



HAL
open science

Contributions à la modélisation, la simulation et l'analyse de systèmes déterministes et stochastiques par approches événementielles

Leonardo Brenner

► **To cite this version:**

Leonardo Brenner. Contributions à la modélisation, la simulation et l'analyse de systèmes déterministes et stochastiques par approches événementielles. Informatique [cs]. Université de reims Champagne-Ardenne, 2024. tel-04647788

HAL Id: tel-04647788

<https://hal.univ-reims.fr/tel-04647788v1>

Submitted on 15 Jul 2024

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

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

HABILITATION À DIRIGER DES RECHERCHES



présenté à

L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

École doctorale Sciences du Numérique et de l'Ingénieur

Leonardo BRENNER

Contributions à la modélisation, la simulation et l'analyse de systèmes déterministes et stochastiques par approches événementielles

Soutenue publiquement le 9 Juillet 2024

Composition du jury :

<i>Président du jury :</i>	LUIZ ANGELO STEFFENEL	Professeur à l'Université de Reims Champagne-Ardenne
<i>Rapporteurs :</i>	KAMEL BARKAOUI	Professeur au Conservatoire National des Arts et Métiers
	MOHAMED GHAZEL	Directeur de recherche à l'Université Gustave Eiffel
	SÉBASTIEN LAHAYE	Professeur à l'Université d'Angers
<i>Examineurs :</i>	HERVÉ MARCHAND	Chargé de recherche HDR à l'INRIA Rennes
<i>Directrice :</i>	ISABEL DEMONGODIN	Professeure à l'Aix-Marseille Université

Laboratoire d'Informatique en Calcul Intensif et Image pour la Simulation (LICIIS)

À mes filles, Sophie et Alice.

Remerciements

Ce document ne se contente pas de retracer mon parcours scientifique, mais il témoigne également d'une riche expérience humaine, faite d'échanges, d'amitiés et de confiance. À cet égard, je souhaite exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à son aboutissement.

Tout d'abord, je souhaite remercier ma garante, Isabel Demongodin, pour sa confiance et son soutien tout au long de ces quatorze dernières années. Tes conseils ont été essentiels à la réalisation de ce travail.

Je souhaite également exprimer ma reconnaissance aux membres du jury, aux examinateurs, Hervé Marchand et Luiz Angelo Steffanel, et tout particulièrement aux rapporteurs, Kamel Barakaoui, Mohamed Ghazel et Sébastien Lahaye, pour avoir pris le temps d'évaluer mon travail. Vos observations et remarques précieuses ont enrichi ce document et m'ont permis de l'améliorer considérablement.

Une grande partie des travaux présentés ici ont été réalisés pendant que j'étais maître de conférences à Aix-Marseille Université. Je ne pourrais oublier de remercier toute l'équipe MOFED du LIS : Claudia Fridman, Rim Saddem, Amine Hamri, et particulièrement Rabah Ammour et Isabel Demongodin (encore une fois). Malgré mon départ vers des contrées moins ensoleillées, j'espère que nos collaborations seront toujours au rendez-vous. Nos échanges m'ont permis un enrichissement personnel et professionnel. En plus de mon équipe, j'aimerais remercier tous mes collègues du LIS, en particulier, Sana Sellami et Mohamed Zerrougui, un grand merci pour votre bonne humeur et toutes les discussions autour d'un café.

À mes collègues de l'Université de Reims et en particulier du LICIS, que j'ai eu l'opportunité de côtoyer il y a plus de dix ans lors de mon année d'ATER et que je retrouve à nouveau depuis mon retour, merci de l'accueil toujours chaleureux et j'ai hâte de réaliser pleines de nouvelles collaborations.

Je souhaite également remercier Radhia Gaddouri, Wenjing Yang et Ruotian Liu, qui m'ont fait confiance pour le co-encadrement de leurs thèses. C'est grâce à vous que j'ai pu réaliser une partie de ce document.

Je tiens aussi à exprimer ma gratitude envers les collègues avec qui j'ai eu l'opportunité de collaborer au cours de ma carrière. Merci à Alexandro Giua, Dimitri Lefebvre et Said Amari pour vos discussions et conseils toujours pleins de sagesse.

Je souhaite également exprimer toute ma gratitude à mes amis marseillais. Notre séjour à Marseille a été beaucoup plus amusant en vos compagnies. Je veux également dire un très grand merci à Luiz Angelo et Manuele pour toutes ces années d'amitiés.

Je remercie ma famille et ma belle-famille, en particulier mes parents, Neumar et Nelsi. Merci pour votre soutien tout au long de ma vie. Vous êtes des modèles pour moi.

Enfin, je remercie infiniment les trois amours de ma vie, Marina, Sophie et Alice, pour leur amour et leur soutien inconditionnel. Vous avez été ma source de force et de motivation durant toutes ces années.

À toutes et à tous, un grand merci.

Table des matières

Chapitre 1 : Introduction	1
I Modélisation de systèmes déterministes	5
Chapitre 2 : Synthèse des travaux sur les formalismes déterministes	6
2.1 Les Réseaux de Petri	6
2.2 Commande des systèmes hybrides	9
2.3 Sécurité dans les systèmes cyber-physiques	11
Chapitre 3 : Réseaux de Petri Lots Triangulaires	17
3.1 Définition des Réseaux de Petri Lots Triangulaires	17
3.2 Réseaux de Petri Lots Triangulaires Contrôlés	38
3.3 Évolution des RdPLots Triangulaires Contrôlés	42
Chapitre 4 : Modélisation du trafic routier par RdPLots Triangulaires	48
4.1 Contexte d'étude du trafic routier	48
4.2 Quelques notions et concepts du trafic routier	49
4.3 Exemple de modélisation du trafic routier	53
Chapitre 5 : Simuleau : un simulateur pour les Réseaux de Petri Hybrides et Lots	59
5.1 Historique	59
5.2 Architecture logicielle	60
5.3 Description d'un modèle	63
5.4 Sortie de la simulation	65

Chapitre 6 : Bilan des travaux sur les formalismes déterministes	67
II Modélisation de systèmes stochastiques	70
Chapitre 7 : Synthèse des travaux sur les modèles stochastiques	71
7.1 Réseaux d'automates stochastiques	71
7.2 Modélisation d'un algorithme de vol de tâches	74
7.3 Prédiction de l'état du trafic routier basée sur des motifs et les chaînes de Markov	77
Chapitre 8 : Propagation de l'influence par cascade indépendante	81
8.1 Principes de base du modèle de cascade indépendante	81
8.2 Calcul de la probabilité d'activation	86
8.3 Maximisation de l'influence par activation des liens	97
Chapitre 9 : Bilan des travaux sur les formalismes stochastiques	109
Chapitre 10 : Conclusion et perspectives	112
10.1 Conclusion générale	112
10.2 Perspectives	113
Bibliographie	116
Annexe A : Curriculum Vitae	123
A.1 Identification	123
A.2 Parcours professionnel	123
A.3 Formation	124
A.4 Charges collectives	125
Annexe B : Activités liées à l'enseignement	127
B.1 Responsabilités administratives	127
B.2 Matières enseignées	128
B.3 Encadrement de stages et TER	129

Annexe C : Activités liées à la recherche	131
C.1 Encadrements	131
C.2 Rayonnement et diffusion scientifique	133
C.3 Liste de publications	135

Table des figures

2.1	Réseau de Petri avec 3 places et 3 transitions	7
2.2	Noeuds d'un Réseau de Petri Lots Généralisé	8
2.3	Un modèle simple d'un RdPLots généralisé	9
2.4	Schéma d'attaque à un système cyber-physique	12
2.5	Exemple d'un OutSynPN	13
2.6	Évolution du marquage suite à la séquence d'événements aa	14
2.7	LFAI de l'OutSynPN de la figure 2.5	15
3.1	Relation flux-densité dans une place lot triangulaire	19
3.2	États des lots contrôlables par rapport à la densité critique.	20
3.3	Noeuds des Réseau de Petri Lots Triangulaires	21
3.4	Exemple d'un RdPLots Triangulaire	22
3.5	Éclatement du lot contrôlable $C\beta_i^r(\tau)$ en déplacement sans contact	35
3.6	Relation flux-densité triangulaire après variation de la vitesse maximale de la place p_i	39
3.7	États des lots contrôlables après une diminution de la vitesse maximale de p_i ($v_i > v'_i$)	40
3.8	États des lots Contrôlables après une augmentation de la vitesse maximale de p_i : ($v_i < v'_i$)	42
3.9	Algorithme d'évolution d'un modèle RdPLots	44
3.10	Un noeud du graphe d'évolution d'un modèle RdPLots Triangulaire	45
3.11	Graphe d'évolution du modèle RdPLots triangulaire de la figure 3.4.	46
4.1	Diagramme fondamental triangulaire proposé par Daganzo	50
4.2	Tronçon d'autoroute A51 de Marseille vers Aix-en-Provence	54
4.3	Disposition des boucles de détection sur le tronçon de A51	54
4.4	Flux d'entrée en cas réel entre 15h58 et 17h34	55
4.5	RdPLots Triangulaire pour le tronçon de l'autoroute A51	55
4.6	Diagramme fondamental triangulaire de la place lot p_2	56

4.7	Variation du flux de sortie par simulation (lot de sortie) et en cas réel (détecté par B3)	58
5.1	Simuleau tool class diagram	61
5.2	Menu principal de Simuleau	62
5.3	Menu de simulation de Simuleau	62
7.1	Modèle RAS avec 2 automates	72
7.2	Chaîne de Markov équivalente au modèle RAS de la figure 7.1.	74
7.3	Exemple de modélisation : architecture à 3 processeurs	76
7.4	Temps d'exécution	77
7.5	Modèle markovien de chaque région	79
8.1	Exemple d'un réseau social représenté par un graphe.	83
8.2	Exemple d'un modèle de cascade indépendante	85
8.3	Une cellule du graphe d'évolution de la <i>path method</i>	87
8.4	Graphe d'évolution du modèle de la figure 8.2 obtenu par <i>path method</i>	90
8.5	Un modèle de cascade indépendante avec 7 noeuds.	100
8.6	EIP calculée par les quatre algorithmes de sélection pour le réseau <i>Advise-seek</i> avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$	107
8.7	EIP calculée par les quatre algorithmes de sélection pour le réseau <i>HighTec</i> avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$	107

Liste des tableaux

4.1	États du trafic de 15h58-17h34	55
4.2	Liste des événements contrôlés	57
7.1	Ensemble de données de la mobilité des taxis	80
7.2	Précision moyenne globale	80
7.3	Précision moyenne par état du trafic	80
8.1	Résultats d'une simulation du modèle de cascade indépendante de la figure 8.2.	86
8.2	Résultats de la propagation de l'influence du modèle de la figure 8.2	96
8.3	Comparaison des ensembles E_a et des EIP pour les quatre algorithmes de sélection pour le modèle de la figure 8.5	105
8.4	Comparaison des temps de calcul pour la sélection des liens avec les quatre algorithmes de sélection pour le réseau <i>Advise-seek</i> avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$	108
8.5	Comparaison des temps de calcul pour la sélection des liens avec les quatre algorithmes de sélection pour le réseau <i>HighTec</i> avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$	108

Chapitre 1

Introduction

Ce mémoire, rédigé en vue d'obtenir un Habilitation à Diriger des Recherches (HDR), présente mes activités depuis l'obtention de mon titre de doctorat à l'Institut Polytechnique de Grenoble en 2009 jusqu'à présent.

Durant la période de 2009 à 2024, j'ai exercé en tant que maître de conférences à Aix-Marseille Université dans l'équipe MoFED (Modèles et Formalismes à Événements Discrets) au Laboratoire d'Informatique et Systèmes (LIS) où j'ai développé mes travaux de recherches dans le domaine des systèmes à événements discrets. Depuis septembre 2023, je suis sur un poste de Chaire Professeur Junior à l'Université de Reims Champagne-Ardenne. Vu mon changement récent, ce document retrace ma carrière uniquement sur la période 2009 à 2023.

Même si je présente dans ce document uniquement mes travaux de recherche réalisés après ma thèse, mon intérêt pour la recherche remonte à la fin des années 90 où, à la fin de ma licence, j'ai eu l'opportunité de travailler dans l'équipe PEG (*Performance Evaluation Group*) à la PUCRS (*Pontífica Universidade Católica de Rio Grande do Sul*) au Brésil sous la coordination du Professeur Paulo Fernandes. Les travaux de l'équipe portaient sur l'évaluation de performance par les formalismes des Réseaux d'Automates Stochastiques (RAS) et les Réseaux de Files d'Attente. J'ai continué ces travaux avec l'équipe PEG jusqu'au début de mon doctorat en France en 2004. Pendant mon doctorat, j'ai continué mes recherches sur le formalisme RAS dans l'équipe MESCAL (*Middleware efficiently scalable*) sous l'orientation de la Professeure Brigitte Plateau.

Ce n'est qu'après mon recrutement à Marseille que j'ai commencé à m'intéresser de plus près aux formalismes déterministes et les collaborations avec la Professeure Isabel Demongodin m'ont amené à travailler sur différentes classes de Réseaux de Petri.

Tous ces formalismes sont utilisés pour modéliser des Systèmes à Événements Discrets (SED). Ces systèmes sont caractérisés par un changement d'état uniquement lorsqu'un événement se produit. Le mot *discret* fait donc référence à la dynamique événementielle du système.

Pour présenter mes travaux dans ce document, j'ai séparé mes recherches sur les SED en

deux catégories de formalismes. Mes travaux sur les formalismes déterministes regroupent plusieurs recherches sur le formalisme des Réseaux de Petri, plus particulièrement les Réseaux de Petri Lots, et plus récemment les Réseaux de Petri synchronisés avec sorties et constituent la première catégorie. La deuxième catégorie concerne mes travaux sur des formalismes stochastiques et sont basés essentiellement sur les formalismes de Réseaux d'Automates Stochastiques et les Chaînes de Markov.

En ce qui concerne mes recherches sur les formalismes déterministes, elles sont centrées sur le développement du formalisme des Réseaux de Petri (RdP). Au cours des années, plusieurs classes ou extensions des RdP ont été proposées dans la littérature. L'objectif de la plupart de ces extensions est d'adapter le formalisme de base au besoin du système afin de mieux décrire son comportement. Un formalisme adapté permet une meilleure description, plus compacte et plus réaliste. Ce développement plus théorique est la base pour le déploiement des applications plus réelles.

Dans ce contexte de développement théorique comme base nécessaire à un développement applicatif, nous avons proposé une extension du formalisme des Réseaux de Petri Lots (Rd-PLots) pour prendre en compte une relation flux/densité triangulaire que nous trouvons fréquemment dans les modèles dédiés au trafic routier. Cette nouvelle relation flux/densité nous a permis de représenter des phénomènes de congestion et de décongestion observés dans le trafic routier et de calculer avec précision la longueur de congestion.

Toujours sur le formalisme de RdPLots, nous nous sommes intéressés aux méthodes de contrôle d'un SED avec l'objectif de diriger le système vers un état stationnaire. C'est-à-dire, un état où les marquages de places et flux des transitions restent constants. Des méthodes de contrôle pour les SED ne sont pas une nouveauté dans la communauté, mais les méthodes existantes reposent sur une discrétisation du pas de contrôle. L'originalité de notre approche repose sur des méthodes de contrôle purement événementielles. Le contrôle est mis à jour uniquement lorsque certains événements se produisent. Plusieurs méthodes ont été proposées et une comparaison des performances au niveau du nombre d'événements ainsi que le temps nécessaire pour amener le système à l'état stationnaire a été faite.

Plus récemment, mes recherches se sont tournées vers le domaine de la sécurité des systèmes cyber-physiques. Ce type de système est caractérisé par une partie informatique qui collabore pour le contrôle, ou commande, des parties physiques. Toujours avec un point de vue SED, notre objectif a été de proposer une extension des RdP qui permet de modéliser les entrées (commandes) du système ainsi que les sorties (retour des capteurs) de la partie physique. Ces recherches ont abouti à la proposition du formalisme de Réseaux de Petri Synchronisés avec sortie. Sur cette nouvelle classe des RdP, nous avons proposé plusieurs méthodes qui nous indiquent le niveau de sécurité de ce type de systèmes.

Mes travaux sur la deuxième catégorie de formalismes exploitent plus souvent les formalismes de Réseaux d'Automates Stochastiques (RAS) et les Chaînes de Markov avec un objectif d'évaluer les performances d'un système avec une dynamique aléatoire.

Les travaux sur l'évaluation de performances des systèmes parallèles et distribués sont pré-

sents dans mes recherches depuis le début de ma carrière. L'objectif général de ces travaux est de proposer des indices de performances pour des systèmes avec grand espace d'états. Nous retrouvons mes contributions dans ce domaine sur plusieurs publications réalisées au long de ma carrière. Dans ce document, j'utilise les RAS pour présenter un modèle stochastique pour un algorithme d'ordonnancement des tâches dans un système parallèle.

