répertoriée selon deux critères : le premier est relatif au temps de maintenance noté T.A.T (Turn Around Time), le deuxième est relatif à la classification des actions de maintenance, il porte sur la localisation de l'élément concerné par l'action de maintenance.

a) Le premier critère relatif au temps de maintenance (T.A.T) enregistre et comptabilise la totalité du temps entre l'interruption et la remise en service du véhicule. Le T.A.T permet donc de distinguer la maintenance en ligne de la maintenance lourde :
   - La maintenance en ligne se caractérise par un T.A.T relativement faible (entre quelques minutes et quelques heures). Dans le cas des vols commerciaux, la maintenance en ligne consiste à garantir la sécurité du vol et des passagers et assurer la réussite de la mission. Les intervenants font partie de la compagnie aérienne ou d'un sous traitant MRO. Leur mission consiste à vérifier, réparer et/ou remplacer un ensemble d'équipements décrits par une liste

précise d'éléments de type L.R.U (Line Replaceable Unit). La maintenance en ligne dite aussi légère ne nécessite généralement pas de déplacement du véhicule dans l'atelier ; c'est pourquoi, dans la plus part des cas elle est effectuée sur la piste de l'aérogare.

- En revanche, la maintenance lourde exige de plus gros moyens et une haute qualification des intervenants. Les actions entreprises durant ce type de maintenance sont pointues et très précises, c'est pourquoi elles sont strictement gérées par une documentation détaillée et précise émise par les constructeurs. L'avion est déplacé dans le hangar-atelier de réparation. Dans ce cas, le T.A.T peut varier de quelques jours à quelques mois. Ce type de maintenance est effectué selon une gestion rigoureuse et exige des outils et pièces de rechange très précis. A noter aussi, qu'au cours de ce T.A.T, plusieurs actions s'exécutent de façon parallèle.

b) Le second critère de classification des actions de maintenance sur un avion identifie la composante de l'avion qui sera maintenue. Selon ce deuxième critère on distingue trois types de maintenance : la maintenance de la cellule, la maintenance du moteur et la maintenance des composants.

- la maintenance de la cellule (Airframe Maintenance) : consiste à inspecter, réparer et/ou remplacer des éléments de la structure de l'avion. Elle englobe essentiellement des actions à faible charge de travail mais à fréquences élevées. La plupart des actions de maintenance de la cellule sont effectuées en aérogare par les compagnies d'exploitation ou par des sous-traitant MRO.

  Cependant la structure de l'avion nécessite aussi des actions de maintenance lourde faite en atelier ; ce qui exige un temps d'arrêt plus ou moins long pouvant aller de un jour à plusieurs semaines.
- La maintenance du moteur (Engine Maintenance) : elle peut comporter des actions de maintenance légère, mais aussi des actions de maintenance lourde à forte charge de travail (caractérisée par un TAT particulièrement élevé allant de 40 à 60 jours). Une telle action est un ensemble de procédures complexes qui sont régies par une documentation spécifique décrivant les détails des instructions à suivre. Les intervenants sont des experts en maintenance moteur et sont en général envoyés spécialement par le constructeur afin d'exécuter ces opérations.
- La maintenance des composants (Components Maintenance) : de manière similaire aux deux maintenances sur la « cellule » et le « moteur », elle comporte d'une part, une maintenance en ligne qui porte sur des éléments de type LRU et d'autre part, une maintenance lourde rigoureusement décrite dans une documentation spécifique qui lui est spécialement dédiée.

La maintenance aéronautique est un procédé complexe régi par des procédures bien définies autour desquelles plusieurs documents sont disponibles. Ces documents peuvent être des consignes de réparation ou des rapports d'interventions émis par les experts à différents niveaux.

Dans le secteur aéronautique, aux différentes données de la maintenance se rajoutent des prélèvements et des données d'utilisation formant ainsi un volume d'informations très important dont l'exploitation peut apporter une plus value pour la conception, la production ou l'exploitation des avions. La section suivante décrit les multiples sortes d'informations disponibles autour d'un avion en service.

4 Les types de données

A tout système complexe nécessitant une surveillance spécifique de fonctionnement et une maintenance, lui sont associées des données de vie et de réparations. Ces données peuvent être le résultat d'une surveillance pointue, dans ce cas elles sont dites « embarquées » ou simplement des transcriptions des événements de sa vie, ce sont des données d'utilisation ou de vie. Les données d'utilisation se rapportent aux différents régimes et phases de vie du système et contiennent généralement les historiques de sa mise en service et des opérations de maintenance appliquées. Les données embarquées sont le résultat d'une surveillance pointue des paramètres de fonctionnement et des composants critiques à travers le déploiement d'un réseau physique de capteurs embarqués.

Spécialement dans l'aéronautique, un avion est associé à un grand volume d'informations acquises en vol ou en atelier se rapportant à son utilisation, mais aussi des données de référence énoncées par les constructeurs et les différentes autorités.

a) *Les données de vie* portent sur :

- L'historique des opérations de vol qui décrit les missions effectuées par le véhicule et les compteurs associés à ses équipements,
- Les informations annexes aux missions de vol qui incluent : les conditions environnementales décrivant des zones atmosphériques traversées par le véhicule, le mode d'utilisation c'est-à-dire le but ou la nature de la mission (qui peut être militaire, de sauvetage, un charter,...) et les conditions d'utilisation se rapportant aux différentes possibilités de « paramétrage » du véhicule (type d'huile dans le moteur,...),
- Les tâches de maintenance appliquées sur le véhicule et ses équipements,
- Les comptes rendus, qui sont des rapports manuscrits, proviennent de différents intervenants : le pilote qui transcrit les différents événements imprévus survenus durant la mission,

l'opérateur qui transcrit les constatations d'usure sur le véhicule relevées durant les escales et l'inspecteur qui consigne ses remarques concernant le véhicule lors de son entrée à l'atelier de maintenance,

- La configuration des appareils, car au fil des différentes opérations de maintenance appliquées la configuration matérielle du véhicule change. La vitesse et le comportement d'usure peuvent varier selon la configuration puisqu'ils dépendent de la manière dont les équipements interagissent entre eux. Cet aspect est un critère d'utilisation à considérer dans la caractérisation des modèles d'usure,
- Le plan de vol, déposé préalablement à la mission : c'est le programme prévu pour les prochaines utilisations du véhicule.

b) *Les données embarquées* sont des relevés de capteurs embarqués sur les équipements des avions. Ces données sont généralement très volumineuses, par exemple, dans le cadre du projet RECORDS [1] elles sont de l'ordre de 4Go par vol et par avion et concernent 28 paramètres. Les données relevées varient selon le type des capteurs déployés :

- Les capteurs de surveillance qui relèvent le signal d'un paramètre et déclenchent une alerte ou un enregistrement si un seuil, configuré par un expert, est dépassé.
- Les capteurs d'enregistrement qui relèvent et enregistrent les valeurs d'un paramètre avec une fréquence fixe définie par un expert. Généralement, ces capteurs sont munis d'une mémoire embarquée ou sont reliés à un équipement de stockage au sol par le biais d'une liaison radio.
- Les capteurs de diagnostics qui effectuent des tests de fonctionnement sur les équipements et sont reliés à un calculateur central. Dans certains cas, ces capteurs peuvent communiquer entre eux.

c) Finalement *les données de référence* représentent l'ensemble des manuels de configurations, d'utilisation, et de réparation des véhicules. Ils sont énoncés par les constructeurs ou les autorités certifiées et regroupent les documents suivants :

- Pilot Instruction Manual (PIM) : c'est le manuel d'utilisation de l'appareil dédié aux pilotes. Il décrit deux aspects de l'utilisation du véhicule :
  - Les consignes générales à respecter lors de l'utilisation du véhicule dans les différentes conditions (milieu maritime, vitesse du vent, ...), durant les différentes phases (décollage, roulage, mise sous tension ....) et les limites d'emploi de chacun des composants le

1. Projet conduits de 2008 à 2010 par différents intervenants français pour développer une infrastructure de service sécurisée afin d'assurer le suivi et l'analyse des conditions d'utilisation d'aéronefs légers ou de véhicules terrestres complexes www.2moro.com/web/Solutions/57-records.php.

constituant,

- Les procédures d'urgence (atterrissage d'urgence, moteur au sol en feu, moteur en vol en feu, ...).

- Aircraft Maintenance Manual (AMM) : Le manuel de maintenance constructeur indique les tâches de maintenance à effectuer pour chaque type d'équipement identifié par un Part Number (P/N). Chaque entrée de l'AMM associe une tâche de maintenance à un P/N et à une fréquence de répétition. Dans ce manuel, les tâches sont organisées selon deux structures : la structure fonctionnelle et la nomenclature.

Les données présentées ci-dessus peuvent être exploitées à différents niveaux pour surveiller les performances des équipements ou leur dégradation.
Ainsi, plus que dans les autres domaines tels que le marketing ou la médecine, les données se rapportant à la vie d'un système en aéronautique sont particulièrement hétérogènes et très volumineuses. Il est pratiquement impossible de part leur volume et l'importance de leur flux, à un expert ou à une équipe d'experts de pouvoir les analyser et les exploiter dans un délai raisonnable.

Il y a donc lieu d'essayer d'automatiser la compilation, l'organisation et la rationalisation de cette masse d'information afin d'en tirer profit pour assurer un meilleur service, réduire les coût et garantir la sécurité des transport aériens. Pour tenter d'atteindre ces objectifs, plusieurs techniques de fouilles de données et de système expert sont utilisées à différents niveaux pour le diagnostic, l'aide à la maintenance, la gestion de flotte ou le design d'équipements. La section suivante présente quelques travaux qui visent à améliorer la maintenance en réduisant le temps de diagnostic et/ou en faisant de la prévention de pannes. Certains de ces travaux ont été consignés dans des brevets et d'autres dans le cadre de recherche industrielle.

5 Extraction de connaissances

Dans le domaine de l'aéronautique, comme dans tous les domaines industriels, les constructeurs et les exploitants de flottes cherchent à améliorer la qualité de leurs produits et services, réduire les coûts et augmenter la sécurité des transports. Ces améliorations passent généralement par des interventions au niveau de la maintenance, puisqu'une meilleure gestion de la maintenance permet de réduire le temps d'interruption des véhicules et donc d'augmenter la disponibilité et d'améliorer ainsi la qualité du service. Aussi, la prévision et le diagnostic de l'état d'endommagement des équipements permettent d'anticiper et de prévenir les interruptions imprévues, et aussi de garantir plus de sécurité.

Prenant en compte le grand volume d'informations disponibles concernant l'utilisation et l'exploitation des véhicules, il y a lieu de les exploiter par l'application d'outil statistique et d'analyse de tendance afin d'étudier le comportement des équipements, d'en améliorer la gestion et d'en optimiser la maintenance, le mode d'utilisation et même le design.

Cette section présente quelques travaux industriels qui tentent de perfectionner la maintenance et le design. Les travaux présentés portent essentiellement sur la réduction du temps et du coup de la maintenance en ligne en utilisant différentes techniques d'analyses statistiques et de fouilles de données sur des données embarquées, des données de vie ou des données de références ou la combinaison de ces trois types d'informations. Les travaux présentés sont issus de brevets[2] industriels ou de publications scientifiques.

Thalès, AirbusS et Honeywell déploient un réseau physique de capteurs de diagnostic embarqués. Les capteurs sont greffés sur un ensemble d'équipements faisant partie de la MEL (en anglais Minimum Equipment List), il s'agit de la liste minimale des équipements nécessaires à vérifier pour l'autorisation de décollage des avions. Les capteurs surveillent les paramètres et appliquent régulièrement des tests de diagnostic de bon fonctionnement. Ce test permet d'informer sur l'état de dégradation de l'équipement, il est répandu dans l'industrie et est appelé BITE pour le terme anglais Built In Test Equipment. Les capteurs de diagnostics embarqués communiquent entre eux et leurs résultats sont transmis à un calculateur central qui gère l'ensemble des messages. Au sein du calculateur le traitement diffère selon les besoins.

Les travaux publiés par Thalès combinent les messages reçus avec la configuration du véhicule et les relations fonctionnelles entre les équipements pour estimer l'état général du système. Ils identifient l'équipement défaillant et prévoient la tâche de maintenance à appliquer. Le système permet une automatisation du diagnostic lors de la maintenance en ligne et réduit le temps inter vol.

Dans un premier temps, les travaux d'Airbus transforment les messages reçus sous forme temporelle linéaire. Ils combinent les chaînes de Markov à un score de vraisemblances pour former un modèle de filtrage des diagnostics de base. Dans un deuxième temps, ils connectent à leur système une base de données hétérogène qui intègre des données de maintenance, des données de vie mais aussi une forme électronique des manuels de maintenance. Ainsi le modèle décrit précédemment devient le premier maillon d'un enchaînement de modules qui permettent d'améliorer le diagnostic et de faire un pronostic des opérations de maintenance. Il est suivi par : un module d'analyse fonctionnelle qui calcule la capacité de fonctionnement d'un équipement, lui-même suivi par un

2. les brevet sont disponible sur l'emplacement web : http://fr.espacenet.com

module de pronostic précurseur de panne, finalement un module de contextualisation ajuste le pronostic émis précédemment à des conditions de vol spécifiques.
Les travaux de Honeywell, quant à eux, utilisent des outils de régressions et d'analyses statistiques afin de corréler symptômes et codes d'alertes. La corrélation se base sur la correspondance entre les messages des diagnostics BITE et les rapports de maintenance. Une classification des symptômes permet une estimation plus précise des alertes à venir et donc des dommages subis. Dans une application sur des avions militaires le système a été étendu aux circuits électriques et mécaniques.

Dans le cadre de l'estimation des dommages et du diagnostic à partir de simples capteurs de surveillances, Boeing exploite des enregistrements de capteurs munis de mémoire. Les données enregistrées sont stockées dans une base de données qui contient la configuration des dépendances fonctionnelles des équipements surveillés et les rapports de maintenance les concernant. Une hiérarchie sous forme de réseaux Bayéziens permet de calculer pour chaque équipement un seuil de nombre de messages d'alerte à partir duquel l'équipement est considéré en état critique. Également, Eurocopter embarque des capteurs de surveillance de paramètres et enregistre des données de pannes des équipements afin de fixer des seuils de déclenchement d'alertes à partir de la surveillance de certains paramètres. Les alertes sont enrichies par des estimations de dommages des parties concernées. Cette estimation est calculée sur la base de comparaison et d'interpolation de données historiques.
D'autres part, les données historiques de maintenance telles que, par exemple, les rapports d'inspection et de maintenances sont mis à profit à l'aide d'outils statistiques afin d'associer symptômes et actions correctives (IBM). Ces derniers exploitent l'aspect temporel pour affecter des priorités et classer les opérations correctives par probabilité décroissante. Ou encore de construire un case base Model a partir de la combinaison de données de vie issues de la maintenance et de données de référence.
Dans [Fau07], un historique de rapports d'interruptions est exploité afin d'extraire des connaissances utiles à partir de règles d'association. Les rapports d'interruptions décrivent les interventions de maintenance en ligne qui provoquent un délai de retard sur l'exploitation des véhicules. Les règles sont extraites à partir d'un réseau Bayésien représentant les connaissances du domaine (dépendances physique et fonctionnelle entre les équipements du véhicule...). L'approche utilisée dans ces travaux nécessite plusieurs interventions d'un expert humain, ce dernier apporte ses connaissances lors de la construction du réseau mais aussi dans le filtrage des règles extraites et leur annotation.

Dans le cadre d'un projet IDS, le NRC CNRC a conduit des travaux de recherche sur le pronostic de pannes de véhicules complexes et spécialement sur le démarreur APU (pour le terme anglais Auxiliary Power Unit) d'avions de types AIRBUS 320. Les travaux exploitent des données de maintenance (rapport de maintenance et journal de bord) mais aussi des données issues de capteurs embarqués pour surveiller les paramètres critiques des équipements (gaz, pression...) [WOHD97]. L'approche consiste à mettre en place deux stratégies de pronostic. La première est locale à l'équipement et/ou au paramètre cible, elle met en place un modèle de pronostic qui analyse à l'aide d'outils de fouille de données les tendances de comportements avant et après l'occurrence d'une panne afin de prévoir les futures défaillances. La deuxième stratégie est globale et consiste à fusionner les pronostics des modèles locaux en utilisant une méthode de vote.
Les modèles locaux effectuent un pré-traitement des données, supervisé par un expert [LYD+05], qui consiste à sélectionner les données pertinentes, les classifier, [SFS97] et les labelliser selon des critères d'éloignement temporel par rapport aux occurrences de pannes [PLF99]. Le modèle de pronostic local extrait des règles et utilise les arbres de décision [YL07, YLZS10], les réseaux bayésiens [FS99, YL07, YLZS10] ou le Rought set algorithmes [PLF99] selon la nature des données en entrée.
La méthode est appliquée sur deux cas d'études [LYD+05], le premier concerne la maintenance des rails de train dans le cadre du projet WildMinner (wheel Impact Laod Detector) [3] et le second concerne la maintenance du démarrage d'un APU dans le cadre du projet ADAM [4]. Elle a aussi été appliquée dans [YLZS10] pour l'adaptation des limites des paramètres de fonctionnement décrit dans FMEA pour Fault Mode and Effect Analysis. C'est un document de référencement émis par les constructeurs (OEM).

Les performances de la méthode, calculées en combinant un score de performance et un score de pronostic ont été plus satisfaisantes pour le projet WilD Minner que pour le projet ADAM à cause de la complexité et de la corruption des données aéronautiques. Le procédé de validation a été amélioré dans [YL07] en rajoutant une évaluation des impacts du pronostic sur les coûts et les bénéfices réalisés. Aussi, dans [YLZS10] l'approche a été étendue pour mettre à jour des documents de référence émis par les constructeurs afin qu'ils soient plus adaptés aux vieillissements des équipements et à leurs conditions et fréquences d'utilisation.

3. lien du projet : http://apnatech.com/railways/wheel-impact-load-detector
4. lien du projet : http://www.nrc-cnrc.gc.ca/eng/ibp/iit/past-projects/aerospace-dataminer.html

6 Conclusion

La maintenance aéronautique répond à des procédures complexes rigides et bien règlementées. Cependant, ces procédures présentent des défaillances dues à la complexité de la configuration des avions et à la nécessité permanente d'expertise pointue.

Profitant de la grande masse d'informations disponibles autour de la vie d'un véhicule aérien en particulier et des véhicules complexes en général, des stratégies d'amélioration de la gestion de la maintenance ont été mises en places. Elles se basent sur des techniques de fouille de données pour extraire de la connaissance utile à partir des différentes sources de données disponibles.

Une grande partie des travaux présentés, dans la dernière section du chapitre, utilisent les données embarquées. Ils permettent pour la plupart la mise en place de seuils de surveillance d'équipements indépendants et le déclenchement automatique d'alerte. Ces alertes concernent une partie particulière des équipements de l'avion (MEL [Fau07], APU [SFS97, PLF99]). Les données de surveillance sont aussi corrélées à un historique de rapports de maintenance (qui font partie des données de vie des véhicules) pour estimer l'endommagement d'une partie du système. Dans ce cas, la mise en place des méthodes utilisées (réseaux bayésien, arbre de décision ...) nécessite l'intervention d'un expert pour configurer les dépendances fonctionnelles ou de configuration entre les paramètres ou les équipements.

L'objectif des travaux présentés dans la deuxième partie de ce mémoire, est de proposer une solution qui tente d'aider à améliorer la gestion de la totalité ou au moins de la majorité des opérations de maintenance applicables à une flotte en utilisant l'ensemble des données utiles disponibles. La stratégie générale de l'approche a été présentée dans [BZ09] et dans [BZMM+09]. Elle devra s'intégrer à un ensemble de services de gestion d'équipements, de flottes et d'ordonnancement de la maintenance qui se doit d'être indépendant des interventions des experts humains, mais qui apportera une aide aux divers intervenants.

Chapitre 2

Les motifs séquentiels

Sommaire

1 Introduction

Ce chapitre présente un état de l'art des approches et techniques d'extraction de séquences fréquentes (*ESF*). A partir de bases de données séquentielles décrivant des comportements successifs ou simultanés de spécimens du monde réel, ces techniques permettent d'extraire les comportements répandus qui ne sont pas visibles par des analystes humains.

La première section présente les définitions générales utilisées dans l'ESF et expose la problématique d'extraction de séquences fréquentes.

La seconde section présente les méthodes d'extraction rencontrées dans la littérature où se distinguent deux principales approches : l'approche d'extraction par niveau et l'approche d'extraction en profondeur.

La troisième section expose les différentes stratégies de gestion du paramètre temporel dans les séquences. Nous distinguons deux types de données séquentielles temporelles : celles estampillées discrètement et celles estampillées par intervalles.

Enfin, la quatrième section présente les techniques d'optimisation de l'extraction par restriction du résultat aux séquences fréquentes optimales. Nous concluons dans la cinquième section.

2 Définitions et problématique

Cette section présente les définitions et les principes généraux dans l'extraction de motifs séquentiels. Soit $\omega = \{e_1, e_2, \ldots, e_p\}$ un ensemble d'évènements. Chaque évènement représente soit une action, une caractéristique, un paramètre, un attribut ... du monde réel. L'ensemble des évènements décrit un domaine précis.

Une transaction est un ensemble d'évènements simultanés où chacun apparaît une seule fois. notée $I = \{e_1, e_2, \ldots, e_p\}$. Intuitivement une transaction représente un ensemble d'actions, de caractéristiques ou d'attributs simultanés associés à un spécimen. Ce dernier est une entité à laquelle sont rattachées des données séquentielles, elle peut être un client s'il s'agit d'une description du domaine du marketing ou un malade si c'est une description du domaine médical.

Exemple 1. *Les tableaux 2.1 présentent deux bases de données. Le tableau 2.1a décrit la base Marketing, elle représente des évènements d'achats effectués par des clients. Le tableau 2.1b décrit la base de données Médecine qui représente des symptômes observés sur des patients.*

Chaque ligne des deux tableaux associe une liste d'évènements à un spécimen ($client_i$ ou $patient_i$). Les évènements d'une ligne sont associés à la même temporalité telle que chaque ligne

Date	Spécimen	évènement
1	$Client_1$	$article_1$, $article_2$
1	$Client_4$	$article_2$
2	$Client_1$	$article_5$, $article_2$
3	$Client_3$	$article_1$
4	$Client_3$	$article_2$
5	$Client_4$	$article_3$
5	$Client_2$	$article_1$, $article_4$
6	$Client_1$	$article_3$, $article_4$
8	$Client_2$	$article_2$, $article_3$

(a) Base de données : Marketing

1	$malade_1$	$sympthome_1$, $sympthome_2$
3	$malade_2$	$sympthome_5$
5	$malade_1$	$sympthome_3$,$sympthome_4$
6	$malade_2$	$sympthome_1$
7	$malade_2$	$sympthome_1$,$sympthome_2$, $sympthome_5$

(b) Base de données : Médecine

Tableau 2.1 – Exemples de bases de données

est une transaction. La première ligne du tableaux 2.1a est la transaction associée au client$_1$ qui a acheté les produits article$_1$ et article$_2$, en même temps, à la date 1. Aussi la première ligne du tableau 2.1b représente le fait que le malade$_1$ présente simultanément les deux symptômes sympthome$_1$ et sympthome$_2$ au jour 1.

Une séquence est une représentation de l'évolution du comportement d'un spécimen. C'est une suite de transactions chronologiquement ordonnées et se rapportant à un même sujet (spécimen). Elle est notée $S = \langle I_1, I_2, \ldots, I_n \rangle$ où I_1 apparait avant $I_2 \ldots I_{n-1}$ apparaît avant I_n. Une séquence qui contient n transactions est dite une n-séquence.

Une séquence temporelle est une séquence dont les transactions sont estampillées temporellement c'est à dire que chaque transaction est associée à un paramètre temporel qui indique l'instant où ses évènements se sont produits. Une séquence temporelle est notée : $S = \langle (t_1, I_1), (t_2, I_2), \ldots, (t_n, I_n) \rangle$ où t_i est la temporalité (l'estampille temporelle) de la transaction I_i et telle que $1 \leq i \lneq n$, $t_i \lneq t_{i+1}$.

Dans une séquence **temporelle**, les temporalités des transactions sont relatives au moment d'apparition de la première transaction de la même séquence, telle que la valeur temporelle t_k associée à la transaction I_k représente le décalage temporel entre la temporalité absolue de la transaction k et la temporalité absolue de la transaction I_1. On a alors $1 \leq i \leq n$, $t_i = t_i - t_1$

Exemple 2. *Si on reprend l'exemple 1, à partir du tableau 2.1a on peut identifier deux séquences*

associées chacune aux spécimens $client_1$ *et* $client_2$.

Le premier, $client_1$ *a acheté dans un premier temps les articles* 1 *et* 2*, dans un deuxième temps les articles* 5 *et* 2 *et dans un troisième temps les articles* 3 *et* 4*. La séquence correspondante est* $client_1 = \langle (article_1, article_2)\ (article_5, article_2)\ (article_3, article_4)\rangle$.

La deuxième séquence représente le comportement du $client_2 = \langle (article_1, article_4)(article_2, article_3)\rangle$*. Elle relate le fait que : « le* $client_1$ *a acheté d'abord les* $article_1$ *et* $article_2$*, ensuite il a acheté les* $article_2$ *et* $article_5$*, finalement il a acheté les* $artcle_3$ *et* $article_4$ *».*

Si on prend en compte les estampilles temporelles des évènements, le comportement du $client_1$ *sera représenté par la séquence temporelle suivante :* $client_1 = \langle (0, article_1, article_2)\ (1, article_5, article_2)(5, article_3, article_4)\rangle$*. Le premier achat du spécimen* $(article_1, article_2)$ *s'est produit à la date absolue* 1*, c'est la référence temporelle des transactions de la séquence. La première transaction est estampillée avec une valeur égale à* 0 $(1-1)$*, la seconde (correspondant aux deuxième achat) est estampillée avec une valeur temporelle égale à* 1 $(2-1)$*, la troisième transaction a une estampille égale à* 5 $(6-1)$*. Pour résumer, La sequence temporelle* $client_1$ *véhicule l'information suivante :« le* $client_1$ *a acheté les* $article_1$ *et* $article_2$*, une unité temporelle après, il a acheté* $article_5$ *et* $article_2$ *et cinq unités temporelles plus tard, il a acheté* $article_3$ *et* $article_4$ *».*

Une base de séquences est une collection de séquences où chaque élément est identifié par un code unique *id_ sequence*. Dans une base de séquence, une transaction est identifiée par le couple (id_sequence, id_transaction), où *id_ transaction* est son identifiant au sein de la séquence (il correspond à son estampille temporelle si les séquences sont *temporelles*) et id_sequence, l'identifiant de la séquence à laquelle la transaction appartient.

Exemple 3. *Dans le tableau 2.1a de l'exemple 1, à chaque client correspond une séquence. Nous obtenons alors la base de séquences temporelles qui répertorie les comportements d'achats par client décrite dans le tableau 2.2. Par exemple la transaction* $(3, article_2, article_3)$ *dans la base de séquence « Marketing » est identifiée par le couple* $(client_2, 3)$ *où* $client_2$ *est l'identifiant de la séquence à laquelle la transaction appartient et* 3 *représente l'identifiant de la transaction au sens de la séquence (son estampille).*

Nous définissons maintenant les notions de sous séquence et de support. Intuitivement, S est une sous séquence de S' si le comportement décrit par S est aussi décrit par S' et donc l'enchainement des évènements de la première séquence apparait aussi dans la seconde. Dans ce cas on peu dire que S' supporte S ou que S est une sous séquence de (contenue dans) S'.

ID_séquence	Séquence
$client_1$	$\langle(0, article_1, article_2)\ (1, article_5, article_2)(5, article_3, article_4)\rangle$
$client_2$	$\langle(0, article_1, article_4)\ (3, article_2, article_3)\rangle$
$client_3$	$\langle(0, article_1)(1, article_2)\rangle$
$client_4$	$\langle(0, article_2)(4, article_3)\rangle$

Tableau 2.2 – Base de séquences « Marketing » utilisée pour l'exemple 3

Une sous séquence : une séquence $S = \langle I_1, I_2, \ldots, I_n\rangle$ est une sous séquence (contenue dans) d'une autre séquence $S' = \langle I'_1, I'_2, \ldots, I'_p\rangle$ notée $S' \supseteq S$ si et seulement si il existe $1 \leqslant y_1 \leqslant y_2, \ldots, \leqslant y_n \leqslant p$ tels que $I_1 \subseteq I'_{y1}, I_2 \subseteq I'_{y2}, \ldots, I_n \subseteq I'_{yn}$. S' est une super-séquence de S.

Lorsque les séquences sont temporelles, en plus du simple enchainement entre groupe d'évènements simultanés, la notion de sous séquence temporelle implique que les décalages entres les transactions successives dans les deux séquences soient les mêmes.

Une sous séquence temporelle Soient deux séquences temporelles $S = \langle(t_1, I_1)(t_2, I_2)\ldots(t_n, I_m)\rangle$ et $S' = \langle(t'_1, I'_1)(t'_2, I'_2)\ldots(t'_m, I'_m)\rangle$ telles que $m \leq n$. S' est contenu dans (supportée par) S si et seulement si $\exists 1 \leq y_1 \leq y_2 \cdots \leq y_m \leq n$ tels que :

- $I'_1 \subseteq I_{y1}, I'_2 \subseteq I_{y2}, \ldots, I'_n \subseteq I_{yn}$
- $t'_2 - t'_1 = t_{y2} - t_{y1}, \ldots t'_i - t'_1 = t_{yi} - t_{y1} \ldots t'_m - t'_1 = t_{ym} - t_{y1}$

Exemple 4. *Considérons les deux séquences non temporelles issues du tableau 2.1a* $client'_2 = \langle(article_1, article_4)\ (article_2, article_3)\rangle$ *et* $client'_3 = \langle(article_1)(article_2)\rangle$*, la séquence* $client_3$ *est une sous séquence de* $client_2$ *car* $article_1 \in \{article_1, article_4\}$ *et que* $article_2 \in (article_2, article_3)$ *aussi l'ordre d'apparition des transactions dans les deux séquences est le même. De la même manière, la séquence* $client_4 = \langle(article_2)\ (article_3)\rangle$ *est une sous séquence de* $client_1 = \langle(article_1, article_2)(article_5, article_2)\ (article_3, article_4)\rangle$ *puisque* $article_2 \in \{article_5, article_2\}$ *et* $article_3 \in \{article_3, article_4\}$ *et que l'ordre d'apparition des transactions est préservé.*

D'autre part, si on considère les estampilles temporelles, la séquence $client_3 = \langle(0, article_1)\ (1, article_2)\rangle$ *n'est pas incluse dans la séquence temporelle* $client_2 = \langle(0, \{article_1, article_4\})\ (3, \{article_2, article_3\})\rangle$ *car malgré le fait que* $(article_1) \in (article_1, article_4)$ *et que* $article_2 \in (article_2, article_3)$ *le décalage temporel entre les transactions de* $clients_3$ *n'est pas égal au décalage*

temporel entre les transactions correspondantes de $client_2$ $(1-0 \neq 3-0)$*. La séquence temporelle* $client_4$ *est contenue dans la séquence temporelle* $Client_1$ *car* $article_2 \in (article_5, article_2)$ *et* $article_3 \in (article_5, article_2)$ *et que le décalage entre les deux transactions de la première séquence (5-1) est égal à celui des transactions de la deuxième transaction (4-0).*

Nous définissons dans la suite la notion de support et de fréquence. Selon le type de motifs fréquents à extraire (séquences ou transactions) et les expertises du domaine d'application, il existe différentes méthodes de comptage du support. Les auteurs dans [JKK99], énumèrent trois méthodes principales.

- La première consiste à calculer pour un motif toutes ses occurrences dans les objets de la base en comptabilisant une occurrence par taille de fenêtre. La taille de la fenêtre représente une distance qui permet de partitionner les objets de la base [MTV97a].
- La deuxième comptabilise aussi une occurrence par taille de fenêtre mais réduit le nombre de fenêtres sur un objet au minimum (e.g que les fenêtres ne se chevauchent pas).
- La troisième méthode comptabilise une seule apparition par séquence. Elle est utilisée pour l'extraction de séquences fréquentes ([AS96, YCJCWCSY10, FVNN08]).

Nous présentons dans ce qui suit la définition du support conformément à cette dernière méthode de calcul.

Le support Nous distinguons entre le support absolu et le support relatif :
Le support absolu d'une séquence S dans une base de séquences BDS est le nombre de séquences de BDS qui supportent S, il est noté

$$support_{a,BDS}(S) = \mid S'; S \subset S' \text{et} S \in BDS \mid$$

Le support relatif d'une séquence S dans une base de séquences BDS est le ratio de séquences de BDS qui contiennent S par rapport au nombre total de séquences dans la base. C'est un paramètre réel contenu dans l'intervalle $[0, 1]$. En effet, au moins S n'est contenue dans aucune séquence de la base, dans ce cas son support relatif est nul et au plus S est contenue dans toutes les séquences de la base, dans ce cas son support relatif est maximal et égal à 1. Il est noté

$$support_{r,BDS}(S) = \frac{\mid S'; S \subset S' \text{et} S \in BDS \mid}{\mid S; S \in BDS \mid}$$

La fréquence d'une séquence S dans une base de séquences BDS est exprimé en fonction de son support relatif. Une séquence S est **fréquente** dans BDS si $support_{BDS}(S)$ est supérieur ou égal à une valeur de support minimal $minsupp$, noté $support_{BDS}(S) \geq minsupp$.

Exemple 5. *Considérons toujours le même exemple, le support absolu de la séquence* $client_4$ *dans la base de séquences « Marketing » est égal à 2. Elle est supportée par les séquences* $client_4$ *et* $client_1$*. Son support relatif est donc égal à* 50% *(2/4). Si le seuil de fréquence minsupp est égal à* 50%*,* $client_4$ *est alors considérée comme une séquence fréquente.*

Propriété 1. *(Anti-monotonie [AS95]) Soient* S *et* S' *deux séquences et une base de séquences* BDS *telles que* $\{S, S'\} \in BDS$ *et* $S' \subset S$ *alors* $support_{BDS}(S') \geq support_{BDS}(S)$*. Cette propriété permet de dire qu'une séquence non fréquente ne peut être contenue dans une séquence fréquente et que par conséquent une séquence fréquente ne contient pas de sous séquence non fréquente. La fréquence est une contrainte* anti-monotone.

Problématique L'extraction de séquences fréquentes identifie les comportements fréquents dans une population de spécimens. Elle permet de dégager à partir d'une masse importante de données des informations difficilement visibles par des analystes humains et de mettre en avant les motifs répétés dont la fréquence est supérieure à un seuil minimal. La problématique d'extraction de séquences fréquentes consiste à extraire à partir d'une base de séquences tous les motifs qui sont supportés par au minimum *minsupp* séquences de la base tel que :

$$ESF(BDS) = \{S; S \subset S_i \mid S_i \in BDS \text{ et } \mid support_{BDS}(S) \mid \geq minsupp\}$$

La manière la plus naïve pour calculer les séquences fréquentes contenues dans un collection de séquences est d'identifier toutes les combinaisons possibles d'enchaînement d'évènements et d'en calculer le support pour ne garder par la suite que celles qui sont fréquentes relativement à un support minimal. Une telle approche est très coûteuse puisque pour un ensemble $\mid \omega \mid = n$ décrivant les évènements d'une base de séquences on peut créer $\mathcal{O}(n^k)$ séquences de longueurs maximales k.

La section suivante présente les principales techniques utilisées pour l'extraction de séquences fréquentes.

3 Technique d'extraction

La problématique d'extraction de motifs séquentiels fréquents peut être complexe et très coûteuse. Sa complexité dépend du nombre d'évènements qui décrivent la base (agissent sur le nombre de combinaison), mais aussi de la taille de cette dernière et de la longueur moyenne de ses séquences (agissent sur le temps de parcours). Les techniques d'extraction, vues dans la littérature exploitent la propriété 1 d'anti-monotonie qui permet d'optimiser l'extraction en

réduisant le nombre de séquences « potentiellement » fréquentes. Effectivement, cette propriété stipule qu'une séquence fréquente ne peut contenir une sous séquence non fréquente.

Toutes les techniques rencontrées, utilisent cette propriété et appliquent le même principe général : d'abord, l'ensemble des 1-séquences fréquentes est identifié (évènements fréquents, c'est l'ensemble le moins coûteux à extraire). Par la suite, un procédé récursif extrait les séquences fréquentes de longueur k à partir de séquences fréquentes de longueur $k-1$ et élimine à priori les autres (celles qui contiennent des sous séquences non fréquentes).

Toutefois, les techniques d'exploration de séquences fréquentes (de longueur k à partir des $k-1$-fréquents) se distinguent en deux approches : L'approche d'extraction par niveau et l'approche d'extraction en profondeur. La première consiste à extraire l'ensemble de toutes les séquences fréquentes de longueur k à partir de toutes les séquences fréquentes de longueur $k-1$. La seconde approche extrait à partir d'une 1-séquence fréquente l'ensemble de toutes les séquences fréquentes qui peuvent l'étendre. Cette section présente successivement un état de l'art succin de chacune des deux approches et effectue une comparaison des deux méthodes.

3.1 Approches d'extraction par niveau

Cette approche extrait les motifs en appliquant un procédé récursif. À chaque itération k, l'ensemble de tous les motifs de longueur k (k-motifs) sont extraits à partir de l'ensemble de tous les motifs de longueur $k-1$ ($(k-1)$-séquences fréquentes). L'extraction applique deux phase : une phase de génération des candidats et une phase d'élagage des candidats non fréquents.

GSP (Generalized Sequential Patterns) est l'algorithme pionnier de cette approche présenté dans [AS96], c'est une extension de l'algorithme d'extraction de transactions fréquentes (*Apriori*) présenté dans [AS94]. Il fournit toutes les séquences fréquentes à partir d'une base de séquence BDS en respectant un support minimal *minsupp*.

L'algorithme GSP parcourt plusieurs fois les données en suivant un procédé itératif horizontal. L'extraction débute par un premier parcours de la base afin d'identifier l'ensemble des évènements fréquents (les 1-séquences fréquentes). Par la suite, à chaque itération ($k \geq 1$) deux étapes sont appliquées : (1) La génération des candidats construit l'ensemble C_k des séquences candidates à partir des L_{k-1} et (2) la phase d'élagage effectue une première sélection des candidats en éliminant celle qui contiennent des sous séquences non fréquente et comptabilise par la suite le support des candidats restant et n'en garde que les fréquents. Cette dernière phase fournit l'ensemble des fréquents L_k.

– La génération de candidats C_k se fait par auto-jointure des fréquents L_{k-1} (fournis par

l'itération précédente). Elle identifie à partir de L^2_{k-1} tous les couples de séquences (s, s') telles que s et s' soit équivalentes en enlevant à la première son premier évènement et à la seconde son dernier évènement. À partir de chaque couple une k-séquence est construite en ajoutant le dernier élément de s' à s. La totalité des « nouvelles » séquences représente C_k (les k-séquences candidates).

– La phase d'élagage, élimine à partir de C_k les séquences non fréquentes. Une première phase de sélection élimine les candidats qui contiennent des sous séquences non fréquentes éliminé à l'itération précédente. Par la suite, le calcul de support des séquences restantes est calculé, pour cela, la base est parcourue une foi. Les auteurs de [AS96] proposent deux techniques pour optimiser ce parcours : la réduction du nombre de candidats, en utilisant une table de hachage et la transformation de la représentation de la base de séquence. CEs deux technique seront détaillées plus loin. Une fois l'ensemble des fréquents L_k identifié, une nouvelle itération est lancée.

L'appel récursif de ces deux phases est stoppé lorsque l'une des deux conditions suivantes est vérifiée : (1) il n'y a plus de fréquents (phase 2) ou aucun candidat n'est généré (phase 1).