Dans un côté plus applicatif, mes contributions dans les prédictions de l'état du trafic urbain mélangent les modèles de Chaîne de Markov et l'exploration de données réelles à la recherche des patrons de déplacements des taxis. La combinaison de ces deux éléments nous ont permis de proposer un modèle simple mais qui restitue des meilleures prédictions que les modèles précédents.

D'autres contributions qui mélangent les côtés théoriques et applicatifs se trouvent dans le domaine de la propagation de l'influence sur des réseaux sociaux. Cela consiste dans le changement des opinions ou des comportements d'un individu en conséquence des interactions avec les autres. Le principe de propagation de l'influence est largement utilisé en stratégie de marketing pour déclencher un "effet viral" et faire connaître des nouveaux produits à moindre coût. La difficulté est de trouver les individus dans un réseau qui vont maximiser l'influence et faire que le plus grand nombre adopte la nouveauté. Mes recherches dans ce domaine portent, d'un côté, sur la définition des nouvelles méthodes de calcul de la propagation qui sont plus précises ou plus rapides que les méthodes existantes. D'un autre côté, nous avons proposé une approche innovante pour la maximisation de propagation qui est basée sur l'analyse des liens qui connectent deux individus d'un réseau.

Organisation du document

Ce document s'articule autour de deux parties, intitulées *Modélisation de systèmes déterministes* et *Modélisation de systèmes stochastiques* et qui font références aux deux catégories de formalismes mentionnés ci-avant.

La première partie retrace mes contributions sur les systèmes déterministes modélisés par des Réseaux de Petri. Le chapitre 2 introduit le formalisme des Réseaux de Petri et présente brièvement quelques contributions sur les méthodes de contrôle de flux et la sécurité de systèmes cyber-physiques. Les chapitres 3 et 4 détaillent mes recherches sur l'intégration d'une nouvelle relation flux/densité au formalisme de Réseaux de Petri Lots ainsi que son application à la modélisation d'un cas réel du trafic routier. Le chapitre 5 présente l'outil de simulation des Réseaux de Petri que j'ai développé ces dernières années. Je termine cette partie (Chapitre 6) par un bilan global de mes contributions sur cette catégorie de formalismes.

Dans la deuxième partie de ce document, je présente mes contributions sur la modélisation de systèmes stochastiques. De façon similaire à la première partie, je commence par introduire, dans le chapitre 7, le formalisme de Réseaux d'Automates Stochastiques et je présente quelques contributions sur la modélisation d'un algorithme d'ordonnancement de tâches et sur la prédiction de l'état du trafic urbain. Le chapitre 8 est consacré aux travaux sur la propagation

de l'influence. Plusieurs méthodes de calcul de la propagation sont présentées et comparées dans l'objectif de trouver le bon compromis en précision des résultats et vitesse de calcul. La deuxième partie de ce chapitre est dédiée à la maximisation de l'influence sous la contrainte d'un budget maximum pour l'activation des liens d'un réseau. Le dernier chapitre de cette partie (Chapitre 9) présente un bilan global de mes contributions.

Dans le chapitre 10, je dresse une conclusion générale de mes travaux de recherche ainsi que quelques perspectives et travaux à venir.

Première partie

Modélisation de systèmes déterministes

Chapitre 2

Synthèse des travaux sur les formalismes déterministes

Mes travaux sur le formalisme des Réseaux de Petri explorent plusieurs thèmes et champs d'applications. Les thématiques développées sur la base des Réseaux de Petri sont issues principalement des recherches dans le cadre des encadrements de thèse ainsi que dans différents projets auxquels j'ai pu participer.

Je synthétise dans ce chapitre mes travaux sur la commande des systèmes hybrides (Section 2.2) et la sécurité des systèmes cyber-physiques (Section 2.3). Les travaux sur les Réseaux de Petri Lots Triangulaires seront détaillés dans les chapitres suivants.

Afin de présenter mes travaux sur les formalismes déterministes, je commence par une brève introduction des Réseaux de Petri.

2.1 Les Réseaux de Petri

Les Réseaux de Petri sont une classe de formalismes pour la modélisation, la simulation et l'analyse des SED. Un modèle en réseau de Petri est représenté par un graphe biparti composé de deux types de noeuds : *places* et *transitions*. Des arcs orientés relient les places aux transitions et les transitions aux places.

Formellement, un réseau de Petri est défini par :

Définition 2.1. (*Réseau de Petri*)

Un réseau de Petri est un 4-uplet $N = (P, T, Pre, Post)$ tel que :

- *P est un ensemble fini de places ;*
- *T est un ensemble fini de transitions ;*

- $Pre, Post : (P \times T \rightarrow \mathbb{N})$ sont respectivement, les matrices de pré-incidence et post-incidence, désignant le poids des arcs, des places vers les transitions et des transitions vers les places. ■

Lorsque nous associons un nombre entier naturel à chaque place $p \in P$, nous obtenons un réseau de Petri marqué. Le marquage, ensemble des marques du réseau, correspond typiquement à la quantité de ressources disponibles et sont représentés par des jetons.

Exemple 2.1. Un exemple de réseau de Petri avec trois places ($P = \{P_1, P_2, P_3\}$) et trois transitions ($T = \{t_1, t_2, t_3\}$) est présenté dans la figure 2.1. Dans ce modèle, la place P_1 est marquée avec un jeton. La valeur 2 associée à l'arc de t_1 à P_2 indique que 2 jetons seront ajoutés à la P_2 lorsque la transition est franchie. De façon similaire, lorsque la transition t_3 est franchie, 2 jetons sont retirés de la place P_3 mais un seul est ajouté à la place P_1 .

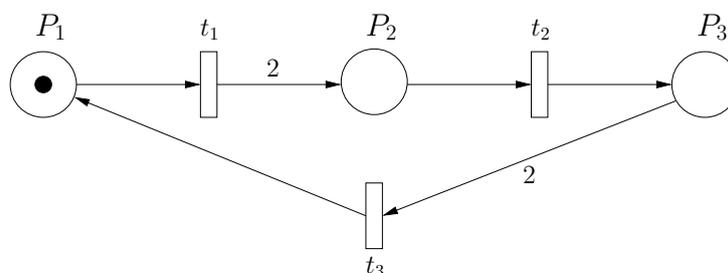


FIGURE 2.1 – Réseau de Petri avec 3 places et 3 transitions ■

L'évolution du marquage au fil du temps se fait par le *franchissement* des transitions. Une transition n'est considérée *franchissable* que si les marquages de toutes les places d'entrée sont supérieurs ou égale aux poids des arcs des places vers la transition. Une fois la transition franchie, les jetons sont retirés des places d'entrée de la transition et ajoutés dans toutes les places de sorties de la transition.

Nous pouvons associer à l'ensemble des transitions une durée qui représente l'attente avant que la transition puisse être franchie et dans ce cas, nous obtenons un réseau de Petri T-temporisé.

Lorsque nous considérons le franchissement des transitions comme un flux de jetons associé à des variables continues, nous obtenons des Réseaux de Petri Continus. Nous avons donc des *places continues* où les marquages de chaque place ne sont plus des entiers naturels mais de valeurs réelles et des *transitions continues* où un flux maximal de franchissement remplace la durée des transitions, dites discrètes [34].

Les Réseaux de Petri Hybrides mélangeant des places et des transitions discrètes et continues ont été introduits par LeBail et al. dans [71]. La littérature est abondante sur les Réseaux de Petri et plusieurs classes et extensions ont été proposées au fil des années [34, 55]. Mes travaux

s'intéressent à une classe des Réseaux de Petri Hybrides Temporisés, appelée Réseaux de Petri Lots Généralisés (RdPLots Généralisés) [35, 37, 41].

Les RdPLots Généralisés enrichissent la classe des Réseaux de Petri Hybrides [34, 55] par l'introduction de nouveaux types de noeuds, appelés noeuds *lots*. Les transitions lots agissent comme les transitions continues avec une sémantique de serveur finis. Les places lots implémentent une dynamique hybride définie par trois caractéristiques continues : une longueur, une vitesse de transfert et une densité maximale. Basé sur le concept de lots en tant que marquage, par exemple, un groupe d'entités se déplace dans une place lot à une certaine vitesse, les RdPLots Généralisés permettent de représenter des flux continus avec des délais variables. Le comportement hybride du modèle repose sur la bascule entre l'évolution en temps continu et une dynamique événementielle. La figure 2.2 présente l'ensemble des noeuds des RdPLots Généralisés.

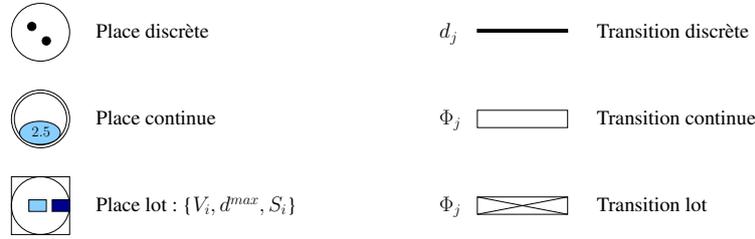


FIGURE 2.2 – Noeuds d'un Réseau de Petri Lots Généralisé

Formellement, un RdPLots Généralisé est défini comme suit :

Définition 2.2. (Réseau de Petri Lots Généralisé [35])

Un réseau de Petri Lots Généralisé (RdPLots Généralisé) est un 6-uplet $N = (P, T, Pre, Post, \gamma, Time)$ tel que :

- $P = P^D \cup P^C \cup P^B$ est un ensemble fini de places partitionné en trois types de classes : discrètes P^D , continues P^C et lots P^B .
- $T = T^D \cup T^C \cup T^B$ est un ensemble fini de transitions partitionné en trois types de classes : discrètes T^D , continues T^C et lots T^B .
- $Pre, Post : (P^D \times T \rightarrow \mathbb{N}) \cup ((P^C \cup P^B) \times T \rightarrow \mathbb{R}_+)$ sont respectivement, les matrices de pre-incidence et post-incidence, désignant le poids des arcs de places vers les transitions et de transitions vers les places.
- $\gamma : P^B \rightarrow \mathbb{R}_+^3$ est la fonction caractéristique de la place lot. Elle associe à chaque place lot $p_i \in P^B$ le triplet $\gamma(p_i) = (V_i, d_i^{max}, S_i)$ qui représente respectivement, la vitesse, la densité maximale et la longueur de cette place lot.
- $Time : T \rightarrow \mathbb{R}_+$ associe un nombre non négatif pour chaque transition :
 - si $t_j \in T^D$, alors $Time(t_j) = d_j$ désigne le délai de franchissement associé à la transition discrète;

- si $t_j \in T^C \cup T^B$, alors $Time(t_j) = \Phi_j$ désigne le flux maximal de franchissement associé aux transitions continue et lot.

■

Le marquage de chaque place lot est constitué par un ensemble de lots ordonnés par leurs positions de tête. Chaque lot β_i^r est décrit par sa longueur, sa densité et la position de tête.

Exemple 2.2. La figure 2.3 présente un modèle d'un RdPLots généralisé composé d'une place discrète Dp , une place continue Cp et une place lot Bp . La place lot Bp est caractérisée par une vitesse de 20, une densité maximale de 100 et une longueur de 10. Nous pouvons constater que le marquage de la place lot est composé d'un lot β_{Bp}^1 de longueur 5, densité 50 et position 8.

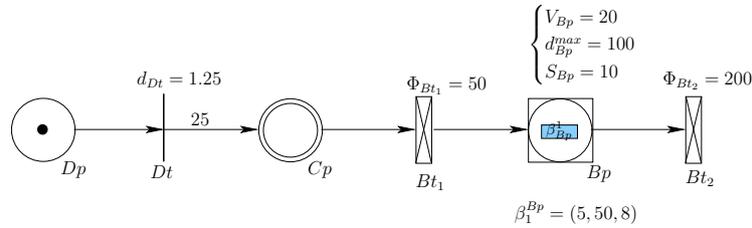


FIGURE 2.3 – Un modèle simple d'un RdPLots généralisé

Cet exemple peut représenter, dans un système manufacturier, une caisse de 25 produits qui arrive dans le système. La caisse est désassemblée par une machine et chaque produit est ajouté dans un convoi de transport. La place Dp et la transition Dt caractérisent la caisse et son délai d'arrivée. Les nombres de produits dans la caisse est caractérisé par le poids de l'arc de transition Dt à la place Cp . Cette place Cp caractérise la machine de désassemblage. La place lots Bp représente le convoi et ces caractéristiques. Les transitions lots Bt_1 et Bt_2 caractérisent le flux maximal de la machine et du convoi.

Je présente dans le chapitre 3 la sémantique et la dynamique associées aux RdPLots, bien qu'adaptées aux RdPLots Triangulaire, les concepts restent les mêmes.

Les sections suivantes décrivent brièvement trois thématiques qui concernent les Réseaux de Petri et que j'ai explorés pendant les dernières années.

2.2 Commande des systèmes hybrides

Le problème considéré dans ces recherches est la commande de systèmes hybrides étudiés du point de vue de la théorie des SED. L'objectif est de déterminer une trajectoire transitoire pour atteindre un état stationnaire à partir d'un état initial. Nous avons proposé des approches

événementielles dans lesquelles l'action de contrôle est mise à jour uniquement lorsque des événements spécifiques se déclenchent.

Afin d'éviter l'introduction de plusieurs définitions et concepts nécessaires à une présentation formelle du problème, je vais présenter les problèmes traités ainsi que les méthodes proposées de façon informelle. Toutes les définitions formelles sont dans les publications associées à ce sujet [74, 75, 76, 77].

Ces travaux ont été développés sur le formalisme de Réseaux de Petri Lots Généralisés Contrôlés (RdPLots Généralisés Contrôlés) [35] sans les noeuds discrets. Autrement dit, nos modèles n'ont pas des places et des transitions discrètes.

La définition d'un état stationnaire en RdPLots Généralisés Contrôlés a été initialement présenté dans [39] et les conditions d'existence et du choix optimal ont été introduites dans [40]. En contrôlant uniquement le flux des transitions continues et lots, nous pouvons guider le système vers un état stationnaire $(\mathbf{m}^s, \boldsymbol{\varphi}^s)$, où le marquage \mathbf{m}^s et le vecteur de flux de franchissement instantané $\boldsymbol{\varphi}^s$ restent constant pour $\tau \geq \tau_s$ où τ_s représente le temps où le système a atteint l'état stationnaire.

Nous avons proposé trois lois de contrôle purement événementielles pour amener le système à un état stationnaire sous hypothèses que le modèle RdPLots généralisé contrôlé soit conservatif, consistant et sans noeud discret.

2.2.1 Contrôle par le flux stationnaire

La première stratégie, appelée loi de contrôle par flux stationnaire (loi SF) [75], est basée sur le principe de transitions ON/OFF [99]. Autrement dit, sous certaines conditions, une transition a son état mis à ON et son flux de franchissement maximal est limité par son flux de franchissement stationnaire. Si les conditions ne sont pas remplies, la transition est mise à OFF et son flux de franchissement est mis à zéro. Le choix du flux de franchissement stationnaire comme seuil permet de construire le marquage régulier [39] des places lots. Il a été démontré qu'un RdPLots dans un état stationnaire peut avoir uniquement deux formes de marquages possibles, ce que nous appelons marquages réguliers. Ces marquages réguliers sont :

1. un seul lot de sortie : $m_i^s = \{\beta_i^o\}$ avec $\beta_i^o = (S_i, d_i^o, S_i)$ et la densité $d_i^o \leq d_i^{\max}$. Autrement dit, la longueur du lot est égale à la longueur de la place p_i .
2. deux lots : $m_i^s = \{\beta_i^e, \beta_i^o\}$ avec $\beta_i^e = (l_i^e, d_i^e, l_i^e)$, $\beta_i^o = (l_i^o, d_i^{\max}, S_i)$ et $l_i^e + l_i^o = S_i$.

Néanmoins, un inconvénient de cette première loi de commande est qu'un problème de blocage peut survenir, c'est-à-dire que la variation du vecteur quantité de marquage peut rester nul et, par conséquent, l'état stationnaire cible peut ne pas être atteint [64, 98]. Une condition sur la quantité de marquage initial est proposée pour éviter les états bloqués et atteindre l'état stationnaire par la loi de contrôle SF.

2.2.2 Contrôle par le flux maximum

Afin d'améliorer les performances temporelles, une deuxième loi de contrôle, appelée loi de contrôle par flux maximal (loi MF) [76], est développée en utilisant le flux maximal de franchissement comme seuil.

La loi de contrôle MF permet d'atteindre l'état stationnaire plus rapidement que la loi de contrôle SF malgré le nombre plus élevé d'événements de contrôle générés.

2.2.3 Contrôle par région d'attractivité

La troisième stratégie appelée loi de contrôle basée sur la quantité de franchissement (loi ZF) [77], est dédiée au cas où la condition de non-blocage sur la quantité de marquage initial n'est pas satisfaite. Dans ce cas, une région attractive de l'état stationnaire est définie et il a été prouvé que toute trajectoire commençant dans cette région est garantie de converger vers l'état stationnaire en utilisant la loi SF de contrôle proposée précédemment. Ainsi, cette troisième loi de contrôle est construite à partir du vecteur de franchissement minimal nécessaire pour atteindre la région attractive à partir d'un marquage initial bloquant.

Mes recherches sur la commande des systèmes hybrides ont commencé avec la thèse de Ruotian Liu, étudiant que j'ai co-encadré avec Isabel Demongodin et Rabah Ammour et qui a soutenu en 2021.

2.3 Sécurité dans les systèmes cyber-physiques

Mes recherches sur les systèmes cyber-physiques ont commencé récemment dans le cadre des deux projets (CPSecurity I et II) INS2I portés par Rabah Ammour. Nous considérons qu'un système cyber-physique est un système complexe dans lequel des éléments informatiques collaborent pour le contrôle et la commande d'entités physiques (Figure 2.4).

Les problèmes de sécurité se posent lorsqu'un attaquant a accès au réseau de communication entre la partie physique (infrastructure physique) et la partie cyber (contrôleur). L'attaquant peut dans ce cas modifier les commandes d'entrée fournies par le contrôleur mais aussi modifier les sorties générées par les actionneurs de la partie physique. Nous appelons une attaque *furtive* lorsque l'attaquant arrive à modifier les entrées et les sorties sans qu'il soit détecté par le contrôleur.

L'objectif de ces projets a été de proposer un formalisme SED qui puisse modéliser plusieurs aspects de ce type de système cyber-physique pour pouvoir réaliser une analyse et détecter des problèmes de sécurité. Nous avons proposé une nouvelle extension pour le formalisme des Réseaux de Petri, appelée Réseaux de Petri synchronisés avec sortie (*Output Synchronized Petri Nets - OutSynPNs*), qui prend en compte des entrées et des sorties du système physique.

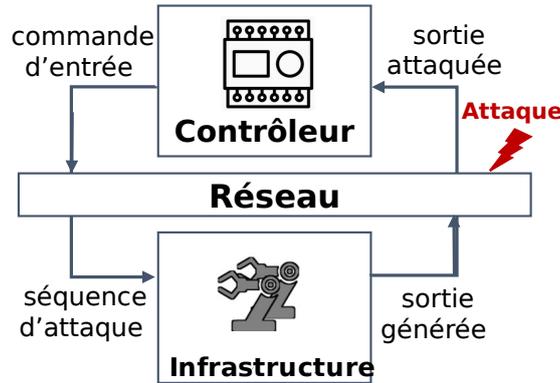


FIGURE 2.4 – Schéma d'attaque à un système cyber-physique

Cette nouvelle extension des Réseaux de Petri a été initialement définie dans [9] comme suit.

Définition 2.3. (*Réseau de Petri synchronisé avec sortie [9]*)

Un Réseau de Petri synchronisé avec sortie (*OutSynPN*) est une structure $N_{os} = \langle N, E, f, \Sigma, \Gamma, Q, g \rangle$. Un *OutSynPN* marqué est $\langle N_{os}, M_0 \rangle$ tels que :

- $N = \langle P, T, Pre, Post \rangle$ est un réseau de Petri, où P est un ensemble de m places, T est un ensemble de n transitions, $Pre : P \times T \rightarrow \mathbb{N}$ et $Post : P \times T \rightarrow \mathbb{N}$ sont les matrices d'incidences *pre-* et *post-* qui définissent les poids des arcs des places aux transitions et vice versa. $C = Post - Pre$ est la matrice d'incidence.
- E est un alphabet des événements d'entrée externe.
- $f : T \rightarrow E_\lambda = E \cup \{\lambda\}$ est une fonction qui associe à chaque transition t un événement d'entrée externe $f(t) \in E$ ou λ , où λ représente l'événement "se produit toujours".
- Un marquage est un vecteur $M : P \rightarrow \mathbb{N}^m$ qui attribut à chaque place un entier non-négatif. $M(p)$ représente le marquage de la place p .
- $\Sigma \subseteq \{\uparrow M(p_i), \downarrow M(p_i) \mid p_i \in P\}$ est un ensemble non-vide de conditions liées aux changements de marquage des places, où \downarrow et \uparrow représente, respectivement, une réduction et une augmentation du marquage d'une place.
- $\Gamma \subseteq \{M(p_i) \sim h, \mid p_i \in P, h \in \mathbb{N}, \sim \in \{=, \neq, \geq, \leq, >, <\}\}$ est un ensemble des conditions liées aux valeurs des marquages des places.
- Q est un alphabet des événements de sortie.
- $g : Q \rightarrow \{0, 1\}$ est une fonction de sortie tel que $\forall q_i \in Q, g(q_i) = \Upsilon(B_\Gamma(q_i)) \wedge \Theta(B_\Sigma(q_i))$ où $B_\Gamma(q_i) \in \mathcal{B}_{|\Gamma|}$, $B_\Sigma(q_i) \in \mathcal{B}_{|\Sigma|}$ et :
 - $\Upsilon : \mathcal{B}_{|\Gamma|} \rightarrow \{0, 1\}$ est une fonction booléenne décrivant les conditions sur les valeurs de marquage des places pour générer la sortie q_i et $\Upsilon(\cdot) = 1$ lorsqu'aucune condition sur les valeurs de marquage n'est impliquée pour la sortie q_i ;

- $\Theta : \mathcal{B}_{|\Sigma|} \rightarrow \{0, 1\}$ est une fonction booléenne décrivant les conditions des événements de changement de marquage pour générer la sortie q_i et $\Theta(\cdot) = 0$ lorsqu'aucun événement de changement de marquage n'est impliqué pour la sortie q_i .
- M_0 est le marquage initial. ■

Nous sommes partis du formalisme de Réseaux de Petri Synchronisés précédemment défini dans [34, 78, 87] et nous avons ajouté des fonctions de sortie lors des changements des marquages.

Les Réseaux de Petri Synchronisés ont la particularité d'associer le même événement d'entrée à plusieurs transitions. La dynamique associée à ce formalisme établit que, lors de la détection de l'événement d'entrée, toutes les transitions actives seront franchies de façon synchronisée.

Exemple 2.3. La figure 2.5 présente un OutSynPN avec $P = \{p_1, p_2, p_3, p_4, p_5\}$, $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, $E = \{a, b\}$. La fonction f associée à chaque transition est donnée par $f(t_1) = f(t_5) = a$, $f(t_2) = b$ et $f(t_3) = f(t_4) = f(t_6) = \lambda$. La fonction de sortie associée à l'alphabet de sortie $Q = \{A, B, C\}$ est $g(A) = (\downarrow M(p_1))$, $g(B) = ((\uparrow M(p_4)) \wedge (M(p_3) = 1))$ et $g(C) = (\downarrow M(p_5))$.

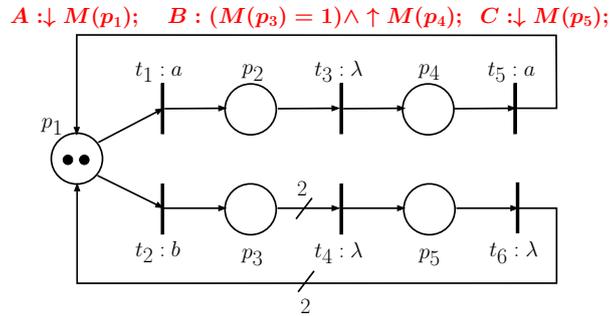


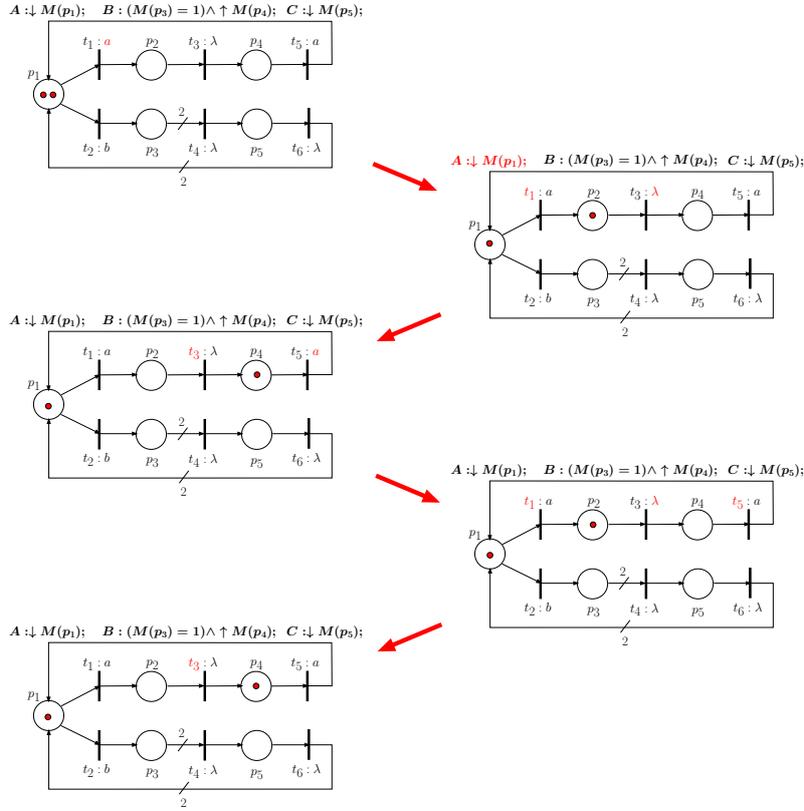
FIGURE 2.5 – Exemple d'un OutSynPN

Évolution du marquage de ce réseau, à partir du marquage initial $M_0 = (2\ 0\ 0\ 0\ 0)^T$ lequel peut être exprimé par $M_0 = 2p_1$ et pour une séquence d'entrée $i = aa$, est la suivante :

$$2p_1 \xrightarrow{a|t_1} p_1p_2 \xrightarrow{\lambda|t_3} p_1p_4 \xrightarrow{a|[t_1, t_5]} p_1p_2 \xrightarrow{\lambda|t_3} p_1p_4 \text{ et génère la séquence de sortie } o = A.$$

La figure 2.6 montre l'évolution du marquage suite à l'envoi des événements d'entrée. ■

Pour représenter cette dynamique et pouvoir l'exploiter, le graphe des marquages atteignables est représenté par une classe particulière d'automates finis appelée Automates Finis Labellisés avec Entrée (*Labelled Finite Automaton with Input - LFAI*).

FIGURE 2.6 – Évolution du marquage suite à la séquence d'événements aa **Définition 2.4.** (Automates finis labellisés avec entrée [9])

Un automate fini labellisé avec entrée (LFAI) est un 6-tuple $G = (X, E_\lambda, \delta, x_0, Q, Obs)$, où :

- X est un ensemble fini d'états,
- E est un ensemble fini de symboles (i.e., événements d'entrée) et $E_\lambda = E \cup \{\lambda\}$,
- $\delta : X \times E_\lambda \rightarrow X$ est une fonction de transition (possiblement partiellement définie),
- $x_0 \in X$ est un état initial,
- Q est un ensemble fini de label (i.e., événements de sortie et $Q_\varepsilon = Q \cup \{\varepsilon\}$,
- $Obs : X \times E_\lambda \rightarrow 2^Q \cup \{\varepsilon\}$ est une fonction d'étiquetage. ■

La dynamique complète ainsi que les règles d'obtention de cet automate sont décrites en détail dans [9].

Exemple 2.4. Le LFAI obtenu à partir du OutSynPN de la figure 2.5 est représenté dans la figure 2.7. Les états pointillés dans la figure représentent les états instables, c'est-à-dire, des états où les systèmes peuvent passer à l'état suivant sans intervention externe (événement d'entrée).

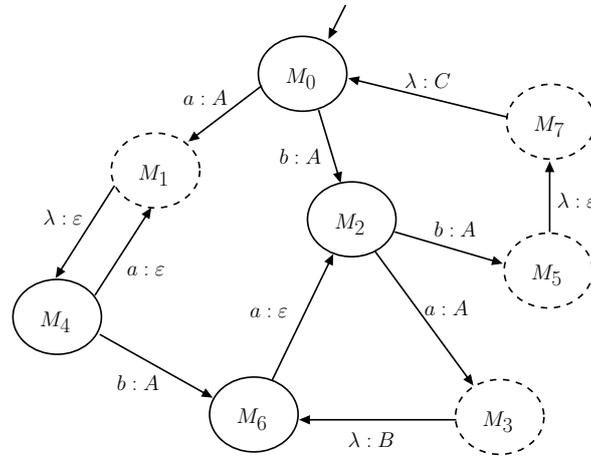


FIGURE 2.7 – LFAI de l'OutSynPN de la figure 2.5

■

Dans l'objectif de détecter des attaques du type furtif, où l'attaquant essaie de ne pas être détecté, plusieurs travaux ont suivi.

Dans [9], nous avons proposé des observateurs basés l'analyse des symboles d'entrée et de sortie. Ces observateurs sont utilisés dans la suite pour estimer l'état courant du système uniquement sur la base des informations des séquences d'entrée et de sortie.

Dans [11, 12], nous introduisons la notion de coût pour détecter des attaques furtives. Nous associons un coût différent à chaque action menée par l'attaquant. Ces actions peuvent être l'ajout, la modification ou la suppression d'une entrée ou d'une sortie du modèle. Chaque action de l'attaquant sur les entrées ou sur les sorties engendre un coût total qui est la somme de toutes les actions entreprises par l'attaquant. Du point de vue de l'attaquant, nous cherchons la séquence d'attaque (modification des entrées et des sorties) la moins coûteuse pour amener le système d'un état x à un état y quelconque. Cet état y étant souvent un état considéré dangereux, par exemple, un état de blocage.

Dans [10], nous faisons une composition des LFAIs, du point de vue de l'attaquant et du contrôleur, pour permettre d'avoir une vision globale des actions possibles mélangeant les actions de contrôle et d'attaque. Des observateurs sont construits par la suite pour fournir une estimation fine de l'état du système en fonctionnement normal ou sous attaque.

Dans [60], nous nous plaçons toujours du point de vue de l'attaquant. Dans la supposition que l'état actuel du système est inconnu de l'attaquant, nous cherchons une séquence de synchronisation, qui est un ensemble d'événements de contrôle d'entrée, qui peut être utilisée pour conduire le système à un état connu. Nous proposons une méthode pour calculer une séquence de synchronisation d'attaque qui mène le système dans un état de blocage en choisissant la meilleure séquence.

Dans la continuité de ce sujet, le projet ANR-PRC MENACES (*Timed Event-Based Methods for Networked Control Systems Security*), porté par Isabel Demongodin, a été démarré en Février 2023 et vise à explorer la notion de temps dans ces modèles.

Mon objectif dans ce chapitre était de présenter une vision globale de mes travaux sans trop détailler les aspects formels ainsi que les méthodologies utilisées. Cela couvre plus de 11 ans de recherches dans le domaine des formalismes déterministes, plus particulièrement, différentes classes de Réseaux de Petri. Cela a engendré plusieurs collaborations ainsi que des participations à différents projets.

Dans le chapitre suivant, je détaille mes travaux sur le formalisme de Réseaux de Petri Lots Triangulaires où l'introduction d'une nouvelle relation flux/densité nous a permis de modéliser des phénomènes de congestion/décongestion qui n'étaient pas encore pris en compte dans le formalisme.

Chapitre 3

Réseaux de Petri Lots Triangulaires

Dans ce chapitre, je présente en détail les travaux développés sur le formalisme de Réseaux de Petri Lots Triangulaires (RdPLots Triangulaires). Les RdPLots Triangulaires sont une extension des RdPLots Généralisés avec lots contrôlables déjà existants [36], où nous avons l'objectif de modéliser une relation flux/densité inspirée des comportements du trafic routier. Le flux dans ce type de relation augmente lorsque la densité augmente jusqu'à atteindre une *densité critique*. À partir de cette densité critique, une augmentation de la densité fait diminuer le flux. Le flux maximal est donc obtenu lorsque la densité du lot arrive à la densité critique de la place.

Ces travaux ont débuté dans le cadre de la thèse de Radhia Gaddouri (2012 – 2015). Ce chapitre a été donc adapté à partir des travaux de sa thèse. Afin de ne pas surcharger ce manuscrit, je ne présenterai pas les preuves des propositions présentées tout au long de ce chapitre. Le lecteur intéressé peut les trouver dans [48, 50, 51, 53].