Base de Séquences	
$client_1$	$\langle(0, article_1, article_2)\ (1, article_5, article_2)(5, article_3, article_4)\rangle$
$client_2$	$\langle(0, article_1, article_4)\ (3, article_2, article_3)\rangle$
$client_3$	$\langle(0, article_1)(1, article_2)\rangle$
$client_4$	$\langle(0, article_2)(4, article_3)\rangle$

Tableau 2.3 – Base de séquences « Marketing » utilisée dans l'exemple 6

Exemple 6. *Considérons la base de séquences du tableau 2.3 et un support absolu minimal égal à 3 (support relatif égal à 75%), l'algorithme* GSP *extrait d'abord les évènements fréquents* $L_1 = \{article_1, article_2, article_3, article_4\}$. *Ensuite, une première itération est lancée :*

La jointure de de L_1 *avec* L_1 *fournit l'ensemble des candidats de longueur* 2 *présentés dans la première colonne du tableau 2.4a. L'ensemble* L_2 *est obtenu par calcul du support des candidats de* C_2 *car aucun des candidats ne contient de sous séquence non fréquente à cette étape.* L_2 *contient deux séquences fréquentes (dernière colonne du même tableau).*

Par la suite, la deuxième itération est lancée : La génération de candidats (tableau 2.4b) fournit la séquence $\langle(article_1)(article_2)(article_3)\rangle$ *par concaténation du dernier évènement du motif* $\langle(article_2)(article_3)\rangle$ *avec la fin de* $\langle(article_1)(article_2)\rangle$. *Cette séquence contient la sous séquence* $\langle(article_1)(article_2)\rangle$ *qui n'est pas fréquente, elle est alors éliminée sans en compter*

le support. La deuxième séquence candidate est une fusion de la transaction $(article_3)$ *avec la dernière de la séquence* $\langle(article_1)(article_2)\rangle$. $\langle(article_1)(article_2, article_3)\rangle$ *est éliminée sans compter son support car sa sous-séquence* $(article_2, article_3)$ *est non fréquente. La phase d'élagage ne comptabilise aucun fréquent donc la troisième itération n'est pas lancée et l'extraction terminée.*

C_2	support	L_2
$\langle(article_1, article_2)\rangle$	1	
$\langle(article_1)(article_2)\rangle$	3	$\langle(article_1)(article_2)\rangle$
$\langle(article_1, article_3)\rangle$	0	
$\langle(article_1)(article_3)\rangle$	2	
$\langle(article_2, article_3)\rangle$	1	
$\langle(article_2)(article_3)\rangle$	3	$\langle(article_2)(article_3)\rangle$

(a) Itération 1

C_3	support	L_3
$\langle(article_1)(article_2)(article_3)\rangle$	-	$\emptyset$
$\langle(article_1)(article_2, article_3)\rangle$	-	

(b) Itération 2

Tableau 2.4 – Exécution de l'algorithme GSP sur la base de séquence « Marketing ».

Optimisation du parcours de la base Les auteurs dans [AS96] proposent deux techniques différentes pour optimiser le calcul de support des candidats :

- La première évite le parcours multiple de la base de séquences et organise les candidats en une arborescence de tables de hachage. La base est parcourue une seule fois par itération. Pour chaque séquence l'arborescence est explorée (en appliquant une fonction de hachage) afin d'identifier tous les candidats contenus dans la séquence et d'incrémenter leur support. A la fin du parcours tous les candidats sont vérifiés dans toutes les séquences de la base. Cette technique réduit le temps de calcul puisque la base est parcourue une seule foi par itération. Cependant la taille de l'arborescence de hachage (gardée en mémoire) est relativement importante puisque chaque nœud peut contenir plusieurs candidats.
- La seconde technique consiste à transformer la représentation de la base de séquences afin d'éviter de parcourir toutes les transactions d'une séquence pour retrouver un évènement.

Pour chaque séquence, chaque évènement est associé aux différentes temporalités des transactions ou il apparait. Cette technique n'évite pas un parcours multiple de la base mais permet d'alléger la recherche d'évènement en les indexant.

PSP (pour *Prefix Tree for Sequential Patterns*) a été introduit dans [MCP98] il applique le même principe d'extraction que l'algorithme *GSP*. Toutefois, il en améliore les performances et met en place une structure hiérarchique différente pour représenter les candidats et permettre de prendre en compte le changement de temporalité entre les évènements. La structure utilisée est une arborescence, où chaque nœud représente un évènement et chaque arc est un label de la relation entre deux évènements d'une séquence (s'il appartiennent à une même transaction ou pas). Une branche complète de l'arborescence (de la racine à la feuille) représente une séquence candidate dont le support est répertorié au niveau de la feuille. Ainsi plusieurs séquences peuvent partager la partie d'une branche : C'est leur « préfixe » commun. L'avantage de cette structure par rapport à celle de *GSP* est le gain considérable en espace mémoire puisque les sous séquences communes à plusieurs candidats ne sont représentées qu'une seule foi.

SPADE Présenté dans [Zak01], L'algorithme *SPADE* (pour le terme anglais Sequential PAttern Discovery using Equivalent classes), utilise le même principe général d'extraction *Apriori* que celui utilisé pour *GSP* et *PSP*. Son originalité réside dans la transformation de la base de séquence initialement représentée de manière horizontale en une représentation verticale. Un premier parcours permet d'associer à chaque item une liste d'identifiants (id_séquence, id_transaction) qui répertorie toutes ses apparitions dans la base et donc de calculer son support. Les évènements non fréquents ne sont associés à aucune liste. Par la suite, la génération de candidats et l'élagage de ceux qui ne sont pas fréquents sont faits non pas en parcourant la base de séquences mais en utilisant sa représentation verticale (croisement des listes d'apparitions des sous séquences fusionnées pour calculer le support d'un candidat). Par rapport aux algorithmes présentés plus haut *SPADE* réalise un gain important du temps de calcul et de l'utilisation de l'espace mémoire puisque les séquences candidates sont représentées sous forme de treillis.

3.2 Approches d'extraction en profondeur : « *FP- growth* »

La deuxième approche applique une exploration en profondeur des séquences fréquentes, elle est aussi appelée *FP- growth* pour le terme anglais Frequent Pattern Growth. L'extraction est récursive telle qu'a chaque itération k toutes les k-séquences fréquentes sont identifiées à partir d'*une même* $(k-1)$*-séquence fréquente* en ajoutant un seul évènement fréquent à chaque foi.

Définitions et notions utilisées

Avant de présenter les détails de l'approche, nous définissons la notion de préfixe et de suffixe utilisées par les algorithmes d'extraction *en profondeur*.

Intuitivement, un préfixe d'une séquence S par rapport à une séquence S' est la sous séquence de S qui commence à son début et se termine à l'apparition de la dernière transaction de S'. Le suffixe correspondant est la sous séquence, qui concaténée à la fin du préfixe forme S.

Définition 1 (Préfixe, Suffixe). *Soit une séquence $S = \langle I_1 I_2 \dots I_n \rangle$ et $S' = \langle I'_1 \dots I'_m \rangle$ une de ses sous-séquences . Pour $1 \leq p \leq n$ tel que I_p contient I'_m marquant la fin de l'apparition de S' dans S.*

- *$\langle I_1 I_2 \dots I_p \rangle$ est un suffixe de S par rapport à S'.*
- *$\langle I_m \setminus I'_m \dots I_n \}$ est le suffixe correspondant de S par par rapport à S'*
  Notons par $prefixe(S, S')$ et $suffixe(S, S')$ respectivement l'ensemble des préfixes et des suffixes de S par rapport à S'.

Intuitivement le suffixe représente l'ensemble des évènements candidats qui peuvent être concaténés à S' pour former une super-séquence (plus longue) de S'. La projection d'une séquence par rapport à une de ses sous séquence.

Définition 2. *Soit une base de séquence BDS et un motif S', la projection de BDS par S' est définie par $BDS_{|S'} = \{suffixe(S, S') | S' \sqsubseteq S \in BDS\}$*

Remarque Lorsque les séquences sont temporelles, en plus des égalités des transactions, les équivalences des estampilles doivent être considérées.

Exemple 7. *La séquence $\langle (0, article_1, article_2)\ (1, article_5, article_2) \rangle$ est le préfixe de $client_1 = \langle (0, article_1, article_2)\ (1, article_5, article_2)(5, article_3, article_4) \rangle$ et $\langle (4, article_3, article_4) \rangle$ est le suffixe correspondant. Considérons les séquences $client_2$ et $client_3$ du tableau 2.3 (rappelons que $client_3 \subseteq client_2$). L'ensemble des préfixes de $client_2$ par rapport à $client_3$ est $Prefixe(client_2, client_3) = \langle (article_1, article_4)\ (article_2) \rangle$. L'ensemble des suffixes de $client_2$ par rapport à $client_3$ est : $suffixe(client_2, client_3) = \langle (article_3) \rangle$.*

Nous présentons dans ce qui suit l'algorithme *PrefixSpan* un des principaux algorithmes qui appliquent l'approche de l'extraction en profondeur.

Algorithmes

PrefixSpan Introduit dans [PHMA+01] est l'algorithme pionner de l'approche d'extraction en profondeur, il exploite la propriété 1 et les notions de *préfixe* et de *suffixe* pour définir l'extraction

de touts les motifs séquentiels fréquentes à partir d'une base de séquences en appliquant une exploration verticale.

Le procédé est récursif tel qu'une 1-séquence fréquente contenant un seul évènement permet d'extraire toutes les séquences fréquentes qui l'étendent. Plus généralement, à chaque itération k, à partir d'une seule $(k-1)$-séquence, toutes les séquences de longueur k qui peuvent l'étendre sont extraites.

Le procédé général d'extraction débute par l'identification des évènements fréquents, chacun génère une itération telle qu'une itération k applique deux étapes :

- Une projection de la base de séquences sur la 1-séquences de L_1 qui a généré l'itération est calculée. Le résultat représente une base de séquence « réduite » qui contient les éventuelles continuations du $k-1$-préfixe dans chaque séquence.
- La seconde étape identifie l'ensemble des 1-Séquences fréquentes à partir de la base résumée. Si les séquences de la base sont temporelles à tout évènement fréquent est associée une estampille représentant les moments de ses occurrences dans la base. Chaque 1-STI identifiée est concaténée à la séquence extraite à l'itération précédente $(k-1)$ pour construire une nouvelle k-séquence fréquente. Dès lors, une nouvelle itération est exécutée.

La récursivité est stoppée si l'une des deux conditions suivantes est satisfaite : (1) Lorsque la projection est vide (étape 1) ou (2) lorsqu'aucune 1-séquence fréquente n'est identifiée (étape 2).

ID_séquence	Séquence
$client_1$	$\langle(0, article_1, article_2)\ (1, article_5, article_2)(5, article_3, article_4)\rangle$
$client_2$	$\langle(0, article_1, article_4)\ (3, article_2, article_3)\rangle$
$client_3$	$\langle(0, article_1)(1, article_2)\rangle$
$client_4$	$\langle(0, article_2)(4, article_3)\rangle$

Tableau 2.5 – Base de séquence représentant les achats de clients : La base « Marketing » utilisée dans l'exemple 8

Exemple 8. *Si on considère la base de séquence « Marketing » reproduite dans le tableau 2.5 et un support absolu minimal égal à 3 séquences (support relatif égal à 75%)* PrefixSpan *commence par extraire l'ensemble des évènement fréquents* $L_1 = \{article_1, article_2, article_3\}$.

Par la suite, pour chaque évènement fréquent de l'ensemble, une projection de la base sur cet évènement est calculée. Les tableaux 2.6a (respectivement 2.6b et 2.6c) représentent les itérations générées par les traitements des évènements fréquents $artcile_1$ *(respectivement* $artcile_2$ *et*

Projection	fréquents	projection
$client_1 : \langle(0, article_2)(1, article_2)(5, artticle_3)\rangle$	$\emptyset$	$\emptyset$
$client_2 : \langle(3, article_2, article_3)\rangle$		
$client_2 : \langle(1, article_2)\rangle$		

(a) Itération correspondant au préfixe $(0, article_1)$

Projection	fréquents	projection
$client_1 : \langle(5, article_3)\rangle$	$\emptyset$	$\emptyset$
$client_1 : \langle(4, article_3)\rangle$		
$client_2 : \langle(0, article_3)\rangle$		
$client_4 : \langle(4, article_3)\rangle$		

(b) Itération correspondant au préfixe $(0, article_2)$

Projection	fréquents	projection
$client_1 : \langle(0, article_4)\rangle$	$\emptyset$	$\emptyset$
$client_2 : \emptyset$		
$client_4 : \emptyset$		

(c) Itération correspondant au préfixe $(0, article_3)$

Tableau 2.6 – Déroulement de l'algorithme $PrefixSpan$ sur la base de séquences « Marketing » ($minsupp$= 75%).

$artcile_3$*).*

Considérons le traitement du fréquent $article_2$ *(tableau 2.6b), la projection de la séquence* $client_1$ *sur* $article_2$ *produit deux séquences puisque ce dernier y apparait deux fois, dans la première et la deuxième transaction. Cet évènement apparait aussi une foi dans les séquences* $client_2$*,* $client_3$ *et* $client_4$*.*

Dans la projection de la base « Marketing » (qui contient donc cinq séquences résumées) aucun évènement n'est fréquent. En effet, lorsque les séquences sont temporelles, la seule fréquence d'apparition de l'évènement n'est pas suffisante, il faut en plus qu'il apparaisse à des instant égaux. Ainsi, malgré le fait que $article_3$ *apparaisse fréquemment (4 fois), l'évènement = n'est pas considéré comme fréquent puisque les temporalités de ses apparitions ne sont pas égale.*

De la même manière dans les projections de la base « Marketing » sur $article_3$ *et sur* $article_1$ *aucun évènement n'est fréquent. Alors, à la fin de l'exécution de l'algorithme les séquences fréquentes sont :* $\langle(0, article_1)\rangle$*,* $\langle(0, article_2)\rangle$*,* $\langle(, article_3)\rangle$*.*

Remarque *Si les séquences ne sont pas temporellement estampillées, le paramètre temporel ne sera pris en compte pour l'identification des évènements fréquents. Dans ce cas les séquences fréquentes extraites sont :* $\langle(article_1)\rangle$, $\langle(article_2)\rangle$, $\langle(article_3)\rangle$, $\langle(article_1)(article_2)\rangle$ *et* $\langle(article_2)(article_3)\rangle$.

Pour l'approche d'extraction en profondeur, les projections de bases de séquences au fil des itérations permettent de réduire l'espace de recherche des fréquents et donc le temps de leur calcul.Cependant, les chargements en mémoire des différents espaces de recherches représentent un coût important.

Afin de palier cet inconvénient, les auteurs dans [PHMA+01] proposent deux solutions : La première est la bi-projection et la seconde est la pseudo-projection.

- la bi-projection consiste à remplacer la projection décrite plus haut par une projection sur une 2-séquence fréquente et où les préfixes sont étendus de deux évènements à chaque itération. A chaque itération k des séquences fréquentes de longueur $2(k-1)$ sont construites à partir de séquences de longueur $2(k-2)$. En effet, lorsque le nouvel espace de recherche est calculé et les évènements fréquents trouvés un croisement entre les occurrences des couples d'évènements permet d'identifier les 2-séquences fréquents et donc de regrouper deux « 1-extensions » en une seule. Cette méthode réduit considérablement le nombre d'itération et donc le temps de calcul général de l'algorithme.
- La pseudo-projection réduit le temps de calcul et l'utilisation de l'espace mémoire en évitant de faire une copie physique en mémoire de la base projetée à chaque itération. En effet, lors de la projection, chaque suffixe est indexé par deux paramètres : l'identifiant de la séquence à laquelle il appartient et sa position de départ dans celle ci.

SPAM introduit dans [AFGY02], cet algorithme extraits les séquences fréquentes et applique l'approche « FP-Growth » . Il pallie le problème d'occupation mémoire par les différentes projection et met en place une représentation binaire de la base des séquences et de ses évènements.

Cette représentation permet d'indexer les transactions des séquences et associe à chaque évènement de la base un « BitMap » qui permet d'identifier sa présence ou non dans la transaction. La construction des motifs fréquents se fait en appliquant à chaque niveau de récursivité « FP-Growth » trois étapes :

- La génération de candidats sélectionne pour un motifs fréquents deux types d'extensions : La *S-extension* et la *I-extension*. La première, ajoute à la fin du motif une transaction qui contient un seul évènement. La seconde ajoute à la dernière transaction du motif un évènement.

- Une phase d'élagage basé sur la propriété apriori (propriété 1). Cette étape élimine selon ce principe les extensions candidates qui contiennent des sous séquences précédemment identifiées comme non fréquentes.
- Une phase de comptage de support sélectionne à partir des candidats élagués ceux qui sont effectivement fréquents.

Ces trois étapes sont récursivement appliquées en étendant chaque motif contenant k évènements pour obtenir tous les motifs contenant $k+1$ évènements. L'extension d'une branche s'arrête lorsqu'aucun fréquents n'est identifié. Alors, La branche suivante est explorée.

La représentation binaire initiée par *SPAM* contourne le problème d'occupation mémoire des algorithmes à la « FP-Growth » classique. Il permet une amélioration considérable des performances de l'extraction en profondeur lorsque les séquences de la base sont longues et / ou nombreuses.

3.3 Bilan

Si on reprend les exemples 6 et 8, on peut facilement voir que si on utilise les mêmes paramètres d'exécution (même base de séquences et même valeur de *minsupp*), les deux algorithmes *GSP* et *PrefixSpan* fournissent le même ensemble de séquences fréquentes. Cependant, vu qu'ils utilisent des approches différentes, ils se dissocies par leurs performances : *GSP* applique une extraction par niveaux alors que *PrefixSpan* applique une extraction en profondeur. L'approche par niveau, garde une seule version physique de la base de séquence en mémoire mais a besoin de la parcourir en totalité au moins une fois par itération. De plus, à chaque itération un nombre de candidats (gardés en mémoire) est crée. Le nombre d'itérations exécutées pour une approche par niveaux est égal à la taille maximale des séquences fréquentes. Concernant l'approche en profondeur, les différentes projections permettent de réduire la taille de la base à parcourir selon la « profondeur » de l'itération. Néanmoins, les nombreuses copies physiques en mémoire des fragments de la base de séquences sont coûteuses. Dans le pire des cas, la taille de la base n'est pas réduite au fil des itérations, alors l'approche en profondeur est aussi coûteuse que l'approche par niveau. Pour tous les autres cas, différents travaux et expérimentations montrent que l'approche en profondeur est moins coûteuse que l'approche par niveau. Cette thèse est soutenue dans [PHMA+01, HKP11, Bay98, PHW02, GHZ10, WH04, KPP03].

Cette section a présenté un état de l'art des techniques d'extraction de séquences fréquentes. A partir des algorithmes étudiés, deux approches d'extractions se distinguent : L'approche d'ex-

traction par niveau et l'approche d'extraction en profondeur.

Ces techniques d'extraction de séquences fréquentes identifient des comportements rependus à partir d'un grand volume de données séquentielles.

Les séquences extraites permettent de dresser des profils de comportements des spécimens étudiés. Selon les domaines d'application et les besoins d'exploitation, les séquences extraites sont plus ou moins précises et le paramètre temporel a une importance variable pour la formulation et l'utilisation des motifs. Par exemple, [HY06] extrait des motifs temporels souple à partir de données se rapportant à des relevés de tremblement de terre au japon, alors que [FVNN08] extrait des séquences temporelles strictes à partir un historique de manipulation d'un robot. En effet, le premier cas d'étude une composante temporelle souple n'affecte pas l'utilisation des motifs pour une prévision approximative alors que pour la deuxième étude la manipulation du robot nécessite une grande précision dimensionnelle et temporelle.

La prochaine section présente un état de l'art succint de la gestion du paramètre temporel dans l'extraction des motifs fréquents. Nous y faisons la distinction entre les séquences temporelles à estampilles temporelles discrètes et les séquences temporelles à estampilles par intervalles.

4 Extraction de séquences temporelles

La pertinence des motifs séquentiels extraits par les méthodes citées précédemment concerne aussi bien leur fréquence que leur précision. La fréquence vérifie la contrainte de support et la précision concerne l'intérêt sémantique des séquence à travers leur formulation temporelle. Par exemple, le motif fréquent $\langle(article_1)(article_2)\rangle$, extrait à partir de la base « Marketing » avec un support relatif égal à 75%, permet de conclure une relation d'association (« corrélation ») entre les achats consécutifs de l'$article_1$ et de l'$article_2$.

Considérons que l'$article_1$ représente des « couches pour bébés » et que $l'article_2$ représente de la « bière » en référence au problème du « panier de la ménagère » [AIS93]. La séquence fréquente $(\langle(couches\_pour\_\text{bébés})(\text{bière})\rangle)$ devient ambiguës car elle manque de précisions temporelles. En effet, en l'absence de précision temporelle deux possibilités d'interprétation peuvent être envisagées : (1) les articles ont été achetés à quelques jour d'intervalle, et dans ce cas la corrélation des deux achats est significative, ou alors (2) ils ont été achetés à quelques mois d'intervalle et dans ce cas la corrélation entres les deux acquisitions n'est pas significative. Pour de palier ce manque de précision des motifs fréquents, l'extraction de séquences temporelles fréquentes fournit des résultats plus adaptés à des besoins d'exploitation « précis ».

Les méthodes d'extraction de séquences temporelles rencontrées dans la littérature distinguent deux types d'estampillages temporels : Les estampilles temporelles discrètes et les estampilles temporelles par intervalles.

- L'estampille discrète associe à chaque transaction de la séquence un paramètre discret qui précise l'instant d'occurrence de ses évènements de manière discrète et simultanée.
- L'estampille par intervalle associe à chaque transaction un intervalle temporel pendant lequel les évènements se produisent de manière continue et simultanée.

Cette section présente un état de l'art sur la gestion de l'aspect temporel selon les deux types d'estampillage dans l'extraction des motifs séquentiels fréquents.

4.1 Extraction de séquences temporelles à estampilles discrète

L'extraction de séquences temporelles fréquentes pallie le problème d'ambiguïté des motifs. Elle fournit des séquences temporelles à estampilles discrètes représentant les comportements des spécimens tels que chaque transaction est associée à une variable temporelle. Cette variable donne un sens temporel à la succession des évènements et clarifie leur interprétation. De telles séquences sont gérées et extraites dans différents domaines tels que : Le profilage du comportement de robots [FVFNMN10], dans le domaine de la finance [PRM+09], dans l' analyse du comportement des tremblements de terre au japon [HY06].

En plus de la précision temporelles des séquences, la pertinence de l'information temporelle qu'ils véhiculent est prise en compte pour juger leur utilité.

Considérons $S_1 = \langle (0, couches\_\text{bébés})\ (2, \text{bière}) \rangle$ et $S_2 = \langle (0, couches\_\text{bébés})(300, \text{bière}) \rangle$, dans le même contexte d'étude que l'exemple précédent la séquence S_1 est pertinente et apporte une information utile à l'expert en marketing, alors que la séquence S_2 n'as aucun intérêt par rapport au contexte. En effet, dans S_2 la distance temporelle entre les deux transactions est « trop » importante et rend l'interprétation de la séquence non « intéressante » pour l'expert.

Les contraintes temporelles sont introduites dans le but de restreindre l'extraction séquences fréquentes temporelle à celles qui sont pertinentes [AS94] pour l'utilisateur, ces contraintes en fixent des règles de formulation des séquences à extraire. Dans ce qui suit nous présentons ces contraintes et les techniques utilisées pour les intégrer aux algorithmes d'extractions.

Les contraintes temporelles

La pertinence et la complétude des résultats fournis représentent des facteurs d'extraction pour les techniques d'extraction de séquences fréquentes. Ils permettent d'estimer l'efficacité

d'une technique d'extraction et la pertinence du résultat qu'elle fournit. Ce dernier doit être complet, pertinent et concis.

Dans cette optique, les contraintes temporelles ont été introduites afin de restreindre les motifs temporels extraits aux représentations datées soumises à des règles de formulations énoncées par les utilisateurs et les experts du domaines d'application de l'extraction. Ces contraintes permettent d'écarter les motifs fréquents jugés « inutiles » de part leurs interprétations.

Différents travaux, redéfinissent la problématique d'extraction initiale [AS94] et introduisent des contraintes temporelles [HY06, FVNN08, Fio06, MCP98] afin d'extraire des motifs à formulations temporelles plus ou moins précises paramétrés par les utilisateurs.

Les contraintes temporelles ont été introduites dans [AS96] pour gérer l'espacement entre transactions successives d'une et le regroupement d'évènements proches.

La première contrainte gère la distance temporelle entre deux transactions successives. C'est le *gap*, elle fixe deux seuils de distance :

- Une distance temporelle minimale, *mingap* qui fixe un seuil à partir duquel l'interprétation de la distance entre deux transactions successives est considérée significative (intéressante) pour l'utilisateur. Si la distance est inférieure à ce seuil les transactions sont considérées comme trop proches et leur enchainement n'est pas significatif.
- Une distance temporelle maximale, *maxgap* fixe le seuil maximal de succession entre deux transactions. Jusqu'à ce seuil la distance entre deux transactions successives est considérée significative pour l'utilisateur. Au delà de ce seuil les deux transactions sont trop éloignées et l'interprétation de la distance qui les sépares n'est pas pertinente pour l'utilisateur.

Les distances séparant les transactions successives des séquences intéressantes extraites doivent varier entre les deux seuils définis par *mingap* et *maxgap*.

La seconde contrainte défini une taille de fenêtre *ws* qui regroupe des évènements de transactions différentes. La contrainte stipule que ces évènements sont assez proches pour être considérés comme étant simultanés. Cette contrainte est une adaptation de la notion d'épisode énoncée dans [MTV97b].

Afin de compléter les règles de formulations temporelles énoncées par [AS96], les auteurs dans [PHMA+01] introduisent une troisième contrainte qui gère l'étendue de la séquence. C'est la contrainte *whole_interval* qui fixe deux seuils :

- Une taille minimale définie par un premier seuil : le *min_whole_interval*. Cette valeur fixe une distance minimale à partir de laquelle une séquence est considérée comme étant

une séquence complète. Si la séquence s'étend sur une durée inférieure au seuil, elle n'est pas considérée comme un comportement significatif « complet ».

– Une taille maximale définie par un second seuil : le $max\_whole\_interval$. Cette valeur fixe une distance maximale jusqu'à laquelle une séquence est considérée comme étant cohérente. Si la durée de la séquence est supérieure à ce seuil, son interprétation n'est pas considérée comme étant comportement indépendant.

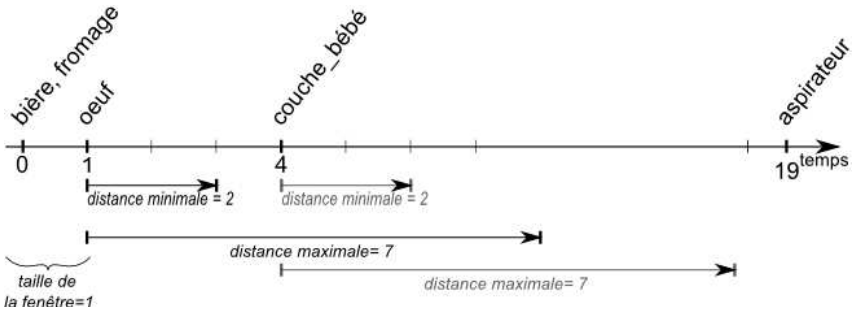


FIGURE 2.1 – Illustration des contraintes temporelles taille de fenêtre et succession

Exemple 9. *Considérons la séquence* $\langle(0, bière, fromage)(1, oeuf)\ (4, couche\_bébé)(9, aspirateur)\rangle$ *et les contraintes* $min\_whole\_interval = 2$ *et* $max\_whole\_interval = 10$, *si deux groupes d'achats sont fait à moins de deux jours d'intervalle, ils sont considérés comme étant trop proches pour être considérés comme étant successifs. Si ils sont effectués à plus de 10 jours d'intervalle ils sont considérés trop éloignes pour être considérés comme directement successifs et l'association de leur succession n'est pas utile. Pour* $ws = 1$ *signifie que si deux achats ou groupes d'achat sont fait à au plus 1 jour d'intervalle on peut les considérer comme simultanés.*

La figure 2.1 illustre les distances gérées par ces contraintes. La première et la deuxième transaction de la séquence peuvent être regroupées puisque la distance temporelle qui les sépare est englobée par la taille de la fenêtre. La quatrième transaction est trop éloignée de la troisième pour être considérée comme une succession « significative ».

Nous présentons maintenant les formulations et les notations qui permettent de concrétiser ces contraintes. L'intégration de ces contraintes à l'extraction de séquences fréquentes sera traitée dans la section. 4.1.

Soit S une séquence temporelle de longueur n telle que $S = \langle(t_1, I_1) \ldots (t_n, I_n)\rangle$. S satisfait les contraintes temporelles : $mingap$, $maxgap$, $min\_whole\_interval$, $max\_whole\_interval$ et la fenêtre glissante ws si et seulement si $\forall\ 1 \leq i \leq n$:

– *Gap* régule les distances temporelles minimale et maximale entre deux transactions successives :

$$mingap \leq (t_{i+1} - t_i) \leq maxgap$$

– *Whole_interval* régule la longueur minimale et maximale d'une séquence :

$$min\_whole\_interval \leq t_n \leq max\_whole\_interval$$

– La *taille de la fenêtre (ws)* permet de regrouper des évènements de transactions successives dans une même transaction en leur associant un intervalle temporel. Elle fixe sa largeur (incertitude) maximale :

$$\text{Si} |t_i - t_{i+1}| \leq ws \text{ alors} I'_i = I_i \cup I_{i+1}$$

Exemple 10. *Soient* $S = \langle (0, A)(1, BC)(4, D) \rangle$ *une séquence temporelle et les contraintes temporelles mingap et maxgap respectivement égales à 2 et 3. S ne satisfait pas mingap puisque* $t_2 - t_1 = 2 - 0 = 1 \lneq 2$. *Par contre, S satisfait maxgap puisque pour toutes ses transactions successives maxgap est satisfaite (*$t_2 - t_1 = 1 - 0 \leq 3$; $t_3 - t_2 = 4 - 1 \leq 3$*). Pour une taille de fenêtre égale à* 2*, la première et la deuxième transaction de S peuvent être fusionnées car* $1 - 0 = 1 \leq 2$. *La séquence S devient* $(ABC)(D)$.

Ces contraintes régulent l'effet du paramètre temporel dans une séquence ; ils gèrent la distance minimale (respectivement maximale) entre deux transactions successives pour que la corrélation entre elles soit significative et que deux transactions trop proches (*mingap*) (respectivement trop éloignées(*maxgap*)) ne soient pas reliées. La taille de la fenêtre gère le regroupement des évènements de transactions adjacentes et la contrainte *whole_interval*, régule la longueur de la séquence afin que la corrélation de toutes les transactions de la séquence soit significative. La plus part des travaux traitant l'extraction de séquences temporelles fréquentes avec des estampilles temporelles discrètes considèrent ces contraintes [AS96, FVNN08, MTV97b, HY06, MPT04, MCP98, Fio06].

Les contraintes peuvent être monotone ou antimonotone tels qu'une contrainte C_m est dite *monotone* si pour une séquence S qui satisfait C_m toute séquence qui contient S satisfait C_m. Une contrainte C_a est dite anti-monotone si pour une séquence S qui satisfait C_a, toute sous séquence de S satisfait C_a.

Les contraintes *maxgap*, *max_whole_interval* et la taille de la fenêtre sont des contraintes leurs vérifications nécessitent un retour en arrière lors du processus d'exploration afin de vérifier qu'elles sont bien satisfaites par un k-motif extrait à partir d'un $(k-1)$-motif(s) qui les satisfait.

Par contre, les contraintes *mingap* et *min_whole_interval* sont des contraintes monotones, si elles sont satisfaites par un $(k-1)$-motif elles sont automatiquement satisfaites par le k-motif(s) qu'il génère.

La partie suivante présente les méthodes utilisées pour intégrer ces contraintes aux méthodes et algorithmes d'extraction

Intégration des contraintes

Dans les travaux rencontrés , la prise en compte des contraintes temporelles peut se faire de deux manière :

- Vérifier que les séquences extraites vérifient les contraintes au cours de l'extraction.
- Sélectionner parmi les séquences initiales celles qui vérifient les contraintes, et appliquer un algorithme d'extraction « classique » par la suite.

Lorsque les contraintes sont intégrées au procédé d'extraction la vérification de leur satisfaction par les séquences extraite dépend de la nature monotone ou anti-monotone de la contraintes. Lorsque la contrainte est anti-monotone , sa vérification nécessite un retour arrière pour vérifier que la séquence étendue satisfait toujours la contrainte. Cette vérification n'est pas nécessaire lorsque la contrainte est monotone.

Pour une extraction par niveau [MCP98, AS96], les contraintes temporelles sont vérifiées au même temps que le comptage des supports des candidats. Les séquences de la base sont alors parcourues en avant et en arrière afin de vérifier la satisfaction des contraintes. Pour une extraction en profondeur [PHMA+01, HY06, FVNN08]les contraintes monotones sont vérifiées à la sélection des évènements fréquents qui étendent les séquences déjà identifiées comme fréquentes. A chaque fois qu'un motif est étendu les contraintes monotones sont vérifiées. Les contraintes anti-monotone sont vérifiée avant d'ajouter le motif à l'ensemble des fréquents.

[MCP98, Fio06] proposent d'organiser les séquences données en graphe appelé GTC (pour le terme anglais graphe Time Constraint). Cette structure représente les séquences qui satisfont les contraintes temporelles (*ws*, *mingap* et *maxgap*). [PRM+09] regroupe de les évènements en appliquant la taille de la fenêtre lors d'une phase de pré-traitement des données sur lesquelles un algorithme d'extraction est par la suite exécuté. Les éléments regroupés sont associés à une estampille temporelle discrète : la plus petite des estampilles des évènements du groupe. Un choix arbitraire qui n'est nullement motivé, si ce n'est pour des considérations de simplification et qui de plus constitue une perte d'information. Ce faisant, le nombre de séquences obtenues est plus grand que le nombre de séquences initiales (une même séquence peut donner lieu à plusieurs re-

groupements différents). Se pose alors le problème de l'interprétation du support d'une séquence. Dans [Tei07], l'auteur intègre aux contraintes temporelles *ws*, *mingap* et *maxgap* la notion de logique flou qui permet une considération flexible de ces contraintes selon des seuils de tolérance spécifiés.

En plus des contraintes temporelles différents autres types de contraintes ont été énoncées selon les besoins d'exploitation des motifs et selon les domaines ou les techniques d'extraction sont utilisées. Par exemple, dans [AS96] et [PRM+09], les auteurs prennent en considération la taxonomie du domaine d'extraction afin de généraliser les séquences extraites. Dans [FVNN08], la notion de séquences multidimensionnelles est considérée pour associer une valeur de contexte aux évènements des motifs. [PHW02] énumèrent différents types de contraintes (agrégation, expressions régulières, ...) et traitent leurs prise en considération dans le procédé d'extraction.

La partie suivante présentera un état de l'art succinct sur l'extraction de séquences fréquente à estampilles temporelles par intervalles.

4.2 Extraction de séquences temporelles à estampille par intervalles

Les techniques d'extraction de séquences temporelles les plus répandues traitent des séquences temporelles avec estampilles ponctuelles. Cependant, les évènements dans le monde réel sont rarement discrets. Typiquement dans le domaine médical, la formulation des symptômes de patients s'apparente plus à une représentation séquentielle par intervalles. Effectivement, des symptômes tel que la toux ou la fièvre apparaissent et persistent pendant un certain temps et leur durée représente un aspect pertinent dans les données à traiter. Afin de traiter des données de ce types, plusieurs travaux ([GQ08, WC07, YCJCWCSY10] et [GNPP06]) ont porté sur l'extraction de séquences temporelles par intervalles.

Une séquence temporelle par intervalles est une suite d'évènements qui se produisent tout au long de d'intervalle estampille.

Tout d'abord, nous présentons la définition des séquences temporelles par intervalles telles qu'elles sont définies dans [GQ08], dans [WC07] et dans [YCJCWCSY10]

Définition 3. *Une séquence temporelle par intervalles S est une suite ordonnée de transaction où chaque transaction est notée $I = (e, [l, u])$ composée d'un évènement e et d'un intervalle non vide $[l, u]$ avec $l < u$ S est définie par : $S = \{(e_i, [l_i, u_i])\}_{i \in \mathbb{N}_p}$ tel que $\forall i, j \in \mathbb{N}0 \leq i < j \leq p$ et $l_i \leq l_j \vee (l_i = l_j \wedge (e_i < e_j \vee (s_i = s_j \wedge u_i < u_j)))$*

Partant de données initialement estampillées par des intervalles, l'extraction de séquences fréquentes est un procédé doublement complexe, de part la difficulté d'identification des intervalles associés aux fréquents extraits et la complexité des relations qui existent entre eux. Nous classons les travaux rencontrés en deux catégories : La première extrait des motifs fréquents avec des estampilles sous forme d'intervalles et la seconde extrait des enchainements représentant des relations entre ces intervalles estampille.

la première catégorie applique un algorithme d'extraction afin d'identifier les séquences fréquentes indépendamment des estampilles et applique par la suite un algorithme de clustering sur les estampilles associées aux occurrences d'un fréquents pour leur attribuer des intervalles temporels. Dans [GNPP06], d'abords toutes les séquences fréquentes (non estampillées) sont extraites et par la suite l'algorithme de clustering *EM* (pour le terme anglais expectation-maximization) [Moo96] est appliqué, il utilise la densité de distribution pour identifier les différents cluster. Les clusters les plus denses sont sélectionnés et le support de leur séquence représentative est recalculé pour vérifier sa fréquence.

Dans [GQ11] l'association entre les évènements fréquents et leur estampille se fait au fil du procédé d'extraction. Les auteurs utilisent un algorithme *FP-Growth* pour extraire les motifs fréquents où à chaque projection les estampilles de chaque évènement fréquent sont représentées et identifiées par l'algorithme de clustering *K-means*. Ce dernier mesure la similarité entre les intervalles et considère le centre du cluster comme intervalle de l'évènement fréquent. Les travaux présentés dans [GQ11] sont une amélioration de [GQ08] qui extrait les séquences fréquentes en utilisent un algorithme d'extraction par niveau, par la suite les estampilles des séquences extraites sont représentées en hypercube afin d'identifier les temporalités associées motif fréquents.

La deuxième catégorie de méthodes exploite la théorie des intervalles d'[All83] pour extraire des relations entres les évènements fréquents séquentiels. La théorie d'Allen énonce treize relations possibles entre intervalles permettant ainsi de représenter toutes les possibilités de chevauchements, d'inclusions et d'intervalles contigus.

[YCJCWCSY10] utilisent ces relations pour identifier des « tranches » de chevauchement fréquents entre les évènements. [WC07] applique un algorithme d'extraction de la méthode *FP-Growth* sur des séquences à estampille temporelle par intervalles et met en place une représentation non ambiguë entre les occurrences des évènements. Le résultat de ces deux travaux consiste en un ensemble de relations *fréquentes* entre évènements et non pas des séquences avec des estampilles associées aux transactions.

On note cependant le travail de [HY06] qui, comme l'approche que nous présentons, considère des séquences à estampilles temporelles discrètes en entrée et extrait des séquences fréquentes par intervalles. Les auteurs utilisent une fonction par paliers qui s'apparente à une fenêtre non glissante. Ainsi, des évènements très proches temporellement peuvent se retrouver dans des groupes différents du fait de l'application des paliers.

La prochaine section aborde la problématique d'extraction de séquences fréquentes optimales. Ce type de séquences est une représentation condensée des fréquents telle que le résultats fourni évite la redondance d'information entre les séquences fréquentes.

5 Les motifs optimaux

L'extraction des motifs fréquents à partir d'une base de séquence est une problématique qui dépend de plusieurs paramètres. Les performances d'extraction et les résultats fournis varient selon la technique d'extraction utilisée, la représentation des données initiales et celle des motifs à extraire mais aussi de la taille et du nombre de séquences de la base. La taille de la base affecte les performances d'extraction de l'algorithme utilisé mais aussi le volume et la forme de l'ensemble des séquences résultat.

En effet, lorsque la taille de la base augmente il y a plus de données à explorer et donc plus d'itération et de vérification a effectuer par l'algorithme. Aussi, lorsque les données sont plus volumineux et les séquences plus longues le résultat fourni l'est aussi. Ce dernier point est posée dans [XHA03] avec l'exemple d'extraction de motifs séquentiels fréquents à partir d'une seule longue séquence $S = \langle (e_1)(e_2)\dots(e_{100})\rangle$ et un support minimal égal à 1. Le résultat contient $2^{100} - 1$ séquences fréquentes parmi lesquelles toutes représentent une information redondante sauf la plus longue.

Afin d'éviter de telles redondances dans les résultats d'extractions différents travaux [XHA03, WH04, LC05] définissent et extraient les séquences closes et les séquences maximales et redéfinissent la problématique d'extraction pour ramener le résultat aux motifs de ce type. Nous présentons les définitions et les travaux vus en littérature concernant l'extraction des séquences closes et maximales

5.1 Les motifs clos

Définition 4. *Soit une base de séquences BDS et un support minimal minsupp, si une séquence S est fréquente dans BDS et qu'il n'existe dans BDS aucune super-séquence de S avec le même support, alors S est une séquence close. S est aussi dite séquence fermée.*

BDS	
S_1	$\langle(A)(B)(C)\rangle$
S_2	$\langle(A)(D)(B)(C)\rangle$
S_3	$\langle(A)(B)(J)\rangle\}$

Tableau 2.7 – Base de séquences BDS utilisée dans l'exemple 11

Exemple 11. *Considérons la base de séquences décrite dans le tableau 2.7 La séquence* $\langle(A)(B)(C)\rangle$ *est une séquence fréquente close, son support est égal à 2. La séquence* $\langle(A)(B)\rangle$ *est aussi fréquente close, son support est égal à 3.*

CLoSpan Les séquences closes ont été introduites dans [XHA03], les auteurs proposent *CloSpan* (pour le terme anglais CLosed Sequential Patterns mining) c'est un algorithme qui compresse l'ensemble des séquences fréquentes en un ensemble plus petit, celui des séquences closes fréquentes. L'algorithme est une amélioration de l'algorithme *PrefixSpan*. Il permet d'éviter un nombre d'itérations récursives tel que chaque motif fréquent est comparé aux motifs clos précédemment extraits si le fréquent est une sous séquence d'un fréquent alors son extension est stoppée. Aussi le principe d'ordre lexicographique et de relation rependue entre évènements permet d'éviter des itérations intermédiaires (principe inspiré du principe de bi-projection). Cependant, la nécessité de maintenir en mémoire l'ensemble des motifs clos découverts, associés à des signatures de leurs projections, rend l'algorithme gourmand en espace mémoire.