Dans la section 3.1, j'introduis la définition des RdPLots Triangulaires. La section 3.2 étend les RdPLots Triangulaires afin d'intégrer des événements de contrôle. Dans la section 3.3, je présente la dynamique associée au formalisme.

Dans la section suivante nous avons proposé une extension du formalisme des Réseaux de Petri Lots (RdPLots) où cette relation flux/densité caractérisée par le diagramme fondamental triangulaire est intégrée dans les places lots.

3.1 Définition des Réseaux de Petri Lots Triangulaires

Les Réseaux de Petri Lots Triangulaires (RdPLots Triangulaires) implémentent une nouvelle relation flux-densité dans les places lots. Cette nouvelle relation est clairement inspirée du diagramme fondamental triangulaire du trafic routier mais peut représenter tout type de système où une augmentation de la densité n'induit pas forcément une augmentation du flux, par exemple, les modèles de foules de piétons.

Notre principale contribution par rapport au formalisme RdPLots défini par Demongodin [35, 36] se situe principalement au niveau de la dynamique interne des places lots où maintenant trois régimes d'évolution sont modélisés : *libre*, *congestion* et *décongestion*.

De façon à simplifier la lecture, je présente dans cette section, la définition des places lots triangulaires et j'adapte les concepts liés à la dynamique du réseau des travaux originaux de Demongodin [35, 36] afin de proposer au lecteur une vision globale du formalisme.

3.1.1 Place lot triangulaire

Pour pouvoir intégrer la nouvelle relation flux-densité dans les places lots, nous avons ajouté un paramètre supplémentaire à la fonction caractéristique d'une place lot. Ce paramètre, le flux maximal de la place, permet de calculer d'autres paramètres importants à cette relation flux/densité, tels que la densité critique et la vitesse de propagation de congestion.

La nouvelle fonction caractéristique γ de la *place lot triangulaire* est définie comme suit :

Définition 3.1. (*Fonction caractéristique d'une place lot triangulaire*)

La fonction caractéristique $\gamma : P^T \rightarrow \mathbb{R}_{>0}^4$ associe à chaque place lot triangulaire $p_i \in P^T$ le quadruplet $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ tel que :

- V_i : la vitesse maximale (unité de longueur / unité de temps);
- d_i^{max} : la densité maximale (entité / unité de longueur);
- S_i : la longueur (unité de longueur);
- Φ_i^{max} : le flux maximal (entité / unité de temps). ■

Dans une relation flux-densité, que nous appelons triangulaire, deux paramètres sont indispensables pour la définition de la relation. Nous associons à chaque place lot triangulaire une densité critique et une vitesse de propagation de congestion.

Définition 3.2. (*Densité critique pour une place lot triangulaire*)

À chaque place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$, une densité critique d_i^{cri} est associée et donnée par :

$$d_i^{cri} = \frac{\Phi_i^{max}}{V_i} \quad (3.1)$$

■

La densité critique nous indique le point de bascule où l'augmentation de la densité induit à un réduit du flux. Elle représente aussi le point où le flux est maximal.

Définition 3.3. (*Vitesse de propagation de congestion pour une place lot triangulaire*)

À chaque place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$, une vitesse de propagation de congestion W_i est associée à p_i et définie par :

$$W_i = \frac{\Phi_i^{max} \cdot V_i}{d_i^{max} \cdot V_i - \Phi_i^{max}} \quad (3.2)$$

■

La vitesse de propagation de congestion nous permet de savoir, dans le cas d'un phénomène de congestion ou décongestion, à quelle vitesse la file d'attente des entités composant la congestion va augmenter ou se dissiper.

À partir de ces informations, nous pouvons définir la relation flux-densité comme suit :

Définition 3.4. (*Relation flux-densité associée à une place lot triangulaire*)

La relation flux-densité associée à une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ est définie par :

$$\phi = \begin{cases} d \cdot V_i & \text{si } 0 \leq d < d_i^{cri} \\ W_i \cdot (d_i^{max} - d) & \text{si } d_i^{cri} \leq d \leq d_i^{max} \end{cases} \quad (3.3)$$

■

Cette relation est représentée dans la figure 3.1 ci-après.

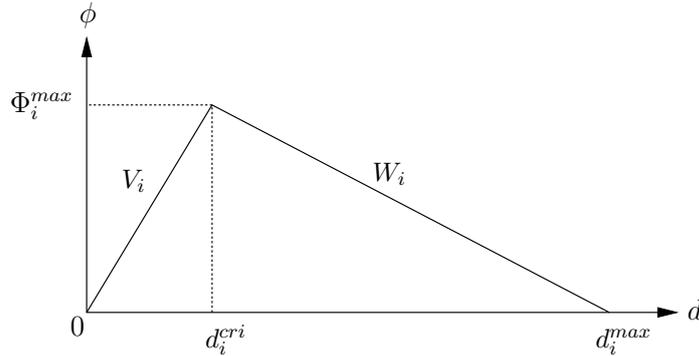


FIGURE 3.1 – Relation flux-densité dans une place lot triangulaire

Cette nouvelle relation régie donc le comportement des lots contrôlables qui composent le marquage d'une place. Le marquage d'une place lot triangulaire est un ensemble de lots contrôlables ordonnés par leurs positions de tête : si $p_i \in P^T$ alors $m_i = \{C\beta_i^r, \dots, C\beta_i^h\}$.

Les *lots contrôlables* [36] sont un groupe d'entité qui se déplace à sa propre vitesse à l'intérieur d'une place lot. La définition d'un lot contrôlable est donnée par la suite :

Définition 3.5. (*Lot contrôlable*)

À l'instant τ , un lot contrôlable $C\beta_i^r(\tau)$ associé à une place lot triangulaire $p_i \in P^T$, avec $p_i = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ est défini par un quadruplet, $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ avec :

- $l_i^r(\tau) \in \mathbb{R}^+$ sa longueur,
- $d_i^r(\tau) \in \mathbb{R}^+$ sa densité,
- $x_i^r(\tau) \in \mathbb{R}^+$ sa position de tête,
- $v_i^r(\tau) \in \mathbb{R}^+$ sa vitesse.

■

Par rapport à la densité critique de la place, nous pouvons classer chaque lot en deux états : libre ou congestionné. Ce classement est représenté dans la figure 3.2.

Définition 3.6. (*États des lots contrôlables*)

Soit, à l'instant τ , un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ d'une place lot triangulaire $p_i \in P^T$, avec $p_i = (V_i, d_i^{max}, S_i, \Phi_i^{max})$.

- $C\beta_i^r$ est dit lot contrôlable libre ou un lot contrôlable dans un état libre si sa densité est inférieure ou égale à la densité critique de la place p_i : $d_i^r(\tau) \leq d_i^{cri}$ vérifiant $\phi_i^r(\tau) = d_i^r(\tau) \cdot V_i$;
- $C\beta_i^r$ est dit lot contrôlable congestionné ou un lot contrôlable dans un état congestionné si sa densité est strictement supérieure à la densité critique de la place p_i : $d_i^r(\tau) > d_i^{cri}$ vérifiant $\phi_i^r(\tau) = W_i \cdot (d_i^{max} - d_i^r(\tau))$.

■

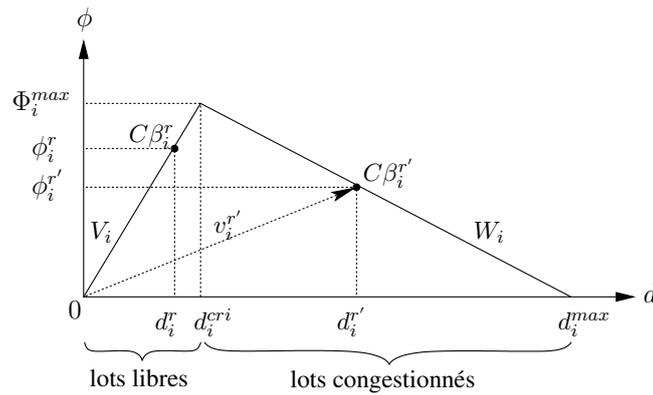


FIGURE 3.2 – États des lots contrôlables par rapport à la densité critique.

D'après la relation triangulaire flux-densité (Définition 3.4) et selon la valeur du flux de sortie $\phi_i^{out}(\tau)$ (voir définition 3.16) de la place lot triangulaire p_i , nous pouvons déduire deux caractéristiques de congestion (densité et vitesse de congestion) pour tout lot contrôlable congestionné dans cette place p_i .

Définition 3.7. (*Densité et vitesse de congestion*)

Soit à l'instant τ une place lot triangulaire $p_i \in P^T$, avec $p_i = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ et $\phi_i^{out}(\tau)$ son flux de sortie.

— la densité de congestion $d_i^{cong}(\tau)$ est donnée par :

$$d_i^{cong}(\tau) = d_i^{max} - \frac{\phi_i^{out}(\tau)}{W_i} \quad (3.4)$$

— la vitesse de congestion $v_i^{cong}(\tau)$ est donnée par :

$$v_i^{cong}(\tau) = \frac{W_i \cdot (d_i^{max} - d_i^{cong}(\tau))}{d_i^{cong}(\tau)} \quad (3.5)$$

■

La densité ainsi que la vitesse de congestion seront utilisées dans la création des nouveaux lots au moment de définir l'évolution des lots en congestion ou décongestion (voir Définition 3.24, page 35).

3.1.2 Définition d'un réseau de Petri lots triangulaire

Dans l'objectif d'intégrer la nouvelle place lot triangulaire au RdPLots, nous étendons les RdPLots Généralisé [35] et les RdPLots avec lots contrôlables [36] en remplaçant les places lots "classiques" par des places lots triangulaires. Les nouveaux noeuds d'un RdPLots Triangulaire sont représentés dans la figure 3.3.

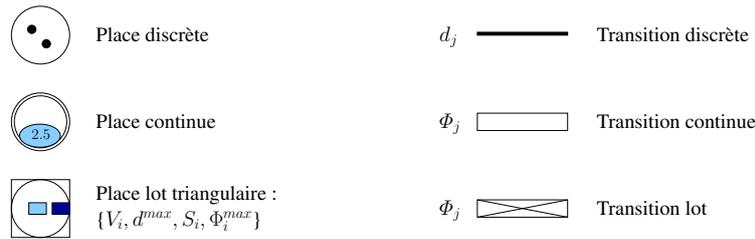


FIGURE 3.3 – Noeuds des Réseau de Petri Lots Triangulaires

Nous définissons un RdPLots Triangulaire comme suit :

Définition 3.8. (*Réseau de Petri Lots Triangulaire*)

Un réseau de Petri Lots Triangulaire (RdPLots Triangulaire) est un 6-uplet $N = (P, T, Pre, Post, \gamma, Time)$ tel que :

— $P = P^D \cup P^C \cup P^T$ est un ensemble fini de places partitionné en trois types de classes : discrètes P^D , continues P^C et lots triangulaire P^T .

- $T = T^D \cup T^C \cup T^B$ est un ensemble fini de transitions partitionné en trois types de classes : discrètes T^D , continues T^C et lots T^B .
- $Pre, Post : (P^D \times T \rightarrow \mathbb{N}) \cup ((P^C \cup P^T) \times T \rightarrow \mathbb{R}_+)$ sont respectivement, les matrices de pre-incidence et post-incidence, désignant le poids des arcs des places vers les transitions et de transitions vers les places.
- $\gamma : P^T \rightarrow \mathbb{R}_+^4$ est la fonction caractéristique des places lot triangulaire. Elle associe à chaque place lot $p_i \in P^T$ le quadruplet $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ qui représente respectivement, la vitesse maximale, la densité maximale, la longueur et le flux maximal de cette place lot triangulaire.
- $Time : T \rightarrow \mathbb{R}_+$ associe un nombre non négatif à chaque transition :
 - si $t_j \in T^D$, alors $Time(t_j) = d_j$ désigne le délai de franchissement associé à la transition discrète ;
 - si $t_j \in T^C \cup T^B$, alors $Time(t_j) = \Phi_j$ désigne le flux maximal de franchissement associé aux transitions continues et lot.

■

Exemple 3.1. La figure 3.4 présente un RdPLots Triangulaire avec une place p_1 et une transition discrète t_1 , deux places continues p_2 et p_3 , une place lot triangulaire p_4 ainsi que deux transitions lots t_2 et t_3 .

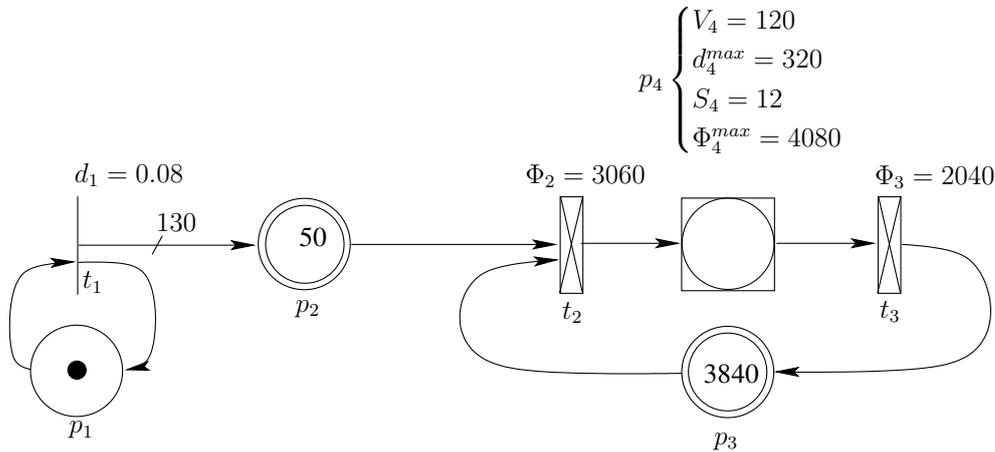


FIGURE 3.4 – Exemple d'un RdPLots Triangulaire

■

Nous considérons les notations suivantes pour la suite de ce chapitre.

✎ **Notation**

$n = P $	le nombre de places du modèle ;
$\bullet t_j$	l'ensemble des places d'entrée de la transition t_j tel que $\{p_i \in P \mid Pre(p_i, t_j) > 0\}$;
t_j^\bullet	l'ensemble des places de sortie de la transition t_j tel que $\{p_i \in P \mid Post(p_i, t_j) > 0\}$;
${}^{(d)}t_i$	l'ensemble des places discrètes d'entrée de la transition t_i tel que ${}^{(d)}t_i = \bullet t_i \cap P^D$;
${}^{(c)}t_i$	l'ensemble des places continues d'entrée de la transition t_i tel que ${}^{(c)}t_i = \bullet t_i \cap P^C$;
${}^{(t)}t_i$	l'ensemble des places lots d'entrée de la transition t_i tel que ${}^{(t)}t_i = \bullet t_i \cap P^T$;
$\bullet p_i$	l'ensemble des transitions d'entrée de la place p_i ;
p_i^\bullet	l'ensemble des transitions de sortie de la place p_i ;
${}^{(d)}p_i$	l'ensemble des transitions discrètes d'entrée de la place p_i tel que ${}^{(d)}p_i = \bullet p_i \cap T^D$;
${}^{(c)}p_i$	l'ensemble des transitions continues d'entrée de la place p_i tel que ${}^{(c)}p_i = \bullet p_i \cap T^C$;
${}^{(b)}p_i$	l'ensemble des transitions lots d'entrée de la place p_i tel que ${}^{(b)}p_i = \bullet p_i \cap T^B$;

La matrice d'incidence regroupe nos différents types de noeuds selon la définition 3.9 ci-après.

Définition 3.9. (*Matrice d'incidence d'un RdPLots Triangulaire*)

Si nous numérotons d'abord les noeuds discrets, puis les noeuds continus et enfin les noeuds lots, la matrice d'incidence d'un réseau de Petri Lots $C = \mathbf{Post} - \mathbf{Pre}$, a la forme suivante :

$$C = \begin{array}{ccc} & T^D & T^C & T^B \\ \begin{bmatrix} C^{DD} & \mathbf{0} & \mathbf{0} \\ C^{CD} & C^{CC} & C^{CB} \\ \mathbf{0} & \mathbf{0} & C^{TB} \end{bmatrix} & P^D & P^C & P^T \end{array}$$

où C^{DD} est une matrice de nombres entiers, et C^{CD} , C^{CC} , C^{CB} , C^{TB} sont des matrices de nombres réels. ■

Définition 3.10. (*RdPLots Triangulaire bien formé*)

Un réseau de Petri Lots Triangulaire est dit bien formé si les conditions suivantes sont vérifiées :

- une place discrète peut être connectée uniquement à une transition continue ou lot par une boucle, c'est-à-dire, pour toute place $p_i \in P^D$ et pour toute transition $t_j \in T^C \cup T^B$, il convient que $Pre(p_i, t_j) = Post(p_i, t_j)$;

- l'ensemble des *Pre* et *Post* des places lots contient seulement des transitions lots, c'est-à-dire, pour tout $p_i \in P^T$, il convient que $\bullet p_i \cup p_i^\bullet \subseteq T^B$.

■

Définition 3.11. (Marquage d'un réseau de Petri Lots Triangulaire)

Le marquage d'un RdPLots Triangulaire à l'instant τ est défini par $\mathbf{m}(\tau) = [m_1(\tau), \dots, m_i(\tau), \dots, m_n(\tau)]$ tel que :

- si $p_i \in P^D$ alors $m_i \in \mathbb{N}$, c'est-à-dire, le marquage de la place discrète est un entier non négatif;
- si $p_i \in P^C$ alors $m_i \in \mathbb{R}_+$, c'est-à-dire, le marquage de la place continue est un réel non négatif;
- si $p_i \in P^T$ alors $m_i = \{C\beta_i^h, \dots, C\beta_i^r\}$, c'est-à-dire, le marquage de la place lot triangulaire est une série de lots contrôlables ordonnés par leur position de tête.

■

⌚ Soit

\mathbf{m}_0 le marquage initial d'un RdPLots Triangulaire.

Nous supposons que, dans les RdPLots Triangulaires, un lot contrôlable ne peut pas dépasser un autre, quel que soit sa densité ou sa vitesse.

Définition 3.12. (Vecteur de quantité de marque)

Le vecteur de la quantité de marque $\mathbf{q} = \mu(\mathbf{m}) \in \mathbb{R}_+$ associé au marquage \mathbf{m} est défini comme suit :

$$q_i = \begin{cases} m_i & \text{si } p_i \in P^D \cup P^C, \\ \sum_{C\beta_i^r \in m_i} l_i^r \cdot d_i^r & \text{si } p_i \in P^T \end{cases}$$

où

- $\mu(\mathbf{m})$ est une fonction injective qui associe un marquage \mathbf{m} à un seul vecteur de quantité de marque \mathbf{q} .

■

⌚ Soit

$Q_i = S_i \cdot d_i^{max}$ la capacité maximale d'une place lot triangulaire $p_i \in P^T$, avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$.

La capacité maximale d'une place lot triangulaire nous permet de définir le concept d'une place lot triangulaire pleine. Cette définition sera utile dans le calcul des flux instantanés de franchissement introduit dans la section 3.1.3.

Définition 3.13. (*Place lot triangulaire pleine*)

Une place lot triangulaire $p_i \in P^T$, avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ est dite place lot triangulaire pleine si la quantité de marque q_i est égale à sa capacité maximale de Q_i . ■

Le concept de lot contrôlable de sortie est lié à la position du lot dans une place lot triangulaire. Lorsqu'un lot en déplacement à l'intérieur d'une place lot atteint la fin de la place, il est considéré comme un lot contrôlable de sortie.

Définition 3.14. (*Lot contrôlable de sortie*)

Soit $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau)) \in m_i(\tau)$ un lot contrôlable d'une place lot $p_i \in P^T$, avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$. $C\beta_i^r$ est dit lot contrôlable de sortie nommé $OC\beta_i^r$ si sa position de tête est égale à la longueur de la place lot triangulaire p_i , c'est-à-dire, $x_i^r(\tau) = S_i$. ■

Chaque place ne peut avoir qu'un seul lot contrôlable de sortie. Nous pouvons donc définir la densité de sortie de la place comme la densité du lot contrôlable de sortie, s'il y en a. Sinon elle sera égale à zéro.

Définition 3.15. (*Densité de sortie d'une place lot triangulaire*)

Soit une place lot triangulaire $p_i \in P^T$, avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ et un lot contrôlable de sortie $OC\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), S_i, v_i^r(\tau))$. La densité de sortie $d_i^{out}(\tau)$ associée à p_i est définie à un instant τ ainsi, si la place lot triangulaire p_i a un lot contrôlable de sortie $OC\beta_i^r(\tau)$, alors $d_i^{out}(\tau) = d_i^r(\tau)$, sinon $d_i^{out}(\tau) = 0$. ■

Cette notion de lot contrôlable de sortie est aussi importante dans le calcul des flux instantanés de franchissement car même si une place lot triangulaire a des lots dans son marquage, cela n'implique pas forcément avoir un flux de sortie si aucun des lots n'est pas un lot de sortie. Nous pouvons constater cela dans les conditions de franchissement présentées dans la section ci-après.

3.1.3 Dynamique du réseau

La dynamique d'un RdPLots Triangulaire est gouvernée par le franchissement des transitions validées et par une dynamique hybride à l'intérieur des places lots triangulaires. Cette dynamique interne engendre un changement de marquage.

Nous introduisons la notion de flux instantané de franchissement (*IFF - Instantaneous Firing Flow*).

Le flux instantané de franchissement d'une transition lot ou continue représente la quantité de marques par unité de temps qui franchissent cette transition.

Définition 3.16. (*Flux instantané de franchissement*)

Le flux instantané de franchissement $\varphi_j(\tau) \leq \Phi_j$, associé aux transitions continues et lots $t_j \in T^C \cup T^B$ représente la quantité de franchissement de la transition t_j par unité de temps. Le vecteur des flux instantanés de franchissement à un instant τ est noté par $\varphi \in \mathbb{R}^{|T^C|+|T^B|}$.

Le flux d'entrée (resp., de sortie) de la place continue ou lot p_i à un instant τ est la somme de tous les flux entrants (resp. sortants) de la place et qui peut être écrit, respectivement, comme suit :

$$\begin{aligned} - \phi_i^{in}(\tau) &= \sum_{t_j \in \bullet p_i} Post(p_i, t_j) \cdot \varphi_j(\tau) = Post(p_i, \cdot) \cdot \varphi(\tau). \\ - \phi_i^{out}(\tau) &= \sum_{t_j \in p_i \bullet} Pre(p_i, t_j) \cdot \varphi_j(\tau) = Pre(p_i, \cdot) \cdot \varphi(\tau). \end{aligned} \quad \blacksquare$$

Conditions de validation

Les conditions et règles de validation et franchissement des transitions discrètes sont celles du réseau de Petri discret T-temporisé. Pour la validation et franchissement des transitions continues, elles sont les mêmes conditions que les réseaux de Petri hybrides [4, 34, 71] c'est-à-dire, nous distinguons les transitions faiblement et fortement validées.

Pour les transitions discrètes, le principe de fonctionnement à vitesse maximale et mono serveur est utilisé. Nous utilisons la notion de marques réservées pour les places discrètes. Dès qu'une transition discrète temporisée est validée, des jetons ou des marques nécessaires à son franchissement sont réservés et ne sont plus disponibles pour un autre franchissement pendant tout le délai associé à la transition. Lorsque la durée est écoulée, la transition est effectivement franchie et les jetons réservés à son franchissement sont ensuite retirés de ses places d'entrée et des jetons non-réservés sont ajoutés dans ses places de sortie. Enfin, une transition discrète ne peut être validée un autre moment durant son franchissement.

Ainsi, dans un RdPLots Triangulaire, la notion de retard déterministe associé à la transition discrète temporisée, introduit une décomposition du marquage global. À tout moment, le marquage \mathbf{m} est la somme des deux marquages \mathbf{m}^r et \mathbf{m}^n , où \mathbf{m}^r est le marquage réservé et \mathbf{m}^n est le marquage non réservé : si $p_i \in P^D$ ou P^C alors $m_i(\tau) = m_i^r(\tau) + m_i^n(\tau)$. Du fait que les conditions structurelles ne permettent pas à une place lot triangulaire d'être une place d'entrée pour une transition discrète, la notion de marquage réservé est inadéquate pour le marquage lot.

Les conditions de validation permettent de prendre en compte l'état de marquage des places discrètes et continues (marques réservés et non réservés). Pour les transitions lots, ces conditions dépendent aussi de l'existence d'un lot de sortie à l'intérieur du marquage des places lots.

Condition 3.1. (*Validation d'une transition discrète*)

Une transition discrète, $t_j \in T^D$, est validée à \mathbf{m} si pour toutes places $p_i \in \bullet t_j$, $m_i \geq \text{Pre}(p_i, t_j)$.

■

Condition 3.2. (Validation d'une transition continue)

Une transition continue $t_j \in T^C$ est valide à \mathbf{m} si pour toute place $p_i \in {}^{(d)}t_j$, $m_i \geq \text{Pre}(p_i, t_j)$. Nous disons qu'une transition continue est :

- fortement validée si $\forall p_i \in {}^{(c)}t_j$, $m_i > 0$;
- faiblement validée si $\exists p_i \in {}^{(c)}t_j$, $m_i = 0$.

■

Condition 3.3. (Validation d'une transition lot)

Une transition lot $t_j \in T^B$ est validée à \mathbf{m} si :

- $\forall p_i \in {}^{(c)}t_j$, $m_i \geq \text{Pre}(p_i, t_j)$.
- $\forall p_i \in {}^{(t)}t_j$, $d_i^{\text{out}} > 0$.

Nous disons qu'une transition lot est :

- fortement validée si $\forall p_i \in {}^{(c)}t_j$, $m_i > 0$.
- faiblement validée si $\exists p_i \in {}^{(c)}t_j$, $m_i = 0$.

■

Conditions de franchissement

Pour le franchissement des transitions discrètes et continues, les marques sont retirées et ajoutées (entier ou réel non négatif), respectivement de leurs places d'entrée et dans les places de sortie discrètes et continues. Nous rappelons que pour un réseau de Petri Lots Triangulaire, pour le franchissement d'une transition discrète, il faut respecter l'hypothèse de mono serveur et le fonctionnement à vitesse maximale.

Condition 3.4. (Franchissement d'une transition discrète)

Soit une transition $t_j \in T^D$ et $\text{Time}(t_j) = d_j$. La transition discrète t_j est franchie à $\tau + d_j$, c'est-à-dire :

- $\forall p_i \in \bullet t_j$, $m_i(\tau + d_j) = m_i(\tau) - \text{Pre}(p_i, t_j)$;
- $\forall p_i \in t_j^\bullet$, $m_i(\tau + d_j) = m_i(\tau) + \text{Post}(p_i, t_j)$.

■

Condition 3.5. (Franchissement d'une transition continue)

Soit une transition $t_j \in T^C$ et $\varphi_j(\tau)$ le flux instantané de franchissement de la transition t_j à la date τ . La transition continue t_j est franchie à τ pendant un délai Δ , c'est-à-dire :

- $\forall p_i \in \bullet t_j, m_i(\tau + \Delta) = m_i(\tau) - \varphi_j(\tau) \cdot Pre(p_i, t_j) \cdot \Delta;$
- $\forall p_i \in t_j^\bullet, m_i(\tau + \Delta) = m_i(\tau) + \varphi_j(\tau) \cdot Post(p_i, t_j) \cdot \Delta.$ ■

Lors du franchissement d'une transition lot, les lots sont détruits à l'intérieur des places lots triangulaire d'entrée et créés dans les places lots triangulaire de sortie.

Condition 3.6. (*Franchissement d'une transition lot*)

Soit une transition $t_j \in T^B$ et $\varphi_j(\tau)$ son flux instantané de franchissement à la date τ . La transition lot t_j est franchie à τ pendant un délai Δ , c'est-à-dire :

- $\forall p_i \in {}^{(c)}t_j, m_i(\tau + \Delta) = m_i(\tau) - \varphi_j(\tau) \cdot Pre(p_i, t_j) \cdot \Delta;$
- $\forall p_i \in {}^{(t)}t_j, m_i(\tau + \Delta) = m_i(\tau) \setminus \{OC\beta_i^r(\tau)\},$ avec $OC\beta_i^r(\tau)$ est le lot contrôlable de sortie de la place lot triangulaire p_i ;
- $\forall p_i \in t_j^{(c)}, m_i(\tau + \Delta) = m_i(\tau) + \varphi_j(\tau) \cdot Post(p_i, t_j) \cdot \Delta;$
- $\forall p_i \in t_j^{(t)}, m_i(\tau + \Delta) = m_i(\tau) \cup \{C\beta_i^r(\tau)\},$ avec $C\beta_i^r(\tau + \Delta)$ est le lot contrôlable créé dans la place lot triangulaire p_i . ■

Les équations qui gouvernent la destruction et la création des lots sont présentées dans la section 3.1.4.

Calcul des flux instantanés de franchissement La méthode utilisée pour le calcul des flux instantanés de franchissement dans le RdPLots Triangulaire est celle proposé par Demongodin et Giua dans [38]. Elle est basée sur la technique de programmation linéaire qui prend en compte la structure du réseau et son état actuel. Dans [51], nous avons adopté cette méthode en ajoutant une contrainte dédiée aux lots en état congestionné.

Proposition 3.1. (*Calcul de flux instantané de franchissement des transitions continues et lots*) Soit, à l'instant τ , un RdPLots Triangulaire de matrice d'incidence \mathbf{C} et un marquage $\mathbf{m}(\tau)$:

- $T_N(\mathbf{m}) \subset T^C \cup T^B$ est l'ensemble des transitions continues et lots qui ne sont pas validées en \mathbf{m} ;
- $P_\emptyset(\mathbf{m}) = \{p_i \in P^C \mid m_i = 0\}$ est l'ensemble des places continues vides ;
- $P_F^T(\mathbf{m}) = \{p_i \in P^T \mid q_i = Q_i\}$ est l'ensemble des places lots triangulaires pleines.

À un instant τ , chaque vecteur admissible φ , pour un marquage \mathbf{m} , est une solution admissible du système linéaire suivant :

$$\left\{ \begin{array}{lll} (a) & 0 \leq \varphi_j(\tau) \leq \Phi_j & \forall t_j \in T^C \cup T^B \\ (b) & \varphi_j(\tau) = 0 & \forall t_j \in T_N(\mathbf{m}) \\ (c) & C(p_i, \cdot) \cdot \varphi_j(\tau) \geq 0 & \forall p_i \in P_\emptyset(\mathbf{m}) \\ (d) & C(p_i, \cdot) \cdot \varphi_j(\tau) \leq 0 & \forall p_i \in P_F^T(\mathbf{m}) \\ (e) & Post(p_i, \cdot) \cdot \varphi_j(\tau) \leq V_i \cdot d_i^{cri} & \forall p_i \in P^T \\ (f) & Pre(p_i, \cdot) \cdot \varphi_j(\tau) \leq V_i \cdot d_i^{out}(\tau) & \forall p_i \in P^T \\ (g) & Pre(p_i, \cdot) \cdot \varphi_j(\tau) \leq V_i \cdot d_i^{cri} & \forall p_i \in P^T \end{array} \right. \quad (3.6)$$

■

De façon à adapter le programme linéaire aux RdPLots Triangulaires, nous avons modifié la contrainte e de manière que le flux maximal ne soit plus calculé par rapport à la densité maximale mais par rapport à la densité critique. La contrainte g établit que flux de sortie d'une place lot triangulaire p_i ne dépasse pas le flux maximal Φ_i^{max} lorsque le lot de sortie est un lot congestionné.

3.1.4 Dynamique hybride des places lots triangulaires

La dynamique hybride d'une place lot triangulaire est définie par trois régimes : *libre*, *de congestion* et *de décongestion*. Pour comprendre l'évolution des lots contrôlables à l'intérieur d'une place lots triangulaire, je commence par introduire l'ensemble des fonctions statiques qui peuvent être appliquées aux lots à n'importe quel instant τ . Dans la suite, je présente une série de définitions et de propositions nécessaires à la construction des équations qui gouvernent l'évolution des lots selon chaque régime.

Pour rappel, je ne présente pas les preuves des propositions. Le lecteur intéressé peut les retrouver dans [48].