BIDE Pour pallier les inconvénients de CloSpan, les auteurs dans [WH04] présentent l'algorithme BIDE (pour le terme BI-Directional Extension) qui est une amélioration de PrefixSpan ; Cet algorithme ne retourne que les motifs fréquents clos. L'algorithme utilise l'approche *FP-growth* pour identifier les séquences fréquentes. A chaque itération deux étapes permettent de vérifier qu'un motif est clos : une vérification en avant et une vérification en arrière. La première vérifie si le motif fréquent est extensible, cette étape consiste à s'assurer si dans la projection il existe des évènements fréquents ayant le même support que la séquence. La seconde vérifie si le motif est extensible à son début ou à son milieu. Pour cela une recherche d'évènements fréquents dans les préfixes qui lui sont associées dans la base. *BIDE* permet d'extraire tous les motifs fréquents clos sans maintenir en mémoire cet ensemble au cours de l'extraction. Ses performances sont améliorées en utilisant la pseudo-projection.

5.2 les motifs Maximaux

Définition 5. *Soit une base de séquences BDS et un support minimal minsupp, si une séquence S est fréquente dans BDS et qu'il n'existe dans BDS aucune super-séquence fréquente de S, alors S est une séquence maximale dans BDS.*

Exemple 12. *Si on reprend l'exemple précédent, la séquence $\langle(A)(B)(C)\rangle$ est maximale alors que la séquence $\langle(A)(B)\rangle$ ne l'est pas, car elle est contenue dans la première.*

L'extraction de séquences fréquentes maximales a été introduite dans [LC05]. Les auteurs proposent *MSPX* (pour Maximal Sequential patterns by using multiple samples), c'est un algorithme qui extrait les séquences maximales en deux phase : le *bottum-up* et le *Top-Down*. La première est une extraction par niveau qui fournit une approximation des séquences fréquentes de longueur supérieur à 3 dont les supports sont calculés à partir d'un échantillon de la base. La seconde effectue une vérification descendante (des plus longs au plus courts) des potentiels fréquents maximaux. La phase de *Top-Down* exploite la propriété des séquences maximales qui stipule qu'une séquence maximale ne contient aucune sous séquence maximale. Cependant, cette phase nécessite le maintien en mémoire de l'ensemble des non fréquents maximaux.

6 Conclusion

Les motifs séquentiels fréquents représentent des comportements répétés d'un ou de plusieurs spécimens décrits dans une base de séquences. Les techniques d'extraction identifient ceux qui sont fréquents. Deux principales approches d'extraction se distinguent dans les travaux vus dans la littérature : la méthode d'extraction par niveau et la méthode d'extraction en profondeur appelée aussi *FP-Growth*. La première extrait les fréquents en appliquant le double procédé génération- élagage. La seconde extrait les séquences fréquences en effectuant des projections répétées permettant de résumé a chaque fois l'espace de recherche des extensions.

L'extraction des séquences fréquentes peut être affinées par l'application de contraintes, dans ce chapitre nous nous somme concentré sur les contraintes temporelles qui permettent de gérer la distances entre les transactions successives, l'étendue totale de la séquence mais aussi une taille de fenêtre qui permet de fusionner les transaction.

Certains travaux proposent de résumer les séquences fréquentes extraites et définissent les séquences maximales et les séquences closes.

L'extraction de séquences fréquente présente une palette de technique souples et adaptable aux domaines d'application. Ces techniques permettent de spécifier les besoins de formulations,

d'extraction et d'exploitation des résultats fournis peu de restrictions sur les données initiales sont nécessaire.

A notre connaissance, aucune initiative de recherche n'a été entreprise sur la relaxation temporelle des séquences fréquentes temporelles à estampilles discrètes.

Nos travaux se sont attachés à mettre en place une stratégie de sélection plus souple des fréquents. Cette stratégie vise à sélectionner comme fréquents des évènements localement fréquents et dont l'enchaînement ne l'est pas forcément tout en préservant l'estampillage temporel des séquences et permettre une exploitation avec une incertitude contrôlée du paramètre temporel.

Nous tentons dans la deuxième partie de ce mémoire de mettre à profit de la maintenance des données aéronautiques hétérogènes via l'application de techniques d'extraction de séquences fréquentes. Les motifs récupérés représentent des « utilisations typiques » qui apparaissent fréquemment avant les applications des tâches de maintenance. Bien formulées, ces utilisations typiques permettrons de prédire les prochaine applications de maintenance au vue d'utilisation récentes ou futures en se basant sur un historique de mise en services d'aéronefs de même type.

Deuxième partie

Contribution

Introduction

Dans le domaine de l'aéronautique, différents acteurs sont impliqués dans la programmation et le suivi de la maintenance des équipements.

L'OEM (Original Equipment Manufacturer -Le fabriquant de l'équipement) émet, à la livraison de son produit, un programme de maintenance l'accompagnant et édictant les contrôles et vérifications régulières à faire sur cet équipement au cours de la vie de celui-ci. L'opérateur qui utilise cet équipement doit s'assurer qu'il respecte ces conditions d'utilisation et les visites régulières en atelier recommandées par l'OEM selon l'usage qui en est fait. Le centre de maintenance assure la réalisation des contrôles sur l'équipement, voire applique des tâches de maintenance. A ce titre, l'exécution d'une action de maintenance ne peut être réalisée que par un centre certifié lequel doit également répondre à des exigences édictées par les autorités compétentes (EASA, FAA, ...).

Différents niveaux de qualifications des centres de réparation existent selon le degré d'intervention technique sur l'équipement. Par exemple, on retrouve communément ce genre de niveaux d'interventions :

- Line Maintenance : la maintenance est effectuée « sous l'aile » pour maintenir l'équipement en service. On parle de maintenance en ligne. On peut avoir comme type de maintenance en ligne : un test, une inspection, une réparation sur un moteur, un remplacement de LRU (Line Replaceable Unit), idem pour inverseur et nacelle, une interrogation des systèmes avions, une recherche de panne (avec valise de test), ...
- Customer Maintenance ou Operator Shop : par exemple, un moteur est déposé de l'avion et mis sur chariot. La maintenance est effectuée dans les locaux du client. On peut avoir comme type de maintenance chez le client : remplacement d'un moteur ou d'un module, contrôle endoscopique, essais moteurs au banc opérateur, ...
- Shop Maintenance ou Repair Shop : par exemple, un moteur ou un train d'atterrissage

est déposé et envoyé en maintenance dans un atelier de réparation. On peut avoir comme type de maintenance en atelier : remplacement de P/N, workscoping, réglage régulateur au banc, ...

Comme dans tout autre domaine, et particulièrement en maintenance aéronautique, le diagnostic efficace et rapide des usures et des défaillances des véhicules et de leurs composants nécessite une bonne connaissance des comportements des équipements et de leurs éventuelles défaillances. Une telle expertise acquise à travers l'expérience des intervenants de la maintenance, peut être spécifique à un type de véhicules ou d'équipements dans des situations particulières.

Nos travaux s'inscrivent dans ce contexte industriel et tentent d'apporter une plus value aux intervenants de la maintenance aéronautique en fournissant un service d'aide à la décision pour la gestion de la maintenance des équipements en particulier et des véhicules en général. Le service que nous proposons se base sur l'analyse des historiques d'utilisations datées et hétérogènes se rapportant à la multitude d'évènements apparaissant durant les vies des véhicules. L'analyse permet de mettre en place un procédé d'extraction de connaissances utiles adaptées à la maintenance aéronautique ce qui facilite la prévision des tâches de maintenance selon les utilisations (conditions, fréquences, ...) des véhicules et des observations relevées.

Partant des données historiques hétérogènes collectées, notre travail comporte principalement trois étapes. La première est une rationalisation et une mise en relation des différents types de données collectées. La seconde étape consiste à organiser les données et à les regrouper de manière à faciliter l'exploitation des aspects les plus intéressants pour notre étude. Enfin, la troisième étape est une analyse des données organisées qui permettent d'extraire de nouvelles connaissances utiles à l'amélioration de la maintenance et la prévision des éventuelles pannes et défaillances.

Cette partie présente dans un premier chapitre la préparation et le prétraitement des données pour l'extraction. Dans une deuxième chapitre, nous définissons un nouveau type de séquences baptisées les « Séquences Temporelles par Intervalles d'incertitude (*STI*) ». Elles représentent les comportements chronologiques ordonnés en intégrant une souplesse temporelle locale aux évènements qui la composent. Pour extraire de telles séquences, nous définissons l'algorithme *STI-PS* et présentons ses principales fonctionnalités. Un troisième chapitre présente les expérimentations permettant d'évaluer les performances techniques et fonctionnelles de notre méthode.

Chapitre 3

Données de l'étude : description et pré-traitement

Sommaire

1 Introduction

Les techniques d'extraction des connaissances à partir de données hétérogènes nécessitent une étape de préparation de l'information initiale. Ce prétraitement consiste à consolider les données, les organiser de manière à favoriser et faciliter leur analyse. Par la suite l'extraction choisie met en avant les connaissances utiles ciblées.

Ce chapitre présente l'étape de prétraitement des données historiques se rapportant à l'exploitation d'une flotte d'avions. La première section détaille les données hétérogènes considérées par notre étude et décrit le processus de leur consolidation. La seconde section présente la stratégie adoptée pour les organiser et les structurer.

2 Description des données

Cette section décrit d'abord les données disponibles pour notre étude et propose par la suite une méthodologie de leur unification et consolidation.

2.1 Présentation des données disponibles

Pour chaque véhicule, nous disposons de deux types de données : Les données de vie et les données de référence. Les données de vie d'un véhicule décrivent d'une part, l'historique de son exploitation et en spécifie les détails d'utilisation et, d'autre part, les réparations mineures et majeures réalisées sur tous ses équipements. Les données de référence sont essentiellement représentées par l'AMM lequel détaille la stratégie de maintenance adoptée par les constructeurs en aéronautique.

Plus précisément, pour chaque véhicule, des données d'utilisations, de maintenance et de référence suivantes sont disponibles :

- L'historique des opérations décrit les missions effectuées. Pour chaque sont indiqués : le matricule représentant l'identifiant de l'avion, la date de la mission, l'heure de décollage, le point de départ et le point d'arrivée, la durée du vol est spécifiée en minute, la charge du véhicule ainsi que le taux de carburant à l'aller et au retour.
- Les conditions d'utilisation : des tableaux de références avec les valeurs qui peuvent êtes prises en considération par les conditions environnementales, les types des missions, les types d'huiles moteurs utilisées.
- Un historique de l'application des tâches de maintenance. A chaque réparation effectuée, est associé le matricule du véhicule, la date de la réparation, le(s) référence(s) de(s) la partie(s) du véhicule concernées par la réparation, les valeurs des compteurs de vie, TSN (Time Since New) et CSN (Cycles Since New) associées à l'équipement ainsi que le type de la tâche appliquée.
- Pour chaque tâche effectuée, une entrée dans le tableau des données indique les informations sur sa prochaine échéance. Les champs suivants sont indiqués : le matricule de l'avion, la

référence de la tâche, son type, son statut qui peut être « applicable », « non applicable », ou « Non défini », le TSN et le CSN de l'équipement correspondant ou la date à laquelle elle doit être appliquée.

– Les « Aircraft Maintenance Mannual » (AMM) sont produits par les constructeurs des véhicules. Ils décrivent les tâches de maintenance préventives à réaliser sur chaque équipement en indiquant leurs fréquences par rapport à un compteur d'utilisation. L'ensemble des opérations de maintenance est organisé en tenant compte d'un standard de description de l'appareil (ATA par exemple).

L'ensemble de ces informations représente des données hétérogènes et disponibles issues de sources diverses. Les historiques de vie représentent des données évolutives fournies par l'exploitant : il s'agit des données de missions issues de journaux de bord et plan de vol et des données de maintenance issues de rapports et compte rendu de maintenance. Les données de références représentent des informations figées gérant la politique de maintenance.

Afin d'exploiter cette palette de données et les mettre à profit pour l'amélioration de la gestion de la maintenance par la prévision de pannes, il est nécessaire de les consolider, de les rationaliser et de les organiser dans une base de données qui facilite leur exploitation. La section suivante présente le procédé utilisé pour unifier cette masse d'information.

2.2 Alignement et mise en correspondance

Les connaissances extraites sont utilisées pour la « contextualisation » de l'aide à la décision concernant la maintenance des véhicules. Pour cela il est important, dans chaque cas, de consolider toutes les données de vie et toutes les données de référence de la maintenance afin de prendre en compte tous les phénomènes qui ont pu affecter les comportements fonctionnels des véhicules ou de leurs équipements ce qui est de nature à affecter le déroulement de la politique de maintenance prévue.

Dans un premier temps, cette section présente la mise en correspondance des données de vie issues de deux sources différentes. La première concerne les intervenants opérationnels des véhicules et inclut les missions, les plans de vols et les différents rapports d'équipage. La seconde concerne les obtenus de la maintenance, elle inclut les rapports des inspecteurs intervenant lors de la maintenance en ligne et les rapports de maintenance lourdes délivrés par les acteurs de la maintenance en atelier.

Dans un deuxième temps, nous intégrons les données de référence qui régissent la politique de maintenance générale à cette consolidation.

La mise en relation et la consolidation de la totalité des informations disponibles nous permettent de mettre en avant l'écart entre la politique de maintenance à suivre et la ligne de vie effective des véhicules et de mettre en avant le décalage entre les comportements attendus et les comportements réels.

Les évènements de vie d'un véhicule, d'une partie de ses équipements ou d'un seul de ses ceux-ci peuvent être regroupés selon trois principaux paramètres : L'état du véhicule ou de l'équipement, le temps et les compteurs de vie. La figure 3.1 modélise ces trois paramètres.

- **Le critère temporel** permet d'aligner des données datées issues des différentes sources. Étant donné que l'ensemble des données de vie collectées présente une information temporelle, la mise en relation temporelle est la plus logique. L'axe temporel représente donc la vie d'un véhicule de sa mise en service jusqu'à la fin potentielle de sa vie.
- **L'état du véhicule** représente les étapes par lesquelles il peut passer tout au long de sa vie, à savoir :
  - *En usine* cet état représente la première étape de la vie du véhicule pendant laquelle il est en cours de production. Elle est représentée dans le schéma d'alignement malgré le fait que dans le cadre de nos travaux, des données se rapportant à cette phase de production n'ont pas été exploitées.
  - *Au sol*, cet état correspond à deux situations différentes :(1) l'avion est en hangar s'il n'y a pas de missions en cours et donc pas d'information à collecter, (2) l'avion est en aérogare, il est dit entre deux vols et est alors sujet à des inspections et des opérations de maintenance en ligne. Les données collectées se rapportent alors à ces deux dernières actions.
  - L'avion subit des réparations lourdes, il est en *Atelier* et cette étape peut avoir été programmée ou non. Dans le premier cas, les réparations sont prévues conformément à la politique de maintenance en place. Dans le deuxième cas, l'interruption est imprévue et la réparation est appliquée suite à une grosse défaillance ou à un dysfonctionnement du système. Les deux cas fournissent des rapports de maintenance et d'inspection.
  - *En vol*, il s'agit de l'état de fonctionnement du véhicule. Plusieurs informations sont disponibles incluant la description de la mission (détaillée dans la section 2.1) mais aussi les rapports édités par l'équipage et le journal de bord.

  Ce critère permet de catégoriser les données selon la situation dans laquelle elles ont été acquises.
- **Les compteurs de vie** du système représentent les paramètres qui permettent de comptabiliser, selon le type de la mesure, les heures ou les cycles de fonctionnement du système.

Il s'agit des paramètres utilisés pour le pilotage de la maintenance et le calcul des vieillissement des avions.

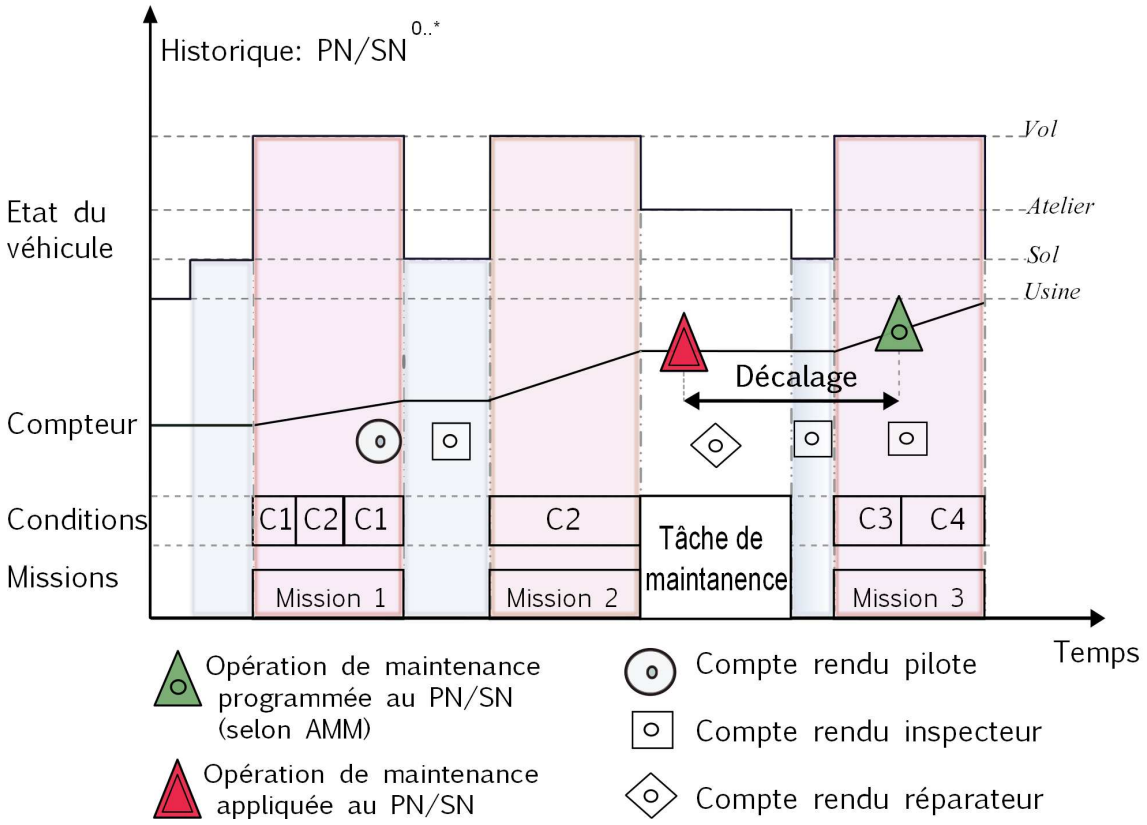


FIGURE 3.1 – Schématisation de la mise en correspondance et de l'alignement temporel des données de vies d'un avion/équipement d'avion

La figure 3.1 illustre la mise en relation des données hétérogènes selon les critères énoncés. Le paramètre temporel représente l'axe horizontal et crée une relation chronologique entre les données de vie. L'état du système est schématisé par un nivelé en haut du schéma, les données recueillies à chaque état se trouvent dans le couloir vertical correspondant. Les opérations de maintenance effectuées (représentées par des triangles rouges) sont placées, selon le moment d'occurrence de chacune sur la courbe du compteur de vie du système (représentée au milieu du schéma).

Les données de référence concernent la maintenance programmée selon la politique de prévention énoncée par les organismes certifiés. Elle est exprimée en fonction des compteurs de vie des équipements et des avions en nombre d'heures de fonctionnement ou du nombre de cycles de fonctionnement du système correspondant. La représentation de ces données est intégrée à la

consolidation de l'ensemble des informations disponibles en se référant aux courbes de compteurs de vie des équipements et des véhicules.

En résumé, les courbes correspondantes aux compteurs de vie des systèmes (avion/équipements) référencent les applications effectives de maintenance et les moments où ces maintenances ont été programmées par la politique de prévention en place. Ce critère permet ainsi de mesurer le décalage entre la « théorie » représentée par la planification de la maintenance et la « pratique » représentée par la maintenance réalisée. Ce décalage, mis dans son contexte d'occurrence et après analyse adaptée de ce contexte, permet d'identifier les comportements d'exploitation qui accélèrent plus ou moins la dégradation du système étudié.

La représentation synthétique de la mise en relation des données historiques et des données de référence est représentée par la figure 3.1. Elle peut regrouper les données par avion ou par type d'équipement. A chaque véhicule identifié par son matricule correspond un alignement chronologique des données selon les quatre critères. De plus, en aéronautique chaque pièce est répertoriée par une paire d'identifiants le P/N (Part Number) représentant le type de la pièce et un numéro de série S/N (Serial Number), qui couplé au P/N identifie de manière unique la pièce. Sachant que chaque pièce est associée à son propre compteur, elle peut être démontée d'un véhicule, être remplacée par une autre, être réparée et remontée ou juste stockée en hangar en attendant d'être remontée. L'alignement des données est autant valable pour un avion que pour une de ses pièces identifiables par un couple (P/N ;S/N).

Le but de l'exploitation de ces données est d'évaluer pour un véhicule en général (ou un équipement en particulier) l'échéance de son fonctionnement et de faire de l'aide à la décision pour l'ordonnancement de sa maintenance. Pour cela, nous tentons de prévoir, avec une certaine précision, les opérations de maintenance à appliquer relativement à un mode d'utilisation.

La consolidation des données hétérogènes, présentée dans ce paragraphe, nous permet de corréler les configurations et contextes d'utilisation et de mettre en avant le décalage entre l'application réelle des opérations de réparations et leur planification par la politique de maintenance. Ce décalage permet, dans un premier temps, d'évaluer selon le contexte d'exploitation du véhicule (de l'équipement) le degré d'applicabilité de la politique de maintenance adoptée et, dans un deuxième temps, un ajustement de la maintenance prévisionnelle afin de réduire les interruptions des véhicules, d'en augmenter l'exploitation et de minimiser les coûts de maintenance.
La section suivante présente le procédé d'organisation des données déjà consolidées sous forme séquentielle chronologique.

3 Organisation

Après avoir consolidé les données hétérogènes issues de différentes sources en un flux de données séquentielles datées, cette section décrit le procédé général de leur organisation. En effet, afin de prévoir l'échéance de fonctionnement de chaque système et/ou sous système (avion/équipement), les données se rapportant à l'ensemble des véhicules de la flotte doivent être organisées de manière à pouvoir dresser les profils comportementaux de chaque élément de l'étude.

La première partie de cette section présente le procédé général d'organisation et étale l'intérêt d'un tel procédé. La seconde partie détaille les définitions et les propriétés qui régissent cette organisation.

3.1 Procédé général

Le but de nos travaux est de mettre en place une aide à la décision pour la planification des tâches de maintenance à travers l'étude « contextualisée » de l'historique d'utilisation et des données de référence se rapportant à une flotte de véhicules aériens. En effet, dans ces travaux, nous nous focalisons sur une analyse « haut niveau » des modes d'utilisations, des défaillances observées, des maintenances appliquées et des maintenances prévues relatives aux avions et à leurs équipements. Il est entendu que cette analyse ne concerne pas l'étude technico-fonctionnelle des systèmes concernés puisque cette dernière est largement étudiée par les spécialistes dans l'industrie [1] et rigoureusement détaillée dans les documents de description des systèmes et dans les politiques de maintenance associées.

Association entre utilisations et maintenance Le principe de notre étude consiste à relier un ensemble d'historiques d'utilisations à l'application d'une tâche de maintenance avec la relation de corrélation suivante : « Telles utilisations dans telles conditions sont typiques (fréquentes) avant l'application de la tâche de maintenance M_1 ». Cette association fournit aux utilisateurs une prévision de l'application de M_1 (avec un certain degré de confiance) lorsque ces utilisations « typiques » sont prévues dans un futur plan de vol ou observées dans un historique récent.

Afin d'atteindre cet objectif, nous organisons les données chronologiquement alignées de manière à assigner les historiques d'utilisations aux historiques de maintenance. Dans une étape ultérieure, une analyse de fréquence des données est appliquée afin d'identifier les modèles de prévisions (Cette étape est détaillée dans le chapitre 4).

1. Etude réalisée sur des brevets industriels, elle est présentée dans la section 5

Pour chaque tâche de maintenance, nous proposons d'identifier les profils « d'utilisation typique » tels que chaque « utilisation typique » associe une chronologie d'utilisation à un délais d'application de la tâche.

Pour un système, à partir d'un historique d'utilisations et d'un ensemble de profils d'utilisations typiques associés aux tâches applicables, si l'historique correspond à un des profils disponibles, alors le moment de la prochaine occurrence de la tâche correspondante peut être évalué d'une manière plus ou moins précise et avec une confiance calculée.

Partant du principe que lorsqu'une réparation est effectuée sur un équipement, il est considéré cent pour cent fonctionnel, les utilisations qui suivent une réparation n'ont alors aucune incidence sur le dysfonctionnement déjà réparé. Les utilisations concernent plutôt les dégradations à venir qui provoqueront la prochaine application de la même maintenance. En se basant sur cette hypothèse, une chronologie d'historiques d'utilisations est assignée à une application d'une tâche de maintenance en amont dans le même historique de données. Chaque association représente une séquence de données reliant utilisations et maintenance.

Décomposition séquentielles De par la complexité des véhicules aériens et la complétude de la maintenance aéronautique, il existe un grand nombre de tâches de maintenance présentant des périodicités différentes et traitant d'équipements très variés. D'où le fait qu'un ensemble local d'utilisations peut affecter plusieurs équipements et en provoquer des dysfonctionnements à des degrés différents. Il convient donc de décomposer l'historique de vie d'un avion en des séquences telles que chacune commence à la première utilisation après l'application d'une tâche de maintenance et se termine à la prochaine application de la même tâche.

Ainsi, pour chaque avion, les données chronologiquement alignées seront découpées en séquences telles que chaque séquence représente l'application d'une tâche de maintenance et les utilisations (en amont) susceptibles d'avoir provoqué la dégradation correspondante.

Décomposition hiérarchique Si on considère le nombre de tâches de maintenance applicables sur un type d'avion, une analyse de fréquence des « comportements d'usure » (séquences) regroupée de toutes les séquences se rapportant à l'ensemble des tâches ne sera pas significative. D'une part, parce qu'une chronologie d'utilisation pouvant affecter plusieurs tâches de maintenance peut se retrouver dans plusieurs séquences, le calcul de sa fréquent n'est donc pas significatif ; D'autre part, les tâches de maintenance ont des périodicités différentes et très variées et donc dans un historique la fréquence d'une tâche semestrielle ne peut être comparée à la fréquence d'une tâche annuelle. Nous regroupons donc les séquences selon les tâches de maintenance qu'elles

concernent.

Pour créer de tels groupes de données, nous nous basons sur la structure décrite dans le document de référence l'AMM (Aircraft Maintenance Mannual). Ce document détaille l'ensemble des tâches de maintenance applicables à un véhicule et les organise en une structure arborescente. La racine de l'arbre représente l'avion comme un ensemble complet et englobe toutes les opérations de maintenance qui peuvent lui être appliquées. Les niveaux intermédiaires décomposent progressivement le véhicule en parties en affinant la granularité de description des tâches de maintenance à des ensembles de plus en plus spécifiques et où chaque feuille de l'arbre décrit une seule tâche de maintenance.

La figure 3.2 décrit l'organisation hiérarchique d'un avion selon le standard ATA (Air Transport Association). Un avion est décomposé avec une granularité de plus en plus précise en : avion, chapitre, section, segment, type de tâche et tâche.

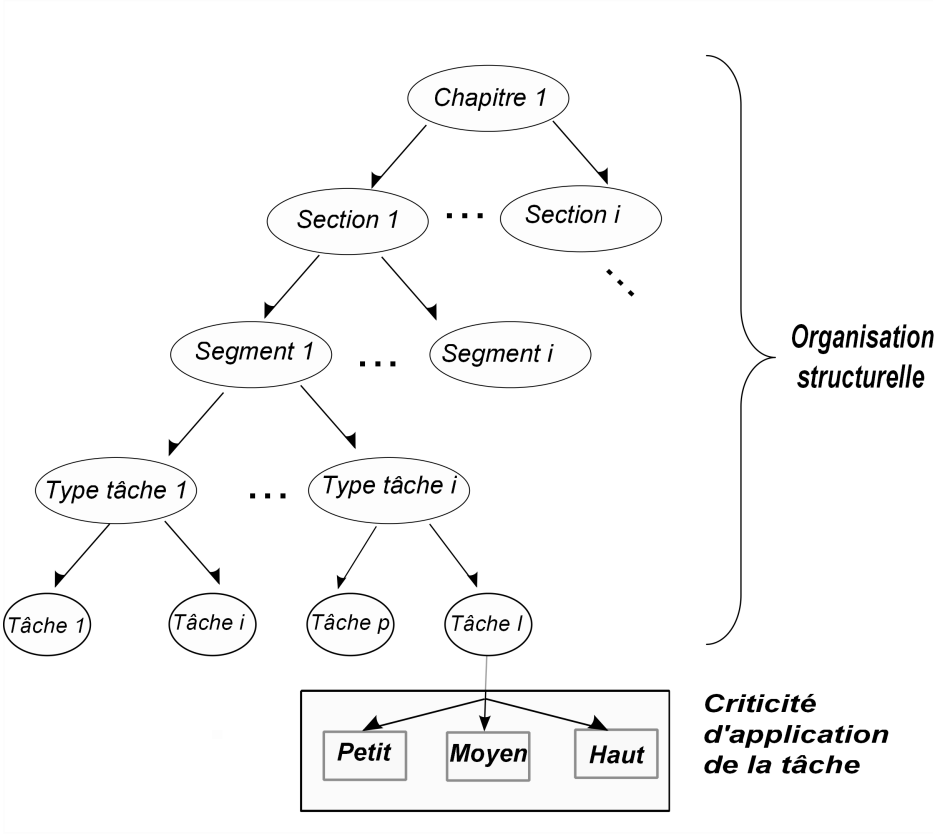


FIGURE 3.2 – Hiérarchie d'organisation des données

Les données sont décomposées en séquences telles que chaque nœud contient des historiques d'utilisations entre deux tâches de maintenance applicables à la partie du véhicule décrite par le nœud en question.

Criticité de la maintenance Nous étendons l'arborescence choisie pour organiser nos données selon un dernier niveau qui ajoute une granularité plus fine que celle des feuilles de la structure de l'AMM (tâche de maintenance). La précision ajoutée, détaille pour chaque tâche de maintenance l'état d'urgence dans lequel elle a été appliquée. Nous définissons pour cela le niveau de « criticité » selon le décalage entre la date d'application dans l'historique de vie et sa planification prévue par l'AMM (les données de références). Nous identifions alors trois niveaux de « criticité » tels que :

- une « criticité » forte regroupant les tâches de maintenance appliquées avec un décalage par rapport à la date de sa planification selon l'AMM supérieur à un seuil de criticité forte.
- une « criticité » moyenne regroupant les tâches de maintenance appliquées avec un décalage par rapport à la date de sa planification selon l'AMM supérieur à un seuil de criticité moyen et inférieur au seuil de criticité forte.
- une « criticité » faible regroupant les tâches de maintenance présentant un faible décalage entre leurs applications et les dates de leurs planifications. Ce décalage doit être au plus égal au seuil de criticité faible.

La précision ajoutée permet de distinguer pour chaque tâche de maintenance les utilisations typiques selon leur impact de dégradation sur l'équipement concerné et la criticité de leur application. Les seuils de criticité sont définis par les experts de la maintenance et sont donc propres à chaque tâche de maintenance

Chaque base de séquences de l'organisation décrite plus haut permet (après analyse) de caractériser les utilisations typiques qui ont un impact sur la dégradation de l'équipement ou sur l'ensemble des équipements décrits par le nœud correspondant.

Cette organisation fournit une arborescence de base de séquences telle que les impacts des utilisations typiques peuvent être identifiés pour les véhicules dans leurs ensembles mais aussi pour leurs parties et sous parties jusqu'au plus simple de leurs équipements. Ce découpage donne le choix du niveau de granularité de l'analyse contextualisée des impacts des utilisations.

Discussion Dans le cadre de notre étude, nous avons choisi d'organiser les données historiques selon la structure hiérarchique de la maintenance émise par les constructeurs. Ce choix est motivé par la nature des données disponibles et les besoins de validation de notre approche. Cependant, la méthode d'organisation et de regroupement des données présentée est flexible et adaptable. En effet, il est possible de choisir d'autres structures reliant de manière différente les équipements

entre eux. Les bases de données qui correspondent aux équipements peuvent être aménagées selon les fonctionnalités de ces derniers, les matériaux qui les composent ou même selon leurs paramètres techniques. Aussi, l'organisation mise en place peut être appliquée à tout système de complexité variable nécessitant l'application d'une politique de maintenance bien organisée. Parmi les applications possibles nous pouvons citer tous les véhicules complexes tel que les trains, les tanks, les drones, les machines industrielles. Ces applications sont aussi possibles pour les véhicules moins complexes tels que les bus, les trams ....

Cette section présente le principe d'organisation des données et la décomposition des historiques séquentiels en bases de séquences se rapportant aux véhicules selon différentes granularité. La section suivante présente les définitions et les propriétés qui gèrent les relations entre les différents éléments de l'organisation.

3.2 Définitions et propriétés

Dans cette partie, nous présentons les définitions et propriétés applicables aux structures d'organisation des données énoncées dans la section précédente. Tout d'abord, nous formalisons les notions de tâches de maintenance et de groupes de tâches de maintenance tels qu'ils sont organisés par la structure de l'AMM. Par la suite, nous fixons la notion de séquence qui régit le découpage du flux de données historiques selon les positions et les « significations » des éléments de l'arborescence et nous présentons les propriétés qui gèrent les interactions entre les différentes bases de séquences de la hiérarchie.

Considérons la hiérarchie d'organisation des tâches de maintenance décrite dans le paragraphe précédent et illustrée dans la figure 3.2, elle organise les réparations applicables à un type de véhicules tel que chaque nœud correspond à un groupe de tâches de maintenance applicable à une partie de l'avion. La précision de la décomposition couvre aussi bien l'avion dans son ensemble (nœud racine), ses chapitres (premier niveau), ... qu'un équipement en particulier ou encore les « criticités » d'une tâche de maintenance (feuille de la structure). Nous formalisons dans ce qui suit la représentation des tâches de maintenance selon cette structure d'organisation.

Définition 6. *Considérons l'arborescence de représentation structurelle d'un avion, et $\hat{T}$ l'ensemble des tâches de maintenance applicables aux véhicules. Nous définissons les tâches associées à chaque nœud tel que :*

- *$\hat{T}$ décrit les tâches associées au nœud racine : il s'agit de l'ensemble des réparations applicables à l'avion*

- *Pour chaque nœud intermédiaire* $\hat{T_u} = \cup_j \hat{T_{uj}}$ *tel que* $\forall j$, $\hat{T_{uj}}$ *est un nœud fils de* $\hat{T_u}$ *tel que* $\hat{T_{uj}} \in \hat{T_u}$
- *Chaque feuille de l'arborescence décrit les occurrences d'une même réparation appliquée avec le même niveau de « criticité ».*

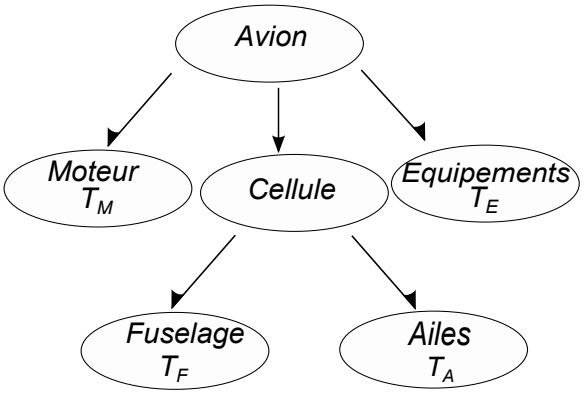


FIGURE 3.3 – Exemple de structure simplifiée d'un avion

Exemple 13. *La figure 3.3 décrit une structure simplifiée d'un véhicule. Cette structure organise l'ensemble des tâches de maintenance* $\hat{T} = \{T_a, T_f, T_m, T_e\}$ *sur trois niveaux. Le premier niveau contient le nœud racine qui profile la totalité du véhicule. Le second niveau contient trois nœuds dont deux (moteurs et équipements) feuilles et un nœud intermédiaire qui est représenté par* $\hat{T_c} = \{T_a, T_c\}$*(c pour cellule). Le troisième et dernier niveau contient les feuilles filles du nœud représentant la cellule de l'avion (structure).*

Notre but étant d'organiser les données conformément à cette structure afin de pouvoir (après analyse) identifier les corrélations entre les utilisations, leurs contextes et les applications des tâches de maintenance. Pour ce faire, nous définissons une séquence données comme étant l'association d'un historique d'utilisations à la réparation d'une ou d'un ensemble de dégradation(s) éventuellement causée(s) par ces utilisations. Une telle séquence associe un ensemble d'historiques d'utilisations à une ou plusieurs tâches de maintenance ; Elle est donc composée de deux parties : l'utilisation et la maintenance

- L'utilisation représente la première partie de la séquence. Considérant un flux historique de données et une ou un groupe de tâche(s) de maintenance $\hat{T_u}$, les utilisations associées sont sélectionnées dans deux cas :
  - Si elles apparaissent dans l'historique entre le commencement du flux de données d'utilisation et la première application de maintenance décrite dans $\hat{T_u}$.
  - Si elles apparaissent entre deux applications successives de maintenance décrite dans $\hat{T_u}$.

La partie utilisation est notée : $SU = \langle (t_1, I_1) \dots (t_{n-1}, I_{n-1}) \rangle$.

- La partie maintenance marque la fin d'une séquence donnée. C'est la dernière transaction de la séquence représentant l'application d'une tâche de maintenance incluse dans $\hat{T_u}$. La maintenance est notée : $T = (t_n, T_i)$

Une séquence $S = \langle SU(t, T) \rangle$ est une implication entre une suite d'utilisations (sa partie utilisation) et l'application d'une (ou plusieurs) tâche(s) de maintenance (sa partie maintenance) $SU \rightarrow (t, T)$

Exemple 14. *Considérons un historique d'utilisation d'un avion où V_i indique que l'avion concerné a effectué le vol V_i et T_i indique que la tâche de maintenance T_i lui a été appliquée. L'historique est représenté par des données chronologiquement ordonnée comme suit :*

$$\langle V_1, V_2, V_3, V_1, T_F, V_2, V_1, V_3, V_4, T_A, V_1, V_4, V_3, V_2, V_1, T_A, V_1, T_E \rangle$$

A cet avion correspond la structure simplifiée décrite dans l'arborescence de la figure 3.3. La séquence $\langle V_1, V_2, V_3, V_1, V_2, V_1, V_3, V_4, T_A \rangle$ est une corrélation entre les huit premiers vols effectués par le véhicule et la tâche de maintenance T_A. Cette séquence sera contenu dans la base de la feuille Ailes car elle se termine par une tâche applicable aux ailes de l'avion. La séquence V_2, V_1, V_3, V_4, T_A quant à elle sera contenue dans la base du nœud cellule, car elle commence après l'application de la tâche T_F applicable au fuselage et se termine par l'application de la tâche T_A applicable aux ailes. Cette dernière séquence entraine la corrélation suivante : « lorsque les vols V_2, V_1, V_3, V_4 ont été effectués successivement, une tâche de maintenance de la cellule (T_A) a été appliquée.

Définition 7. *Soit une séquence donnée S et un ensemble de tâches de maintenance $\hat{T}$, nous définissons σ un opérateur qui fournit une base de séquences telle que :*

$$\sigma(S, \hat{T}) = \{S_1 \dots S_n / S_i = \langle e_1 \dots e_n \rangle \subseteq S; e_n \in \hat{T}; e_0 \in \hat{T} \cup \{\emptyset\}; \forall i 1 \leq i \leq n-1; e_i \notin \hat{T}\}$$

Remarque : Pour un ensemble de tâches $\hat{T} = \cup_i T_i$ et une séquence contenant des applications de tâches inclues dans $\hat{T}$, nous avons :

- Chaque base de séquences associées à un nœud est une nouvelle décomposition des données séquentielles de départ telle que : $\sigma(BT, \hat{T}) \neq \bigcup_i \sigma(BT, T_i)$
- Pour une tâche $T_i \in \hat{T}$ le nombre d'occurrences de T_i dans la séquence de départ est inférieur à celui des occurrences de toutes les tâches décrites par $\hat{T}$, d'où nous pouvons dire que $|\sigma(BT, \hat{T})| \gneq |\sigma(BT, T_i)|$

Pour un flux d'historiques chronologiques SH et un ensemble de tâches de maintenance $\hat{T} = \cup_i T_i$, une séquence contenue dans $\sigma(SH, T_i)$ contient une séquence de $\sigma(SH, \hat{T})$. En effet,

lorsque dans SH deux occurrences de T_i se succèdent, le découpage de la séquence historique fournit une séquence qui sera à la fois dans $\sigma(SH, T_i)$ et dans $\sigma(SH, \hat{T})$. Dans le cas contraire, lorsque dans SH entre deux occurrences d'une tâche T_i apparait l'application d'une autre tâche alors le découpage de l'historique fournit une séquence pour $\sigma(SH, T_i)$ qui contient celle pour $\sigma(SH, \hat{T})$.

Propriété 2. *Soit $\hat{T} = \cup_i T_i$ alors $\forall S \in \sigma(BT, \hat{T})$, $\exists S' \in \sigma(BT, T_i); T_i \in \hat{T}$ telle que $S \subseteq S'$.*

Preuve. Soit un ensemble d'évènements $\omega = \{e_1 \dots e_k\}$, un ensemble de tâches de maintenance $\hat{T} = \cup_i T_i$ et une séquence temporelle $SH = \langle e_{j_1}, e_{j_2}, \dots e_{j_m} \rangle$, $\forall 1 \leq i \leq m$ $e_{j_i} \in \{e_1 \dots e_k\} \cup \hat{T}$.

Pour $s \in \sigma(BT, \hat{T}) = \{S | S = \langle e_{j_k} \dots e_{j_n} \rangle\}$ avec $e_{j_{k-1}} \in \hat{T} \cup \{\emptyset\}$, $e_{j_n} \in \hat{T}$ et $\forall k \leq i \leq n-1; e_{j_{n-1}} \notin \hat{T}$ Pour une tâche de maintenance $T_u \in \hat{T}$, deux possibilités de configurations existent pour s :

1. $e_{j_{k-1}} \in \{T_u, \emptyset\}$ et $e_{j_n} = T_u$ alors il existe $s' \in \sigma(BT, T_u)$ telle que $s' = s$ et donc $s \subseteq s'$
2. $e_{j_{k-1}} \in \hat{T} \setminus \{T_u\}$ et $e_{j_n} \in T_u$ il existe alors $e_{j_p} \in BT \cup \{\emptyset\}$ tel que e_{j_p} apparaît avant $e_{j_{k-1}}$ et $e_{j_p} = T_u$ tel que $\langle e_{j_{p+1}} \dots e_{j_{k-2}}, e_{j_k}, \dots e_{j_n} \rangle = \langle e_{j_{p+1}} \dots e_{j_{k-2}}, s \rangle = s'$. Notons que $s' \in \sigma(BT, T_u)$ et donc $s' \supseteq s$

Nous pouvons donc conclure que pour chaque séquence $s \in \sigma(BT, \hat{T})$ existe $s' \in \sigma(BT, T_i)$ tels que $T_i \in \hat{T}$ et $s \subseteq s'$ □

Exemple 15. *Les applications de l'opérateur σ sur le flux de données historiques SH selon les tâches de maintenance décrites dans l'exemple (Figure 3.3) sont illustrées dans le tableau 3.1. Conformément à la structure organisationnelle, le tableau contient six bases de séquences, chacune assignée à un nœud de l'arborescence. Nous pouvons remarquer que chacune des séquences de la base du nœud avion est contenu dans au moins une des autres bases de séquences. De même il faut noter que, que les bases de séquences du haut de la hiérarchie contiennent des séquences plus courtes et plus nombreuses que les bases du plus bas niveau.*

$$SH = \langle V_1, V_2, T_E, V_3, V_1, T_F, V_3, V_1, T_M, V_3, V_4, T_A,$$
$$V_4, V_3, T_E, V_2, V_1, T_A, V_1, V_4, V_3, T_E, T_F, T_M, V_2, V_3, V_4, T_A \rangle$$

4 Conclusion

Ce chapitre a présenté d'abord les données disponibles pour notre étude, il a détaillé par la suite le procédé utilisé pour les consolider, les unifier et les organiser afin de permettre leur analyse et leur exploitation. L'organisation présentée consiste à aligner sur un axe temporel les

<table>
<tr><td colspan="3">Avion</td></tr>
<tr><td colspan="3">$\langle V_1, V_2, T_E\rangle$,$\langle V_3, V_1, T_F\rangle$, $\langle V_3, V_1, T_M, \rangle$ $\langle V_3, V_4, T_A\rangle$, $\langle V_4, V_3, T_E, \rangle$, $\langle V_2, V_1, T_A\rangle$, $\langle V_1, V_4, V_3, T_E, T_F, T_M, \rangle$, $\langle V_2, V_3, V_4, T_A\rangle$</td></tr>
<tr><td>Moteur</td><td>Cellule</td><td>Équipements</td></tr>
<tr><td>$\langle V_1, V_2, V_3, V_1, V_3, V_1, T_M\rangle$, $\langle V_3, V_4, V_4, V_3, V_2, V_1,$ $V_1, V_4, V_3, T_M\rangle$</td><td>$\langle V_1, V_2, V_3, V_1, T_F\rangle$ $\langle V_3, V_1, V_3, V_4, T_A\rangle$ $\langle V_4, V_3, V_2, V_1, T_A\rangle$, $\langle V_1, V_4, V_3, T_F\rangle$ $\langle V_2, V_3, V_4, T_A\rangle$</td><td>$\langle V_1, V_2, T_E\rangle$ $\langle V_3, V_1, V_3, V_1, V_3, V_4, V_4, V_3\rangle$ $\langle V_2, V_1, V_1, V_4, V_3, T_E\rangle$</td></tr>
<tr><td colspan="2">Fuselage</td><td>Ailes</td></tr>
<tr><td colspan="2">$\langle V_1, V_2, V_3, V_1, T_F\rangle$, $\langle V_1, V_4, V_3, T_F\rangle$</td><td>$\langle V_3, V_1, V_3, V_4, T_A\rangle$,$\langle V_4, V_3, V_2, V_1, T_A\rangle$, $\langle V_2, V_3, V_4, T_A\rangle$</td></tr>
</table>

Tableau 3.1 – Décomposition de l'historique SH la structure d'organisation de la figure 3.3

données initialement hétérogènes, à les découper en séquences de sorte à associer utilisations et maintenance et à regrouper ces séquence selon la granilarité de la maintenance qu'elles décrivent. Aussi une précision a été apportée à cette organisation afin de différencier le degré d'urgence « faible », « moyen » ou « fort » d'application des tâches de maintenance par rapport à la politique de maintenance initialement prévue.

La méthode proposée pour le pré-traitement des données est flexible et peut être adaptée selon le sujet et le domaine d'application de l'étude puisqu'il n'y aucune contrainte posée par rapport à la décomposition des données. La méthode de consolidation des données présentée dans ce chapitre a fait l'objet d'une présentation lors d'un séminaire se rapportant à l'industrie aéronautique [BZR11]

A l'issue d'une telle décomposition, nous pouvons extraire, pour chaque nœud et donc pour chaque degré de précision de l'avion, les utilisations typiques qui peuvent affecter ses composants. La méthode d'extraction utilisée met en en avant les comportements qui seraient extraits par des experts humains. la formalisation et le procédé d'extraction des utilisations typiques sont présentés dans le chapitre suivant.

Une fois les données hétérogènes consolidées et organisées, la prochaine étape consiste à les exploiter et à les analyser afin d'extraire des bases de données cibles utiles et qui permettent d'extraire les comportements typiques pour les tâches de maintenance décrites dans la base analysée.

Chapitre 4

Extraction de séquences temporelles par intervalles

Sommaire

1 Introduction

La réduction du coup de la maintenance des véhicules est un des challenges des plus importants pour les exploitants d'aéronefs. Effectivement, toute interruption d'un véhicule représente un coup élevé en termes de main d'œuvre, d'équipements, de logistique diverse mais aussi en terme de temps « non exploité » de l'avion.

Notre travail fait partie d'une initiative industrielle qui vise à apporter une valeur ajoutée aux exploitants de flotte et leur permettre de réduire les temps d'immobilisation et baisser les coûts de la maintenance. Cette optimisation des réparations permet de réduire le coût de la maintenance via un ordonnancement plus « économique » et « intelligente » des tâches initialement prévues et celles non prévues.

Notre travail se focalise sur l'extraction des prévisions d'applications de tâches de maintenance à partir d'un volume de données datées et hétérogènes disponibles autour de la vie de la flotte. Ces données tracent les évènements de vie des avions et les organisent selon le décalage entre les évènements de maintenance réellement appliqués et ceux prévus par la politique de maintenance.

Afin d'exploiter au mieux les données, une phase de pré-traitement (présentée au chapitre 3) met en relation les utilisations et les tâches de maintenance de chaque avion et les organise en séquences représentant une relation de corrélation ($utilisation \longrightarrow maintenance$).

A partir de ces bases de séquence, nous identifions les « utilisations typiques » qui représentent les utilisations fréquemment associées à l'application d'une tâche de maintenance en particulier et permettre d'estimer sa prévision.

Dans une première section, sont présentés les besoins de formulation et d'exploitation des « utilisations typiques » de manière à fournir des séquences représentant le plus fidèlement possible les besoins des futurs utilisateurs. Ces besoins motivent les choix que nous portons pour définir les « séquences temporelles par intervalles d'incertitude (STI) ».

Dans une deuxième section, nous présentons la définition des séquences temporelles par intervalles d'incertitudes, leurs propriétés et les contraintes qui s'appliquent à ces séquences.

Dans la troisième section est présenté « *STI-PS* » l'algorithme mis en place pour extraire les « utilisations typiques » à partir d'une base de séquences en prenant en compte un support minimal et des contraintes temporelles. Finalement, nous concluons dans une quatrième section.

2 Motivation

Pour que la prévision des tâches de maintenance soit efficace et pertinente, les utilisations typiques doivent relater au mieux les comportements répétés « significatifs ». Ainsi, les séquences extraites à partir des historiques de vies doivent se rapprocher au mieux des comportements qui seraient relevés par un expert humain du domaine.

Pour identifier ces comportements, nous avons réalisé une phase de consultation d'experts qui nous a permis d'identifier trois critères principaux que doivent présenter les utilisations typiques.

Le premier critère concerne la fréquence de la chronologie des évènements qui forment les motifs, le second concerne l'importance de l'information temporelle reliées aux évènements dans un motif et le troisième concerne la souplesse de la chronologie « locale » des évènements. La combinaison de ces critères fait que les motifs extraits représentent des « utilisations typiques » qui ont une forte probabilité d'entraîner l'application de la tâche de maintenance à laquelle ils

sont associés. Nous détaillons dans ce qui suit chacun de ces trois critères.

Fréquence Le premier critère représente la fréquence de l'ensemble de la séquence extraite. Un comportement n'est considéré typique, et donc la séquence correspondante n'est identifiée en tant que fréquente, que si elle apparaît un nombre suffisant de fois. Ce nombre, dit la fréquence, est fixé par l'utilisateur du système et s'exprime en terme de pourcentage de présence dans les séquences de la base.

Ancrage temporel Le second critère concerne l'aspect temporel des motifs extraits. Lorsque les utilisations typiques (séquences fréquentes) relevées ne présentent pas d'informations temporelles qui renseigne le décalage entre les évènements qui la constitue, la pertinence de prévision de cette dernière est moins pertinent. En effet, la correspondance entre les parties utilisations des deux séquences non temporelles peut faire correspondre des comportements différents et lorsque la mise en relation des deux parties utilisations est temporellement acceptable les séquences ne fournissent aucune information sur le moment auquel elle devra l'être.

S_1	$\langle(12/01/2000, V_2)(13/01/2000, V_3V_5)(15/01/2000, V_1)\rangle$
S_2	$\langle(12/01/2002, V_2)(13/05/2002, V_3V_5)(15/10/2001, V_1)\rangle$

Tableau 4.1 – Exemple de séquences d'utilisations d'avions

Exemple 16. *Supposons que V_i désigne un vol particulier et M_i une tâche de maintenance. Soit $SI = \langle(V_2)(V_3V_5)(V_1)(M_2)\rangle$ une séquence extraite représentant une utilisation typique. Elle est associée à la fréquence σ. SI véhicule l'information suivante : « Dans au moins σ pour cent des cas rencontrés, la séquence des utilisations $(V_2)(V_3V_5)(V_1)$ est suivie par l'application de la tâche de maintenance M_2 ». Les deux séquences décrites dans le tableau 4.1 correspondent à SI. Chacune d'elles permet de prédire une prochaine application de la maintenance M_2. Cependant, si l'on regarde de plus près les datations des évènements de S_1 et S_2, on peut constater que les deux comportements qui y sont représentés sont très différents : dans S_1, les vols sont effectués avec quelques jours d'intervalle, alors que dans S_2, ils sont effectués avec des mois d'intervalle. Les deux situations n'ont pas le même impact sur la dégradation des équipements concernés par la tâche de maintenance M_2.*

La présence de l'estampille temporelle dans les séquences fréquentes (« utilisations typiques ») permet d'éviter de telles ambiguïtés.

Souplesse des datations locales Le troisième critère concerne la souplesse de l'ordre local des évènements. Ce critère permet de considérer comme étant une succession fréquente une suite d'évènements « proches » qui apparaissent dans un ordre différents dans la base.

S_1	$\langle(0, V_1)(1, V_2)(2, V_3)(5, M_1)\rangle$
S_2	$\langle(0, V_1)(1, V_3)(2, V_2)(6, M_1)\rangle$

Tableau 4.2 – Exemple de séquences de corrélation entre utilisations et la tâche de maintenance d'avions M_1

Exemple 17. *Considérons la base de séquences $\mathcal{S}$ décrite dans le tableau 4.2. Ces séquences relient des évènements de vie d'un avion à la tâche de maintenance M_1. Considérons une fréquence égale à 100%, $SI = \langle(V_1)(V_2V_3)(M_1)\rangle$ est un comportement utile a relever. Cette séquence représente la fréquence de l'ordre global d'apparition de ses évènements. Cependant, elle véhicule une souplesse sur l'ordre d'apparition des évènements de la deuxième transaction. Celle-ci les vols V_2 et V_3 et les considère comme simultanés alors que dans les données ces deux vols sont alternés mais proches.*

Afin de concilier les trois critères lors de l'extraction de séquences « intéressantes » pour un expert en maintenance aéronautique, nous proposons de *fusionner* des évènements consécutifs mais proches temporellement en un ensemble d'évènements simultanés et de leur associer une estampille temporelle sous forme d'intervalle. Cette estampille permet de préserver l'ancrage temporel des évènements des séquences tout en autorisant une souplesse chronologique à travers les intervalles d'incertitude. De cette manière, les transactions temporellement proches qui contiennent des évènements considérés comme ponctuels seront fusionnées et leur fusion associée à une estampille temporelle exprimée par un intervalle. Cet intervalle représente la tranche temporelle durant laquelle les évènements des transactions fusionnés peuvent apparaître. C'est une incertitude sur leurs moments d'apparitions. Pour gérer cette incertitude, nous proposons une taille de fenêtre qui fixera un seuil maximal de regroupement et donc une incertitude maximale.

Exemple 17 (Suite)**.** *Avec une taille de fenêtre égale à 2 on peut extraire la séquence suivante : $SI_1 = \langle([0, 0]V_1)([1, 2]V_2V_3)([5, 6]M_1)\rangle$. Elle relate « l'utilisation typique » suivante : « Le vol V_1 est effectué, il est suivi par les vols V_2 et V_3 qui sont effectués* dans n'importe quel ordre *dans un intervalle de largeur 1. Par rapport à V_1, V_2 et V_3 apparaissent au plus tôt une unité temporelle après. V_2 et V_3 sont eux-mêmes suivis par la tâche M_1 après un laps de temps au*

moins égal à 3 et au plus égal à 5 ». Cette séquence répond à tous les critères relevés lors de la phase d'interrogation d'expert en maintenance aéronautique.

Dans la section suivante, nous définissons les séquences temporelles par intervalles et présentons leurs propriétés et les contraintes temporelles qui leur sont appliquées.

3 Définitions et propriétés

Dans un premier temps nous rappelons les principales définitions énoncées dans plusieurs travaux traitant l'extraction des séquences fréquentes (ESF) à partir de séquences temporelles ([HY06, FVNN08, PHMA+01]).

Définition 8. *Une séquence temporelle S notée $S = \langle (t_1, I_1), (t_2, I_2) \dots (t_n, I_n) \rangle$ avec $n \in \mathbb{N}$ où $\forall 1 \leq i \leq n$, I_i est une transaction et t_i est son estampille temporelle.*

Le support absolu d'une séquence S dans une base de séquences BDS est noté $support_{a(BDS)}(S)$. Il correspond au nombre de séquences de BDS qui contiennent S. Son support relatif est noté $support_{r(BDS)}(S)$. Il correspond au pourcentage de séquences de D qui contiennent S. S est fréquente dans BDS si et seulement si son support (absolue ou relatif) est supérieur ou égal à un seuil minimal minsupp fixé par l'utilisateur.

Dans le reste de ce chapitre nous utilisons le support absolu. Pour des raisons de simplification, il sera noté $support_{BDS}(S)$.

Nous définissons maintenant les séquences temporelles par intervalles d'incertitude et présentons leurs propriétés.

Définition 9 (Séquence temporelle par intervalles (STI)). *Soit un ensemble d'évènements $\omega = \{e_1, e_2, \dots, e_k\}$. Une séquence temporelle par intervalles (STI) S de longueur n est notée : $S = \langle ([m_1, M_1], I_1), ([m_2, M_2], I_2) \dots ([m_n, M_n], I_n) \rangle$ où $([m_i, M_i], I_i)$ est une transaction à estampille temporelle sous forme d'intervalle telle que :*

- *$\forall 1 \leq i \leq n : m_i \leq temps\_occurrence(e_j) \leq M_i$. pour tout $e_j \in I_i$;*
- *S est cohérente si et seulement si : $m_1 = 0$ et $\forall 1 \leq i \leq n-1$; $M_i \leq M_{i+1}$ et $m_i \leq m_{i+1}$*

Exemple 18. *Soit $S1 = \langle ([0,1], A)([2,2], BC) \rangle$. Elle exprime le fait suivant : « A se produit entre les instant 0 et 1. Par la suite BC se produisent simultanément entre les instants 2 et 2 (donc à l'instant 2). Relativement à A, B et C se produisent au plus tôt une unité temporelle après l'évènement A et au plus tard deux unités temporelles après ».*

$S2 = \langle ([0,3], A)([1,2], B)([2,5], C) \rangle$ n'est pas cohérente, puisque la borne supérieure du second intervalle est inférieure à la borne supérieur du premier intervalle ($M_1 = 2 \lneq m_2 = 3$).

Une séquences temporelle par intervalles véhicule des incertitudes sur les moments exactes des occurrences des évènements de leurs transactions tel que les borne inférieure et supérieure de l'intervalle d'une transaction décrivent les moments au plus tôt et au plus tard où les évènements de cette transaction se produisent par rapport à ceux de la première transaction de la même séquence.

Propriété 3. *Pour une* STI $S = \langle([m_1, M_1], I_1), ([m_2, M_2], I_2) \dots ([m_n, M_n], I_n)\rangle$ *et* $\forall 1 \leq i \leq n$ *tel que* $([m_i, M_i], I_i) \in S$ *on a :*

Chaque évènement de I_i *apparaît au plus tôt* $m_i - M_1$ *(unités temporelles) après l'apparition du dernier évènement de* I_1 *et au plus tard* M_i *(unités temporelles) après l'apparition du premier des évènements de* I_1 *tel que :*

$$\forall e_j \in I_i,\ (m_i - M_1) \leq temps(e_j) \leq (M_i - m_1) \text{ sachant que } m_1 = 0.$$

Exemple 19. *Soit* $S = \langle([0, 1], AD)([3, 5], BC)\rangle$. *La succession des évènements de cette séquence est interprétée comme suit « Les évènements* A *et* D *se produisent entre les instants* 0 *et* 1. *Les évènements* B *et* C *apparaissent au plut tôt* $3 - 1 = 2$ *unités temporelles après* A *et* D*, et au plus tard* $5 - 0 = 5$ *unités temporelles après* A *et* D *».*

Remarque Plus généralement, soient $([m_j, M_j]I_j)$ et $([m_k, M_k]I_k)$ des transactions de S avec $j < k$. Les évènements de I_k ont lieu au plutôt $m_k - M_j$ unités temporelles après ceux de I_j et au plus tard $M_k - m_j$ unités temporelles après ceux de I_j.

Concernant les séquences temporelles à estampilles discrètes, chaque transaction est associée à un instant temporel ponctuel pendant lequel ses évènements ont lieu simultanément. Il n'y a donc aucune incertitude sur les moments de leurs occurrences.

Remarque Chaque séquence temporelle à estampille discrète S tel que : $S = \langle(t_1, I_1) \dots (t_n, I_n)\rangle$ peut être transformée en une *STI* dont la largeur des intervalles est nulle (l'incertitude est nulle), elle est donc notée : $STI(S) = \langle([m_1, M_1], I_1), \dots, ([m_n, M_n], I_n)\rangle$, avec $\forall 1 \leq i \leq n, m_i = M_i = t_i$.

Afin d'adapter les délais temporels des séquences temporelles par intervalles aux besoins spécifiques d'extraction et de formulation, nous définissons dans ce qui suit les contraintes temporelles qui peuvent leur être appliquées. Comme pour les séquences temporelles à estampilles discrètes (voir section 4.1) les contraintes applicables aux STI gèrent : les distances temporelles minimales et maximales entre couples de transactions successives, fixent l'étendue temporelle minimale et maximale d'une sequence, et fixent un seuil maximal d'incertitude. Cette dernière contrainte est

assimilable aux regroupements d'évènements utilisés dans [PRM+09, MTV97a, AS95, MPT09].

Soit S une STI de longueur n. S satisfait les contraintes temporelles : *mingap*, *maxgap*, *min_whole_interval*, *max_whole_interval* et la fenêtre glissante ws si et seulement si $\forall\ 1 \leq i \leq n$:

- $mingap \leq (m_i - M_{i-1}) \leq maxgap$ La contraintes de *Gap* régule la distance temporelle « au plut tôt » minimale et maximale entre deux transactions successives.
- $min\_whole\_interval \leq M_n \leq max\_whole\_interval$ La contraintes *Whole_ interval* régule l'étendue temporelle minimale et maximale d'une séquence :
- $|M_i - m_i| \leq ws$ La *Fenêtre glissante* permet de regrouper des évènements de transactions successives dans une même transaction en leur associant un intervalle temporel. Elle fixe une incertitude maximale correspondant à la largeur maximale des intervalles des transactions :

Exemple 20. *Soit* $SI = \langle([0,1],A)([2,3],BC)([6,10],D)\rangle$. *Les contraintes temporelles mingap et maxgap sont respectivement égales à 2 et 3. SI ne satisfait pas mingap puisque* $m_2 - M_1 = 2 - 1 = 1 \leq 2$. *Par contre, SI satisfait maxgap puisque pour toutes ses transactions successives maxgap est satisfaite (*$m_2 - M_1 = 2 - 1 \leq 3$; $m_3 - M_2 = 6 - 3 \leq 3$*). Pour une taille de fenêtre égale à 1, SI ne respecte pas la taille de la fenêtre puisque* $M_3 - m_3 = 10 - 6 \gneq 1$. *D'autre part, pour une fenêtre égale à 4, la contrainte est respectée par tous les intervalles de la séquence.*

Ces contraintes régulent la sémantique temporelle des *STIs* ; Le *gap* gère la distance temporelle entre deux transactions successives et fixe la distance « au plus tôt » minimale (*mingap*) respectivement maximale (*maxgap*) entre deux transactions successives pour que la corrélation (i.e la mise en relation des co-occurrences) entre elles soit significative.

Concernant *mingap*, lorsque deux transactions sont trop proches, leur succession n'est pas considérée sémantiquement significative car les évènements sont susceptibles de ne pas être considérés comme « non simultanés ». Ce seuil permet donc de fixer la limite temporelle au delà de laquelle les évènements sont dissociés.

Lorsque cette même distance est trop importante (supérieure à *maxgap*) leur corrélation directe n'a pas de sens, car les évènements sont considérés trop éloignés pour être sémantiquement directement reliés.

La contrainte *whole_interval*, régule la longueur de la séquence afin que la corrélation de toutes les transactions soient significatives. Une séquence peu étendue n'est pas considérée comme représentant un comportement complet. *min_whole_interval* permet de ne pas considérer des « sous comportement » fréquents, qui isolé de leur contexte ne véhiculent pas d'information utile.

Dans le cas contraire, une séquence trop « longue » ($max\_whole\_interval$) peut contenir des évènements qui ne se rapportent pas au même comportement et risque de contenir du « bruit ».

La taille de la fenêtre gère le regroupement des évènements et régit l'incertitude sur leurs occurrences. Elle permet d'associer des évènements qui au départ sont successifs et les considérer comme simultanées. ils seront associés à un intervalle d'incertitude qui balaye la période dans laquelle se sont produites les transactions de départ. Cet intervalle permet de préserver l'information temporelle de départ, sa taille maximale est fixée par la taille de la fenêtre.

Remarque Lorsque ws fixe un seuil maximal de regroupement des évènement, $mingap$ définit la distance minimale à partir de laquelle les évènements sont dissociés dans des transactions différentes. $maxgap$ fixe quant à lui un seuil maximal de la distance entre transaction successives. Pour qu'une séquence qui respecte les contraintes temporelles énoncées plus haut soit cohérente, il faut que la distance de « regroupement » soit inférieure à la distance de « dissociation ». Nous avons alors : $ws < mingap$; $ws < maxgap$ et $mingap < maxgap$.

Une séquence qui ne respecte pas toutes ces contraintes n'est pas considérée comme sémantiquement significative par rapport à ces contraintes et donc représente un comportement inutile pour les utilisateurs de l'application.

Nous nous focalisons dans ce qui suit sur le regroupement des évènements des transaction par l'application de la fenêtre glissante ws.

Dans le but de considérer toutes les possibilités de regroupement des évènements des transactions successives d'une séquence relativement à la taille d'une fenêtre, cette dernière est appliquée d'une manière glissante. Dans ce qui suit nous définissons un opérateur $\diamondsuit$ qui regroupe des transactions successives d'une séquence. À partir d'une position j, $\diamondsuit$ fusionne des transactions qui se succèdent. Il applique une fenêtre dont la taille fixe l'espacement temporel maximal entre la première et la dernière transactions du groupe.

Définition 10. *Soient* $S = \langle([m_1, M_1], I_1)([m_2, M_2], I_2) \dots ([m_n, M_n], I_n)\rangle$, *un entier* $1 \leq j \lneq n$ *et* ws. *On définit l'opérateur* $\diamondsuit_{ws}$ *comme suit :*

$$\diamondsuit_{ws}(S, j) = SI' = \langle([m'_1, M'_1], I'_1)([m'_2, M'_2], I'_2) \dots ([m'_n, M'_n], I'_k)\rangle$$

- *où* $\forall 1 \leq i \lneq j$: $([m'_i, M'_i], I'_i) = ([m_i, M_i], I_i)$;
- $\exists\, 1 \leqslant l_j \leqslant l_{j+1}, \dots l_i \cdots \leqslant l_{k-1} \leqslant n$ *tels que :*
  - $I'_j = \cup_{p=j}^{l_j} I_p$; $\dots$ $I'_i = \cup_{p=l_{i-1}+1}^{l_i} I_p$; $\dots$ $I'_k = \cup_{p=l_{k-1}+1}^{l_n} I_p$,
  - $m'_j = m_j, M'_j = M_{l_j} \dots m'_i = m_{l_{i-1}+1}, M'_i = M_{l_i} \dots m'_k = m_{l_{k-1}+1}, M'_k = M_n$

– $|m_j - M_{l_j}| \leqslant ws$; $\ldots |m_{l_{i-1}+1} - M_{l_i}| \leqslant ws$; $\ldots |m_{l_{k-1}+1} - M_n| \leqslant ws$.

Exemple 21. *Soit* $SI = \langle([0,2],A)([1,2],B)([3,5],C)([4,6],D)\rangle$ *et* $ws = 3$*, alors* $\diamondsuit_3(SI,1) = \langle([0,2],AB)([3,6],CD)\rangle$*. L'opérateur* $\diamondsuit_3$ *regroupe les transactions de la séquence* S *à partir de la première transaction en appliquant une fenêtre de taille* 3*. Pour ce faire, les évènements des deux premières (respectivement des deux dernières) transactions sont regroupés et les intervalles qui leur sont respectivement associés sont fusionnés car l'intervalle résultant de la fusion des deux premières (respectivement deuxièmes transactions) est autorisé par la fenêtre (*$(2-0) \leqslant 3$ *respectivement* $(6-3) \leqslant 3$*).*

Notons que $\diamondsuit_3(SI,2) = SI$*. En effet, lorsque le regroupement commence à la deuxième position, les intervalles de la deuxième et de la troisième transactions ne peuvent être fusionnés puisque leur union est trop large par rapport à la taille de la fenêtre* ws*. Finalement,* $\diamondsuit_3(SI,3) = \langle([0,2],A)([1,2],B)\ ([3,6],CD)\rangle$ *et* $\diamondsuit_3(SI,4) = SI$*.*

Nous définissons maintenant l'opérateur $\overbrace{\diamondsuit}^{ws}$ qui pour une STI de longueur n et une taille de la fenêtre ws fournit un ensemble de STI's. L'opérateur applique la fenêtre sur une séquence en la faisant glisser pour prendre en compte toutes les possibilités de *fusion* des transactions successives. Intuitivement, $\overbrace{\diamondsuit}^{ws}$ fait glisser la fenêtre de regroupement et fournit l'ensemble de toutes les séquences condensées (toutes les possibilités de combinaison de regroupement des transactions proches) qui peuvent représenter la séquence de départ. C'est l'ensemble des résultats des applications de $\diamondsuit_{ws}$ sur une n-séquence pour toutes les valeurs entière de j dans $[1, n-1]$.

Définition 11. *Soit* $SI = \langle([m_1,M_1],I_1)\ldots([m_n,M_n],I_n)\rangle$ *et* ws *une taille de la fenêtre.*

$$\overbrace{\diamondsuit}^{ws}(SI) = \{SI_1, SI_2, \ldots, SI_{n-1}\} \text{ avec } \forall 1 \leq i \leq n-1\ ;\ I_i = \diamondsuit_{ws}(SI,i)$$

Exemple 22. *Soit* $SI = \langle([0,2],A)([1,2],B)([3,4],C)([4,6],D)\rangle$ *et* $ws = 3$*. Alors* $\overbrace{\diamondsuit}^{3}(SI) = \{\langle([0,2],AB)\ ([3,6],CD)\rangle, \langle([0,2],A)\ ([1,4],BC)\ ([4,6]D)\rangle, \langle([0,2],A)\ ([1,2],B)\ ([3,6]CD)\rangle\}$*. La séquence* $\langle([0,2],AB)([3,4],C)([4,6],D)\rangle$ *n'est pas incluse dans le résultat, par contre* $\langle([0,2],AB)([3,6],CD)\rangle$ *est « induite » de la précédente.*

Les deux opérateurs $\diamondsuit_{ws}$ et $\overbrace{\diamondsuit}^{ws}$ formalisent le regroupement d'évènements qui appartiennent à des transactions différentes par l'application d'une fenêtre glissante laquelle permet de gérer les intervalles associés aux transactions des séquences temporelles par intervalles(STI).

À ce stade, nous pouvons définir la relation d'inclusion entre deux séquences temporelles par intervalles d'incertitude. Intuitivement, SI contient SI', si et seulement si pour chaque

transaction de SI', (i) ses évènements sont contenus dans une ou des transactions de SI et (ii) à chaque fois qu'on a cette inclusion on a aussi l'inclusion des intervalles.

Définition 12. *Soient* $S = \langle([m_1, M_1], I_1), \ldots ([m_n, M_n], I_n)\rangle$, $S' = \langle([m'_1, M'_1], I'_1) \ldots ([m'_k, M'_k], I'_k)\rangle$ *et ws. S contient S', notée* $S \sqsupseteq S'$ *si et seulement si : pour tout* $([m'_j, M'_j]I'_j)$ *de* S' *et tout* $e \in I'_j$, *il existe* $([m_k, M_k]I_k)$ *de S telle que ;* $e \in I_k$ *et* $[m_k, M_k] \subseteq [m'_j, M'_j]$.

$S_1 = \langle([0,2]A)([3,4],B)([5,6]C)\rangle$
$S_2 = \langle([0,4]AB)\rangle$
$S_3 = \langle([0,2]A)([3,6]BC)\rangle$
$S_4 = \langle([0,3]A)([2,6]BC)\rangle$

Tableau 4.3 – Séquences de l'exemple 23

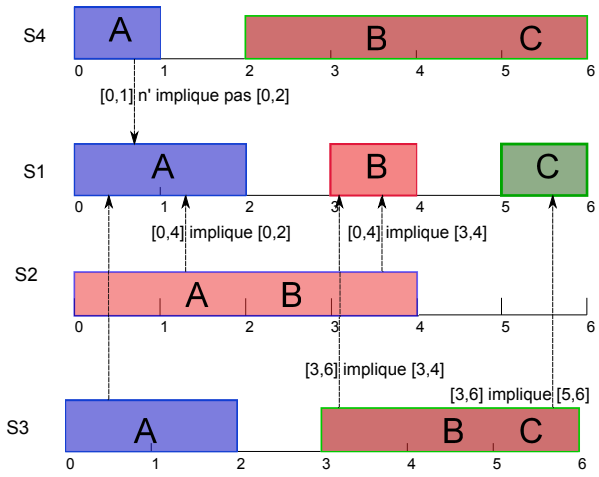


FIGURE 4.1 – Illustration des inclusions entre intervalles des séquences du tableau 4.3

Exemple 23. *Soient les séquences du tableau 4.3.* $S_1 \sqsupseteq S_2$ *puisque* $[0,4]$ *implique* $[0,2]$ *et* $[3,4]$ *et* $S_1 \sqsupseteq SI_3$. *Cependant* $S_1 \not\sqsupseteq S_4$ *puisque* $[0,2] \not\supseteq [0,3]$. *La figure 4.1 illustre l'inclusion des intervalles des séquences de l'exemple.*

Propriété 4. *Soient ws une taille de fenêtre, S une séquence temporelle à estampilles discrètes et SI une STI. S contient SI si et seulement si* $STI(S) \sqsupseteq SI$

Après avoir défini les STIs et l'opérateur d'inclusion, nous pouvons redéfinir la notion de support d'une séquence temporelle par intervalles dans une base de séquences temporelles à estampilles discrètes. Cette nouvelle définition du support permet de prendre en compte l'opérateur d'inclusion des *STI* et les propriétés qui les concernent. Nous considérons que le support d'une

STI dans une base de séquences est le nombre de séquences de la base qui la contiennent au moins une fois.

Définition 13. *Le support d'une STI SI dans D est défini par : $supp_D(SI) = |\{S \in D | S \sqsupseteq SI\}|$. Pour des besoins de simplification, $supp_D(SI)$ sera noté $supp(SI)$.*

Après avoir défini les STIs et exhibé quelques unes de leurs propriétés, nous abordons dans la section suivante l'algorithme permettant de les extraire.

4 Algorithme d'extraction : *STI-PS*

Cette section présente le procédé d'extraction des *STI* (décrites dans la section précédente) à travers l'algorithme *STI-PS* (Séquence Temporelles par Intervalle d'incertitude- PrefixSpan). *STI-PS* applique une extraction en profondeur sur une base de séquences temporelles. Il fournit toutes les séquences fréquentes temporelles par intervalles en considérant les contraintes suivantes : un support minimal *minsupp*, une taille de fenêtre glissante de regroupement *ws* et les contraintes temporelles *mingap*, *maxgap*, *min_whole_interval* et *max_whole_interval*.

Dans un premier temps, nous motivons les choix de construction des différentes fonctionnalités de l'algorithme *STI-PS*. Dans un deuxième temps, nous présentons son fonctionnement général. Par la suite, nous détaillerons ses deux principaux modules : sélection de fréquents et projection des espaces de recherches.

4.1 Choix et motivations

L'algorithme que nous proposons est appelé *STI-PS* (pour Séquences temporelles par Intervalles d'incertitude- *PrefixSpan* [PHMA+01]). Il applique une méthode d'extraction en profondeur des séquences basée sur la projection de l'espace de recherche. Le procédé consiste à extraire les fréquents par branche où chacune explore à partir d'un évènement fréquent (identifié à partir de la base de séquences de départ) l'ensemble des motifs fréquents qui l'étend.

Le choix de cette approche pour la construction de notre algorithme est motivé par deux critères. Le premier concerne les performances d'exécution. En effet, différents travaux de la littérature ([PHMA+01, HKP11, Bay98, PHW02, GHZ10, WH04]) ont montré que la méthode d'extraction en profondeur est dans la majorité des cas (tous sauf le pire des cas voir chapitre 2 section 3.3) plus efficace que la méthode d'extraction horizontale.

Le second critère concerne la difficulté et le coût de l'extraction de motifs temporels en utilisant une extraction horizontale section 3.1). En effet, les algorithmes d'une approche par niveau identifient les séquences fréquentes en appliquant les trois phases génération-élagage-test. D'abord, des candidats sont construits, ensuite la phase d'élagage en élimine ceux qui contiennent des sous séquences non fréquentes. Enfin la phase de test effectue un comptage du support des candidats et ne garde que ceux qui sont effectivement fréquents. Or, la génération de candidats temporels augmente le nombre de candidats de manière exponentielle, car elle doit envisager toutes les combinaisons de décalage temporels entre les transactions pour spécifier les estampilles temporelles.

Exemple 24. *Si l'on considère une méthode d'extension par niveau appliquée avec les contraintes* $mingap = 3$, $maxgap = 5$ *et* $ws = 2$.

Considérons qu'au cours de la deuxième itération (deuxième niveau) d'extraction cette méthode dispose des 1-STI fréquentes représentées dans la première colonne du tableau 4.4. Les candidats de longueur 2 (C_2*) sont construits à partir des évènements fréquents de longueur* L_1. *Durant cette phase, le motif* $\langle([0,0]A)\rangle$ *est étendu avec l'évènement du motif fréquent* $\langle([0,0]B)\rangle$. *Cette extension doit considérer toutes les possibilités de concaténation entre les deux éléments en prenant en compte les contraintes temporelles spécifiées. Les différents résultats sont représentés dans la deuxième colonne du tableau 4.4.*

Fréquents L_1	Candidats C_2
$\langle([0,0]A)\rangle$	$\langle([0,0]AB)$
	$\langle([0,0]A)([3,5]B)\rangle$
$\langle([0,0]B)\rangle$	$\langle([0,0]A)([4,6]B)\rangle$
	$\langle([0,0]A)([5,7]B)\rangle$

Tableau 4.4 – Candidats STI de taille 2 générés par une méthode d'extraction par niveau à partir de l'ensemble L_1

Le premier résultat ajoute l'évènement B *à la transaction de* $\langle([0,0]A)\rangle$, *dans ce cas les contraintes temporelles n'entre pas en considération. Chacun des trois autres résultats ajoute à la séquence une deuxième transaction. Chacun de ces trois résultats envisage un décalage temporel différents entre la première et la deuxième transaction. La séquence résultante doit satisfaire les contraintes* $mingap$, $maxgap$, *et* ws.

Par la suite, la phase d'élagage vérifie si les candidats contiennent une sous séquence non fréquente, si c'est le cas le candidat est éliminé. Par la suite, le calcul des supports des candidats

restants permet d'en sélectionné que ceux qui sont effectivement fréquents.

Dans cet exemple plusieurs candidats sont générés, pour chacun la phase d'élagage vérifie sa fréquence a priori, c'est une première sélection. Par la suite, un calcul de support identifie les fréquents. Pour une telle pratique, si les contraintes temporelles représentent des distances larges, le nombre de candidats est très important et occupe un espace mémoire assez important.

Lors d'une extraction en profondeur, l'espace de recherche est parcouru une fois afin d'identifier tous les évènements fréquents qui permettent d'étendre le motif en cours de construction. Alors, aucune structure supplémentaire n'a besoin d'être crée.

La section suivante présente la méthode d'extraction déployée par l'algorithme *STI-PS*.

4.2 Procédé Général

Étant données une base de séquences, un support minimal *minsupp*, une taille de fenêtre ws, l'algorithme *STI-PS* extrait les séquences temporelles par intervalles d'incertitude (*STI*) fréquentes qui satisfont les contraintes temporelles *mingap*, *maxgap*, *min_whole_interval* et *max_whole_interval*. Les séquences extraites présentent une souplesse temporelle relative à la taille de la fenêtre ws. Pour cela, *STI-PS* met en œuvre une méthode d'extraction à la *FP-Growth*.

Dans un premier temps *STI-PS*, décrit dans l'algorithme 1, extrait l'ensemble des 1-STI à partir d'un base de séquence BDS. Cet ensemble est noté $L_1 = \{S; S = \langle([m = 0, M = 0], e)\rangle; supp(e) \geqslant minsupp\}$et associe a chaque évènement fréquent un intervalle nul. Ces évènements représentent les débuts des STI plus longues extraites par les prochaines extractions.