Définition 3.17. (Fonctions statiques pour un lot contrôlable)

À l'instant τ , plusieurs fonctions statiques peuvent être appliquées aux lots contrôlables composant le marquage d'une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$:

- *Création* : si le flux d'entrée d'une place lot triangulaire p_i est non nul, c'est-à-dire, $\phi_i^{in}(\tau) \neq 0$, un lot contrôlable $C\beta_i^r(\tau) = (0, d_i^r(\tau), 0, V_i)$ avec $d_i^r(\tau) = \phi_i^{in}(\tau)/V_i$, est créé et ajouté au marquage de p_i , c'est-à-dire, $m_i(\tau) = m_i(\tau) \cup \{C\beta_i^r(\tau)\}$.
- *Destruction* : si la longueur de lot est nulle ($l_i^r(\tau) = 0$) et il n'est pas un lot de création c'est-à-dire ($x_i^r(\tau) \neq 0$), ce lot peut être détruit. La fonction de destruction est notée par $C\beta_i^r(\tau) = \emptyset$, le lot est supprimé du marquage de la place p_i , c'est-à-dire, $m_i(\tau) = m_i(\tau) \setminus \{C\beta_i^r(\tau)\}$.
- *Fusion* : si deux lots contrôlables sont en contact avec la même densité et la même vitesse, ils peuvent être fusionnés. Considérons $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ et $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$ avec $x_i^h(\tau) = l_i^h(\tau) + x_i^r(\tau)$. Dans ce cas, le lot

en aval de $C\beta_i^r(\tau)$, $C\beta_i^h(\tau)$, devient $C\beta_i^h(\tau) = (l_i^r(\tau) + l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$ et le lot $C\beta_i^r(\tau)$ est détruit : $C\beta_i^r(\tau) = \emptyset$.

- *Éclatement* : il est toujours possible d'éclater un lot en deux lots qui sont en contact et qui ont la même densité et la même vitesse. Considérons à un instant τ un lot $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ éclaté en deux lots : $C\beta_i^r(\tau)$ et $C\beta_i^{r'}(\tau)$. Dans ce cas le lot $C\beta_i^{r'}(\tau)$ est ajouté au marquage de p_i , c'est-à-dire, $m_i(\tau) = m_i(\tau) \cup \{C\beta_i^{r'}(\tau)\}$. ■

Définition 3.18. (Densité d'entrée/vitesse d'entrée, densité de sortie/vitesse de sortie pour un lot contrôlable)

Soit un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ d'une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ ayant un flux de sortie $\phi_i^{out}(\tau)$. À un instant τ :

1. en entrée, une densité $d_{in}^r(\tau)$ et une vitesse $v_{in}^r(\tau)$ sont définies pour lots $C\beta_i^r(\tau)$ comme suit :

- si $C\beta_i^r(\tau) = (0, d_i^r(\tau), 0, v_i^r(\tau))$ est un lot créé :

$$\begin{cases} d_{in}^r(\tau) = d_i^r(\tau) \\ v_{in}^r(\tau) = v_i^r(\tau) \end{cases} \quad (3.7)$$

- si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est un lot en déplacement n'est pas en contact amont :

$$\begin{cases} d_{in}^r(\tau) = 0 \\ v_{in}^r(\tau) = 0 \end{cases} \quad (3.8)$$

- si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est un lot contrôlable qui est en contact amont avec un lot contrôlable $C\beta_i^q(\tau) = (l_i^q(\tau), d_i^q(\tau), x_i^q(\tau), v_i^q(\tau))$:

$$\begin{cases} d_{in}^r(\tau) = d_i^q(\tau) \\ v_{in}^r(\tau) = v_i^q(\tau) \end{cases} \quad (3.9)$$

2. en sortie, une densité $d_{out}^r(\tau)$ et une vitesse $v_{out}^r(\tau)$ lui sont associées comme suit :

- si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est un lot en déplacement :

$$\begin{cases} d_{out}^r(\tau) = 0 \\ v_{out}^r(\tau) = 0 \end{cases} \quad (3.10)$$

- si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est en contact aval avec un lot $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$:

$$\begin{cases} d_{out}^r(\tau) = d_i^h(\tau) \\ v_{out}^r(\tau) = v_i^h(\tau) \end{cases} \quad (3.11)$$

— si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), S_i, v_i^r(\tau))$ est un lot contrôlable de sortie :

$$\begin{cases} d_{out}^r(\tau) = d_i^r(\tau) \\ v_{out}^r(\tau) = \frac{\phi_i^{out}(\tau)}{d_i^r(\tau)} \end{cases} \quad (3.12)$$

■

Proposition 3.2. Flux instantanés d'entrée et de sortie d'un lot contrôlable

Soit un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ ayant une densité d'entrée $d_{in}^r(\tau)$, une vitesse d'entrée $v_{in}^r(\tau)$, une densité de sortie $d_{out}^r(\tau)$ et une vitesse de sortie $v_{out}^r(\tau)$. Le lot contrôlable $C\beta_i^r(\tau)$ appartient au marquage m_i de la place lot triangulaire $p_i \in P^T$ ayant un flux d'entrée ϕ_i^{in} et un flux de sortie ϕ_i^{out} . À un instant τ :

1. en entrée, un flux instantané $\phi_{in}^r(\tau)$ est défini pour le lot $C\beta_i^r(\tau)$ par :

— si $C\beta_i^r(\tau) = (0, d_i^r(\tau), 0, v_i^r(\tau))$ est un lot créé :

$$\phi_{in}^r(\tau) = \phi_i^{in}(\tau) \quad (3.13)$$

— si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est un lot en déplacement n'est pas en contact amont :

$$\phi_{in}^r(\tau) = 0 \quad (3.14)$$

— si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est un lot contrôlable qui est en contact amont avec un lot contrôlable $C\beta_i^q(\tau) = (l_i^q(\tau), d_i^q(\tau), x_i^q(\tau), v_i^q(\tau))$:

$$\phi_{in}^r(\tau) = \phi_i^q(\tau) \quad (3.15)$$

2. en sortie, un flux instantané $\phi_{out}^r(\tau)$ lui est associé comme suit :

— si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est un lot en déplacement :

$$\phi_{out}^r(\tau) = 0 \quad (3.16)$$

— si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ est en contact aval avec un lot $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$:

$$\phi_{out}^r(\tau) = \phi_i^h(\tau) \quad (3.17)$$

— si $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), S_i, v_i^r(\tau))$ est un lot contrôlable de sortie :

$$\phi_{out}^r(\tau) = \phi_i^{out} \quad (3.18)$$

■

D'après la proposition des flux instantanés d'entrée et de sortie d'un lot contrôlable 3.2, nous pouvons définir la variation (avancement) des positions de tête et de fin d'un lot contrôlable.

Définition 3.19. (*Variation de la position de tête et position de fin d'un lot contrôlable*)

Soit un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ appartenant au marquage m_i de la place lot triangulaire $p_i \in P^T$, avec un flux instantané ϕ_i^r , un flux instantané d'entrée $\phi_{in}^r(\tau)$ et un flux instantané de sortie $\phi_{out}^r(\tau)$. La variation de la position de tête $x_i^r(\tau)$ et de la position de fin $z_i^r(\tau)$ est donnée respectivement par les deux équations suivantes :

$$\dot{x}_i^r = \frac{\phi_{out}^r(\tau) - \phi_i^r(\tau)}{d_{out}^r(\tau) - d_i^r(\tau)} \quad (3.19)$$

et

$$\dot{z}_i^r = \frac{\phi_i^r(\tau) - \phi_{in}^r(\tau)}{d_i^r(\tau) - d_{in}^r(\tau)} \quad (3.20)$$

■

Entre deux événements datés, plusieurs équations gouvernent la dynamique hybride des lots contrôlables : entrée, déplacement et sortie. Cependant, la densité et la vitesse des lots contrôlables ne peuvent pas varier dans le temps, par contre leurs valeurs peuvent changer pendant l'occurrence d'un événement. En d'autres termes, ces deux caractéristiques (densité et vitesse) sont constantes par morceaux, alors que la longueur et la position ont des variations linéaires dans le temps.

En conséquence, pour tout lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ appartenant au marquage m_i de la place lot triangulaire $p_i \in P^T$, nous avons :

$$\dot{d}_i^r = \dot{v}_i^r = 0 \quad (3.21)$$

Évolution d'un lot en régime libre

L'évolution d'un lot contrôlable en régime libre est régie par différentes équations selon sa position dans la place. Nous calculons les variations de longueur et de position de tête en fonction des flux instantanés d'entrée/sortie des lot.

Définition 3.20. (*Lot contrôlable en régime libre*)

Un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ dans une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ est en régime libre s'il se déplace librement à sa propre vitesse $v_i^r(\tau)$ qui est égale à la vitesse maximale V_i de la place lot triangulaire. ■

Proposition 3.3. (*Évolution d'un lot contrôlable en régime libre*)

Soit une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$, un flux d'entrée $\phi_i^{in}(\tau)$, un flux de sortie $\phi_i^{out}(\tau)$ et un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ en régime libre. Les équations linéaires qui régissent les caractéristiques de ce lot sont exprimées par :

- *Entrée* : à un instant τ , un lot contrôlable créé $C\beta_i^r(\tau) = (0, d_i^r(\tau), 0, v_i^r(\tau))$ entre librement dans la place p_i conformément à :

$$\dot{l}_i^r = \dot{x}_i^r = v_i^r(\tau) \quad (3.22)$$

- *Déplacement* : à partir de l'instant τ , un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$, qui est en régime libre, se déplace librement dans la place lot triangulaire p_i selon les équations suivantes :

$$\dot{l}_i^r = 0; \quad \dot{x}_i^r = v_i^r(\tau) \quad (3.23)$$

- *Sortie* : à un instant τ , soit le lot contrôlable de sortie $OC\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), S_i, v_i^r(\tau))$, qui a un flux inférieur ou égal au flux de sortie de la place p_i c'est-à-dire $\phi_i^r(\tau) \leq \phi_i^{out}(\tau)$. À partir de cet instant τ , la sortie du lot est gouvernée par :

$$\dot{l}_i^r = -v_i^r(\tau); \quad \dot{x}_i^r = 0 \quad (3.24)$$

■

Évolution d'un lot en régime de congestion/décongestion

Passons maintenant à l'évolution d'un lot en régime de congestion ou décongestion. Commençons par donner les définitions des lots contrôlables en régime de congestion et du même pour le régime de décongestion.

Définition 3.21. (*Lot contrôlable en régime de congestion*)

Un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ dans une place lot triangulaire $p_i \in P^T$ avec un flux de sortie $\phi_i^{out}(\tau)$ est en régime de congestion s'il ne peut pas se déplacer à sa propre vitesse $v_i^r(\tau)$, mais doit la réduire, c'est-à-dire, il commence la congestion. Deux situations peuvent causer ce régime :

- $C\beta_i^r(\tau)$ est un lot de sortie et le flux de sortie de p_i est strictement inférieur au flux instantané du lot de sortie, c'est-à-dire $\phi_i^{out}(\tau) < \phi_i^r(\tau)$;
- $C\beta_i^r(\tau)$ est en contact aval avec un lot contrôlable $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$: $x_i^h(\tau) > x_i^r(\tau)$ qui a une vitesse inférieure c'est-à-dire $v_i^h(\tau) < v_i^r(\tau)$. ■

Remarque 3.1. *Il faut noter que pour un lot $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ qui est en contact aval avec $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$, même si la somme de densité de deux lots est inférieure à la densité maximale de la place p_i et la vitesse de lot $C\beta_i^r(\tau)$ est supérieure à la vitesse de $C\beta_i^h(\tau)$ c'est-à-dire, si $(d_i^r(\tau) + d_i^h(\tau) \leq d_i^{max}$ et $v_i^r(\tau) > v_i^h(\tau))$, le lot $C\beta_i^r(\tau)$ est en régime de congestion. ■*

Définition 3.22. *(Lot contrôlable en régime de décongestion)*

Un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ en état congestionné dans une place lot triangulaire p_i avec un flux de sortie $\phi_i^{out}(\tau)$ est en régime de décongestion s'il peut se déplacer à une vitesse plus grande que sa propre vitesse $v_i^r(\tau)$. Trois situations peuvent causer cette dynamique :

- $C\beta_i^r(\tau)$ est un lot contrôlable de sortie congestionné et le flux de sortie de p_i est strictement supérieur au flux de lot $C\beta_i^r(\tau)$ c'est-à-dire $\phi_i^{out}(\tau) > \phi_i^r(\tau)$;
- $C\beta_i^r(\tau)$ est un lot contrôlable congestionné sans contact aval ni amont (se déplace tout seul dans la place lot triangulaire p_i);
- $C\beta_i^r(\tau)$ est un lot contrôlable congestionné et qui est en contact aval avec un lot contrôlable $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$ qui a une vitesse supérieure c'est-à-dire $v_i^h(\tau) > v_i^r(\tau)$. ■

Nouvelles caractéristiques des lots contrôlables Maintenant que les régimes d'un lot contrôlable sont définis, nous allons déterminer dans cette partie les différentes caractéristiques d'éclatement attribuées à ces lots contrôlables. Contrairement à l'évolution en régime libre, si un lot contrôlable est en régime de congestion ou décongestion, il s'éclate en lui-même et un deuxième lot contrôlable est créé. Les nouvelles caractéristiques de ce deuxième lot sont définies en fonction des critères suivants :

1. la position du lot contrôlable c'est-à-dire soit une lot contrôlable en sortie ou en déplacement;
2. l'état d'un lot contrôlable libre ou congestionné;
3. le régime d'un lot contrôlable libre, congestion ou décongestion;
4. si le lot est en contact aval, en contact amont avec un autre lot ou s'il est sans contact.

Définition 3.23. *(Éclatement d'un lot contrôlable)*

Soit une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ avec un flux d'entrée $\phi_i^{in}(\tau)$ et un flux de sortie $\phi_i^{out}(\tau)$. Pour tout lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ en régime de congestion ou décongestion appartenant au marquage m_i de la place p_i , à un instant τ , le lot $C\beta_i^r(\tau)$ s'éclate en lui-même et en un lot $C\beta_i^{r'}(\tau)$ en aval ayant une longueur nulle et la même position que $C\beta_i^r(\tau)$. Les caractéristiques d'éclatement attribuées au lot $C\beta_i^{r'}(\tau)$ sont alors comme suit :

$$C\beta_i^{r'}(\tau) = (0, d_i^{r'}(\tau), x_i^r(\tau), v_i^{r'}(\tau))$$



La figure 3.5, montre l'éclatement du lot contrôlable $C\beta_i^r(\tau)$ sans contact en régime de congestion ou décongestion. Il faut noter que cet éclatement s'applique de la même manière que pour un lot contrôlable de sortie.

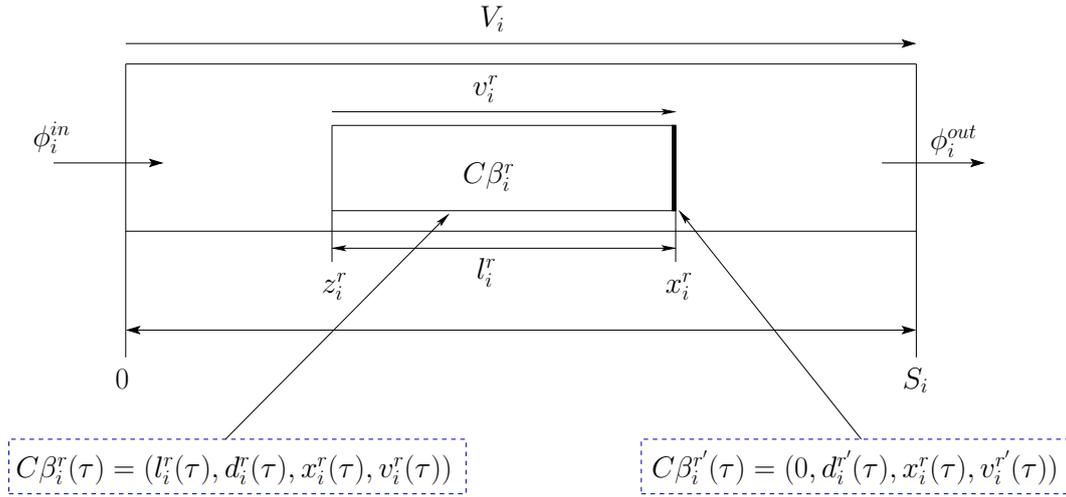


FIGURE 3.5 – Éclatement du lot contrôlable $C\beta_i^r(\tau)$ en déplacement sans contact

Dans la définition qui suit, nous allons déterminer la densité $d_i^{r'}(\tau)$ et la vitesse $v_i^{r'}(\tau)$ qui sont attribuées au lot $C\beta_i^{r'}(\tau)$ selon les quatre critères présentés ci-dessus.

✎ Soit

τ^+ définit la notion d'instant "à partir de" τ . Il faut noter que pendant l'utilisation de τ^+ , le temps n'a pas encore avancé, en d'autres termes le marquage $m_i(\tau)$ et l'état du système n'ont pas changé ($\tau^+ \neq \Delta\tau$).