Ensuite, l'extraction continue récursivement par la construction des $k+1$-séquences à partir d'une k-séquence fréquente. La récursion est décrite par la fonction **Extension** illustrée dans l'algorithme 2. A chaque itération i où une séquence S de longueur k est identifiée comme fréquente, le procédé algorithmique de *STI-PS* applique deux étapes :

- La première étape consiste à identifier dans BDS les 1-séquences fréquentes S'. Pour chaque S' ainsi obtenue, $S \oplus S'$ est une $k+1$-séquence fréquente ($\oplus$ désigne la concaténation). Cette étape est appelée *Sélection de fréquents* et sera détaillée dans la section 4.3 de ce chapitre.
- Une seconde étape consiste à reconsidérer l'espace de recherche. On projette alors, la base sur S pour obtenir une base BDS'. Cette dernière résume les séquences de la base BDS qui contiennent S. Les séquences de la projection sont temporellement par rapport au dernier évènement ajouté à S. Cette étape est appelée la *projection* et est détaillée dans la section

Input : BDS ,$minsupp \in [0,1]$,mingap, maxgap , min_whole_interval, max_whole_interval, ws

Data : Motifs : ensembles des STI fréquentes

L'ensemble Motifs est initialisé à null

premier initialisé à vrai

récupérer dans L_1 l'ensemble des évènement fréquent, appel de la fonction Select_frequents(BDS, $minsupp$, ws, premier)

//***** Pour chaque évènement fréquent ******* //

forall the e *de* L_1 **do**

Préfixe $\oplus([0,0], e)$

Calcul de BDS' la Projection(BDS, e, 0, 0, ws)

//***** Appel à la fonction de récursion ******* //

Extension(BDS', e, 0, 0, minsupp,ws, mingap , maxgap, faux)

if *Préfixe vérifie les contraintes min_whole_interval et max_whole_interval* **then**

Ajouter Préfixe à l'ensemble Motifs

end

end

Algorithme 1: L'algorithme principal **STI-PS** parcourt une première fois la base de séquences passée en paramètre et identifie les évènements fréquents. Pour chacun des fréquents il lance un appel de la fonction récursive **Extension**

4.4.

Le procédé est arrêté si l'une des deux conditions suivantes est satisfaite : (1) Lorsque la considération de l'espace de recherche est vide (étape 1) ou (2) lorsqu'aucune 1-STI n'est identifiée (étape 2).

Dans l'algorithme 1, chaque élément de L_1 génère une itération. D'abord, la projection de la base de séquences initiale par l'évènement fréquent est calculée. Par la suite la récursion décrite plus haut est lancée par appel de la fonction **Extension** (décrite par l'algorithme 2). Cette dernière récupère en entrée un espace de recherche (DBS), le dernier motif fréquent identifié (Prefix), les bornes de l'intervalle de la dernière transaction du motif, les contraintes temporelles et la taille de la fenêtre. D'abord, elle identifie les évènements fréquents valides de l'espace de

Input : BDS, Préfix, e, $borne\_inf$, $borne\_supp$, $minsupp$, ws, $mingap$, $maxgap$, premier , min_whole_interval, max_whole_interval

---

//***** Appel à la fonction de sélection de fréquents ******* //

Calculer l'ensemble E des fréquents par appel de Select_frequents(BDS, $minsupp$, ws, premier, $mingap$, $maxgap$)

forall the $edeE$ **do**

  Préfix $\oplus([t_min, t_max], e)$

  if *Préfix vérifie min_whole_interval et max_whole_interval* **then**

    //***** Appel à la fonction de Projection ******* //

    $BDS|_e$ = Projection(BDS, e, 0, 0, $minsupp$, $mingap$, $maxgap$)

    //***** Appel récursif avec les nouveaux paramètres ******* //

    Extension($BDS|_e$, e, 0, 0, $minsupp$, ws, $mingap$, $maxgap$, faux)

    Ajouter $Prfix$ à l'ensemble des motifs

  end

end

Algorithme 2: L'algorithme de la fonction récursive **Extension** qui appelle la fonction de sélection de fréquents, ensuite projette l'espace de recherche pour chaque 1-STI identifiée et effectue un appel récursif si le nouveau motif satisfait les contraintes $mingap$ et $maxgap$

recherche.

Par la suite, chaque évènement fréquent est concaténé au préfixe pour construire un nouveau motif. Si ce dernier satisfait les contraintes temporelles, la projection est appliquée pour résumer l'espace de recherche. Le résultat est utilisé pour appeler une nouvelle fois la fonction **Extension**.

Les algorithmes 1 et 2 formalisent le procédé d'extraction en profondeur classique tel qu'il est utilisé dans la plupart des algorithmes à la « pattern growth » ([PHMA+01, PHW02, YCJCWCSY10, GQ11, WC07, XHA03]).

Cependant, *STI-PS* se distingue des algorithmes de cette approche par l'autorisation du regroupement des évènements proches et l'application d'une fenêtre glissante qui associe souplesse « locale » de la chronologie des évènements et encrages temporels dans les motifs séquentiels. Cette association caractérise les séquences temporelles par intervalles d'incertitude. La fenêtre glissante est assimilée par STI-PS à deux niveaux du traitement :

1. La sélection d'évènements fréquents ; Un évènement est identifié comme fréquent s'il apparaît un nombre suffisant de fois et que le décalage entre les estampilles de ses occurrences

les plus éloignées est au plus égal à la taille de la fenêtre.

2. La réduction de l'espace de recherche des continuations du motif construit, appelée la projection. Cette fonction intègre un retour en arrière qui s'étend sur un espace temporel égal à la taille de la fenêtre. Ce retour arrière permet de considérer comme fréquents des évènements dont les occurrences sont éparpillées en amont et en aval du dernier évènement du motif fréquent. La projection reconsidère la référence temporelle des séquences résumées. Ces dernières sont référencées par rapport au dernier évènement du motif fréquent sur lequel on projette la base.

BDS	
S_1	$\langle(0,A)(1,B)(2,C)\rangle$
S_2	$\langle(0,A)(1,C)(2,B)\rangle$

Tableau 4.5 – Base de séquences BDS

Exemple 25. *Soient la base de séquences décrite dans le tableau 4.5, un support minimal absolu* $minsupp_a = 2$ *et une taille de fenêtre* $ws = 2$*. La figure 4.2, illustre le schéma de déroulement de la première étape et de la première branche de l'algorithme* STI-PS *sur* BDS*. l'algorithme extrait dans une première étape l'ensemble des évènements fréquents* $L_1 = \{([0,0]A), ([0,0]B), ([0,0]C)\}$ *Par la suite, il extrait un a un l'ensemble des séquences fréquentes qui découlent de chaque élément de* L_1*.*

Dans cet exemple nous nous focalisons sur la branche générée par l'évènement fréquent $([0,0]A)$*. L'algorithme calcule d'abord les résumés des séquences de* BDS *qui contiennent* A*. Nous appelons le résultat* BDS_A*, dans cet espace les références temporelles des séquence sont les apparitions de l'évènement* A*.*

Par la suite, il y identifie la 1-STI fréquentes $([1,2]B)$*.* B *est fréquent car il apparait dans les deux séquences de la base et ses apparitions se produisent* 1 *et* 2 *unités temporelles après celles de* A*. De la même manière* $([1,2]C)$ *est identifiée comme fréquente (les détails de cette identification seront expliqués dans la section 4.3). Chacun de ces fréquents génère une sous branche dans la quelle* BDS_A *est « réduite » et les évènements fréquents dans les nouveaux espaces de recherches seront identifiés.*

Considérons d'abord l'appel récursif généré par le fréquent $([1,2]B)$*. D'abord, la projection est effectuée pour résumer les séquences et en changer la références temporelles. L'instant zéro est*

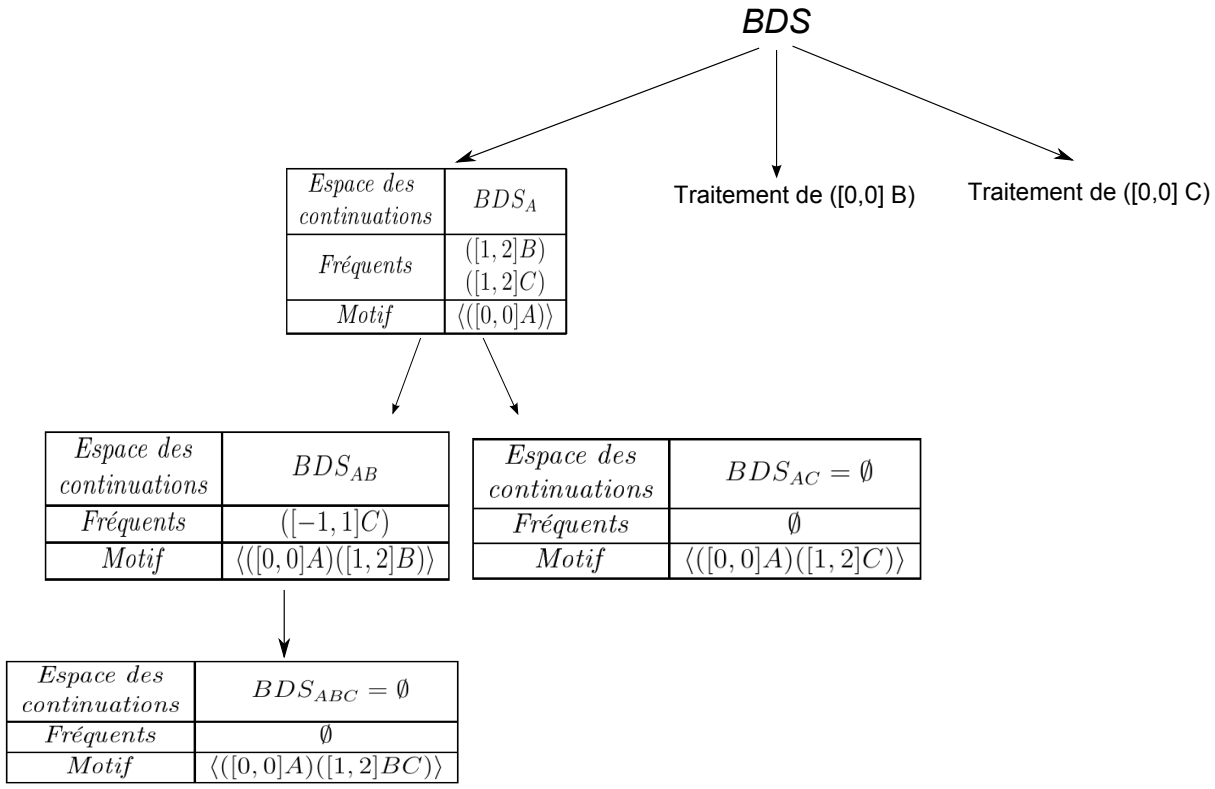


FIGURE 4.2 – Structure de déroulement de $STI - PS$ sur la base BDS.

désormais associé aux occurrences de B dont les estampilles sont initialement dans l'intervalle $[1,2]$. *Nous désignons la nouvelle base par* $BDS_{A,B}$. *Le motif obtenu par la concaténation de* $([0,0]A)$ *avec* $([1,2]B)$ *est* $\langle([0,0]A)([1,2]B)\rangle$. *Par la suite, la sélection de fréquent y identifie* $([-1,1]C)$ *comme fréquent. En effet, dans BDS l'évènement C apparait deux fois ; La première, une unité temporelle après B dans* S_1 *et la seconde, une unité temporelle avant B dans* S_2. *On peut dire que C est fréquent et apparait dans l'intervalle* $[-1,1]$ *par rapport B. La concaténation du précédent motif avec le fréquent* $([-1,1]C)$ *nécessite de référencer les deux éléments par rapport au même point temporel c'est -à dire A. C apparaît au plus tôt une unité temporelle apres la dernière apparition de B (2 par rapport à A), alors C apparait au plus tôt* $(2 + (-1) = 1)$ *unité temporelle après A. Aussi, C apparaît au plus tard une unité temporelle après la première apparition de B (1 par rapport à A), alors C apparait au plus tard* $(1 + (1) = 2)$ *unité temporelle après A. La concaténation fournit alors le fréquent* $\langle([0,0]A)([1,2]BC)\rangle$.

Par la suite la deuxième sous branche de BDS_A *est explorée. Elle est générée par le fréquent* $([1,2]C)$. *La projection de* BDS_A *par cet évènement fournit l'espace de recherche vide* $BDS_{A,C}$. *La sous branche ne sera donc pas étendue.*

Le traitement de l'extension explore une à une les branches et les sous branches générées par les évènements fréquents jusqu'à ce qu'elles ne soient plus extensibles. Un retour à un niveau

supérieur explore les autres extensions.

La section suivante, détaille la fonction de sélection de fréquents appliquée par notre algorithme.

4.3 Sélection de fréquents

La fonction de sélection des évènements fréquents identifie les 1-séquences qui apparaissent suffisamment de fois en appliquant la fenêtre glissante. Elle autorise plus de souplesse sur les estampilles temporelles en passant outre la temporalité stricte utilisée dans la plupart des techniques d'extraction de séquences temporelles. Ainsi, les occurrences « temporellement rapprochées » d'un évènement lui permettent d'être considéré comme fréquent alors qu'il ne le serait pas par les méthodes de sélection de fréquents classiques [PHMA+01, HY06, PHMA+01].

Un évènement est considéré comme fréquent s'il apparaît dans suffisamment de séquences de la base ($minsupp_a$) et que le décalage (la distance) temporelle maximale entre les estampilles ses apparitions est au plus égale la taille de la fenêtre ws. L'intervalle associé à l'évènement fréquents représente l'étendue temporelle sur laquelle il apparait. Si cette distance est inférieure à ws, alors la largeur de l'intervalle le sera aussi. De cette manière l'incertitude des occurrences des évènements est strictement égale au décalage entre ses apparitions les plus éloignées.

La fonction **Select_fréquents** est détaillée dans l'algorithme 3. D'abord, elle évalue le support et la liste des estampilles de chaque évènement qui apparaît dans la base de recherche (BDS). Ensuite, toutes les combinaisons d'intervalles sont identifiées à partir de la liste de ses temporalités telles que : (1) la largeur de l'intervalle est au plus égale à ws et (2) que le nombre d'occurrences associées à ces temporalités soit au moins égal à $minsupp$.

Exemple 26. *Si on revient sur l'exemple 25. Dans BDS, les évènements fréquents sont sélectionnés sans prendre en compte l'estampille temporelle qui leur est associée. Car, à ce stade de l'extraction, chaque évènement fréquent représente le début d'une séquence fréquente. Il est associé à une estampille nulle vu qu'il représente la référence temporelle de tous les évènements fréquents qui vont lui succéder. A la première étape de l'algorithme, les fréquents sont sélectionnés à partir de BDS sur la seule base de leurs apparitions suffisantes par rapport au support minimal.*

Cependant dans l'espace BDS_A représenté dans le tableau 4.6, les évènements fréquents sont

Input : BDS, $minsupp$, ws, premier, $mingap$, $maxgap$,

Data : V : Vecteur qui associe à chaque évènement la liste des temporalités de ses apparitions),

w = ensemble des évènements présents dans BDS,

support : Cumul du support de l'évènement en cours de vérification

Output : ω : Ensemble des 1-STI fréquentes

```
//************ Pour chaque évènement de l'espace de recherche ***** ///
foreach e ∈ w do
    support = 0
    initialiser V
    //************ parcourir toutes les séquences de l'espace de recherche ***** ///
    foreach S ∈ BDS do
        compté = false
        while e ∈ S do
            if non compté then
                //************ une seule occurrence par séquence est comptabilisée *****
                ///
                compté= true
                support ++
            V= ajout_trié(V, time(S,e))
    //**********Si l'évènement est assez fréquent ***** //
    if support ≥ minsupp then
        //********Construire le ou les intervalles qui lui sont associés ***** //
        i=0
        while i < taille(V) do
            j = i+1
            while j < taille(V) do
                if V[j] − V[i] > ws then
                    ω = ω ∪ ([V[i] − V[j]e)
return ω
```

Algorithme 3: La fonction **Select_frequents** parcourt la base de séquences et crée une liste d'estampilles d'occurrences de chaque évènement. Elle utilise la fenêtre glissante pour sélectionner toutes les possibilités d'intervalles pour un évènement fréquent.

BDS_A	
S_1	$\langle(1,B)(2,C)\rangle$
S_2	$\langle(1,C)(2,B)\rangle$

Tableau 4.6 – Base de séquences BDS_A

$([1,2]B)$ *et* $([1,2]C)$. *En effet,* B *apparaît deux fois dans les deux séquences de la base. la première dans* S_1*, il est associé à l'estampille temporelle* 1 *et la deuxième dans* S_2 *ou il est associé à l'estampille temporelle* 2*. La distance entre les estampilles ses occurrences ne dépasse pas la taille de la fenêtre* ws *puisque* $(2-1 \leq 2)$*. On peut donc dire que dans* BDS_A*,* B *est fréquent dans l'intervalle* $[1,2]$ *et permet de construire le motif fréquent* $\langle([0,0]A)([1,2]B)\rangle$*. La 1-séquence* $([1,1]B)$ *n'est pas fréquente car, associée à l'estampille temporelle* 1*, l'évènement* B *n'apparait qu'une seule foi dans* BDS_A*. Par contre on peut dire que* $([1,3]B)$ *est fréquent, car*$3-1 \leq ws = 2$*. Dans ce cas l'incertitude sur l'occurrence de* B *est plus grande que celle strictement associées à ses apparitions dans les données. De la même manière, l'élément fréquent* $([1,2]C)$ *permet d'extraire le motif fréquent* $\langle([0,0]A)([1,2]C)\rangle$.

La section suivante détaille la projection la deuxième étape du procédé d'extraction.

4.4 Projection

La projection permet de réduire la taille de la base d'extraction d'un niveau de récursion à un autre. Elle consiste à résumer la base de séquences dans laquelle le motif S' en cours d'extension a été identifié. Résumer la base de séquences consiste à faire deux choses :

- Supprimer les séquences qui ne contiennent pas S'. Ces séquences ne pouvant pas contenir des super-séquences fréquentes contenant S'.
- Supprimer des autres séquences de la base les occurrences de S' et les sous séquences de la base qui précédent S'.

Dans la plupart des algorithmes d'extraction en profondeur [FVNN08], [PHMA+01], [PHW02] [PRM+09], la projection sélectionne à partir des séquences qui incluent S' les sous-séquences en aval de cette dernière. La projection « classique » ne sélectionne que les évènements qui se produisent après S' dans la base. Les détails de ces méthodes sont précisés sans la section 3.2.

Exemple 27. *Considérons BDS représentée dans le tableau 4.7,* $minsupp = 2$ *et* $ws = 2$*. Nous*

BDS_A	
S_1	$\langle(0,A)(1,B)(2,C)\rangle$
S_2	$\langle(0,A)(1,C)(2,B)\rangle$

Tableau 4.7 – Base de séquences BDS

appliquons à BDS l'algorithme STI-PS *en utilisant la projection classique telle qu'elle est définie dans les algorithmes à la « pattern growth » [PHMA+01].*

BDS'_A	
S'_1	$\langle(1,B)(2,C)\rangle$
S'_2	$\langle(1,C)(2,B)\rangle$

(a) Projection « classique » de BDS sur $([0,0]A)$

BDS'_{AB}	
S''_1	$\langle(1,C)\rangle$
S''_2	$\langle\rangle$

(b) Projection « classique » de BDS'_A sur $([1,2]B)$

BDS'_{AC}	
S''_1	$\langle\rangle$
S''_2	$\langle(1,B)\rangle$

(c) Projection « classique » de BDS'_A sur $([1,2]C)$

Tableau 4.8 – Résultats des projections « classiques » intégrées à STI-PS sur la base initiale de BDS

Dans BDS, A est fréquent, il permet de construire le 1-motif $\langle([0,0]A)\rangle$. *Pour identifier ses extensions, on projette BDS sur A. Le résultat de la projection est appelée* BDS'_A, *il est illustrée dans le tableau 4.8a. Les séquences de* BDS'_A *représentent les évènements de* S_1 *et* S_2 *qui se trouvent après A. La première séquence de* BDS'_A *(respectivement la deuxième) contient les évènements B et C qui apparaissent après A dans* S_1 *respectivement dans* S_2.

Maintenant, on calcule les 1-STI fréquentes dans BDS'_A *pour étendre* $\langle([0,0]A)\rangle$. *On identifie,* $\langle([1,2]B)\rangle$ *et* $\langle[1,2]C)\rangle$ *comme fréquentes.*

On choisit arbitrairement de traiter $\langle([1,2]B)\rangle$ *en premier. Sa concaténation à* $\langle([0,0]A)\rangle$ *fournit* $\langle([0,0]A)([1,2]B)\rangle$. *Pour identifier les extensions du dernier motif extrait on projette* BDS'_A *sur* $([1,2]B)$. *Le résultat de la projection est représenté dans le tableau 4.8b, il est appelée* $BDS'_{A,B}$. *La première séquence dans cet espace contient la transaction* $(1,C)$, *car dans* S_1 , *l'évènement de cette transaction apparait une unité temporelle après B. La deuxième séquence de* $BDS'_{A,B}$ *est vide puisque dans* S_2, *B est le dernier évènement. Dans* $BDS'_{A,B}$ *aucun évènement n'est fréquent, l'extension du motif n'est donc pas possible.*

Reprenons maintenant l'extension de $\langle([0,0]A)\rangle$ *avec* $\langle[1,2]C)\rangle$, *on obtient le motif :* $\langle([0,0]A)([1,2]C)\rangle$.

La projection de BDS'_A par cette 1-STI fournit la base $BDS'_{A,C}$ illustrée dans le tableau 4.8c. Cette base ne contient aucun évènement fréquent, l'extension du motif $\langle([0,0]A)([1,2]C)\rangle$ n'est donc pas possible.

Pour extraire les *STI*, la projection doit permettre de considérer comme continuation d'un motif les évènements qui peuvent présenter un désordre local avec les évènements de sa dernière transaction. Pour être local, ce désordre doit se situer au plus sur une distance temporelle égale à la taille de la fenêtre.

Si on considère l'exemple précédent, la projection classique ne prend en compte que les sous-séquences en amont du dernier évènement du motif extrait. Elle n'autorise donc pas la sélection d'une continuation avec un désordre local.

Lorsqu'une séquence est étendue par une 1-STI, il existe deux possibilités de concaténations :

- La première est une extension de la dernière transaction de la séquence. L'évènement de la 1-STI est ajouté aux évènements de cette transaction et les deux intervalles de la STI et de la dernière transaction de la séquence sont fusionnés.
- La seconde concatène une transaction contenant un seul évènement à la séquence. La 1-STI devient sa dernière transaction.

Exemple 28. *Soit $S = \langle([0,1]A)([2,3]B)\rangle$ qu'on veut étendre par $S' = \langle([4,5]C)\rangle$. Ceci peut se faire de deux manières ; (i) On fait juste la concaténation et on obtient $\langle([0,1]A)([2,3]B)([4,5]C)\rangle$ ou bien (ii) On « fusionne » la dernière transaction de S avec $([4,5]C)$ et on obtient $\langle([0,1]A)([2,5]BC)\rangle$. Bien sûr, cette fusion ne peut se faire que si les deux évènements (celui de S et celui qu'on veut a jouter) sont temporellement « proches ».*

La définition suivante introduit ces deux types d'extensions à travers les opérateurs $\oplus_T$ et $\oplus_S$.

Définition 14. *Soient $S = \langle([m_1, M_1]I_1)\dots([m_n, M_n]I_n)\rangle$ et $S' = \langle([m_i, M_i]I)\rangle$.*

- *La* T-extension *de S par S' est définie par :*

$$S \oplus_T S' = \langle([m_1, M_1]I_1)\dots([m_{n-1}, M_{n-1}]I_{n-1})([m'_n, M'_n]I'_n)\rangle$$

$$où\ m'_n = min(m_n, m_i)\ ,\ M'_n = max(M_n, M_i) et\ I'_n = I \cup I_n$$

- *La* S-extension *de S par S' est définie par :*

$$S \oplus_S S' = \langle([m_1, M_1]I_1)\dots([m_n, M_n]I_n)([m_i, M_i]I)\rangle$$

Exemple 29. *On reprend l'exemple précédent. Ainsi, $S \oplus_T S' = \langle([0,1]A)([2,5]BC)\rangle$ et $S \oplus_S S' = \langle([0,1]A)([2,3]B)([4,5]C)\rangle$.*

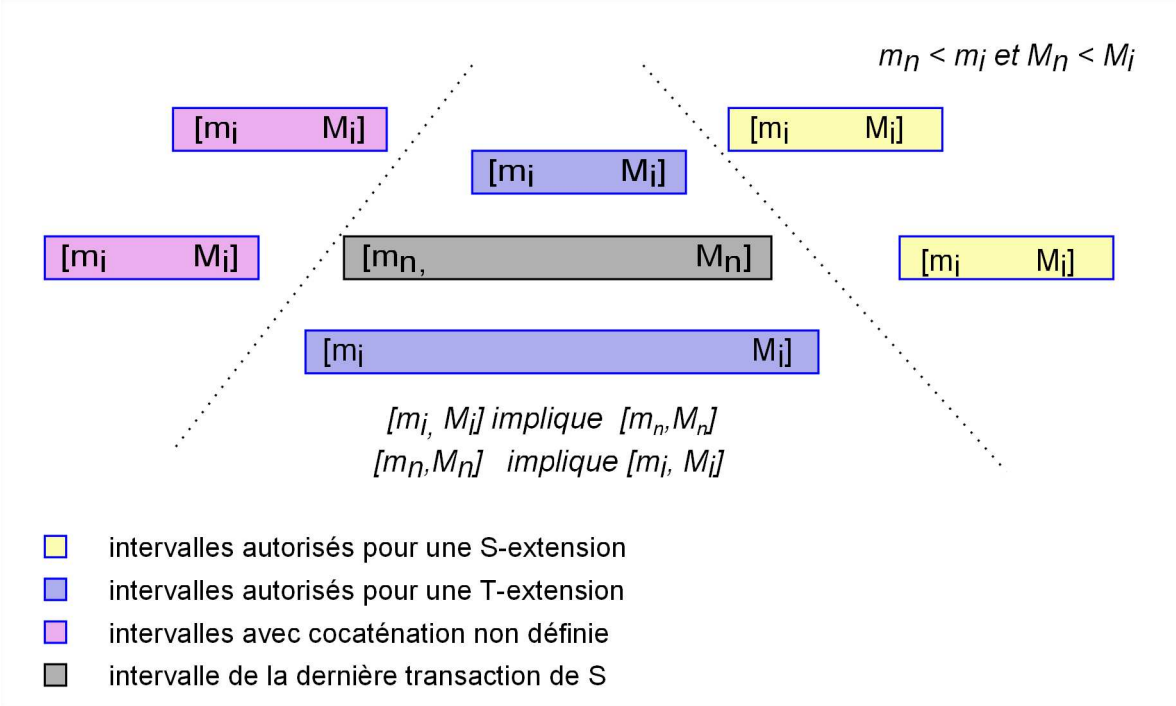


FIGURE 4.3 – Schématisation de la relation entre le dernier intervalle de S et celui de S' et opérateur de concaténation

Notre approche, basée sur les intervalles d'incertitudes, privilégie la fusion quand il est possible de la faire. Ainsi, une concaténation sera traduite par une *T-extension* quand c'est possible, sinon elle le sera par une *S-extension*. Nous définissons plus formellement l'opérateur de concaténation $\oplus$, comme suit :

Définition 15. *Considérons ws,* $S = \langle([m_1, M_1]I_1)\dots([m_n, M_n]I_n)\rangle$ *et* $S' = ([m_i, M_i]I)$ *qui respectent la contrainte ws. La concaténation de* S *avec* S' *est définie par :*

$$S \oplus S' = \begin{cases} S \oplus_T S' & si\ [m_i, M_i]\ implique\ [m_n, M_n] \\ & ou [m_n, M_n] implique [m_i, M_i] \\ S \oplus_S S' & si\ m_n \leq m_i\ et\ M_n \leq M_i \\ S & sinon \end{cases}$$

La figure 4.3 illustre la nature de la relation de concaténation entre S et S' selon le positionnement du dernier intervalle de S et celui de S'.

Exemple 30. *Reprenons l'exemple 28 et considérons* $ws = 2$. $S \oplus_S S' = \langle([0,1]A)\ ([2,3]B)([4,5]C)\rangle$ *est une séquence cohérente car ws est vérifiée. Considérons* $S'' = ([2,4]D)$, $S \oplus_T S'' = \langle([0,1]A)$ $([2,4]BD)\rangle$.

Nous définissons dans ce qui suit la notion de préfixe et de suffixe d'une séquence par rapport à une autre.

Intuitivement, les préfixes de S par rapport à S' sont des sous-séquences de S qui commencent au début de S et dont la dernière transaction contient S'.

Définition 16 (Préfixe). *Soient* $S = \langle (t_1, I_1) \dots (t_n, I_n) \rangle$ *et* $S' = \langle ([m, M], I) \rangle$. *La sous séquence* $\langle (t_1, I_1), (t_2, I_2) \dots (t_j, I_j) \rangle$ *est un Préfixe de* S *par rapport à* S' *si et seulement si* $I_j \supseteq I$ *et* $t_j \in [m, M]$.

Notons par Préfixe(S,S') l'ensemble des préfixes de S *par rapport à* S'. *La figure 4.4 est une schématisation du préfixe de* S *par rapport à* S'.

Exemple 31. *Considérons la séquence temporelle* $S = \langle (0, A)(1, C)(2, B)(4, D) \rangle$ *et la 1-STI* $S' = ([1, 2]B)$. *Le préfixe de* S *par rapport à* S' *est* $\langle (0, A)(1, C)(2, B) \rangle$. $Prfixe(S, S') = \{\langle (0, A)(1, C)(2, B) \rangle\}$, *cet ensemble contient un seul élément car* S' *apparait une seule foi dans* S.

Classiquement, le suffixe permet de définir les continuations en aval d'une séquence. Cependant, vu que nous utilisons des estampilles par intervalles, nous avons besoin d'introduire des continuations en amont. Ainsi, nous redéfinissons la notion de suffixe pour l'adapter à notre contexte.

Un suffixe de S par rapport à S' est une sous-séquences de S qui contient les éventuelles continuations de S' dans S. Il contient donc des évènements dont la concaténation avec S' permet de construire des séquences cohérentes. Ils représentent ses éventuelles *T-extensions* ou *S-extensions* telles que :

- Les évènements qui représentent les éventuelles *T-extension* de S' sont « proches » des évènements qui constituent sa dernière transaction. Ils se trouvent donc dans une distance temporelle égale à ws autour de ces évènements.
- Les évènements qui représentent les éventuelles *S-extension* de la séquence se trouvent « après » ceux de sa dernière transaction. Ils apparaissent, pour la plupart, dans S à une distance supérieure à ws de l'apparition du dernier évènement S'.

Notons que les évènements faisant partie de S ne peuvent faire partie de son extension.

Au vu de la définition de $\oplus$, nous pouvons noter que la concaténation se traduit par une fusion ($\oplus_T$ extension) quand il y a « proximité » entre la dernière transaction et celle qu'on veut adjoindre. Ainsi, étant donnée une séquence S' incluse dans S on peut définir une « zone » dans laquelle se trouvent les transactions qui peuvent donner lieu à une *T-extension* de S'. En effet,

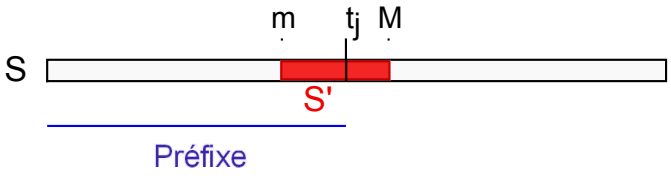


FIGURE 4.4 – Illustration graphique d'un Préfixe de S par rapport à S'

il suffit de prendre celles qui sont à une distance ws de part et d'autre de la dernière transaction de S'. Nous appelons cet ensemble de transactions la « zone T-extension ».

Définition 17. *Soient* $S = \langle (t_1, I_1) \dots (t_n, I_n) \rangle$, $S' = ([m, M]I)$ *et* ws. *Pour* $j \in [1, n]$ *tel que* $t_j \in [m, M]$ *et* $I \subseteq I_j$, *la zone T-extension relative à* j *est définie par :*

$$\{(t_i, I_i) \in S | \exists j : \ t_j \in [m, M], \ I \subseteq I_j et \mid t_i - t_j \mid \leq ws\}$$

Notons par $\delta_{ws}(S, S')$ *l'ensemble des zones des* T-extensions S' *dans* S.

Exemple 32. *Considérons la séquence temporelle* $S = \langle (0, A)(1, C)(2, B)(4, D) \rangle$ *et* $S' = ([1, 2]B)$. *La zone T-extensions de* S' *dans* S *pour* $ws = 1$ *est* $\{(1, C)\}$. *Effectivement la transaction* $(1, C)$ *se trouve à une unité temporelle avant* S' *dans* S *et* $([1, 1]C)$ *est une* T-extension *de* S' *car* $[1, 2]$ *implique* $[2, 2]$.

$\delta_2(S, S') = \{(1, C)\}$ *car il existe une seule occurrence de* S' *dans* S.

Le suffixe d'une séquence S par rapport à S' contenue dans S est l'union des évènements qui représentent les éventuelles *T-extensions* et de ceux qui représentent les *S-extension* de S' dans S. La zone *T-extension* contient tous les évènements qui peuvent être fusionnés dans la dernière transaction de S', ainsi que certaines de ses *S-extensions*. Ces dernières sont les évènements « éloignés » en aval des évènements de S'. Ils sont donc aussi présents dans la sous-séquence de S qui succède à la zone *T-extension*. La figure 4.5 illustre la notion de Préfixe, de suffixe et de δ_{ws}. Nous formalisons maintenant la définition du suffixe de S par rapport à S'.

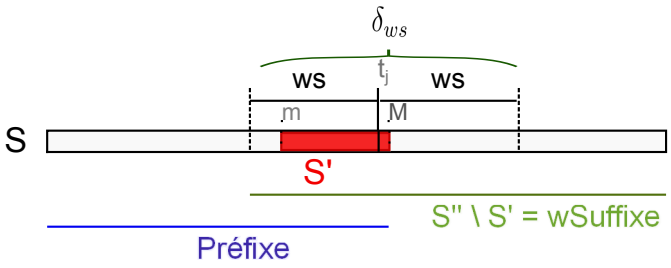


FIGURE 4.5 – Illustration du Préfixe de δ_{ws} et du suffixe de S par rapport à S'

Définition 18 (**wSuffixe**). *Soient* $S = \langle (t_1, I_1) \dots (t_n, I_n) \rangle$ *et* $S' = \langle ([m, M], I) \rangle$.

- *Pour* $j \in [1, n]$ *avec* $I \subseteq I_j$ *et* $t_j \in [m, M]$, *la sous-séquence* $\langle (t'_k, I_k) \dots (t'_j, I_j \backslash \{I\}) \dots (t'_n, I_n) \rangle$ *est un suffixe de* S *par rapport à* S' *si et seulement si :*

  1. $\forall i,\ k \leq i \leq n\ t'_i = t_i - t_j$ *et*

  2. $t'_k \leqslant (t'_j - ws)$ *et* $t'_{k-1} > (t_j - ws)$

- *Sinon, la sous-séquence vide* $\langle \emptyset \rangle$ *est le suffixe de* S *par rapport à* S'.

Notons que la référence temporelle d'un suffixe de S *par rapport à* S' *est l'estampille de la transaction* $I_j \supseteq I$.

$wsuffixe(S, S')$ [1] *dénote l'ensemble des suffixes de* S *par rapport à* S'.

BDS_A	
S_1	$\langle (1, B)(2, C) \rangle$
S_2	$\langle (1, C)(2, B) \rangle$

Tableau 4.9 – Base de séquences BDS_A

Exemple 33. *Considérons la base de séquences* BDS_A *décrite dans le tableau 4.9.*

L'ensemble des suffixes de S_1 *par rapport à* S' *est* $suffixe(S_1, S') = \{\langle (1, C) \rangle\}$. *Cet ensemble contient un seul élément, car* S' *apparait une seule foi dans* S_1. *La référence temporelle de la séquence suffixe est modifiée pour que les estampilles de ses évènements soit relative à ceux de* S'. *Le suffixe véhicule donc l'information suivante :* C *a lieu une unité temporelle après* B.

De la même manière S' *apparait une seule foi dans* S_2 *et* $prfixe(S_2, S') = \{\langle (1, C)(2, B) \rangle\}$. $Suffixe(S_2, S') = \{\langle (-1, C) \rangle\}$, *car dans* S_2, C *apparait une unité temporelle avant* S'.

Nous définissons maintenant la projection. Elle détermine pour une base de séquences BDS et une 1-STI S', qui y est fréquente, l'ensemble de toutes les continuations de S' dans les séquences de BDS. Concrètement, elle calcule l'ensemble des wsuffixes pour chaque séquence S par rapport à S'.

Définition 19 (**wprojection**). *Soit* BDS *une base de séquences et* $S' = ([m, M]I)$ *une 1-STI fréquente dans* BDS, *la wprojection* [2] *de* BDS *par rapport à* S' *est définie par :*

$$wprojection(BDS|S') = \{S''|S'' = wsuffixe(S, S'),\ S \in BDS\}$$

1. wSuffixe pour ws-suffixe : le suffixe qui prend en compte un retour arrière géré par une taille de fenêtre ws.
2. wprojection pour ws-projection : la projection qui considère les wSuffixes

Au vu de la définition de la *wprojection*, nous introduisons une variante de l'algorithme *STI-PS* qui calcule un résumé de la base de séquences en appliquant *wprojection* et identifie, par la suite, des motifs plus longs. Nous appelons cette variante de *STI-PS* $Algo_1$. L'exemple suivant illustre les étapes d'extraction et le déroulement de son application sur la base de séquences BDS utilisée dans l'exemple 4.2.

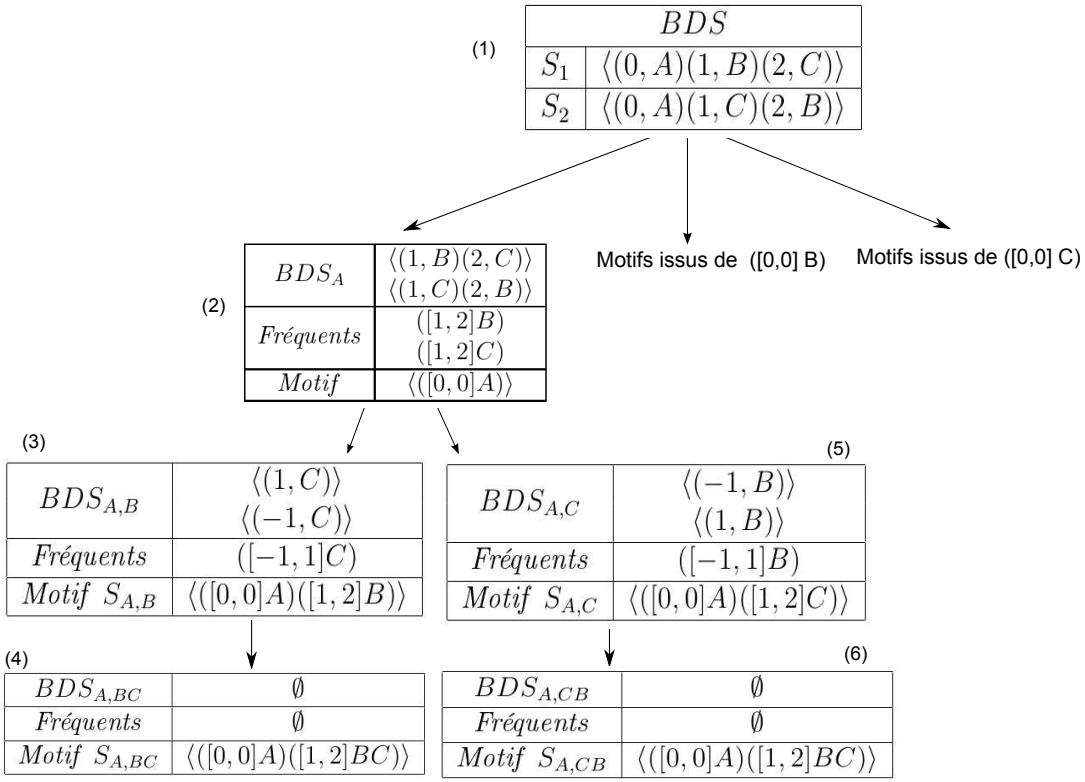


FIGURE 4.6 – Exemple de la structure de déroulement de $STI-PS$ en appliquant *wprojection*.

Exemple 34. *Soit BDS la base de séquences décrite dans le tableau (1) de la figure 4.6. Nous déroulons l'algorithme* $Algo_1$ *avec* $minsupp = 2$ *et* $ws = 2$. *Dans BDS, les évènements fréquents sont A, B et C.*

Nous nous concentrons dans cet exemple sur l'extraction des motifs issus de l'extension de la 1-STI fréquente $\langle([0, 0]A)\rangle$. *Pour cela nous détaillons les étapes de la branche d'extraction correspondante. La figure 4.6 illustre les résultats des différentes étapes d'extraction.*

D'abord, nous projetons BDS sur $\langle([0, 0]A)\rangle$. *La base résultat, que nous appelons* BDS_A *est décrite dans le tableau (2) de la figure. Ses deux séquences représentent respectivement les suffixes de* S_1 *et de* S_2 *par rapport à* $\langle([0, 0]A)\rangle$. *Toutes deux ne comportent pas de retour arrière car A apparaît dès la première transaction dans les deux séquences.*

Par la suite, dans BDS_A, *les 1-STI* $\langle([1, 2]B)\rangle$ *et* $\langle([1, 2]C)\rangle$ *sont identifiées comme fréquentes.*

Ce sont les extensions de $\langle([0,0]A)\rangle$.

Nous traitons d'abord $\langle([1,2]B)\rangle$. *Il s'agit d'une* S-extension *puisque* $0 < 1$ *et* $0 < 2$. *Elle fournit le motif* $S_{A,B} = \langle([0,0]A)\rangle \oplus \langle([1,2]B)\rangle = \langle([0,0]A)\rangle \oplus_T \langle([1,2]B)\rangle = \langle([0,0]A)([1,2]B)\rangle$.

Une nouvelle extraction est lancée à la découverte des extensions de ce dernier motif. Nous projetons BDS_A *sur* $\langle([1,2]B)\rangle$. *La base résultat que nous appelons* $BDS_{A,B}$ *est illustrée dans le tableau (3) de la figure 4.6. Dans cette nouvelle projection,* $\langle([-1,1]C)\rangle$ *est fréquente.*

Pour étendre $S_{A,B}$ *avec cette 1-séquence, les deux éléments doivent avoir la même référence temporelle. Or, l'intervalle associé à* C *est référencé par rapport à* B, *alors que les intervalles de* $S_{A,B}$ *le sont par rapport à* A. *Avant de concaténer il faut référencer l'intervalle de* $\langle([-1,1]C)\rangle$ *par rapport à* A. C *apparait au plus tôt une unité temporelle avant l'apparition au plus tard de* B *(borne inférieure de l'intervalle de* B*), la borne inférieur de l'intervalle associé à* C *référencée par rapport à* A *est égale à* $2 + (-1) = 1$. C *apparaît au plus tard une unité temporelle après l'apparition au plus tôt de* B *(borne supérieure de l'intervalle de* B*), la borne supérieure de l'intervalle associé à* C, *et référencée par rapport à* A, *est égale à* $1 + 1 = 2$.

On obtient alors $\langle([1,2]C)\rangle$. *Ainsi,* $S_{A,B} \oplus_T \langle([1,2]C)\rangle = \langle([0,0]A)([1,2]BC)\rangle$ *que nous appelons* S_{ABC}. *La* $wprojection(BDS_{A,B}, \langle([-1,1]C)\rangle) = \{\emptyset\}$ *représentée dans le tableau (4) de la figure 4.6. L'extension de* $S_{A,B,C}$ *n'est donc pas possible.*

L'extraction remonte alors de deux niveaux puisque $S_{A,B}$ *n'est pas non plus extensible. Nous étendons donc* $\langle([0,0]A)\rangle$ *avec* $\langle([1,2]C)\rangle$ *pour obtenir* $S_{A,C}$. *La projection de* BDS_A *sur* $\langle([1,2]C)\rangle$ *est appelée* $BDS_{A,C}$. *Elle est décrite dans le tableau (5) de la figure 4.6. Cette nouvelle base fournit la 1-STI fréquente* $\langle([-1,1]B)\rangle$, *sa concaténation avec* $S_{A,C}$ *fournit* $S_{A,C,B} = S_{A,C} \oplus_T \langle([1,2]B)\rangle = \langle([0,0]A)([1,2]BC)\rangle$. *La projection de* $BDS_{A,C}$ *sur* $\langle([-1,1]B)\rangle$ *est vide (tableau (6) de la figure) et l'extension de* $S_{A,C,B}$ *n'est pas possible.*

Les motifs extraits par extension de $\langle([0,0]A)\rangle$ *sont tous identifiés. L'exaction continue pour extraire les motifs qui étendent* $\langle([0,0]B)\rangle$ *et* $\langle([0,0]C)\rangle$.

On peut remarquer que les deux sous-branches déroulées dans cet exemple fournissent le même motif fréquent $\langle([0,0]A)([1,2]BC)\rangle$.

Dans l'exemple précédent, le retour arrière de la projection a permis d'identifier les occurrences alternées et rapprochées de B et de C comme étant fréquentes. Cependant, dans les résultats fournis, le motif $\langle([0,0]A)([1,2]BC)\rangle$ est extrait deux fois. La première suite aux extensions $([0,0]A) \oplus_S ([1,2]B) \oplus_T ([1,2]C)$. La seconde fois, suite à la succession d'extensions $([0,0]A) \oplus_S ([1,2]C) \oplus_T ([1,2]B)$. Effectivement, la fusion de $\langle([1,2]C)\rangle$ avec la dernière transaction de $\langle([0,0]A)([1,2]B)\rangle$ revient à fusionner $\langle([1,2]B)\rangle$ avec la dernière transaction de $\langle([0,0]A)([1,2]C)\rangle$ car au sein d'une transaction les évènements ne sont pas ordonnés.

Plus généralement, une séquence $S = \langle([m_1, M_1]I_1)\dots([m_{n-1}, M_{n-1}]I_{n-1})([m_n, M_n]I_n)\rangle$, où $I_n = \{e_1 \dots e_p\}$, est obtenue par $\langle([m_1, M_1]I_1)\dots([m_{n-1}, M_{n-1}]I_{n-1}) \oplus_S ([m_1, M_1]e_1) \oplus_T \cdots \oplus_T ([m_p, M_p]e_p)$. C'est la succession d'une *S-extension* avec $(N-2)$ *T-extensions*. L'ordre dans lequel les *T-extensions* sont effectuées ne change pas la séquence finale, car une transaction est une ensemble d'évènements. De plus, la *T-extension* associe à la « nouvelle » dernière transaction de S la fusion des intervalles extensions.

Lemme 1. *Soient* $S' = \langle([m_1, M_1]I_1)\dots([m_n, M_n]I_n)\rangle$, ws *et* $\alpha = \{([m_1, M_1]I_1), \dots, ([m_p, M_p]I_p)\}$ *l'ensemble des T-extensions de S. Soit* $m = min(m_1 \dots m_p)$ *et* $M = max(M_1 \dots M_p)$ *avec* $M - m \leq ws$. *Les motifs construits par concaténation successives « dans n'importe quel ordre » de tous les évènements de* α *sont équivalents.*

Nous allons montrer ce lemme pour le cas où α contient deux évènements fréquents de type T-extension. Par la suite le traitement un à un des évènements de α permet de généraliser la preuve à n T-extensions

Preuve. Soient $S_p = ([m_p, M_p]I_p)$ et $S_k = ([m_k, M_k]I_k)$ des T-extensions de S. Ces deux évènements sont proches de la dernière transaction de S et sont aussi proches entre eux. Nous allons étudier dans cette preuve les résultats des deux doubles extensions de S par les deux STIs : $S \oplus ([m_p, M_p]I_p) \oplus ([m_k, M_k]I_k) = S \oplus_T ([m_p, M_p]I_p) \oplus_T ([m_k, M_k]I_k)$ et $S \oplus_T ([m_k, M_k]I_k) \oplus_T ([m_p, M_p]I_p)$

- La première possibilité d'extension est d'abord une extension de S par $S_p = ([m_p, M_p]I_p)$ et ensuite une T-extension du résultat par S_k tel que :

$$S \oplus S_p = S \oplus_T S_p = \langle([m_1, M_1]I_1)\dots([m'_n, M'_n]I'_n)$$

avec

$$\begin{cases} m'_n = min(m_n, m_p) \\ M'_n = max(M_n, M_p) \\ I'_n = I_n \cup \{I_p\} \end{cases}$$

La concaténation du résultat avec S_k est une T-extension qui fournit la séquence suivante :

$$S \oplus S_p \oplus S_k = S \oplus_T S_p \oplus_T S_k = \langle([m_1, M_1]I_1)\dots([m''_n, M''_n]I''_n)$$

avec

$$\begin{cases} m''_n = min(m'_n, m_k) = min(m_n, m_p, m_k) \\ M''_n = max(M'_n, M_k) = max(M_n, M_p, M_k) \\ I''_n = I'_n \cup \{e_k\} = I_n \cup \{I_p, I_k\} \end{cases}$$

– La deuxième est d'abord une extension de S par S_k, et ensuite une T-extension du résultat par S_p tel que :

$$S \oplus S_k = S \oplus_T S_k = \langle([m_1, M_1]I_1) \dots ([\acute{m}_n, \acute{M}_n]\acute{I}_n)$$

avec

$$\begin{cases} \acute{m}_n = min(m_n, m_k) \\ \acute{M}_n = max(M_n, M_k) \\ \acute{I}_n = I_n \cup \{I_k\} \end{cases}$$

La concaténation du résultat avec S_p est aussi une T-extension qui fournit la séquence suivante :

$$S \oplus S_k \oplus S_p = S \oplus_T S_k \oplus_T S_p = \langle([m_1, M_1]I_1) \dots ([\grave{m}_n, \grave{M}_n]\grave{I}_n)$$

avec

$$\begin{cases} \grave{m}_n = min(\acute{m}_n, m_k) = min(m_n, m_p, m_k) \\ \grave{M}_n = max(\acute{M}_n, M_k) = max(M_n, M_p, M_k) \\ \grave{I}_n = \acute{I}_n \cup \{e_p\} = I_n \cup \{I_p, I_k\} \end{cases}$$

Nous pouvons facilement déduire que $S \oplus_T S_p \oplus_T S_k = S \oplus_T S_k \oplus_T S_p$. Nous concluons donc que les concaténations successive à une même séquence S de deux évènement avec un ordre alternés fournit la même séquence résultats.

Notons maintenant par S_1 la séquence résultat, et considérons deux autres *T-extension* $S_u = ([m_u, M_u]e_u)$ et $S_i([m_v, M_v]I_v)$ tels que $\{S_v, S_u\} \in \alpha$. Si on étend S_1 de la même manière deux fois la séquence en lui concaténant d'abord S_u et en suite S_v, on obtient alors le même résultat que lorsque on applique l'extension $S \oplus S_v \oplus S_u = S \oplus_T S_v \oplus_T S_u$.

On peut finalement conclure que quelque soit le nombre des *T-extensions*, si on étend la dernière transaction d'une séquence S avec le même ensemble de *T-extensions* en considérant des ordres différents on obtient toujours le même résultat.

□

En appliquant ce type d'extraction, $Algo_1$ extrait certaines STI fréquentes plusieurs fois. Effectivement, lorsque la dernière transaction d'un motif contient N évènements. Ce dernier est obtenu 2^N fois en appliquant a chaque fois un nouvel ordre des évènements pour les T-extensions. Cette redondance dans la construction d'une STI diminue l'efficacité de l'algorithme et augmente son temps de calcul.

Nous proposons dans la suite une solution qui permet de contourner cette problématique. L'idée consiste à ne considérer parmi les T-extensions d'un motif $S \oplus_T S'$ que celle qui n'ont pas déjà été concaténée directement à S.

Concrètement, pour une séquence S et $\alpha = \{S_1 \dots S_N\}$ l'ensemble de ses T-extensions, l'exploration des continuation $S \oplus S_1$ génère une branche d'extension qui au final construit le motif $S \oplus_T S_1 \oplus_T S_2 \cdots \oplus_T S_N$. Alors, lorsque S sera étendu avec $S \oplus_T S_2$, S_1 ne sera pas une extension possible de cette dernière séquence. La branche d'extraction générée par $S \oplus_T S_2$ permet d'identifier $S \oplus_T S_2 \oplus_T S_3 \dots S_N$ comme le motif le plus long de cette branche.

Afin de mettre en place cette solution, nous avons besoins d'établir un ordre de traitement des *T-extensions*. Nous avons choisit de définir $\triangleleft$ qui introduit l'ordre entre les évènements tel que $e_i \triangleleft e_j$ si et seulement si e_i est inférieur à e_j.

Définition 20 (opérateur d'ordre $\triangleleft$)**.** *Soit $\omega = \{e_1, e_2 \dots e_N\}$, nous définissons $\triangleleft$ tel que : $\forall i \leq j$, $e_i \triangleleft e_j$ si et seulement si e_i est inférieur à e_j.*

Propriété 5. *L'opérateur $\triangleleft$ a les propriétés suivantes :*

- *Si $e_i \triangleleft e_j$ et $e_j \triangleleft e_k$ alors $e_i \triangleleft e_k$.*
- *Pour $I = \{e_{j1}, e_{j2} \dots e_{jK}\}$, $e_i \triangleleft I$ si et seulement si $\forall e_j \in I$ $e_i \triangleleft e_j$.*
- *$I_1 \triangleleft I_2$ si et seulement si $\exists e_j \in I_2$ tel que $\forall e_i \in I_1$ $e_i \triangleleft e_j$.*

Exemple 35. *Soit $\omega = \{A, B, C, D, E, F\}$ et $\triangleleft$ qui défini l'ordre entre les éléments de ω tel que $A \triangleleft B \triangleleft C \triangleleft D \triangleleft E \triangleleft F$. Alors on a $A \triangleleft EF$*

Nous utilisons cet opérateur d'ordre pour redéfinir la zone T-extension d'une séquence S' dans une séquence S.

Afin d'éviter les constructions multiples d'une même séquence S', nous définissons une nouvelle la zone T-extensions afin qu'elle représente l'ensemble des transactions de S qui se trouvent à une distance ws de la dernière transaction de S'. Les évènements qui sont inférieur à la transaction de S' sont enlevés de cette zone.

Définition 21 ($\delta_{ws\triangleleft}$)**.** [3] *Soient $\omega = \{e_1 \dots e_N\}$ et $\triangleleft$ l'ordre entre les évènement de ω. Considérons $S = \langle (t_1, I_1) \dots (t_n, I_n) \rangle$, $S' = ([m, M]I)$ et ws. Pour $j \in [1, n]$ tel que $t_j \in [m, M]$ et*

3. $\delta_{ws\triangleleft}$ la zone T-extension qui s'étend sur une distance temporelle égale à ws de part et d'autre de la séquence S' et prend en compte l'ordre définit par $\triangleleft$

$I \subseteq I_t$, la zone T-extension est définie par :

$$\{(t_i, I'_i) | (t_i, I_i) \in S;\ \exists j : t_j \in [m, M], I \subseteq I_j \ et\ \mid t_i - t_j \mid \leq ws \ avec\ I'_i = I_i \setminus \{e_u | e_u \triangleleft I\}\}$$

*Notons par $\delta_{ws}(S, S')$ l'ensemble des zones des*T-extensions *S' dans S.*

Exemple 36. *Considérons la séquence temporelle $S = \langle (0, A)(1, B)(2, C)(4, D)\rangle$, $S' = ([1, 2]B)$ et $\triangleleft$ l'ordre lexicographique entre les évènements de S. $\delta_{2\triangleleft}(S, S') = \{(1, C)\}$. A est supprimé de l'espace $\delta_{2\triangleleft}(S, S')$car il est inférieur à B.*

Maintenant, nous définissons $wsufixe_{\triangleleft}$ l'ensemble des suffixes d'une séquence S par rapport à S' qui prend en compte $\delta_{ws}(S, S')$. Intuitivement un suffixe de S par rapport de S' est l'ensemble des *T-extensions* et des *S-extensions* de S' dans S. Les T-extensions étant redéfinis pour ne considérer que les transactions proches de la dernière transaction de S' et dont les évènements sont supérieurs à ceux de cette dernière (relativement à l'opérateur $\triangleleft$).

Définition 22 *($wsufixe_{\triangleleft}$). Soient un ensemble d'évènements $\omega = \{e_1, e_2 \dots e_m\}$, l'opérateur $\triangleleft$ qui établit l'ordre entre ces éléments, $S = \langle (t_1, I_1) \dots (t_n, I_n)\rangle$ et $S' = \langle ([m, M], I)\rangle$.*

- *Pour j $(1 \leq j \leq n)$ tel que $I \in I_j$ et $t_j \in [m, M]$ $\langle (t'_k, I'_k) \dots (t'_j, I'_j \setminus \{e_r\}) \dots (t'_n, I'_n)\rangle$ est un suffixe de S par rapport à S' si et seulement si :*
  1. *$\forall i,\ k \leq i \leq n,\ t'_i = t_i - t_j$*
  2. *$t'_k \leqslant (t'_j - ws)$ et $t'_{k-1} > (t_j - ws)$*
  3. *$\forall i,\ k \leq i \leq n$ si $t'_i \leq ws$alors$I'_i = I_i \setminus \{e_u | e_u \triangleleft I\}$*
- *Sinon, la sous-séquence vide $\langle \emptyset\rangle$ est le suffixe de S par rapport à S'.*

Notons par $wsuffixe_{\triangleleft}(S_1, S')$ [4] l'ensemble des suffixes de S par rapport à S'.

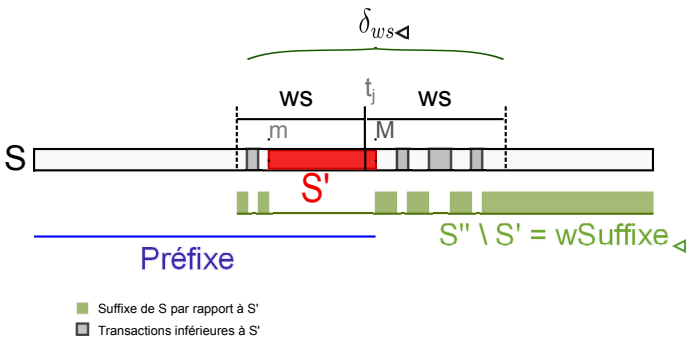


FIGURE 4.7 – Illustration de δ_{ws} et du $wsuffixe_{\triangleleft}$ de S par rapport à S'

4. $wsuffixe_{\triangleleft}$: pour ws-suffixe qui applique l'opérateur d'ordre $\triangleleft$

La figure 4.7 est une illustration graphique de $\delta_{ws\triangleleft}$ et du $wsuffixe_{\triangleleft}$ de S par rapport à S'.

Exemple 37. *Considérons la séquence* $S_1 = \langle(0,B)(1,C)(2,A)(3,D)\rangle$*, l'ordre lexicographique suivant entre les évènements de* S_1 *tel que* $A \triangleleft B \triangleleft C \triangleleft D$ *, une taille de fenêtre* $ws = 2$*,* $S' = \langle([1,2]A)\rangle$ *et* $S'' = \langle([1,2]C)\rangle$*.* $wsuffixe_{\triangleleft}(S_1, S') = \{\langle(-2,B)(-1,C)(1,D)\rangle\}$*, les évènements* B *et* C *apparaissent dans le suffixe de* S_1 *par rapport à* S' *malgré qu'ils en soient proches car* B *et* C *sont supérieur à* A*.*

Dans, $wsuffixe_{\triangleleft}(S_1, S'') = \langle(2,D)\rangle$*, le retour arrière de la projection ne fournit aucun évènement car tous ceux qui sont en « amont proches » de* C*, en sont inférieurs. Parmi les évènements en aval de* C*,* A *ne fait pas partie du suffixe car il est proche et inférieur à* C*.*

Nous pouvons maintenant définir la $wprojection_{\triangleleft}$ qui calcule pour une base de séquences sa projection par rapport à une séquence S' en prenant en compte le $wsuffixe_{\triangleleft}$.

Définition 23 ($wprojection_{\triangleleft}$)**.** *Soit BDS une base de séquences et* $S' = ([m,M]I)$*. Nous définissons la projection* $wprojection_{\triangleleft}$ [5] *de BDS sur SI par :*

$$wprojection_{\triangleleft}(BDS|S') = \{S''|S'' = wsuffixe_{\triangleleft}(S,S'),\ S \in BDS\}$$

Nous définissons une deuxième variante de l'algorithme *STI-PS* qui applique la projection $wprojection_{\triangleleft}$ (définition 23) que nous appelons $Algo_2$. Pour étendre un motif fréquent, $Algo_2$ traite les 1-STI fréquente considérée comme l'extension du motif selon l'ordre défini par l'opérateur $\triangleleft$. Afin de concrétiser les étapes d'extraction effectuées par $Algo_2$ nous présentons un exemple de son application sur la base de séquences de l'exemple 4.2.

Exemple 38. *Reprenons la base de séquences de l'exemple BDS décrite dans le tableau (1) de la figure 4.8, on y applique l'algorithme* $Algo_2$ *avec* $minsupp = 2$*,* $ws = 2$ *et l'opérateur d'ordre assimilé à l'ordre lexicographique des évènements de BDS.*

Comme pour l'exemple 34, nous nous focalisons sur l'extraction des motifs qui étendent $\langle([0,0]A)\rangle$*, les résultats des étapes d'extraction le long de cette branche sont illustrés dans la figure 4.8.*

D'abord, nous projetons BDS sur $\langle([0,0]A)\rangle$*. Le résultat est appelé* $BDS_{A\triangleleft}$ *et est illustré dans le tableau (2) de la figure 4.8.* $\langle([0,0]A)\rangle$ *étant le premier motif exploré la* $wprojection_{\triangleleft}$ *de BDS est égale à celle de wprojection.*

5. $wprojection_{\triangleleft}$: pour ws- projection qui applique l'opérateur d'ordre entre évènements $\triangleleft$

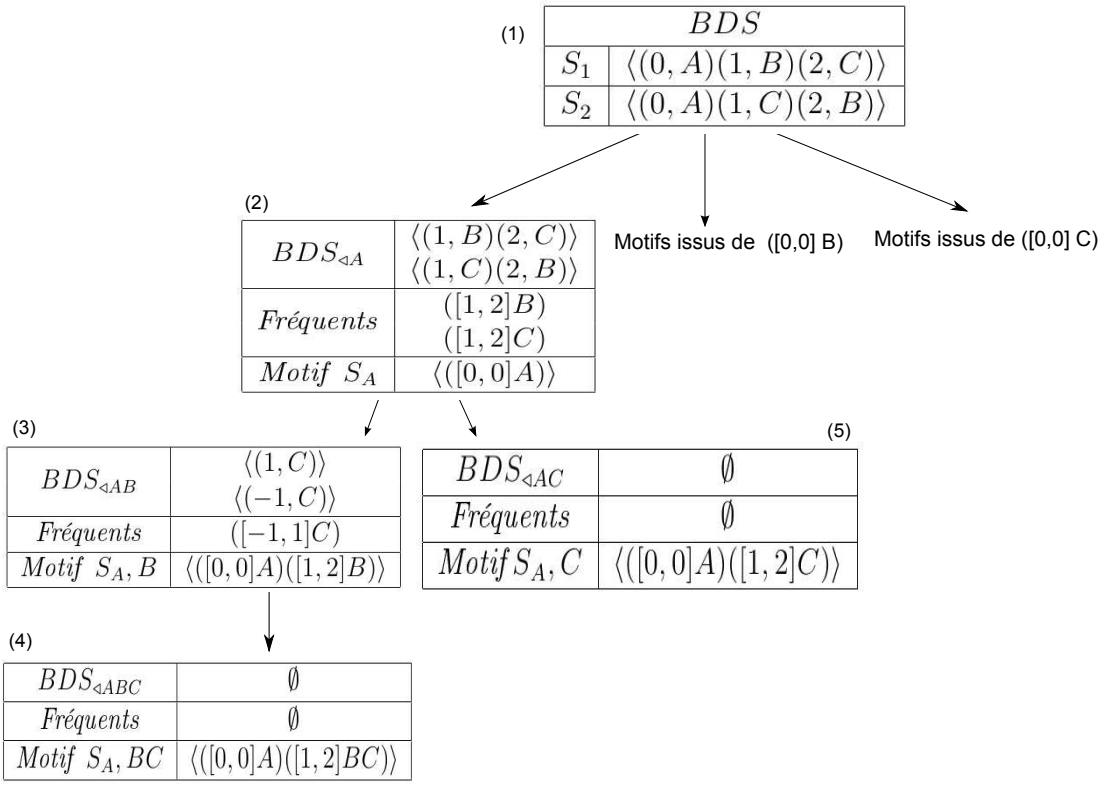


FIGURE 4.8 – Exemple de la structure de déroulement de $STI-PS$ en appliquant $wprojection_{\triangleleft}$.

Dans $BDS_{A\triangleleft}$, *les -STI* $\langle([1,2]B)\rangle$ *et* $\langle([1,2]C)\rangle$ *sont fréquentes. Pour étendre* $\langle([0,0]A)\rangle$, $\langle([1,2]B)\rangle$ *est traité en premier car* $B \triangleleft C$.

On construit alors $S_{A,B} = \langle([0,0]A)\rangle \oplus \langle([1,2]B)\rangle = \langle([0,0]A)\rangle \oplus_S \langle([1,2]B)\rangle$ $(1>0$ *et* $1>0)$ *et* $BDS_{A,B\triangleleft}$ *est calculée , elle est illustrée dans le tableau (2) de la figure 4.8. Vu que* B *est le premier évènement qui étend* $\langle([0,0]A)\rangle$ *aucun évènement inférieur à* B *n'est présent dans le résultat de la projection. Dans* $BDS_{A,B\triangleleft}$, $\langle([-1,1]C)\rangle$ *est fréquent. Ce dernier est ramené à la même référence temporelle que* $S_{A,B}$ *et devient* $\langle([1,2]C)\rangle$. *On peut maintenant le concaténer à* $S_{A,B}$. *C'est une T-extensions car l'intervalle associé à* C *et égal à celui associé à* B, *alors* $S_{A,BC} = S_{A,B} \oplus \langle([1,2]C)\rangle = S_{A,B} \oplus_T \langle([1,2]C)\rangle$. *Par la suite, nous projetons* $BDS_{A,B\triangleleft}$ *sur* $\langle([-1,1]C)\rangle$, *le résultat de la projection illustré dans le tableau (4) de la figure 4.8 est vide. L'extension de* $S_{A,BC}$ *n'est donc pas possible.*

L'extraction remonte donc de deux niveaux pour étendre $\langle([0,0]A)\rangle$ *avec* $\langle([1,2]C)\rangle$. *Cette extension est une S-extension qui fournit le motif suivant :* $S_{A,C} = \langle([0,0]A)\rangle \oplus \langle([1,2]C)\rangle = \langle([0,0]A)\rangle \oplus_S \langle([1,2]C)\rangle$.

La projection de $BDS_{A\triangleleft}$ *sur* $\langle([1,2]C)\rangle$ *est appelée* $BDS_{A,C\triangleleft}$, *elle est représentée dans le tableau (5) de la figure 4.8. Les séquences de cet nouvel espace de recherche sont vides car dans la base résultat, l'évènement* B *n'apparait pas. Ce dernier n'étant présent que dans la zone T-*

extension par rapport à C il n'est pas comptabilisé parmi les extensions puisque $B \triangleleft C$. $BDS_{AC\triangleleft}$ est donc vide et l'ensemble des motifs extraits suite au fréquents $\langle([0,0]A)\rangle$ sont identifié.

Si on compare les exemples 34 et 38, nous constatons que la première branche d'extraction qui identifie le plus long motif est la même. En effet, dans les deux exemple les évènements sont traités selon leurs ordre lexicographique, ce choix est fait afin de bien mettre l'accent sur la différence de comportement entre les deux algorithmes. Nous pouvons donc voir que lorsque $Algo_1$ considère deux fois la relation de proximité entre les évènement B et C, $Algo_2$ ne l'analyse qu'une seule foi. Ce dernier explore une extension en moins, celle qui aboutit à la double extraction du motif $\langle([0,0]A)([1,2]BC)\rangle$ dans $Algo_1$.

On se propose dans ce qui suit de montrer que $Algo_1$ et $Algo_2$ sont équivalents, c'est à dire que appliqués à une même base de séquences, en utilisant les mêmes paramètres d'extraction ($minsupp$, ws, $mingap$, $maxgap$, $min\_whole\_interval$ et $max\_whole\_interval$), les deux algorithmes retournent le même résultat. Pour cela, nous étudions les cas de possibilités d'extension d'un motif.

Montrons d'abord que pour les deux algorithmes, la S-extension d'un motif S ne peut être étendue avec une T-extensions. Nous étudions par la suite les différentes possibilités d'extension d'un motif par $Algo_1$ et $Algo_2$.

Intuitivement, lors de l'extension d'un motif avec une S-extension, la projection résume les séquences de la base autour de la dernière transaction ajoutée à S. Cette transaction étant éloignée de la dernière transaction de S, les T-extensions de S ne font pas partie du nouveau résumé.

Lemme 2. *Soient $S = \langle([m_1, M_1]I_1)\dots([m_n, M_n]I_n)\rangle$, $S_p = \langle([m_p, M_p]I_p)\rangle$ et $S_k = \langle([m_k, M_k]I_k)\rangle$ fréquentes dans BDS la base de séquences résumée associée à S. S_p Respectivement S_k représentent une T-extension respectivement une S-extension de S avec $I_n \triangleleft I_p$*

- *$S \oplus ([m_i, M_i]e_i) \oplus ([m_j, M_j]e_j)$ est un motif fréquent*
- *$S \oplus ([m_j, M_j]e_j) \oplus ([m_i, M_i]e_i)$ n'est pas un motif fréquent.*

Preuve. Nous étudions dans cette preuve les extensions de S avec S_p et S_k extraites par $Algo_1$ et $Algo_2$. Ces séquences représentent les concaténations suivantes : $S \oplus S_p \oplus S_k$ et $S \oplus S_k \oplus S_p$.

Nous traitons d'abord l'extension de S avec S_p, la séquence résultats est $S_1 = S \oplus S_p = S \oplus_T S_p$. Pour extraire l'extension de S_1, $wprojection(BDS, S_p)$ est calculée par $Algo_1$ et $wprojection_\triangleleft(BDS, S_p)$ est calculée par $Algo_2$. Ces deux projections contiennent les *T-extensions* et les *S-extensions* de S_1 dans BDS.

Vu que I_p est proche de I_n, ses S-extensions sont les mêmes que celles de S (et donc de

sa dernière transaction) dans les deux projections calculée par $Algo_1$ et $Algo_2$. Effectivement, ces dernières se trouves en aval de S_p dans les séquences de BDS. Dans le cas où S_p se trouve après la dernière transaction de S, elle en est quand même assez proche pour y être fusionnée. Les transactions qui se trouvent alors entre I_n et I_p sont donc aussi des T-extension et seront fusionnées avec les deux dernières transactions.

On peut donc dire que dans les deux projections effectuées par les deux variantes de *STI-PS*, les S-extensions de S_p sont les mêmes que celles de S. Alors S_k étend S_1 tel que : $S_1 \oplus S_k = S_1 \oplus_S S_k = S \oplus_T S_p \oplus_S S_k$.

Nous concluons donc que $S \oplus_T S_p \oplus_S S_k$ est identifiée comme fréquente par $Algo_1$ et $Algo_2$.

Par la suite nous traitons l'extension de S avec S_k la séquence résultats est $S_2 = S \oplus S_k = S \oplus_S S_k$. Pour extraire l'extension de S_2, $wprojection(BDS, S_k)$ est calculée par $Algo_1$ et $wprojection_{\triangleleft}(BDS, S_k)$ est calculée par $Algo_2$. Ces deux projections contiennent les *T-extensions* et les *S-extensions* de S_k dans BDS.

Étant donné que S_k est une S-extension de S, elles se trouvent donc éloignée en aval de I_p pour les deux projections. Les T-extensions de S_k sont proches de cette dernière et donc forcément éloignées en aval de I_n (la dernière transaction de S) dans les deux projections. Aussi, les S-extensions de S_k en sont éloignées en aval et le sont donc aussi par rapport à I_n dans les deux projections. On peut conclure que quelle que soit la variante de *STI-PS* appliquée dans la projection de BDS sur S_k il n'y a aucune transaction proche de I_n. S_p ne s'y trouve donc pas.

On peut donc conclure que $S_2 \oplus S_p = S \oplus_T S_p \oplus S_k$ n'est identifié fréquente ni par $Algo_1$ ni par $Algo_2$. □

Proposition 1. *Soit une base de séquences BDS, un support minimal minsupp, une taille de fenêtre ws et les contraintes temporelles de gap mingap, maxgap, de longueur de la séquence min_whole_interval et max_whole_interval et $\omega = \{e_1, e_2 \dots e_n\}$ l'ensemble des évènements fréquents dans BDS. On applique à BDS les deux algorithmes $Algo_1$ et $Algo_2$. Alors on a :*

- *L'ensemble des STI fréquentes retournées par les algorithmes sont équivalentes.*
- *$algo_1$ extraits plusieurs fois la même STI alors que $algo_2$ ne l'extrait qu'une seule foi.*

Preuve. Pour un motif $S = \langle([m_1, M_1]I_1) \dots ([m_n, M_n]I_n)\rangle$ identifié comme fréquent, nous étudions dans cette preuve l'ensemble de ses extensions possibles identifiées par $Algo_1$ et $Algo_2$ à partir de α_T et α_S respectivement l'ensemble des T-extensions et des S-extensions, dans BDS. C'est la base de séquences résumée associée aux continuations de S.

Nous identifions les quatre possibilités d'extensions suivantes :

1. $S \oplus S_p \oplus S_k$ avec $\{S_p, S_k\} \in \alpha_T^2$ et $I_n \triangleleft I_p \triangleleft I_k$.

2. $S \oplus S_p \oplus S_k$ avec $S_p \in \alpha_T$ et $S_k \in \alpha_S$
3. $S \oplus S_p \oplus S_k$ avec $\{S_p, S_k\} \in \alpha_S^2$, $I_p \triangleleft I_k$ et S_p proche de S_k.
4. $S \oplus S_p \oplus S_k$ avec $\{S_p, S_k\} \in \alpha_S^2$, $I_p \triangleleft I_k$ et S_p éloignée de S_k.

Les illustrations des positionnements de $([m_n, M_n]I_n)$ par rapport à S_p et S_k sont représentées dans les figures 4.9, 4.10, 4.12 et 4.12. Nous détaillons l'extraction effectuée par $Algo_1$ et $Algo_2$ dans chacun de ces quatre cas :

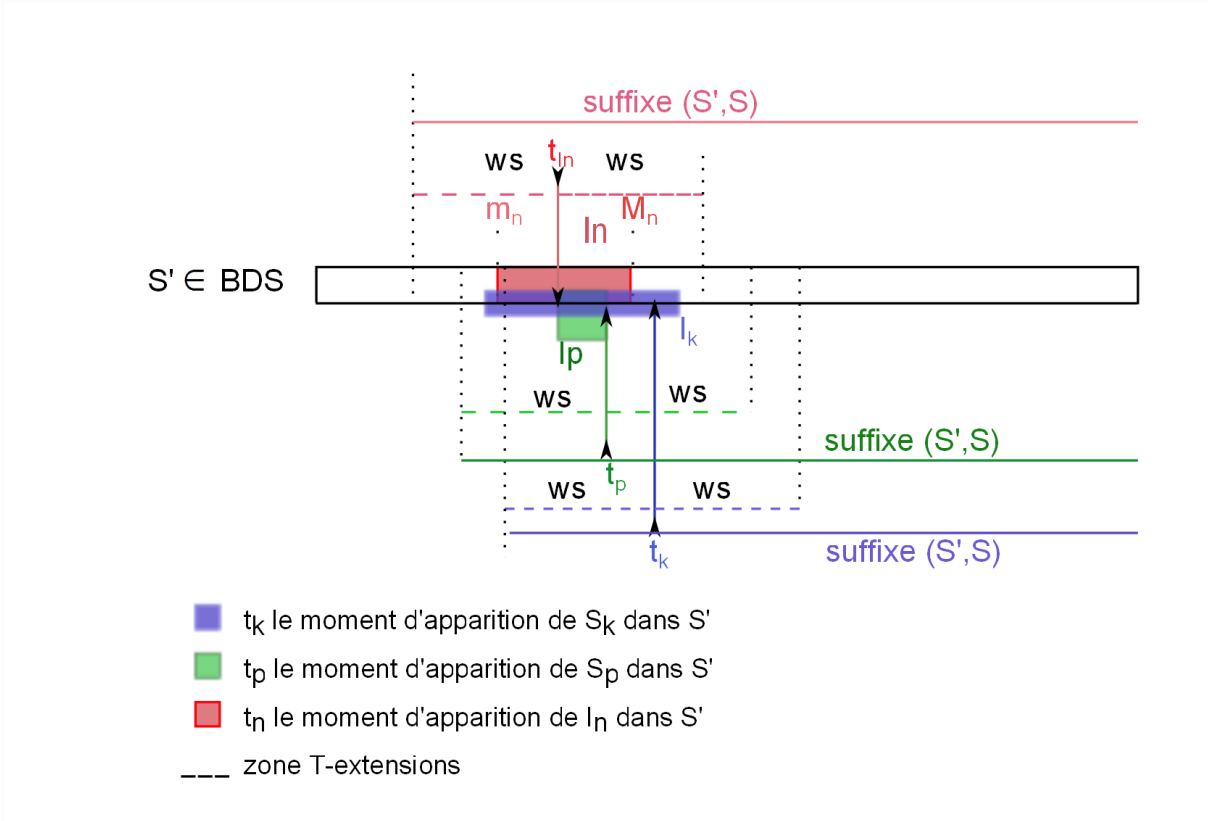


FIGURE 4.9 – Cas 1 : S est étendue avec S_p et S_k deux de ses T-extensions

1. Cas 1 : S peu être étendue avec S_p et S_k où $\{S_p, S_k\} \in \alpha_T^2$ et $I_n \triangleleft I_p \triangleleft I_k$ pour obtenir $S \oplus S_p \oplus S_k = S \oplus_T S_p \oplus_T S_k$. L'illustration graphique du positionnement de S_p, S_k et I_n, pour ce premier cas est représentée dans la figure 4.9.

   Traitons d'abord $S \oplus S_p \oplus S_k$, $Algo1$ et $Algo_2$ concatènent tous deux S_p à S pour obtenir $S \oplus S_p = S \oplus_T S_p$, I_p est fusionné avec la dernière transaction de S. Par la suite, les deux algorithmes construisent la séquence $S \oplus S_p \oplus S_k = S \oplus_T S_p \oplus_T S_k$. Effectivement, puisque S_k et S_p sont proches de S, alors ils sont proches entre eux.

   Par la suite l'extraction continue jusqu'à ce que les extensions de cette dernière séquence soient explorées. Alors, l'extension remonte les niveaux dans les deux algorithmes, pour

étendre S avec S_k. Dans ce cas les traitements effectués par $Algo_1$ et $Algo_2$ différents.

$Algo_1$ étend S avec S_k et construit $S \oplus S_k = S \oplus_T S_k$. De la même manière que l'extraction précédente, il construit $S \oplus S_k \oplus S_p = S \oplus_T S_k \oplus_T S_p$.

$Algo_2$, quand à lui étend S avec S_k et construit le motif $S \oplus S_k = S \oplus_T S_k$. Puisque $I_p \triangleleft I_k$, $Algo_2$ ne permet d'étendre $S \oplus_T S_k$ avec S_p.

D'après le lemme 1 les séquence $S \oplus_T S_p \oplus_T S_k$ et $S \oplus_T S_k \oplus_T S_p$ sont équivalentes.

On peut donc conclure que pour ce premier cas $Algo_1$ extrait deux fois la même séquence fréquente, alors que $Algo_2$ l'extrait une seule foi.

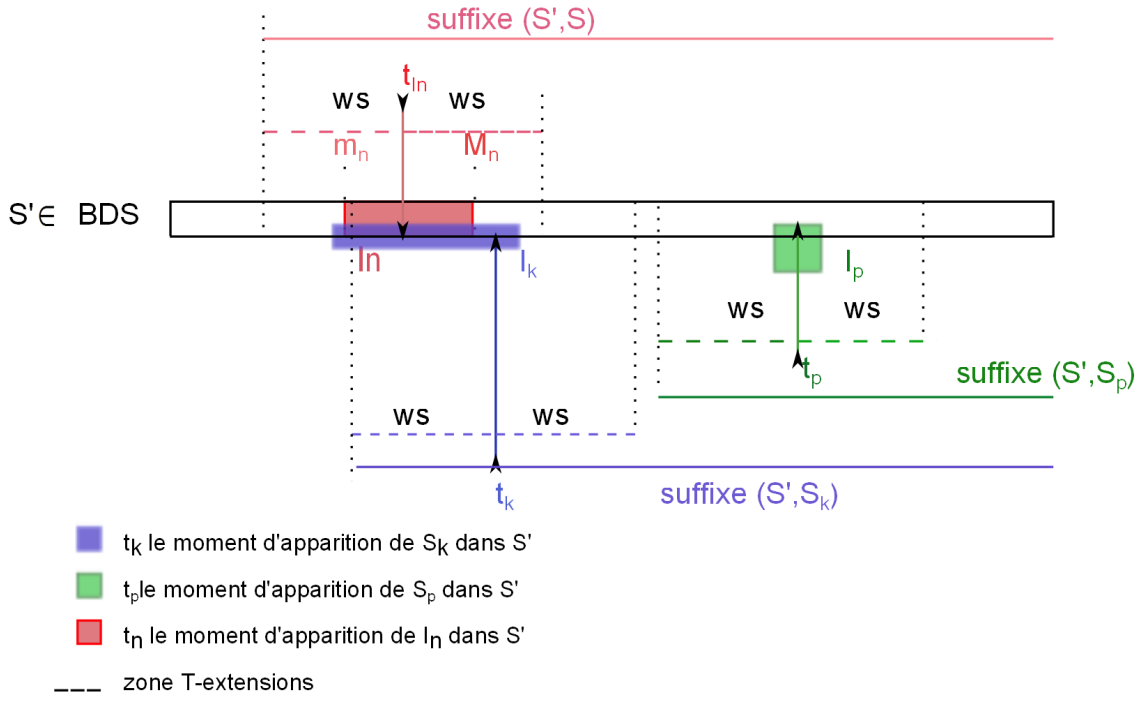


FIGURE 4.10 – Cas 2 : S est étendue avec $S_p \in \alpha_T$ et $S_k \in \alpha_S$

2. Cas 2 : S peut être étendue avec S_p et S_k où $S_p \in \alpha_T$ $S_k \in \alpha_S$ et $I_n \triangleleft I_p \triangleleft I_k$ pour obtenir $S \oplus S_p \oplus S_k = S \oplus_T S_p \oplus_S S_k$. L'illustration graphique du positionnement de S_p, S_k et $(m_n, M_n]I_n)$, pour ce premier cas est représentée dans la figure 4.10.