Définition 3.24. (Caractéristiques de $C\beta_i^{r'}(\tau)$)

À un instant τ , soit un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ de la place lot triangulaire p_i avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ et son flux de sortie $\phi_i^{out}(\tau)$. Si $C\beta_i^r(\tau)$ est en régime de congestion ou décongestion, à un instant τ^+ , ce lot s'éclate en lui-même et en un lot en aval $C\beta_i^{r'}(\tau^+) = (0, d_i^{r'}(\tau^+), x_i^r(\tau^+), v_i^{r'}(\tau^+))$, la densité $d_i^{r'}(\tau^+)$ et la vitesse $v_i^{r'}(\tau^+)$ sont définies :

— si $C\beta_i^r(\tau)$ est en régime de congestion et en contact aval avec $C\beta_i^h(\tau)$ (sachant que $C\beta_i^r(\tau)$ en état libre et $C\beta_i^h(\tau)$ en état congestionné), les caractéristiques du lot $C\beta_i^{r'}(\tau^+)$ sont comme suit.

$$\begin{cases} d_i^{r'}(\tau^+) = d_i^h(\tau) \\ v_i^{r'}(\tau^+) = v_i^h(\tau) \end{cases} \quad (3.25)$$

- si $C\beta_i^r(\tau)$ est un lot en déplacement sans contact et en régime de décongestion, les caractéristiques du lot $C\beta_i^{r'}(\tau^+)$ sont données par :

$$\begin{cases} d_i^{r'}(\tau^+) = d_i^{cri} \\ v_i^{r'}(\tau^+) = V_i \end{cases} \quad (3.26)$$

- si $C\beta_i^r(\tau)$ est en régime de décongestion et en contact aval avec $C\beta_i^h(\tau)$, les caractéristiques de $C\beta_i^{r'}(\tau^+)$ sont discutées en fonction de l'état de $C\beta_i^h(\tau)$:

1. si $C\beta_i^h(\tau)$ est en état libre :

$$\begin{cases} d_i^{r'}(\tau^+) = d_i^{cri} \\ v_i^{r'}(\tau^+) = V_i \end{cases} \quad (3.27)$$

2. si $C\beta_i^h(\tau)$ est en état congestionné :

$$\begin{cases} d_i^{r'}(\tau^+) = d_i^h(\tau) \\ v_i^{r'}(\tau^+) = v_i^h(\tau) \end{cases} \quad (3.28)$$

- si $C\beta_i^r(\tau)$ est un lot de sortie en régime de congestion, les caractéristiques du lot $C\beta_i^{r'}(\tau^+)$ sont données par :

$$\begin{cases} d_i^{r'}(\tau^+) = d_i^{cong}(\tau) \\ v_i^{r'}(\tau^+) = v_i^{cong}(\tau) \end{cases} \quad (3.29)$$

- si $C\beta_i^r(\tau)$ est un lot de sortie en régime de décongestion, les caractéristiques du lot $C\beta_i^{r'}(\tau^+)$ sont données par :

$$\begin{cases} d_i^{r'}(\tau^+) = \frac{\phi_i^{out}(\tau)}{V_i} \\ v_i^{r'}(\tau^+) = V_i \end{cases} \quad (3.30)$$

■

Dynamiques en régime de congestion/décongestion Une fois qu'un lot contrôlable en régime de congestion ou décongestion est éclaté et les nouvelles caractéristiques du deuxième lot sont établies, nous pouvons nous consacrer à la définition des équations qui régissent la dynamique hybride de ces lots contrôlables. En effet, nous allons adopter les équations d'évolution d'un lot contrôlable en régime d'accumulation qui sont définies dans [36] et les étendre pour définir l'évolution (déplacement et sortie) des lots contrôlables en régime de congestion et décongestion dans une place lot triangulaire.

Proposition 3.4. (Déplacement en régime de congestion / décongestion)

Soit un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ de p_i , qui n'est, ni un lot de création, ni un lot de sortie, c'est-à-dire $(x_i^r(\tau) < S_i \text{ et } l_i^r(\tau) \neq 0)$, et en contact aval avec $C\beta_i^h(\tau) = (l_i^h(\tau), d_i^h(\tau), x_i^h(\tau), v_i^h(\tau))$. A un instant τ^+ , le lot contrôlable $C\beta_i^r$ est éclaté en lui-même et en un lot $C\beta_i^{r'}(\tau^+)$:

- $C\beta_i^r(\tau^+) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ et
- $C\beta_i^{r'}(\tau^+) = (0, d_i^{r'}(\tau^+), x_i^r(\tau^+), v_i^{r'}(\tau^+))$

À partir de cet instant τ^+ , l'évolution de deux lots contrôlables $C\beta_i^{r'}(\tau^+)$ et $C\beta_i^r(\tau^+)$ est gouvernée par :

$$\begin{cases} \dot{x}_i^{r'} = v_i^{r'}(\tau^+) \\ \dot{l}_i^{r'} = \frac{d_i^r(\tau)}{d_i^r(\tau) - d_i^{r'}(\tau^+)} \cdot (v_i^{r'}(\tau^+) - v_i^r(\tau)) \end{cases} \quad (3.31)$$

$$\begin{cases} \dot{x}_i^r = \frac{v_i^{r'}(\tau^+) \cdot d_i^{r'}(\tau^+) - v_i^r(\tau) \cdot d_i^r(\tau)}{d_i^{r'}(\tau^+) - d_i^r(\tau)} \\ \dot{l}_i^r = \frac{d_i^{r'}(\tau^+)}{d_i^{r'}(\tau^+) - d_i^r(\tau)} \cdot (v_i^{r'}(\tau^+) - v_i^r(\tau)) \end{cases} \quad (3.32)$$

■

Proposition 3.5. (Sortie en régime de congestion / décongestion)

Soit un lot contrôlable de sortie $OC\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), S_i, v_i^r(\tau))$ de p_i ayant un flux de sortie $\phi_i^{out}(\tau)$. A τ^+ le lot contrôlable $OC\beta_i^r$ est éclaté en lui-même et un lot en aval $OC\beta_i^{r'}(\tau^+)$:

- $C\beta_i^r(\tau^+) = (l_i^r(\tau), d_i^r(\tau), S_i, v_i^r(\tau))$,
- $OC\beta_i^{r'}(\tau^+) = (0, d_i^{r'}(\tau^+), S_i, v_i^{r'}(\tau^+))$;

À partir de l'instant τ , l'évolution des lots contrôlables $C\beta_i^r$ et $OC\beta_i^{r'}$ est gouvernée par :

$$\begin{cases} \dot{x}_i^{r'} = 0 \\ \dot{l}_i^{r'} = \frac{v_i^r(\tau) \cdot d_i^r(\tau) - \phi_i^{out}(\tau)}{d_i^{r'}(\tau^+) - d_i^r(\tau)} \end{cases} \quad (3.33)$$

$$\begin{cases} \dot{x}_i^r = \frac{\phi_i^{out}(\tau) - v_i^r(\tau) \cdot d_i^r(\tau)}{d_i^{r'}(\tau^+) - d_i^r(\tau)} \\ \dot{l}_i^r = \frac{\phi_i^{out}(\tau) - v_i^r(\tau) \cdot d_i^{r'}(\tau^+)}{d_i^{r'}(\tau^+) - d_i^r(\tau)} \end{cases} \quad (3.34)$$

■

Avant de présenter l'algorithme d'évolution des RdPLots Triangulaires et la méthode d'analyse, nous étendons dans la section suivante les RdPLots Triangulaires au RdPLots Triangulaires Contrôlés.

3.2 Réseaux de Petri Lots Triangulaires Contrôlés

Dans le même esprit des RdPLots Contrôlés proposés par Audry et Prunet dans [14], nous définissons dans cette section l'extension de RdPLots Triangulaires Contrôlés par l'ajout des deux événements contrôlés, à savoir, la variation de la vitesse de la place lot et la variation des flux maximaux des transitions continues et lots.

Définition 3.25. (*Réseau de Petri Lots Triangulaires Contrôlés*)

Soit un RdPLots Triangulaire avec une place lot triangulaire $p_i \in P^T$ et $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$. Un réseau de Petri Lots Triangulaire Contrôlé est un RdPLots Triangulaire où la vitesse maximale associée à la place p_i , peut varier : $v_i(\tau)$ avec $0 \leq v_i(\tau) \leq V_i$. Le flux maximal associé aux transitions continues ou lots $t_j \in T^C \cup T^B$, peut aussi varier : $\Phi_j(\tau)$ avec $0 \leq \Phi_j(\tau) \leq \Phi_j$. ■

Définition 3.26. (*Événements contrôlés*)

Soit un RdPLots Triangulaire Contrôlé, pour chaque ensemble de places lots triangulaires $p_i \in P^T$ et pour chaque ensemble de transitions continues ou lots $t_j \in T^C \cup T^B$. Deux événements contrôlés sont définis par :

- l'événement contrôlé de vitesse est un triplet (p_i, v_i, τ) où p_i est une place lot triangulaire, $v_i \in [0, V_i]$ est la vitesse maximale de p_i et τ est la date d'occurrence de cet événement.
- l'événement contrôlé de flux est un triplet (t_j, Φ_j, τ) où t_j est une transition continue ou lot, $\Phi_j \in [0, \Phi_j]$ est le flux maximal de t_j et τ est la date d'occurrence de cet événement.

Il faut noter que la variation de la vitesse de la place lot triangulaire impose une variation de sa densité critique et de son flux maximal. Ce qui nous amène à définir une densité critique variable.

Proposition 3.6. (*Densité critique variable*)

Soit une place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$, de vitesse maximale $v_i(\tau)$. À un instant τ , la densité critique variable $d_i^{cri}(v_i(\tau))$ est donnée par :

$$d_i^{cri}(v_i(\tau)) = \frac{W_i \cdot d_i^{max}}{v_i(\tau) + W_i} \quad (3.35)$$

avec $d_i^{cri} \leq d_i^{cri}(v_i(\tau)) \leq d_i^{max}$ ■

Nous remarquons que la variation de la densité critique impose une variation du flux maximal qui devient égal à $\Phi_i^{max}(v_i(\tau)) = v_i(\tau) \cdot d_i^{cri}(v_i(\tau))$, avec $0 \leq \Phi_i^{max}(v_i(\tau)) \leq \Phi_i^{max}$.

Remarque 3.2. La vitesse maximale $v_i(\tau) \in [0, V_i]$ dépend du temps, d'où la densité d_i^{cri} et le flux maximal Φ_i^{max} qui varient en fonction de cette vitesse $v_i(\tau)$ dépendent aussi du temps τ . Pour raison d'alléger l'écriture de la densité critique variable $d_i^{cri}(v_i(\tau))$ et le flux maximal variable $\Phi_i^{max}(v_i(\tau))$, nous supprimons τ de ces deux écritures, d'où $v_i(\tau)$ devient v_i , $d_i^{cri}(v_i(\tau))$ devient $d_i^{cri}(v_i)$ et $\Phi_i^{max}(v_i(\tau))$ devient $\Phi_i^{max}(v_i)$. ■

La figure 3.6 illustre une variation de la vitesse maximale de la place lot triangulaire p_i . Nous remarquons que la vitesse de propagation de la congestion W_i (Équation 3.2) est supposée constante, par contre une densité critique variable $d_i^{cri}(v_i)$ et un flux maximal variable $\Phi_i^{max}(v_i)$ sont associés à la place p_i .

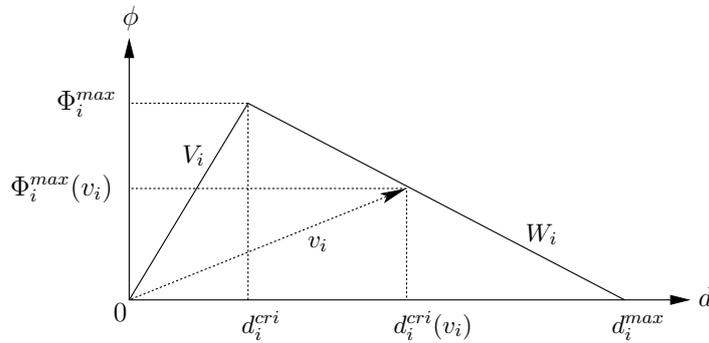


FIGURE 3.6 – Relation flux-densité triangulaire après variation de la vitesse maximale de la place p_i

Nous allons adopter la définition des états des lots contrôlables (voir définition 3.6) en fonction de la variation de la vitesse maximale de la place lot triangulaire.

Définition 3.27. (États des lots contrôlables après variation de la vitesse maximale de la place lot triangulaire)

Soit $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ un lot contrôlable de la place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ et une densité critique variable $d_i^{cri}(v_i)$ associée à la place p_i avec $v_i \in [0, V_i]$ sa vitesse maximale.

- $C\beta_i^r$ est dit lot contrôlable libre ou en état libre si sa densité est inférieure ou égale à la densité critique variable de la place p_i : $d_i^r(\tau) \leq d_i^{cri}(v_i)$;
- $C\beta_i^r$ est dit lot contrôlable congestionné ou en état congestionné si sa densité est strictement supérieure à la densité critique variable de la place p_i : $d_i^r(\tau) > d_i^{cri}(v_i)$. ■

3.2.1 Variation de la vitesse maximale de la place lot triangulaire

Les caractéristiques des lots contrôlables changent quand la vitesse maximale de transfert $v_i \in [0, V_i]$ de la place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ change. Nous

supposons que la vitesse maximale de transfert de la place p_i change de v_i à v'_i avec $v'_i \in [0, V_i]$. Deux situations peuvent être considérées : soit une augmentation, soit une diminution de la vitesse maximale de transfert, c'est-à-dire ($v'_i > v_i$, ou $v'_i < v_i$).

Définition 3.28. (*Caractéristiques d'un lot contrôlable après une diminution de la vitesse maximale de p_i*)

Soit $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ un lot contrôlable de la place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$ et $v_i \in [0, V_i]$ sa vitesse maximale à un instant τ . Si la vitesse de lot $C\beta_i^r(\tau)$ est supérieure à la nouvelle vitesse maximale v'_i : $v_i^r(\tau) > v'_i$ alors le lot doit changer sa vitesse à v'_i , c'est-à-dire $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v'_i)$. ■

Pour le cas de diminution de la vitesse maximale d'une place lot triangulaire p_i , $v'_i < v_i$, nous présentons dans la figure 3.7 les trois différentes situations d'un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$.

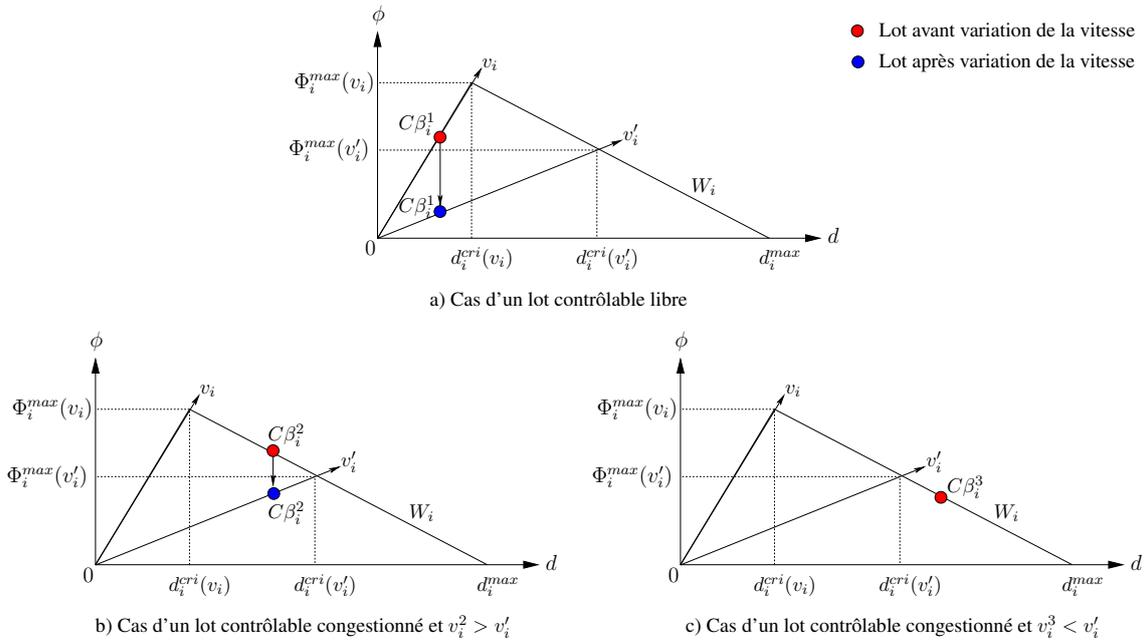


FIGURE 3.7 – États des lots contrôlables après une diminution de la vitesse maximale de p_i ($v_i > v'_i$)

1. Premier cas : $C\beta_i^1(\tau) = (l_i^1(\tau), d_i^1(\tau), x_i^1(\tau), v_i^1(\tau))$ est un lot contrôlable libre. Quand la vitesse maximale de la place lot triangulaire p_i change de v_i à v'_i telle que $v'_i < v_i$, le lot contrôlable $C\beta_i^1$ diminue sa vitesse en gardant la même densité. Il reste en état libre $C\beta_i^1(\tau) = (l_i^1(\tau), d_i^1(\tau), x_i^1(\tau), v'_i)$. Cela est représenté par le cas a) dans la figure 3.7).
2. Deuxième cas : $C\beta_i^2(\tau) = (l_i^2(\tau), d_i^2(\tau), x_i^2(\tau), v_i^2(\tau))$ est un lot contrôlable congestionné avec une vitesse supérieure à v'_i ($v_i^2(\tau) > v'_i$). Le lot $C\beta_i^2(\tau)$ réduit sa vitesse à v'_i tout en gardant sa densité $d_i^2(\tau)$. Il devient en état libre avec les caractéristiques suivantes $C\beta_i^2(\tau) = (l_i^2(\tau), d_i^2(\tau), x_i^2(\tau), v'_i)$. Cela est représenté par le cas b) dans la figure 3.7).