   D'après le lemme 2, $Algo_1$ et $Algo_2$ identifie tous deux le motifs $S \oplus S_p \oplus S_k = S \oplus_T S_p \oplus_S S_k$. Les deux algorithmes n'identifient pas $S \oplus S_k \oplus S_p = S \oplus_S S_k \oplus_T S_p$ comme une extension de S.

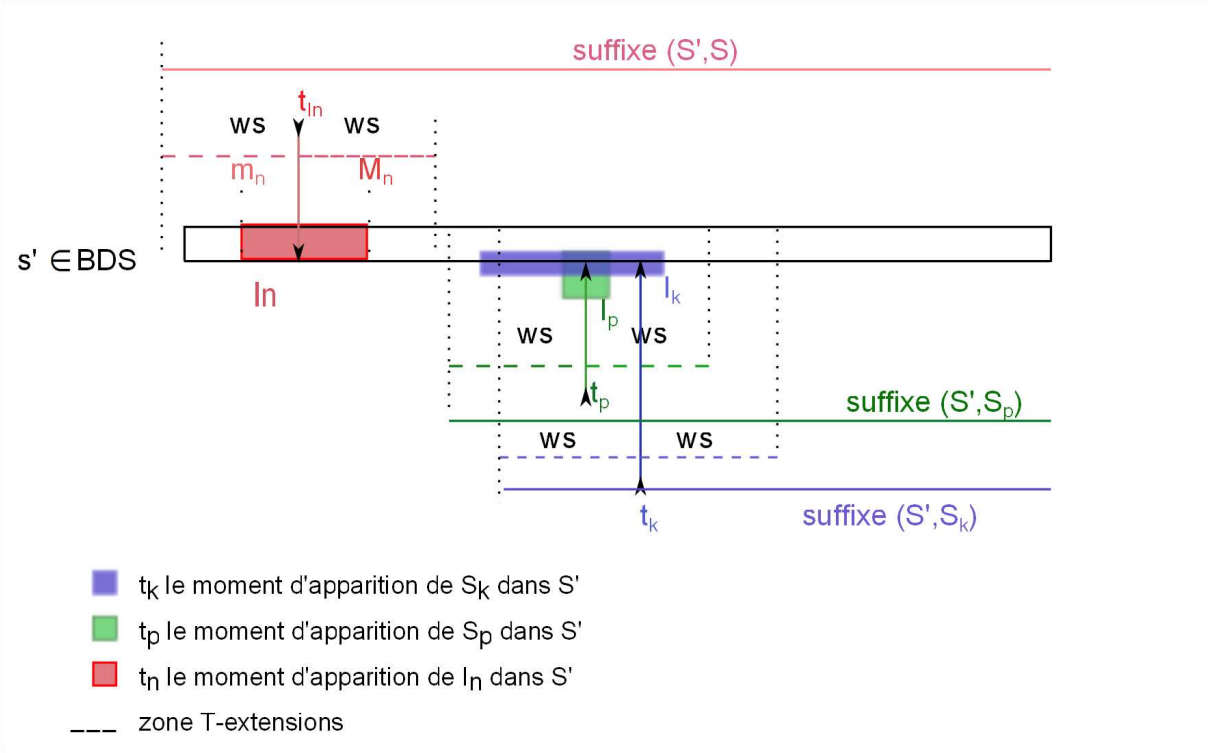


FIGURE 4.11 – Cas 3 : S est étendue avec $S_p \in \alpha_S$ et $S_k \in \alpha_S$ avec S_p et S_k proches

3. Cas 3 : S peut être étendue avec S_p et S_k où$\{S_p, S_k\} \in \alpha_S^2$ avec $I_n \triangleleft I_p \triangleleft I_k$ et S_p et S_k proches. L'illustration graphique ce cas est représentée dans la figure 4.12.

   Les deux algorithmes étendent S avec S_p pour obtenir $S \oplus S_p = S \oplus_S S_p$. Par la suite, S_k est concaténée avec la séquence résultats tel que $S \oplus S_p \oplus S_k = S \oplus_S S_p \oplus_T S_k$. S_k est une T-extension de $S \oplus_S S_p$ car S_p et S_k sont proches. Pour les deux variantes de *STI-PS*, les extensions du dernier motif sont explorées.

   Dans une autre branche d'extraction, S est étendue avec S_k, et les deux algorithmes construisent de la même manière $S \oplus S_k = S \oplus_S S_k$. L'extension de ce dernier motif n'est pas la même pour $Algo_1$ et $Algo_2$.

   $Algo_1$ concatène S_p à $S \oplus S_k = S \oplus_S S_k$, c'est une T-extension car S_p et S_k sont proches. Il fournit alors $S \oplus S_k \oplus S_p = S \oplus_S S_k \oplus_T S_p$. D'après le lemme 1 on a $S \oplus_S S_k \oplus_T S_p$ équivaut à $S \oplus_S S_p \oplus_T S_k$.

   $Algo_2$, quand à lui ne permet d'étendre $S \oplus S_k$ avec S_p car $I_p \triangleleft I_k$ et S_p est proche de S_k. Dans ce troisième cas les deux algorithmes identifient le même ensemble de motifs fréquents. Cependant $Algo_1$ extrait $S \oplus S_k \oplus S_p$ deux fois, alors que $Algo_2$ explore la possibilité de construction de ce motif une seule foi.

4. Cas 4 : S peut être étendue avec S_p et S_k avec $\{S_p, S_k\} \in \alpha_S^2$, $I_n \triangleleft I_p \triangleleft I_k$ et S_p et S_k

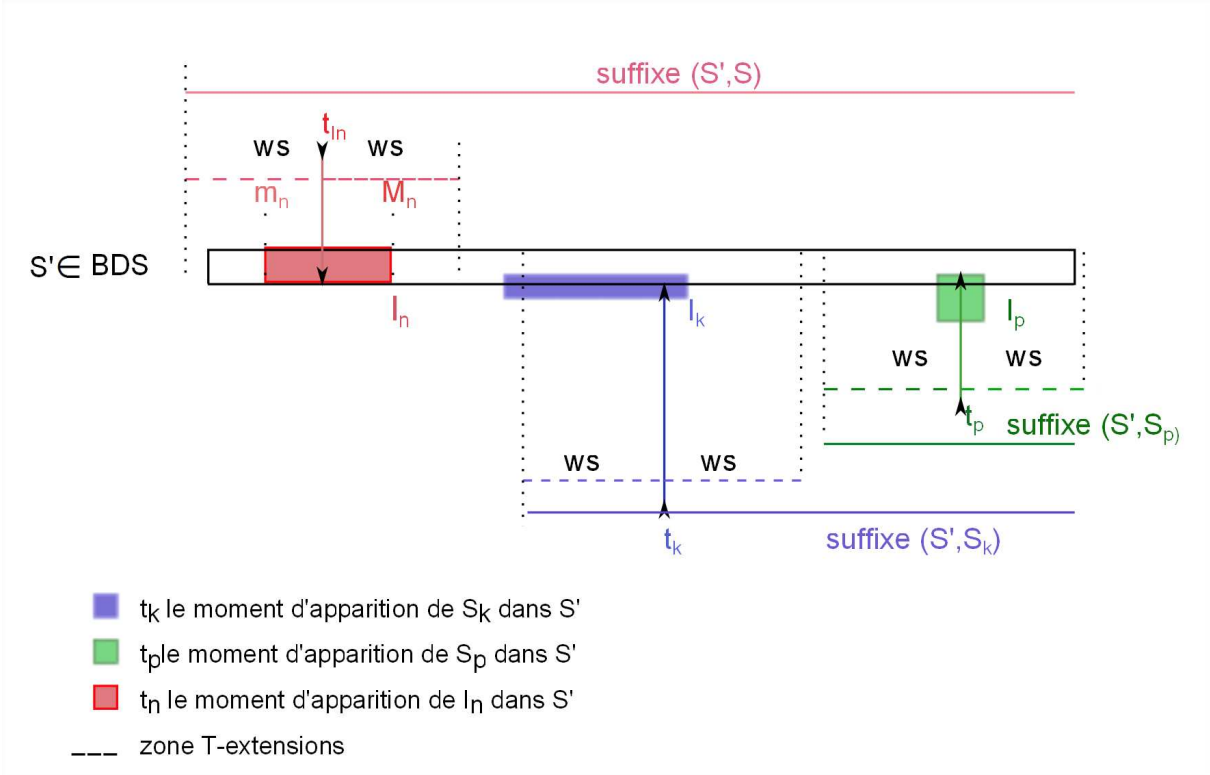


FIGURE 4.12 – Cas 4 : S est étendue avec $S_p \in \alpha_S$ et $S_k \in \alpha_S$ avec S_p et S_k éloignées

sont éloignées aussi entre eux. L'illustration graphique du positionnement de S_p, S_k et I_n, pour ce cas est représentée dans la figure 4.12. Dans la représentation de la figure 4.12, S_k apparait avant S_p.

$Algo_2$ commence par étendre S avec S_p pour obtenir $S \oplus S_p = S \oplus_S S_p$. Dans l'espace associé aux extensions de ce dernier S_k n'apparait pas car il apparait dans les séquence loin avant S_p. Il n'est donc pas possible de construire le motif $S \oplus S_p \oplus S_k$.

Lorsque $Algo_2$ étend S avec S_k, il construit le motif $S \oplus S_k = S \oplus_S S_k$ l'extension de ce dernier permet d'y concaténer S_p pour obtenir $S \oplus_S S_k \oplus S_p = S \oplus_S S_k \oplus_S S_p$. Effectivement, malgré le fait que $I_p \triangleleft I_k$ la construction de ce dernier motif est possible. Puisque S_p et S_k sont éloignées et que la restriction des continuations selon l'ordre ($\triangleleft$) des évènements ne s'applique qu'aux continuations proches.

De la même manière $Algo_1$ construit $S \oplus S_p = S \oplus_S S_p$ et ne peut construire $\S \oplus S_p \oplus S_k$ car S_k est loin en amont de S_p. elle ne fait pas partie de son suffixe. Aussi, de la même manière que $Algo_2$, $Algo_1$ permet de construire le motif $S \oplus_S S_k \oplus S_p = S \oplus_S S_k \oplus_S S_p$.

Dans ce cas les deux algorithmes extraient exactement les mêmes motifs.

Nous avons énuméré, dans cette preuve, toutes les possibilités d'extension d'une séquence S selon les cas de proximité et d'ordre entre les deux types d'extension définis. Les comportements

de $Algo_1$ et $Algo_2$ ont été étudiés pour chaque cas afin de comparer les motifs extraits par les deux algorithmes. Nous pouvons conclure après l'étude de ces cas que les deux algorithmes extraient le même ensemble de motifs. Cependant dans les cas 1 et 3 $Algo_1$ extrait plusieurs fois le même motif alors que $Algo_2$ ne l'explore qu'une seule foi. □

Malgré le fait que les deux variantes de *STI-PS* explorent certaines branches de manière différentes, les ensembles de motifs retournés par les deux algorithmes sont les mêmes dans tous les cas d'extensions d'un motif S. La restriction par l'ordre ($\lhd$) appliqué aux T-extensions dans $wprojection_{\lhd}$. permet d'éviter l'extraction d'un même motif plusieurs fois. En effet, lorsque les évènement sont proches le retour arrière de la projection $wprojection$ considère plusieurs fois la même situation de rapprochement, ce problème est évité par l'instauration de l'ordre dans le traitement des évènements proches. Cette solution permet donc de ne considérée que le rapprochement d'un évènement par rapport à ceux qui lui sont supérieurs.

Pour optimiser l'algorithme *STI-PS*, nous utilisons la $wprojection_{\lhd}$. L'algorithme 4 détaille la fonction de projection appelée par notre algorithme.

Input : BDS, e, $borne\_inf$, $borne\_supp$, $minsupp$, ws, $mingap$, $maxgap$

Output : newBDS : Le nouvel espace de recherche

---

initialiser newBDS

forall the $S \in BDS$ **do**

  if $e \in S$ *et* $time_S(e) \in [borne\_inf, borne\_sup]$ **then**

    newBDS = newBDS $\cup$ $wsuffixe_{\lhd}$(S,e, $borne\_inf$, $borne\_supp$, ws)

  end

end

Algorithme 4: La fonction de **Projection** calcule l'ensemble $wsuffixe_{\lhd}$ pour chaque séquence de BDS par rapport à la 1-STI ($[borne\_inf, borne\_supp]e$) .

Cette section a présenté l'algorithme d'extraction de séquences fréquentes *STI-PS* (Séquences Temporelles par Intervalles d'Incertitude -utilisant PrefixSpan [PHW02]). Cet algorithme extrait des séquences temporelles par intervalles d'incertitude à partir de séquences temporelles à estampilles discrètes et intègre les contraintes temporelles usuelles. Aussi, il instaure un principe de regroupement d'évènements, au départ associés à des estampilles différentes, dans une même transaction estampillée par intervalle temporel afin de préserver l'information temporelle initiale. Un tel regroupement est intégré grâce à l'application d'une fenêtre glissante, qui fixe le degré d'incertitude des évènements par rapport aux intervalles qui leur sont associés.

5 Conclusion

Ce chapitre présente une nouvelle méthode d'extraction de séquences fréquentes. Elle calcule les séquences temporelles par intervalles (STI) d'incertitudes à partir de données séquentielles datées. Ces *STI* présentent des aspects non pris en compte dans les techniques d'extraction rencontrées dans la littérature [PHMA+01, HY06, AS95, YCJCWCSY10, FVNN08]. Elles remontent des comportements « typiques » qui répondent à trois critères : (1) la fréquence de la chronologie des transactions qui décrivent une séquence, (2) les transactions sont associées à des représentations temporelles qui autorisent l'application de contraintes afin de retourner des séquences répondant à une sémantique temporelle décrite par les contraintes et (3) les *STIs* représentent une relaxation de la temporalité d'apparition des évènements au sein d'une même transaction.

STI-PS extrait des séquences fréquentes temporelles avec intervalles d'incertitudes en considérant un support minimal *minsupp*, les contraintes temporelles usuelles *mingap*, *maxgap*, *min_whole_interval* et *min_whole_interval*, et une taille de fenêtre *ws*. Cette fenêtre permet d'appliquer un regroupement progressif des évènements distincts afin de les associer à une même transaction et leur affecter un intervalle d'incertitude sur leur occurrences.

STI-PS est un algorithme à la « pattern growth », qui intègre la fenêtre et la prise en compte de son aspect glissant dans les deux principales étapes de l'approche : La sélection de fréquent et la réduction de l'espace de recherche.

Le chapitre suivant présente une évaluation détaillée des performances de *STI-PS*. Dans un premier temps, il évalue les performances d'exécution de *STI-PS* par rapport à celles d'algorithmes appliquant la même approche d'extraction. Dans un deuxième temps, il présente une évaluation de la qualité des motifs (STI) extraits.

Chapitre 5

Implémentations et Expérimentations

Sommaire

1 Introduction

L'algorithme *STI-PS* extrait à partir d'un historique de données séquentielles, des comportements fréquents datés et « intéressants ». Ce sont les séquences temporelles par intervalles d'incertitude. Elles présentent un ordre chronologique global et une relaxation de l'ordre local des évènements considérés comme étant simultanés.

Le chapitre précédent détaille les différents procédés algorithmiques utilisés pour mettre en œuvre *STI-PS* qui fait partie d'une approche d'extraction de type « FP-Growth ». Il applique un procédé itératif-récursif qui extrait tous les motifs fréquents de longueur k à partir d'un préfixe

fréquent de longueur $k-1$.

Une première section présente la technique utilisée pour l'implémentation de *STI-PS*, détaille les aspects techniques de son déroulement et analyse ses performances en terme d'occupation mémoire seront présentées.

Une seconde section prose, quelques expérimentations comparant la qualité et l'utilité de *STI-PS*. Tout d'abord, elle évalue ses performances techniques par rapport à des algorithmes similaires faisant partie de la même approche « FP-Growth ».

Ensuite, la pertinence des motifs extraits est évaluée en fonction de la qualité et de la quantité des comportements véhiculées par les séquences temporelles par intervalles à partir des données séquentielles initiales estampillées de manière discrète. Cette évaluation consiste à comparer les résultats obtenus par *STI-PS* par rapport à ceux de certaines méthodes similaires.

2 Mise en œuvre

D'abord, nous détaillons les aspects techniques de la mise en œuvre de l'algorithme *STI-PS* et précisons ensuite les améliorations apportées à la première implémentation, afin d'optimiser ses performances en temps de calcul et en consommation mémoire.

2.1 Mise en œuvre naïve de *STI-PS*

Pour extraire les séquences temporelles par intervalles d'incertitude, *STI-PS* applique l'approche « FP- Growth ». Il prend en compte les contraintes $mingap$, $maxgap$, $min\_whole\_interval$, $max\_whole\_\ interval$ et ws.

Il applique une méthode d'extraction dite en « profondeur » qui déploieune structure d'appels récursifs arborescente. Au départ, le nœud racine correspond au niveau zéro d'extraction où aucun motif n'est encore identifié. La base de séquences initiale est parcourue pour identifier les évènements qui y sont fréquents. Chacun des évènements fréquents génère un appel récursif à partir de la racine et l'extraction continue jusqu'à ce que tous les motifs soient identifiés. Une branche complète de l'arbre d'extraction (de la racine à une feuille) correspond à l'extraction de tous les motifs fréquents qui partagent un même préfixe (l'évènement fréquent issu de la racine).

L'implémentation de *STI-PS* que nous présentons dans ce paragraphe est une modification de celle de *SPMF* présentée dans [FVNN08]. Il s'agit d'un algorithme d'extraction de séquences temporelles fréquentes à la manière de ce que réalise « FP-Growth » et qui considère les contraintes

temporelles *mingap*, *maxgap*, *min_whole_interval* et *max_whole_interval*. L'implémentation utilisée est disponible sur l'emplacement WEB [1].

La Figure 5.1 illustre le schéma des itérations des algorithmes « FP-Growth ». Le premier appel de sélection de fréquents ne prend pas en compte l'encrage temporel des évènements pour calculer leurs fréquences. Il identifie l'ensemble des 1-STI fréquentes. Ces dernières représentent aussi les premières transactions des motifs extraits par les prochains appels récursifs.

Par la suite, pour chacune des 1-STI précédemment identifiées, les fonctions de projection et de sélection de fréquents sont successivement appelées. Elles sont représentées dans la figure 5.1 par les blocs colorés. Ce sont les fonctionnalités qui différencient *STI-PS* des autres algorithmes de la même approche.

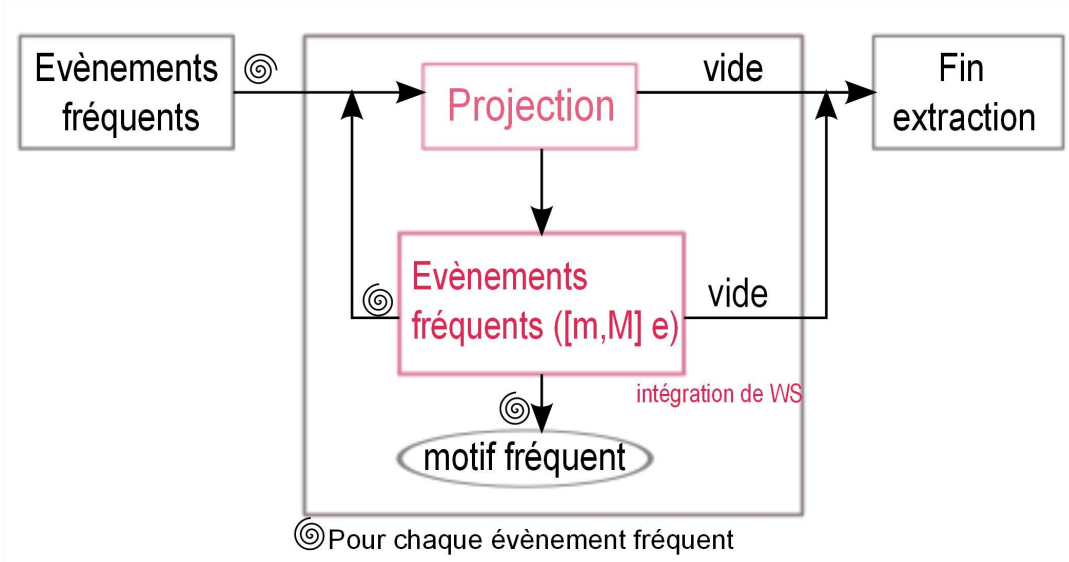


FIGURE 5.1 – Schéma des appels itératif-récursif des modules de l'algorithme *STI-PS*

La succession des appels récursifs fournit les motifs tels que chaque appel étend le motif en cours avec un évènement fréquent identifié par le module de sélection de fréquents. Ainsi, un motif de longueur k est obtenu en effectuant k appels récursifs.

Cette succession d'appels génère une branche d'extraction de longueur k, dans laquelle tout chemin entre la racine de la branche et un de ses nœud représente un motif fréquent, ce sont les sous-séquences du motif construit par la totalité de la branche.

Ainsi l'exploration d'un motif contenant k éléments génère k projections et autant de versions réduites de la base de séquences qui résident en mémoire.

A Chaque projection, une nouvelle base de données physique est crée, sa la suppression

1. http://www.philippe-fournier-viger.com/spmf

intervient lorsque toutes les itérations et les appels récursifs (sous-branches) générées par les autres évènements fréquents sont terminées.

Lorsque la base de données de départ contient des motifs fréquents « long » (de longueur N), leur extraction construit des structures arborescentes « profondes » et donc projette les espaces de recherches en un nombre important de fois.

Selon la configuration de la base de séquences initiale et des contraintes temporelles qui interviennent de motifs à extraire, l'algorithme effectue un nombre variable d'appels récursifs et réalise autant de projections. Par conséquent, l'algorithme nécessite un espace mémoire équivalent à l'espace occupé par les différents espaces de recherches générés et les motifs extraits.

Afin de remédier à cette contrainte d'espace nous présentons dans la section suivante une optimisation de l'implémentation de notre algorithme.

2.2 Amélioration apportée pour *STI-PS*

Cette section présente une des améliorations de notre algorithme qui permet de réduire ses besoin en ressource mémoire.

Tous les motifs ayant le même préfixe sont générés dans le même sous-arbre. Ainsi, au lieu de les stocker individuellement, on peut utiliser une structure d'arbre de préfixe. Nous illustrons notre propos par l'exemple suivant :

Exemple 39. *Considérons la base de séquence BDS décrite dans le tableau 5.1a, elle contient les trois évènements fréquents associés à des intervalles nuls et forment les* 1-STI *suivantes :* $([0,0]A)$, $([0,0]B)$ *et* $([0,0]C)$. *La branche d'extraction correspondante à l'itération générée par le 1-motif fréquent* $([0,0]A)$ *est décrite dans la figure5.2. A partir de cette branche, l'ensemble des motifs extraits partagent le même préfixe* $([0,0]A)$. *Les motifs extraits le long de cette branche sont illustrés dans le tableau 5.1b. Le fait de stocker indépendamment ces motifs consiste à enregistrer quatre fois l'information* $([0,0]A)$, *deux fois* $([1,2]B)$ *et deux fois* $([1,2]C)$.

Nous optimisons l'occupation de l'espace mémoire en utilisant un arbre de préfixes pour stocker les motifs extraits. Cette technique a été utilisée dans différents travaux [AFGY02, WC07, Zak01, XHA03].

Lors de la première étape d'extraction, l'algorithme *STI-PS* élimine les évènements non fréquents de la base de séquence de départ et crée la racine de l'arbre de préfixe laquelle contient le motif vide $\langle\emptyset\rangle$. Par la suite, chaque évènement fréquent identifié à la première étape d'extrac-

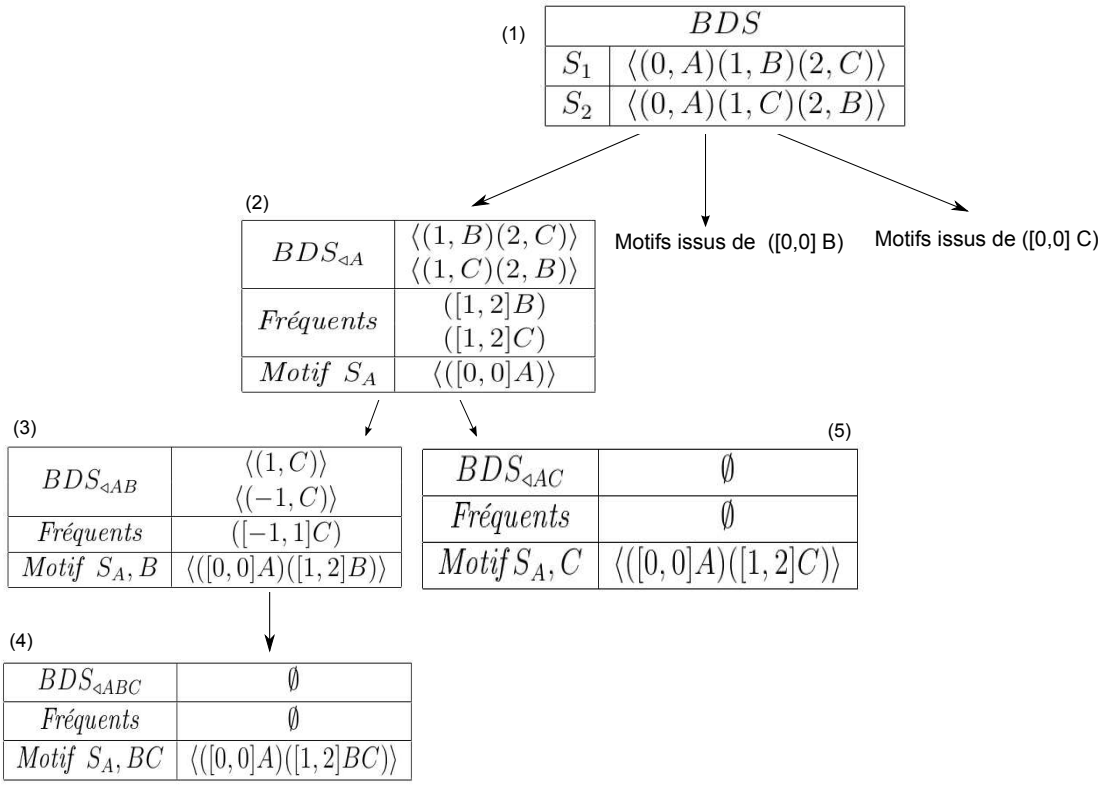


FIGURE 5.2 – Arborescence des appels récursif de *STI-PS* appliqué à BDS

tion construit un nœud fils de la racine. L'heuristique la plus utilisée consiste à trier par ordre croissant les évènements fréquents.

Le long des branches d'extraction récursivement générées par chacun de ces nœuds, tout évènement fréquent identifié dans l'espace projeté permet d'étendre l'arbre des préfixes en ajoutant un fils au nœud en cours, de telle que sorte qu'à chaque appel récursif un nœud représentant l'évènement fréquent est ajouté à l'arbre. Chaque chemin entre la racine et n'importe quel élément de l'arbre des préfixes (feuille ou nœud intermédiaire) représente une séquence fréquente par intervalle. Chaque nœud de l'arbre contient trois informations qui permettent de reconstruire les *STI* fréquentes extraites.

- e : l'évènement fréquent qui a permis de construire la branche ;
- $[m, M]$: l'intervalle temporelle qui représente l'estampille de l'évènement fréquent ;
- $support(([m, M], e))$ le support de la 1-STI fréquente.

De cette manière, à chaque branche de l'extraction correspond une branche dans l'arbre des préfixes. Une branche complète de la racine à la feuille correspond au motif le plus long extrait à partir d'un évènement fréquent identifié à la première étape de l'algorithme. Nous reprenons

BDS_A	
S_1	$\langle(0,A)(1,B)(2,C)\rangle$
S_2	$\langle(0,A)(1,C)(2,B)\rangle$

(a) Base de séquences BDS

Motifs
$\langle([0,0]A)\rangle$
$\langle([0,0]A)([1,2]B)\rangle$
$\langle([0,0]A)([1,2]C)\rangle$
$\langle([0,0]A)([1,2]BC)\rangle$

(b) Motifs extraits suite à la branche générée par A

Tableau 5.1 – Exemple d'extraction et de stockage de motifs

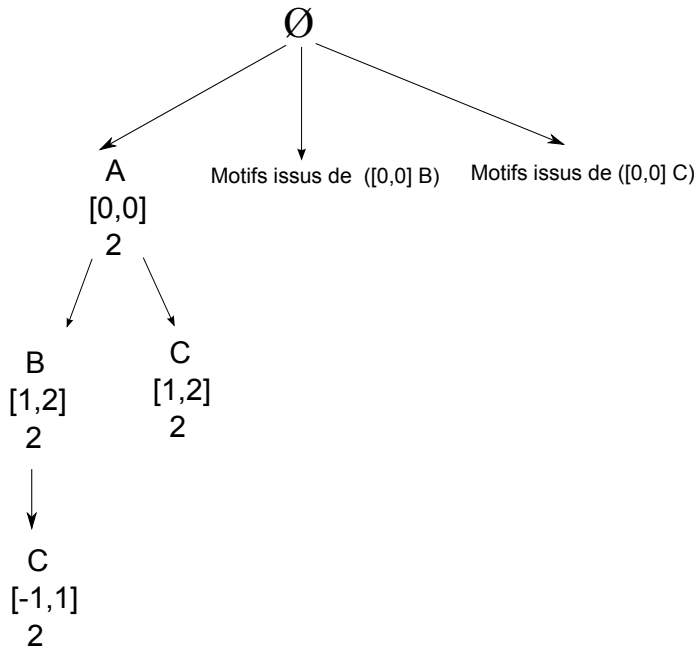


FIGURE 5.3 – Arbre des préfixes pour les motifs issus de l'évènement fréquent A

dans ce qui suit l'exemple 39 pour illustrer cette représentation.

Exemple 40. *Reprenons l'exemple précédant en appliquant l'arbre de préfixe. Nous obtenons la structure de stockage décrite par la figure 5.3. Cette structure permet de stoker une seule fois l'information* $([0,0]A)$ *au lieu de quatre, une seule fois* $([1,2]B)$ *et deux fois* $([1,2]C)$. $([1,2]C)$ *est stockée deux fois car cette information se rapporte à deux sous-branches d'extraction différentes. La première pour l'extension de* $\langle([0,0]A)([1,2]B)\rangle$ *et la seconde pour l'extension de* $\langle([0,0]A)\rangle$

La section suivante présente les expérimentation conduites pour évaluer les performances techniques et fonctionnelles de notre approches.

3 Expérimentation

Cette section présente les expérimentations qui permettent de juger de la qualité et de l'apport fonctionnel de notre algorithme *STI-PS*. Tout d'abord, nous proposons une évaluation des performances techniques de notre algorithme par rapport à des algorithmes similaires faisant partie de l'approche « FP-Growth ».

Ensuite, nous réalisons une évaluation de la pertinence des motifs extraits. Pour cela, la qualité et la quantité des comportements véhiculés par les séquences temporelles par intervalles sont estimées. Cette évaluation consiste à comparer les résultats obtenus par notre algorithme, à ceux obtenus par des méthodes similaires. Enfin, nous concluons par une évaluation de la pertinence des résultats renvoyés par *STI-PS* appliqué à un historique de vie d'une flotte d'avions pour la prévision des applications de tâches de maintenance aéronautique.

3.1 Évaluation des performances de STI-PS

Nous présentons dans ce qui suit une évaluation des performances d'exécution de notre algorithme par rapport aux algorithmes d'extraction de séquences fréquentes présents dans la littérature. Nous comparons les performances d'exécution de l'implémentation de notre algorithme avec celles des algorithmes *PrefixSpan* [PHMA+01] *SPMF* [FVNN08] et *SPAM* [AFGY02]. Les trois algorithmes font partie de l'approche à la « FP-Growth ». Une comparaison des performances de *STI-PS* par rapport à celles de *GSP*, l'algorithme pionnier de l'approche d'extraction par niveau a été présentée dans [BZMMS10].

PrefixSpan est l'algorithme pionnier de l'approche « FP-growth ». L'algorithme *SPMF* est une amélioration du premier. Il se rapproche plus de notre algorithme *STI-PS* puisqu'il intègre les contraintes temporelles *mingap*, *maxgap*, $min\_whole\_interval$ et $max\_whole\_interval$ et extrait des séquences fréquentes temporelles. Finalement, l'algorithme *SPAM* est une amélioration des performances de *PrefixSpan*. Il intègre une représentation binaire de la base de séquences et un arbre de préfixe pour le stockage des motifs fréquents extraits. Cet algorithme extrait des motifs de la même forme que ceux extraits par l'algorithme *PrefixSpan*. La description détaillée de ces algorithmes est présentée dans la partie traitant de l'état de l'art à la section 3.1.

Comme *STI-PS*, les trois algorithmes sont implémentés en JAVA et disponibles sur la page de Philippe-Fournier-Viger [2]. Ils sont exécutés sur une machine Windows 7(64), Intel(R) Core(TM) 3 CPU 2.40 GHz Avec 3 GO RAM.

2. http://www.philippe-fournier-viger.com/spmf/index.php

Nous testons ces algorithmes sur des données synthétiques que nous avons construites de manière aléatoire. Nous évaluons d'abord le temps d'exécution de chacun des algorithmes en faisant varier la valeur du support. La base de séquences utilisée contient mille séquences. Les séquences ont une longueur moyenne de 5 transactions et contiennent 9 évènements différents.

Ètant donnée que les algorithmes *SPAM* et *PrefixSpan* ne gèrent pas les séquences temporelles, les valeurs des contraintes temporelles pour l'algorithme *SPMF* et *STI-PS* sont paramétrées de sorte qu'elles n'appliquent aucune restriction sur les séquences fréquentes extraites. Ainsi, les contraintes *mingap* et $min\_Wohle\_interval$ sont mises a zéro. Les contraintes *maxgap* et $max\_Wohle\_interval$ sont fixées à une même valeur supérieure à la durée de la séquence la plus longue de la base. Ainsi, pour la même raison, la taille de la fenêtre glissante est fixée à zéro pour l'algorithme *STI-PS*.

Les figure 5.5, 5.6, 5.4 et 5.7 illustrent les différents résultats d'exécution des quatre algorithmes. La figure 5.4 montre les variations de leurs temps d'exécution en fonction de différentes valeurs du seuil de support *minsupp*. On peut remarquer que le temps d'exécution de *STI-PS* est moins important que celui de *PrefixSpan* et de *SPMF*. Il est par contre plus élevé que celui de *SPAM*. Effectivement, l'algorithme *SPAM* extrait les mêmes séquences que *PrefixSpan*, cependant, il met en place une représentation binaire de la base de séquences et des items fréquents. Par conséquent, il réalise des opérations binaires moins coûteuses en temps de calcul et en espace mémoire.

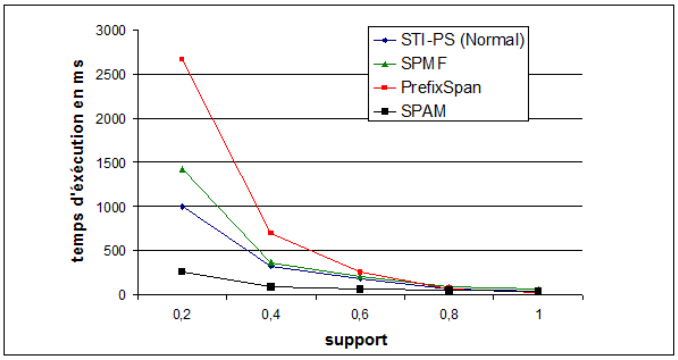


FIGURE 5.4 – Évaluation du temps d'exécution en fonction du support minimal

La figure 5.5 illustre les variations de la mémoire maximale occupée pour chaque exécution des quatre algorithmes. La quantité de mémoire maximale occupée par notre algorithme se rapproche de la mémoire occupée par les deux algorithmes *SPMF* et *PrefixSpan*. Par contre, elle

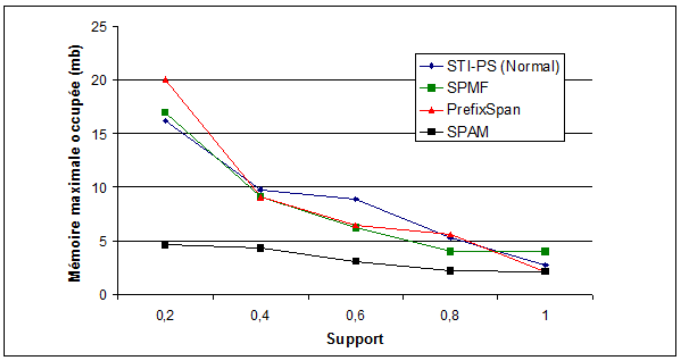


FIGURE 5.5 – Évaluation de la mémoire maximale occupée par les algorithmes du support minimal

est largement supérieure à celle occupée par *SPAM*, car la représentation des données en *BitMap* permet de réaliser un gain de l'ordre de 60%. Pour une taille de fenêtre nulle, le comportement de *STI-PS* se rapproche de celui de *SPMF* et de celui de *PrefixSpan*.

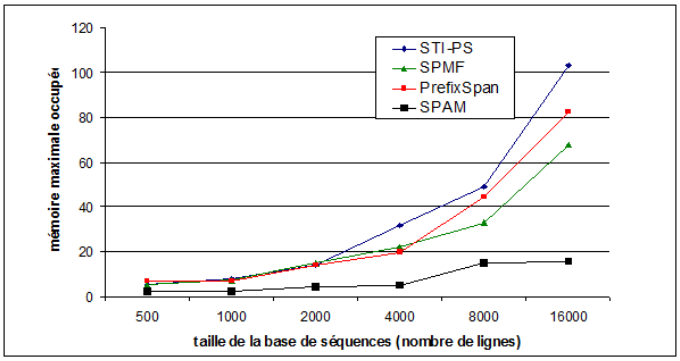


FIGURE 5.6 – Évolution de la mémoire maximale occupée en fonction de la taille de la base de séquences pour $minsupp = 0.5$

La figure 5.6 montre la variation de la mémoire maximale occupée par les quatre algorithmes en faisant varier la taille de la base de séquence. Nous remarquons que les comportements des algorithmes *SPMF* et *STI-PS* sont similaires.

L'évaluation de la variation du temps d'exécution des algorithmes en fonction de la taille de la base de séquence est représentée dans la figure 5.7. La courbe représentative du temps d'exécution de *SPAM* est plus basse pour les trois plus petites bases de données, elle est cependant plus haute pour les autres bases de données. Ceci est dû au coût de construction des représentations binaires des données volumineuses.

Les performances d'exécution de l'algorithme *STI-PS* sont équivalentes à celles de *SPMF*.

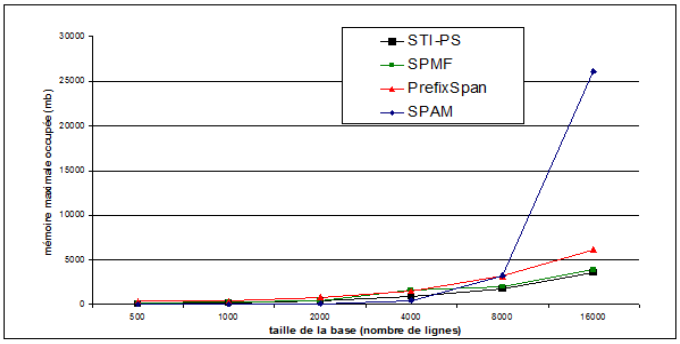


FIGURE 5.7 – Évolution du temps d'exécution en fonction de la taille de la base de séquences pour $minsupp = 0.5$

Elles devancent celles de l'algorithme pionnier de l'approche d'extraction à la « FP-Growth ». Cependant, l'algorithme *SPAM* a de meilleures performances d'exécution que notre algorithme grâce à la représentation binaire des données.

Nous pouvons donc conclure qu'une représentation binaire de la base des séquences et des évènements permettrait d'améliorer significativement notre algorithme.

3.2 Validation des STI fréquentes

Dans cette section, nous présentons, dans un premier temps, une évaluation quantitative et qualitative des motifs extraits par *STI-Ps* par rapport à ceux extraits par l'algorithme *GSPM* [HY06]. Pour cela, nous étudions le nombre et la nature des motifs extraits par les deux algorithmes.

Dans un second temps, nous présentons une estimation de « l'utilité » des *STI-PS* fréquentes pour la prévision des applications des tâches de maintenance à partir d'un historique d'utilisation d'avions.

Évaluation des STI fréquentes

Données synthétiques Cette partie présente une comparaison des séquences extraites par l'algorithme *STI-PS* avec celles obtenues par *GSPM* l'algorithme présenté dans [HY06]. Les deux algorithmes sont basés sur *prefixSpan*. Ils se distinguent par le fait qu'ils utilisent des méthodes différentes pour regrouper les transactions : *GSPM* se base sur une fonction par pallier qui s'apparente à une fenêtre non glissante alors que notre algorithme utilise une fenêtre glissante.

Pour que la comparaison ait un sens, à chaque fois que nous fixons la valeur de la fenêtre glissante ws pour notre algorithme, nous fixons la fonction palier de *GSPM* à $f(t) = \lfloor 1/ws \rfloor$. Nous donnons ci-dessous un exemple expliquant le fonctionnement de *GSPM* et renvoyons le

lecteur à [HY06].

Exemple 41. *Considérons la base de séquences* $\{S_1 = \langle(0, A)(1, B)(2, C)(3, F)(4, B)\ (6, G)\rangle$, $S_2 = \langle(0, A)(1, C)(2, B)(3, D)(4, F)(5, G)\rangle\}$, *un support minimal* $minsupp = 2$, *une fenêtre glissante* $ws = 2$ *et une fonction par paliers* $f(t) = \lfloor t/2 \rfloor$. *Les intervalles associés par* GSPM *seront donc de la forme* $[2 \times f(t), 2 \times (f(t) + 1)[$. *L'algorithme extrait d'abord les 1-séquences fréquentes suivantes A, B, C, F et G (l'estampille de toutes ces 1-séquences correspond à l'intervalle nul). En considérant la séquence B, la projection de la base fournit les séquences suivantes :* $\{S'_1 = \langle(1, C)(2, F)\ (3, B)(5, G)\rangle$, $S''_1 = \langle(2, G)\rangle$, $S'_2 = \langle(2, F)(3, G)\rangle\}$. *A partir de cet ensemble, le motif* $([2, 4[, F)$ *est identifié. Il est considéré comme fréquent car (1) F apparaît dans* S'_1 *et* S'_2 *et (2) dans les deux cas,* $f(t) = \lfloor t/2 \rfloor = 1$. *Pour l'intervalle associé, on applique* $[2.f(t), 2.(f(t)+1)[$ *ce qui donne* $[2, 4[$. *Dans la même projection, G apparaît dans 3 séquences. Dans* S''_1 *et* S'_2 *on a* $f(t) = 1$ *alors que dans* S'_1 *on a* $f(t) = 2$. *Ainsi, seule* $([2, 4[, G)$ *est extraite.* $([4, 6[, G)$ *n'est pas considérée comme fréquente.*

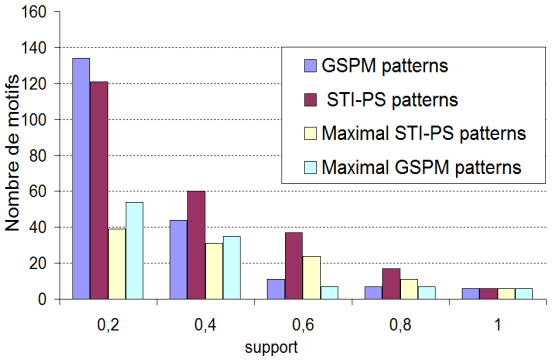
Comme *STI-PS*, *GSPM* est développés en JAVA en utilisant la même version [3] de prefixSpan présentée dans [FVNN08]. Il est aussi implémenté sur la même machine Windows 7(64), Intel(R) Core(TM) 3 CPU 2.40 GHz Avec 3 GO RAM.

Nous comparons les séquences extraites par les deux méthodes en utilisant des données synthétiques. Les séquences contiennent 7 évènements différents et se caractérisent par un écart moyen entre les transactions de 3 unités temporelles et une longueur moyenne de 15 transactions par séquence. Lors de l'extraction, les contraintes mingap (respectivement maxgap, min_whole_interval et max_whole_interval) sont fixées à 0 (resp. 1, 0 et 15). La base de séquences utilisée contient 12 séquences. Nous avons délibérément choisi des jeux de données de petite taille car nous fondons notre étude non pas sur les temps d'exécution (évalués dans l section précédente) mais plutôt sur les résultats obtenus et plus précisément sur les nombres de séquences extraites.

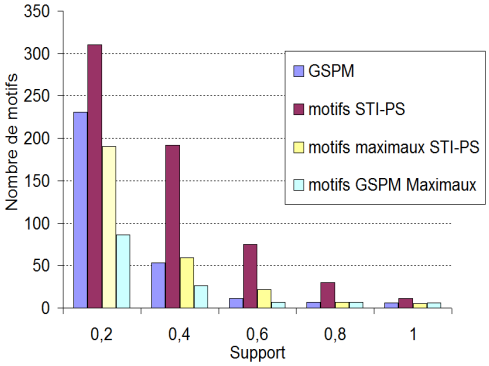
A travers cette expérimentation, notre objectif est de valider notre approche en vérifiant si les résultats extraits sont intéressants dans le cadre de l'application que nous ciblons. En effet, vu que notre approche est plus tolérante vis à vis de la chronologie des évènements, il est naturel de s'attendre à ce qu'on extraie *plus* d'informations. La Figure 5.8 illustre les résultats obtenus en faisant varier la valeur du support et la taille des deux opérateurs de regroupement. Pour chaque combinaison des valeurs de ces paramètres, nous avons mesuré le nombre de séquences extraites

3. http://www.philippe-fournier-viger.com/spmf/index.php

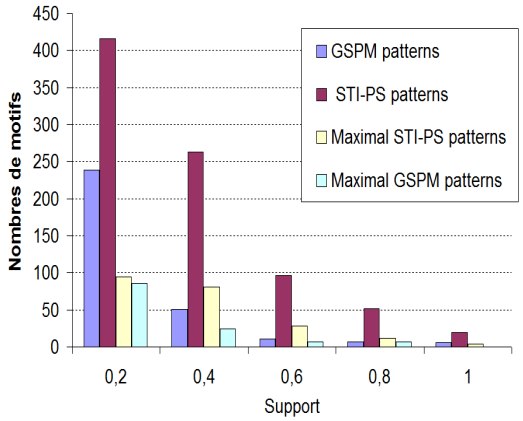
par les deux méthodes. De plus, pour chaque algorithme, nous avons calculé les séquences *maximales* à partir des résultats obtenus.



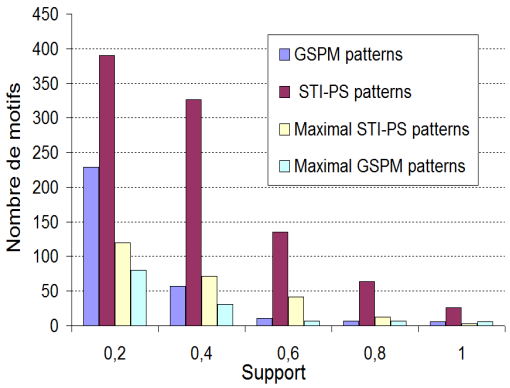
(a) WS=1, $f(t) = \lfloor t/1 \rfloor$



(b) WS=3, $f(t) = \lfloor t/3 \rfloor$



(c) WS=5, f(t)= $\lfloor t/5 \rfloor$



(d) WS=7, f(t)= $\lfloor t/7 \rfloor$

FIGURE 5.8 – Comparaison du nombre de séquences extraites en faisant varier le support, la taille de ws et la largeur du palier de la fonction

Les figures 5.8a (resp. 5.8b 5.8c 5.8d) illustrent les variations des tailles des résultats sous différentes valeurs de regroupement. Elles montrent que le nombre de séquences extraites par *STI_PS* est beaucoup plus important que ceux obtenus par *GSPM*. Ceci est expliqué par l'application de contraintes temporelles plus souples. En effet, la fenêtre glissante regroupe les transactions de proche en proche et permet d'avoir toutes les combinaisons possibles de fusions mais aussi d'extraire des séquences plus longues. Aussi, le retour en arrière lors de la projection permet de prendre en compte plus d'évènements. La souplesse temporelle associe à la fréquence d'un évène-

ment des occurrences décalées puisque les fréquents sont associés à des intervalles. Elle augmente aussi la taille de la projection (retour arrière) en prenant en compte plus d'évènements que la projection déployée par *GSPM*. Ces deux derniers point permettent de révéler des sous séquences fréquentes qui ne le sont pas en utilisant d'autres méthodes d'extraction.

Notons cependant que les résultats présentés dans la figure 5.8a ne sont pas cohérents. D'une part, pour $minsupp = 0.2$ l'algorithme *GSPM* extrait plus de séquences que *STI-PS*. D'autre part, pour les autres valeurs du support, le décalage entre le nombre de motifs retournés par les deux algorithmes n'est pas très élevé. Ceci est dû au fait que le paramètre de regroupement de *STI-PS* $ws = 1$ n'est pas équivalent à celui de *GSPM* $f(t) = \lfloor t \rfloor = 1$. Effectivement, une telle valeur de la fonction n'applique aucune fusion entre les transactions, alors que ws regroupe les transactions espacées d'une unité temporelle.

Le tableau 5.2 illustre en détail les nombres de séquences extraites par les deux méthodes pour une valeur de regroupement variable et un support égal à 0.4. Notons que lorsque les deux méthodes fournissent des séquences de même longueur (correspondance de la figure 5.8a dans le tableau 5.2), les séquences maximales extraites par notre approche sont moins nombreuses que celles obtenues par *GSPM*. Ce cas de figure est illustré dans l'exemple 42. Cependant, lorsque les séquences de *STI_PS* sont plus longues que celles retournées par *GSPM* les maximales sont plus nombreuses et leur nombre est majoritairement représenté par des motifs de longueurs supérieures aux séquences maximales extraites par *GSPM*. Notons enfin que le nombre de séquences maximales extraites par notre approche reste comparables à celles de *GSPM*.

Exemple 42. *Si l'on reprend l'exemple 41, on peut vérifier que les plus longues séquences maximales que* GSPM *extrait sont :* $\langle([0,0[,B)([2,4[,F)\rangle$, $\langle([0,0[,G)\rangle$, $\langle([0,0[,A)\rangle$, $\langle([0,0[,C)\rangle$. STI_PS *extrait la longue séquence suivante* $\langle([0,2],ABC)\ ([3,4],F)\ ([5,6],G)\rangle$. *Si on considère la séquence* $\langle([0,0[,B)([2,4[,F)\rangle$ *extraite par* GSPM, *elle traduit le fait que* F apparaît dans l'intervalle $[2,4[$ après B. *La séquence* $\langle([0,2],ABC)([3,4],F)([5,6],G)\rangle$ *obtenue par* STI_PS *exprime, entre autres, le fait que* F apparaît dans l'intervalle $[3-2=1, 4-0=4]$ après B. *Vu que* $[1,4]$ *contient* $[2,4[$, *on peut donc dire que la séquence maximale extraite par notre approche inclut toutes les séquences maximales extraites par* GSPM *et ce en tolérant plus d'incertitude.*

ws	GSPM maximaux	STI-PrefixSpan	STI-PS maximaux
1	$L_1 = 13$, $L_2 = 21$, $L_3 = 1$	$L_1 = 17$, $L_2 = 39$, $L_3 = 14$	$L_2 = 21$, $L_4 = 14$
3	$L_1 = 9$, $L_2 = 12$, $L_3 = 5$	$L_1 = 7$, $L_2 = 53$, $L_5 = 3$ $L_3 = 96$, $L_4 = 26$,	$L_3 = 30$, $L_4 = 26$, $L_5 = 3$
5	$L_1 = 9$, $L_2 = 13$, $L_3 = 2$	$L_1 = 7$, $L_2 = 55$, $L_3 = 133$, $L_4 = 75$, $L_5 = 9$, $L_6 = 1$	$L_3 = 26$, $L_4 = 42$, $L_5 = 13$, $L_6 = 1$
7	$L_1 = 9$, $L_2 = 19$, $L_3 = 3$	$L_1 = 7$, $L_2 = 51$, $L_3 = 115$, $L_4 = 88$, $L_5 = 21$, $L_6 = 4$	$L_3 = 24$, $L_4 = 29$, $L_5 = 12$, $L_6 = 4$

Tableau 5.2 – Nombre de i-séquences (L_i) extraites en fonction de la variation de la taille de ws et un support égal à 0.4

Évaluation de la pertinence des STI fréquentes

Nous présentons dans ce qui suit une évaluation de la pertinence des *STI* fréquentes extraites par l'algorithme *STI-PS*. Ces *STI* sont extraites à partir d'un historique d'utilisation d'une flotte de véhicules incluant des missions de vols et des réparations sur les véhicules.

A partir de l'historique initial, nous construisons pour chaque tâche de maintenance une base de séquences historiques à laquelle nous appliquons l'algorithme *STI-PS*. Ce dernier, d'extrait les utilisations « typiques », qui corrèlent les séquences d'utilisations fréquentes et l'application de la tâche de maintenance. Les utilisations « typiques » ont la forme de *STI*.

Les séquences extraites permettent de prévoir les éventuelles applications de la tâche de maintenance. En effet, la correspondance entre dans la prévision des applications des tâches de maintenances d'avions. La prévision est possible grâce à la mise en correspondance entre un historique récent et une utilisation typique.

Description des données réelles Nous évaluons dans ce qui suit la pertinence des *STI* fréquentes extraites par notre algorithme à travers l'évaluation de leurs capacités de prévisions des tâches de maintenance. Pour cela nous disposons d'un historique séquentiel datée se rapportant à neuf mois de mise en service et de réparations d'une flotte d'avions comptant six véhicules de petite taille.

L'historique contient des détails sur les missions de vols effectuées et sur les réparations appliquées. Un historique de missions détaille les informations suivantes : L'avion qui l'a effectuée, la durée de la mission, la charge de l'avion et sa consommation en fuel. Un historique de réparation

spécifie l'avion sur lequel a été appliquée la tâche de maintenance, la date de la maintenance, le type de la tâche ainsi que sa référence.

Les données numériques sont transformées pour représenter la durée de vol en trois paliers qui différencient entre le court, moyen et long courrier. De la même manière, la charge de l'avion et sa consommation de fuel sont normalisées en trois valeurs respectivement légère, moyenne et lourde et basse moyenne et haute. La temporalité des différents évènements apparaissant dans l'historique disponible est transformée telle que l'unité temporelle qui gère les estampilles des séquences est le jour (24 heures).

Sur les neufs mois d'historique nous disposons de 182 applications différentes de tâches de maintenance. La plupart d'entre elles n'apparaissent pas un nombre suffisant de fois de telle sorte que, la décomposition de l'historique leur associe des bases de séquences qui ne sont pas suffisamment « consistantes » (ne contiennent pas assez de séquences). Dans ce cas, les « utilisations typiques extraites ne sont pas pertinentes et ne permettent pas de construire des prévisions significatives. Pour évaluer la pertinence des *STI* extraites par notre algorithme, nous nous restreignons à l'extraction des « utilisations typiques » des seize tâches de maintenance les plus redondantes dans nos jeux de données réelles.

Résultats de l'expérimentation Nous partitionnons les données historiques des seize tâches de maintenance en deux jeux de données. Le premier inclut l'historique de cinq avions et est utilisé pour extraire les séquences temporelles par intervalles fréquentes, il s'agit du jeu de données de test. Le second jeu de données est représenté par l'historique du sixième avion, il constitue le jeu de validation qui permettra d'évaluer la pertinence des « utilisations typiques » extraites à partir des données de test.

Ces données sont partitionnées en seize bases de séquences correspondant aux seize tâches de maintenance. L'algorithme *STI-PS* est appliqué à toutes les bases avec les mêmes paramètres d'extraction. Les paramètres d'extraction choisis sont les suivants :$minsupp = 50\%$, $mingap = 0$, $maxgap = 3$, $min\_whole\_interval = 0$ et $max\_whole\_interval = 18$.

Rappelons qu'une « utilisation typique » représente une corrélation entre une séquence d'utilisation et l'application d'une tâche de maintenance. Elle précise pour chaque évènement qui y apparait le moment de son occurrence (par rapport à l'apparition du premier) avec un degré d'incertitude maximal fixé lors de l'extraction à travers la valeur du paramètre taille de la fenêtre glissante.

La validation des « utilisations typiques » se fait par le calcul de la confiance d'une STI dans le

jeu de données de validation. Intuitivement, la confiance d'un comportement critique est le ratio d'implication de la tâche de maintenance qu'elle concerne par les « utilisations typiques » par rapport aux autres tâches de maintenance. Pour un comportement typique $SI = \langle SU([m, M], T)\rangle$ il s'agit du ratio des apparitions de la successions *utilisation* $\rightarrow$ *maintenance* par rapport au nombre total d'apparition de la partie utilisation. Elle est notée

$$Conf(SI) = \frac{|\ \{S' \in S : S'\ SI\}\ |}{|\ \{S' \in S; S' \sqsupseteq SU\}\ |}$$

Partir de chaque base de séquences « test » se rapportant à une des tâches de maintenance étudié, nous appliquons *STI-PS* pour extraire les utilisations typiques. Pour chacune des séquences fréquentes extraite nous calculons sa confiance à partir des données de validation. Nous calculons la confiance de chaque utilisation typique extraite à partir du jeu de données de validation, cette dernière est considérée faible si elle a une valeur dans $]0, 0.5]$, elle est considérée moyenne si sa valeur est incluse dans l'intervalle $]0.5, 0.7]$, elle est haute si sa valeur est incluse dans l'intervalle $]0.7, 1]$.

Le tableau 5.3 visualise les résultats de validation des *STI* fréquentes extraites. La première colonne du tableau représente la référence de la tâche, la seconde le nombre de séquence dans la base de test associée. La troisième colonne représente le nombre d'utilisation typique (STI) extraites. Par la suite, successivement le reste des colonnes du tableau indiquent le pourcentage de STI qui ont une confiance nulle, faible et haute. La confiance moyenne n'est pas représentée car aucune des STI extraites ne présente de confiance faisant partie de cette catégorie.

Les utilisations typiques extraites à partir des bases de séquences relatives aux tâches de maintenance 1, 2 et 4 ont des valeurs de confiances nulles. Ce résultat est dû au fait que les tâches de maintenance correspondantes n'apparaissent pas dans les données séquentielles de validation. On peut remarquer que la confiance des « utilisations typiques » extraites à partir des base de séquences relativement grande est élevées. Ce résultat permet de confirmer la validité des séquences temporelles extraites lorsque la tâche de maintenance à prédire s'est déjà réalisée un nombre suffisant de fois.

4 Conclusion

L'algorithme *STI-PS* est dérivé de l'approche « FP-growth » il extrait des séquences temporelles par intervalles à partir de séquences temporelles estampillées de manière discrète et autorise l'application des contraintes temporelles suivantes : *mingap*, *maxgap*, *min_whole_interval*, *max_whole_interval* et *ws*.

Tâche	Taille de la base	Nombre de STI	STI non valide (%)	STI Valide (%)	Confiance Faible (%)	Confiance Haute (%)
1	4	2506	100	0	0	0
2	38	8949	100	0	0	0
3	115	218	1.3	9.7	24.7	72.9
4	12	358	100	0	0	0
5	62	9274	9.5	90.5	31.5	58.1
6	7	109	10.9	89.9	89.9	0
7	7	103	4.85	91.26	91.26	0
8	7	153	9.1	89.9	89.9	0
9	13	6	16	84	84	0
10	13	30	26	73	73	0
11	15	157	3.8	92.35	92.35	0
12	11	2697	2,96	96,58	96,58	0
13	78	243	1.64	95,47	13,58	81,89
14	75	30	23,33	73,33	73,33	0
15	117	207	2,89	91,30	20,28	71,01
16	99	401	0,49	99,50	2,24	97,25

Tableau 5.3 – Validation des « utilisations typiques » extraites pour seize tâches de maintenance ($1\% \lesssim$ Confiance Faible $\lesssim 50\% \leq$ Confiance Moyenne $\lesssim 70$ % Confiance Haute $\leq 100\%$)

Dans ce chapitre, nous avons d'abord présenté les aspects techniques de l'implémentation de cet algorithme et de l'amélioration de ses performances.

Par la suite, une évaluation des performances d'exécution de notre algorithme par rapport à différents algorithmes faisant partie de l'approche « FP-Growth » permet de conclure que ce dernier devance la plupart de ces algorithmes et présente des performances satisfaisantes en terme de temps d'exécution et d'occupation maximale de la mémoire.

Aussi, nous avons évalué la pertinence des Séquences temporelles par intervalles d'incertitude. Une première comparaison qualitative et quantitative, par rapport à celles extraites par l'algorithme *GSPM* [HY06], montre que l'intégration de la séquence permet d'extraire plus de comportements fréquents et que ces comportements couvrent plus d'informations. En effet, les

STI extraites sont plus nombreuses mais également plus longues. Ce phénomène est dû à la relaxation instaurée par la fenêtre glissante. Ces expérimentations ont fait l'objet de deux publications [BZMMS12a, BZMMS12b].

Une deuxième expérimentation a permis d'évaluer la pertinence des *STI* extraites par notre algorithme à partir de données réelles dans le cadre de prévisions de tâches maintenance d'équipements aéronautiques.

L'historique dont nous disposons ne permet pas de tester la prévision de l'ensemble des 182 tâches de maintenance appliquées. Effectivement, certaines tâches ont une faible fréquence d'application sur une durée de neuf mois (l'ensemble des données de l'étude) génèrent peu de séquences et ne permettent pas d'extraire des « utilisations typiques » significatives.

Sur un historique plus fourni, l'extraction d'utilisation typique associées à ce type de tâches nécessite l'utilisation d'un grand espace mémoire. Effectivement, lorsque les séquences sont longues la taille des projections successives est aussi importante et le nombre d'appels de la fonction récursive de l'algorithme est plus grand. Afin de garantir les performances de notre algorithme lors du traitement de telles données, les futurs travaux porteront sur l'amélioration des performances de l'algorithme en instaurant une représentation binaire des données séquentielles et du traitement des évènements fréquents. En effet, les résultats des expérimentations présentés à la section 3.1 montrent qu'un tel type d'encodage des données permet de réaliser des gains important en termes de temps d'exécution et d'occupation mémoire.

Ainsi, l'optimisation de l'extraction des motifs maximaux permet de réduire les opérations de calcul des évènements fréquents. En effet, il n'est pas efficace d'extraire d'abord, *tous* les motifs et ensuite ne retenir que les maximaux. Une autre amélioration consiste en la distribution des traitements effectués par les principales branches d'extractions.

Conclusion

Le travail de cette thèse a été motivé par les éléments suivants :

- La maintenance aéronautique est un enjeu économique central pour les compagnies aériennes.
- Ces compagnies disposent d'un grand nombre de données sur leurs véhicules, celles-ci sont souvent sous exploitées et leur valorisation est un défi des plus complexes.

La contribution développée dans ce mémoire participe à la résolution de ces deux problématiques en proposant une méthode d'optimisation de la planification d'exécution des tâches de maintenance basée sur l'analyse des données qui sont acquises sur les avions durant leur exploitation.

Préparation et prétraitement des données aéronautique

A partir de données hétérogènes et provenant de différents intervenants de la gestion de la flotte, nous avons étudié les relations fonctionnelles et structurelles entre elles dans le but créer une approche de consolidation et de mise en relation des informations disponibles.

Nous avons, dans un premier temps, mis en relation les données via un alignement temporel. Ce dernier permet de considérer comme un flux séquentiel les informations hétérogènes.

Dans un deuxième temps, une analyse plus approfondie des relations sémantiques et fonctionnelles nous a permis de combiner deux types d'organisation des données « séquentielles » :

- La première découpe le flux de données en séquences temporelles qui associent historiques d'utilisations et historiques d'application de tâche de maintenance.
- La deuxième organise les séquences en bases de séquences orchestrées par une relation hiérarchique.

La première organisation crée une relation d'association entre les données d'utilisation et la tâche de maintenance appliquée. La seconde regroupe d'abord les séquences reliant les utilisations à la même réparation en base de séquences. Par la suite, les bases de séquences sont gérées par une structure arborescente qui illustre la structure des véhicules étudiés.

Ces deux types d'organisations permettent de corréler les données hétérogènes et de les consolider avec la structure fonctionnelle et organisationnelle des systèmes étudiés. Ainsi, l'analyse des données peut se faire à différents niveaux de granularité laissant le choix de la prévision des tâches de maintenance par équipement spécifique ou sous partie du véhicule.

Des améliorations peuvent être apportées pour créer des relations de dépendances plus fortes entre les bases de séquences et permettre de faire des prévisions validées non seulement sur une partie des données mais aussi par l'ensemble des structures organisées.

Définition et extraction des séquences temporelles par intervalles d'incertitude

A partir de chaque base de séquences, les utilisations typiques représentent les séquences fréquentes qui permettent de prédire l'application d'une tâche de maintenance. Ces utilisations, issues des données historiques, doivent représenter au mieux les séquences d'utilisations.

Pour automatiser l'extraction des utilisations typiques, de manière à représenter au mieux celles qui le seraient par des experts de la maintenance aéronautique, nous avons défini les séquences temporelles par intervalles d'incertitudes : les *STIs*. Elles représentent les comportements fréquents apparaissant avant les applications précédentes d'une tâche de maintenance. Ce sont des séquences temporelles car la prévision des tâches de maintenance est plus précise et plus utile lorsqu'elle est associée à une approximation temporelle. Elles représentent aussi une relaxation temporelle des évènements au sein des transactions. Cette relaxation est concrétisée par des intervalles qui représentent une incertitude contrôlée de leurs apparitions ponctuelles. Ce dernier point permet d'extraire des comportements intéressants qui ne sont pas relevés par les méthodes d'extraction vues dans la littérature.

Pour extraire de telles séquences, nous avons défini l'algorithme *STI-PS* qui applique une approche d'extraction de type « FP-Growth ».

Les séquences « intéressantes » sont extraites de manière récursive en parcourant des parties de la base de séquence initiale. *STI-PS* intègre la fenêtre glissante à ses principales fonctionnalités pour identifier et construire les STI. D'une part, la projection classique est modifiée pour prendre en compte un retour arrière qui permet de considérer certains évènements en amont du motif à étendre. De cette façon, nous avons instauré un ordre de traitement des évènements rapprochés afin de n'analyser qu'une seule fois les situations de rapprochement entre deux extensions d'une même transaction. D'autre part, la fonction de sélection d'évènements fréquents identifie les 1-STI fréquentes en considérant des occurrences temporellement distinctes et proches. Nous avons ainsi défini une fonction de concaténation qui permet d'identifier des motifs cohérents. Plusieurs améliorations peuvent être apportées *STI-PS* lesquelles seront présentées et motivées suite au constat réalisé sur les expérimentations de notre algorithme.

Expérimentation et évaluation

Par la suite, nous avons conduit différentes expérimentations pour statuer, d'une part, sur les performances de notre algorithme en termes de temps d'exécution et d'occupation mémoire

et d'autre part, sur la pertinence de la qualité et la quantité des séquences temporelles par intervalles d'incertitudes retournées.
Les évaluations des performances de l'algorithme ont montré que les relaxations de l'extraction des STI n'affectent pas les performances de l'algorithme. En effet, son temps d'exécution et l'occupation mémoire maximale par les différentes projections sont équivalentes à celles des autres algorithmes qui implémentent la même approche d'extraction ([FVNN08] [PHMA+01] et [HY06]). Cependant, ses performances sont moins efficaces que celles de *SPAM* qui met en place une représentation binaire de la base de séquences réduisant ainsi l'occupation de la mémoire et le temps de calcul. Toutefois, cet algorithme ne permet pas d'extraire des séquences temporelles, il ne répond donc pas aux besoins d'extraction de notre méthode.

Cette expérimentation nous a permis de conclure que la représentation binaire des données initiales et des calculs de fréquences est une perspective d'amélioration des performances de *STI-PS*. L'exploration de cette piste nous a conduit à une représentation « bitmap » des séquences de la base et des évènements fréquents ainsi qu'à une indexation des temporalités des transactions. Cependant, elle nécessite une mise au point plus approfondie avant d'être présentée.

Pour évaluer la qualité des séquences temporelles extraites, nous avons comparé les *STI* extraites par *STI-PS* avec les séquences temporelles extraites par une méthode similaires [HY06]. Les résultats montrent que, suite aux relaxations autorisées par la fenêtre glissante, *STI-PS* extrait un nombre plus important de motifs fréquents. Ces derniers sont plus nombreux mais aussi plus longs que ceux extraits par *GSPM* [HY06]. Les résultats montrés dans la section 3.2 amènent la perspective d'extraction des motifs maximaux. Ces derniers permettront de réduire considérablement le nombre des fréquents extraits. D'une part, ils en facilitent l'exploitation et d'autre part ils permettent d'améliorer les performances de l'algorithme en explorant moins de possibilités.

Après avoir validé les résultats obtenus par *STI-PS* selon d'autres méthodes d'extractions sur des données synthétiques, nous avons utilisé un historique de données se rapportant à une flotte d'avions et portant sur neuf mois d'exploitation pour valider la pertinence des STI retournées et statuer sur leurs capacités de prévisions de la maintenance.

Étant donné la richesse et la diversité des opérations de la maintenance dans l'aéronautique, les données dont nous disposons ne nous ont permis de prédire qu'une partie des tâches de maintenance des aéronefs étudiés. Pour ces tâches, l'analyse des données a réussi à prédire les opérations de maintenance dans la plupart des cas. Cependant, une validation plus approfondie sur un volume de données plus consistant sera réalisée dans les travaux futurs.

Le travail présenté dans ce mémoire de thèse a permis de mettre en place une méthode de

prévision des tâches de maintenance en spécifiant une approximation temporelle dont l'incertitude est paramétrable par l'utilisateur. Cette méthode permet de prédire les tâches de maintenances aéronautique avec une confiance acceptable (au dessus de 70%). Plusieurs améliorations peuvent y être apportées, en outre les améliorations techniques citées plus haut qui optimiseraient les performances de l'algorithme *STI-PS*, et des améliorations fonctionnelles permettant notamment de mieux quantifier et qualifier la confiance du résultat obtenu.

Nous citons, par exemple, l'utilisation de calcul de score pour l'affinage de la correspondance entre « historique récent » et utilisation typique. Ce score permettra une quantification plus significative de la pertinence des prévisions apportées par notre méthode. De plus, une exploitation plus approfondie de la structure arborescente des données permettra de créer des relations de dépendances entre les utilisations typiques.

Liste des abréviations

AMM Aircraft Maintenance Manual

ATA Air Transport Association

BITE BuIlt In Test Equipment

CSN Cycle Since New

ESF Extraction de Séquences Fréquentes

FP-Growth Frequent Pattarns Growth

IPC Illustrate Part Catalogue

LRU Line Replaceable Unit

MRO Maintenance Repair and Overhaul

P/N Part Number

S/N Serial Number

STI-PS Séquence Temporelle Par Intervalle d'incertitude-PréfixSpan

TSN Time Since New

STI Séquence Temporelle Par Intervalle d'incertitude

Bibliographie

[AFGY02] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435, Edmonton, Alberta, Canada, 2002. ACM.

[AIS93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216, Washington, D.C., 1993. ACM.

[All83] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26, 1983.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of VLDB conference*, pages 487–499, Santiago de Chile, Chile, 1994.

[AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceeding of ICDE conference*, pages 3–15, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[AS96] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns : Generalizations and performance improvements. In *Proceedings of EDBT conference*, pages 3–17, Avignon, France, 1996.

[Bay98] Roberto J. Bayardo. Efficiently mining long patterns from databases. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, volume 27, pages 85–93, Seattle, Washington, USA, 1998. ACM.

[BS08] Abdeljabbar Ben Salem. *Modèles probabilistes de séquences temporelles et fusion de décisions. Application à la classification de défauts de rails et à leur classification.* PhD thesis, Université Henry Poincaré - Nancy 1, 2008.

[BZ09] Asma Ben Zakour. Fouille de données datées hétérogènes optimisant les opérations et la maintenance de véhicules. In *Actes du XXVIIème Congrès INFORSID*, pages 455–456, Toulouse, France„ Mai 2009.

[BZMM+09] Asma Ben Zakour, Sofian Maabout, Mohamed Mosbah, Marc Sistiaga, and Julien Revault. Heterogeneous and dated data for optimizing operations and maintenance on aircrafts, san diego, ca. In *PHM*, Prognostic and Health Managment, 2009.

[BZMMS10] Asma Ben Zakour, Sofian Maabout, Mohamed Mosbah, and Marc Sistiaga. Time constraints extension on frequent sequential patterns. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval KDIR*, pages 281–287, Valencia, Spain, 2010.

[BZMMS12a] Asma Ben Zakour, Sofian Maabout, Mohamed Mosbah, and Marc Sistiaga. Extraction de séquences fréquentes avec intervalles d'incertitudes. In *EGC*, Revue des Nouvelles Technologies de l'Information, pages 213–224, Bordeaux, FRance, 2012. Hermann-Éditions.

[BZMMS12b] Asma Ben Zakour, Sofian Maabout, Mohamed Mosbah, and Marc Sistiaga. Uncertainty interval temporal sequences extraction. In *Internationa conference on information systems technology and managment ICISTM*, volume 285, page 259. Springer, March 28-30 2012.

[BZR11] Asma Ben Zakour and Eva Randria. Case based model to enhance aircraft fleet management and equipment performance. 15th CASI Astronautic conference Astro, Montréal, Canada, April 2011.

[Fau07] Clément Fauré. *Découvertes de motifs pertinents par l'implémentation d'un réseau Bayésien : application à l'industrie aéronautique.* PhD thesis, Institut des sciences appliquées de Lyon, Novembre 2007.

[Fio06] Céline Fiot. Extended time constraints for generalized sequential patterns. Technical report, 2006. Version étendue de l'article soumis à la revue IJWET.

[FS99] Famili Fazel and Létourneau Sylvain. Monitoring of aircraft operation using statistics and machine learning. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference*, pages 9–11, Chicago, Illinois, USA, November 1999.

[FVFNMN10] Philippe Fournier-Viger, Usef Faghihi, Roger Nkambou, and Engelbert Mephu Nguifo. Exploiting sequential patterns found in usersŠ solutions and virtual

tutor behavior to improve assistance in its. *Subscription Prices and Ordering Information*, page 13, 2010.

[FVNN08] Philippe Fournier-Viger, Roger Nkambou, and Engelbert Mephu Nguifo. A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems. In *Proceeding of the 7th Mexican International Conference on Artificial Intelligence,*, pages 765–778, 2008.

[GHZ10] Karam Gouda, Mosab Hassaan, and Mohamed J. Zaki. Prism : An effective approach for frequent sequence mining via prime-block encoding. *Journal of Computer and System Sciences*, 76(1) :88–102, 2010.

[GNPP06] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, and Fabio Pinelli. Mining sequences with temporal annotations. In *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC)*, pages 593–597. ACM, 2006.

[GQ08] Thomas Guyet and René Quiniou. Mining temporal patterns with quantitative intervals. In *Workshops Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, pages 218–227, Pisa,Italy, 2008. IEEE Computer Society.

[GQ11] Thomas Guyet and Rene Quiniou. Extracting temporal patterns from interval-based sequences. In *IJCAI*, pages 1306–1311, Barcelona, Catalonia, Spain, 2011.

[HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining : concepts and techniques*, chapter Mining Stream, Time-Series, and Sequence Data, pages 500–534. Morgan Kaufmann Pub, 2011.

[HY06] Yu Hirate and Hayato Yamana. Generalized sequential pattern mining with item intervals. *JCP*, 1(3) :51–60, 2006.

[JKK99] Mahesh V. Joshi, George Karypis, and Vipin Kumar. A universal formulation of sequential patterns, 1999.

[KPP03] Walter Kosters, Wim Pijls, and Viara Popova. Complexity analysis of depth first and fp-growth implementations of apriori. In Petra Perner and Azriel Rosenfeld, editors, *Machine Learning and Data Mining in Pattern Recognition*, volume 2734 of *Lecture Notes in Computer Science*, pages 77–119. Springer Berlin / Heidelberg, 2003.

[KT09] Jerry Kiernan and Evimaria Terzi. Constructing comprehensive summaries of large event sequences. *ACM Trans. Knowl. Discov. Data*, 3(4) :21 :1–21 :31, dec 2009.

[LC05] Congnan Luo and Soon M. Chung. Efficient mining of maximal sequential patterns using multiple samples. In *Proceeding of the 2005 SIAM international conference on data mining (SDMŠ05), Newport Beach, CA*, pages 415–426, 2005.

[LYD+05] Sylvain Létourneau, Chunsheng Yang, Chris Drummond, Scarlett Elizabeth, Julio Valdés, and Marvin Zaluski. A domain independent data mining methodology for prognostics. In *Conference Proceedings : Essential Technologies for SuccessfulPrognostics*, volume 59, pages 18–21, Virginia Beach, Virginia, USA, April 2005. the Machinery Failure Prevention Technology Society.

[MCP98] Florent Masseglia, Fabienne Cathala, and Pascal Poncelet. The psp approach for mining sequential patterns. In *PKDD*, pages 176–184, Nante, France, 1998.

[Moo96] T.K. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6) :47–60, 1996.

[MPT04] Florent Masseglia, Pascal Poncelet, and Maguelonne Teisseire. Pre-processing time constraints for efficiently mining generalized sequential patterns. In *Temporal Representation and Reasoning,TIME International Symposium*, pages 87–95, Tatihou, Basse Normandie, France, 2004. IEEE.

[MPT09] Florent Masseglia, Pascal Poncelet, and Maguelonne Teisseire. Efficient mining of sequential patterns with time constraints : Reducing the combinations. *Expert Syst. Appl.*, 36(2) :2677–2690, 2009.

[MTV97a] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *DMKD*, 1(3) :259–289, 1997.

[MTV97b] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3) :259–289, 1997.

[PHMA+01] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. Prefixspan : Mining sequential patterns by prefix-projected growth. In *Proceedings of the 17th International Conference on Data Engineering ICDE*, pages 215–224, 2001.

[PHW02] Jian Pei, Jiawei Han, and Wei Wang. Mining sequential patterns with constraints in large databases. In *Proceedings of the eleventh international*

conference on Information and knowledge management CIKM, pages 18–25, Toronto, Ontario, Canada, 2002. ACM.

[PLF99] José M. Péna, Sylvain Létourneau, and Fazel Famili. Application of rough sets algorithms to prediction of aircraft component failure. *Advances in Intelligent Data Analysis*, pages 473–484, 1999.

[PRM+09] Quang-Khai Pham, Guillaume Raschia, Noureddine Mouaddib, Régis Saint-Paul, and Boualem Benatallah. Time sequence summarization to scale up chronology-dependent applications. In *EDBT 2008, 11th International Conference on Extending Database Technology*, pages 1137–1146, Hong Kong, China, 2009.

[RBP09] Julien Rabatel, Sandra Bringay, and Pascal Poncelet. So_mad : Sensor mining for anomaly detection in railway data. In *ICDM*, pages 191–205, 2009.

[SFS97] Letourneau Sylvain, Famili Fazel, and Matwin Stan. Discovering useful knowledge from aircraft operation/maintenance data. In *Workshop on Machine Learning in the Real World*, pages 34–41, London, England, July 1997.

[Tei07] Maguelonne Teisseire. *Autour et alentours des motifs séquentiels.* Hdr, Université Montpellier II - Sciences et Techniques du Languedoc, Dec 2007.

[WC07] Shin-Yi Wu and Yen-Liang Chen. Mining nonambiguous temporal patterns for interval-based events. *IEEE Trans. on Knowl. and Data Eng.*, 19 :742–758, June 2007.

[WH04] Jianyon Wang and Jiawei Han. Bide : Efficient mining of frequent closed sequences. In *International Conference on Data Engineering ICDE*, pages 79–90, Boston, MA, USA, 2004. IEEE.

[WOHD97] A.Rob Wylie, Robert Orchard, Micheal Halasz, and François Dubé. Ids : Improving aircraft fleet maintenance. In *14th National Conference Aritificial Intelligence and Innovative Applications of Artificial Intelligence IAAI*, Rhode Island, USA, july 1997.

[XHA03] Yan Xifeng, Jiawei Han, and Ramin Afshar. Clospan : Mining closed sequential patterns in large datasets. In *SIAM International Conference on Data Mining*, pages 166–177, 2003.

[YCJCWCSY10] Chen Yi-Cheng, Jiang Ji-Chiang, Peng Wen-Chih, and Lee Suh-Yin. An efficient algorithm for mining time interval-based patterns in large database. In *ACM*, Proceedings of CIKM conference, pages 49–58, Hong Kong, China, 2010.

[YL07] Chunsheng Yang and Sylvain Létourneau. Model evaluation for prognostics : Estimating cost saving for the end users. In *ICMLA*, pages 13–15, Florida, USA, December 2007.

[YLZS10] Chunsheng Yang, Sylvain Létourneau, Marvin Zaluski, and Elizabeth Scarlett. Apu fmea validation and its application to fault identification. In International Design Engineering Technical Conference, Computer, and Information in Engineering Conference, editors, *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, pages 15–18, Montreal, Quebec, Canada, August 2010.

[Zak01] Mohammed Javeed Zaki. Spade : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2) :31–60, 2001.