3. Troisième cas : $C\beta_i^3(\tau) = (l_i^3(\tau), d_i^3(\tau), x_i^3(\tau), v_i^3(\tau))$ est un lot contrôlable congestionné avec une vitesse inférieure à v'_i ($v_i^3(\tau) < v'_i$). Le lot contrôlable $C\beta_i^3(\tau)$ garde sa vitesse et sa densité en restant en état congestionné. Cela est représenté par le cas c) dans la figure 3.7).

De façon similaire, une augmentation de vitesse induit à un changement des caractéristiques des lots.

Définition 3.29. (Caractéristiques d'un lot contrôlable après une augmentation de la vitesse maximale de p_i)

Soit $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$ un lot contrôlable de la place lot triangulaire $p_i \in P^T$ avec $\gamma(p_i) = (V_i, d_i^{max}, S_i, \Phi_i^{max})$, de vitesse maximale $v_i \in [0, V_i]$ et densité critique variable $d_i^{cri}(v_i)$. À un instant τ , la vitesse maximale de la place lot change de v_i à v'_i avec $v_i < v'_i$, si la densité du lot $C\beta_i^r(\tau)$ est inférieure à la nouvelle densité critique variable $d_i^{cri}(v'_i)$, le lot contrôlable change sa vitesse à v'_i c'est-à-dire $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v'_i)$. Si la densité de $C\beta_i^r(\tau)$ est supérieure à la densité critique variable $d_i^{cri}(v'_i) : (d_i^{cri}(v'_i) < d_i^r < d_i^{cri}(v_i))$, d'après l'équation 3.4 le lot $C\beta_i^r(\tau)$ change sa vitesse à :

$$v_i^r(\tau) = \frac{W_i \cdot (d_i^{max} - d_i^r(\tau))}{d_i^r(\tau)}$$

■

Pour le cas de l'augmentation de la vitesse maximale de la place lot triangulaire p_i , $v'_i > v_i$, nous présentons dans la figure 3.8 les trois différentes situations d'un lot contrôlable $C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$.

1. Premier cas : $C\beta_i^1(\tau) = (l_i^1(\tau), d_i^1(\tau), x_i^1(\tau), v_i^1(\tau))$ est un lot contrôlable libre et sa densité est inférieure à $d_i^{cri}(v'_i)$. Quand la vitesse de la place lot triangulaire p_i change de v_i à v'_i telle que $v'_i > v_i$, le lot contrôlable $C\beta_i^1$ augmente sa vitesse jusqu'à v'_i en gardant sa densité. Il reste en état libre $C\beta_i^1(\tau) = (l_i^1(\tau), d_i^1(\tau), x_i^1(\tau), v'_i)$. Cela est représenté par le cas a) dans la figure 3.8).
2. Deuxième cas : $C\beta_i^2(\tau) = (l_i^2(\tau), d_i^2(\tau), x_i^2(\tau), v_i^2(\tau))$ est un lot contrôlable libre avec une densité supérieure à $d_i^{cri}(v'_i)$. Quand la vitesse maximale de la place p_i change de v_i à v'_i telle que $v'_i > v_i$, le lot contrôlable $C\beta_i^2(\tau)$ garde sa densité mais en augmentant sa vitesse jusqu'à $v'_2(\tau)$ avec $v'_2(\tau) = \frac{W_i \cdot (d_i^{max} - d_i^r(\tau))}{d_i^r(\tau)}$ d'après l'équation 3.5. Ce lot devient un lot contrôlable congestionné avec les caractéristiques suivantes $C\beta_i^2(\tau) = (l_i^2(\tau), d_i^2(\tau), x_i^2(\tau), v'_2(\tau))$. Cela est représenté par le cas b) dans la figure 3.8).
3. Troisième cas : $C\beta_i^3(\tau) = (l_i^3(\tau), d_i^3(\tau), x_i^3(\tau), v_i^3(\tau))$ est un lot contrôlable congestionné. Quand la vitesse maximale de la place lot triangulaire p_i change de v_i à v'_i telle que $v'_i > v_i$, ce lot contrôlable ne change pas d'état et il reste un lot congestionné en gardant toutes ses caractéristiques. Cela est représenté par le cas c) dans la figure 3.8).

Il faut noter que, lors d'un changement de vitesse d'une place lot triangulaire p_i , tous les lots contrôlables qui composent le marquage m_i de place doivent s'adapter à la nouvelle vitesse.

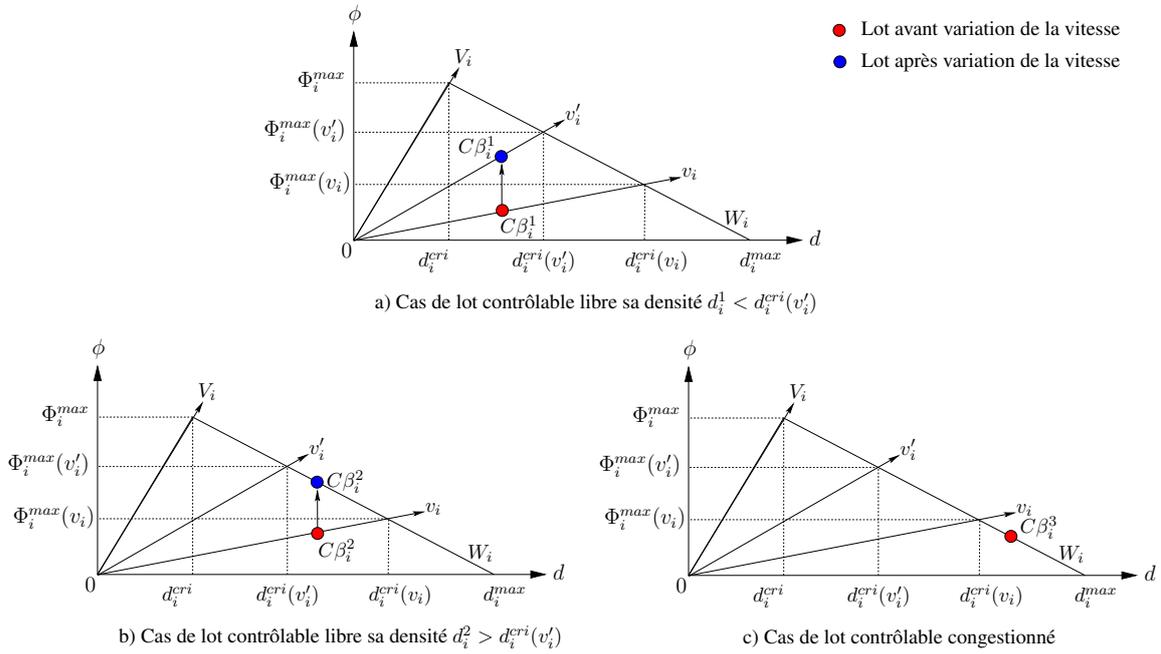


FIGURE 3.8 – États des lots Contrôlables après une augmentation de la vitesse maximale de p_i : ($v_i < v'_i$)

3.3 Évolution des RdPLots Triangulaires Contrôlés

L'évolution d'un RdPLots triangulaire contrôlé se fait par franchissement des transitions discrètes, continues et lots. La dynamique des RdPLots Triangulaires Contrôlés est basée sur une approche à événement discret avec une évolution linéaire (constante) entre deux événements datés. Cette évolution est gouvernée, pour les noeuds discrets et continus, par les règles d'évolution des RdPLots Généralisés [35] et repris dans la section 3.1.3. Pour les places lots triangulaires, l'évolution dynamique des lots contrôlables est celle déjà vue dans les sections précédentes.

Entre deux événements datés, l'état d'un RdPLots triangulaire contrôlé est invariant (IB-state), et défini comme suit :

Définition 3.30. (État invariant d'un RdPLots triangulaire contrôlé)

L'état invariant d'un RdPLots triangulaire contrôlé correspond à une période du temps où :

- le marquage des places discrètes est constant ;
- les marques réservées dans les places discrètes et continues sont constantes ;
- le vecteur de flux instantané de franchissement des transitions continues ou lots est constant ;
- la densité de sortie d'une place lot est constante. ■

3.3.1 Liste des événements

Plusieurs types d'événements peuvent survenir pendant la dynamique. Cette liste définit les trois types d'événements : les *événements internes* (c'est-à-dire, les événements qui se produisent à l'intérieur d'une place lot triangulaire $p_i \in P^T$), les *événements externes* (c'est-à-dire, les événements qui se produisent au niveau des RdPLots) et les *événements contrôles* (c'est-à-dire, les événements de contrôle à partir de l'extérieur du modèle).

L'état invariant d'un RdPLots triangulaire contrôlé change si et seulement si un (ou éventuellement plusieurs en même temps) des événements suivants se produit :

- Événements internes :
 - i.1 - un lot devient lot de sortie $C\beta_i^r = OC\beta_i^r$;
 - i.2 - rencontre de deux lots ;
 - i.3 - un lot devient dense ;
 - i.4 - destruction d'un lot $C\beta_i^r = \emptyset$.
- Événements externes :
 - e.1 - une transition discrète est franchie ;
 - e.2 - une place continue devient vide ;
 - e.3 - une transition discrète devient valide ;
 - e.4 - un lot devient lot de sortie (événement i.1 ci-dessus) ;
 - e.5 - un lot de sortie est détruit (événement i.4 ci-dessus).
- Événements contrôlés :
 - c.1 - le flux maximal d'une transition continue ou lot est modifié : $\Phi_j(\tau) = \Phi_j'(\tau)$;
 - c.2 - la vitesse maximale d'une place lot triangulaire est modifiée : $v_i(\tau) = v_i'(\tau)$.

Finalement, l'équation fondamentale qui gouverne le comportement dynamique d'un RdPLots triangulaire en terme de vecteur de quantité de marquage est :

$$\mathbf{q}(\tau) = \mathbf{q}(\tau_0) + \mathbf{C} \cdot \mathbf{z}(\tau) \quad (3.36)$$

où $\mathbf{z}(\tau) \in \mathbb{R}^+$ est le vecteur caractéristique qui indique le nombre de fois que les transitions discrètes ont été franchies et les quantités franchies par les transitions continues et lots pendant $[\tau_0, \tau]$.

3.3.2 Algorithme d'évolution

L'algorithme d'évolution des RdPLots Triangulaires Contrôlés est représenté dans la figure 3.9. Cet algorithme est basé sur l'évolution des RdPLots Généralisés que nous avons modifiée pour prendre en compte les lots contrôlables à l'intérieur des places lots triangulaires.

Nous pouvons séparer l'algorithme en deux parties : l'évolution du réseau proprement dit et le calcul des dates futures. L'évolution du réseau consiste basically à faire une mise à jour de l'état des places, faire évoluer les lots, calculer l'état des transitions et ses flux instantanés de franchissement et, mettre à jour les caractéristiques des places dans le cas où le modèle a des événements contrôlés. La deuxième partie, que nous appelons *échancier*, consiste à calculer les dates futures de tous les événements possibles du modèle à partir de la date courante et de faire avancer l'horloge à la date de l'événement le plus proche. J'ai décidé de ne pas aborder le calcul des dates futures dans ce document mais le lecteur peut trouver les équations ainsi que les preuves dans [48].

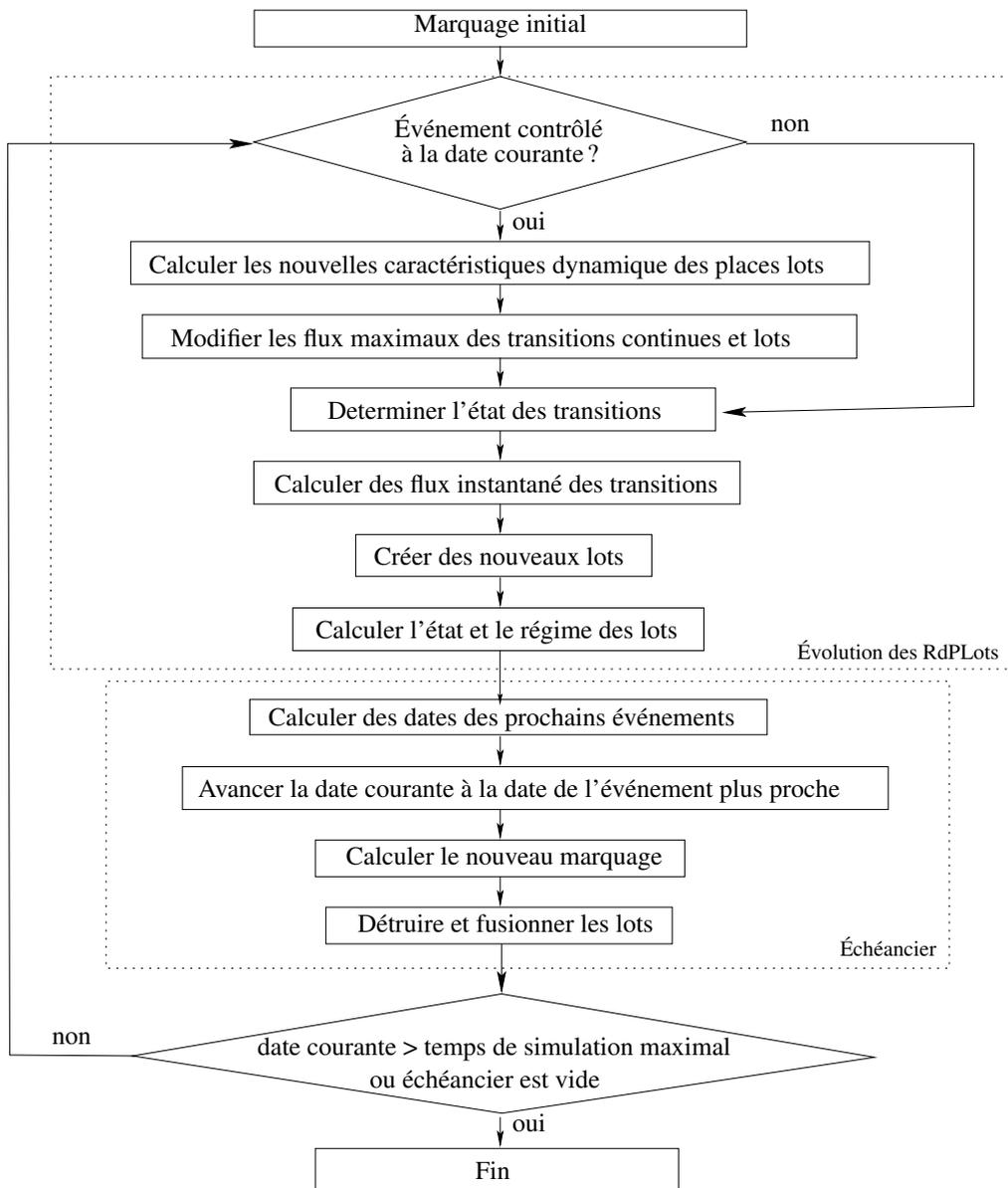


FIGURE 3.9 – Algorithme d'évolution d'un modèle RdPLots

3.3.3 Graphe d'évolution

Le graphe d'évolution des RdPLots Triangulaires permet de réaliser une analyse du comportement dynamique du système en déduisant des indices de performances tels que : le temps de cycle, les états transitoires et stables, ...).

Le graphe d'évolution est une série de noeuds qui représentent les états invariants du système et qui sont liés par des arcs avec des transitions étiquetées par le ou les événements qui ont causé le changement de l'état invariant.

Le noeud d'un graphe d'évolution des RdPLots Triangulaires est décomposé en trois parties correspondant à un état invariant, défini par la constance des marquages des places discrètes, des marques réservées des places discrètes et continues qui se trouvent en amont des transitions discrètes, l'état et le régime d'un lot contrôlable dans une place lot triangulaire, des flux instantanés de franchissement des transitions continues et lots.

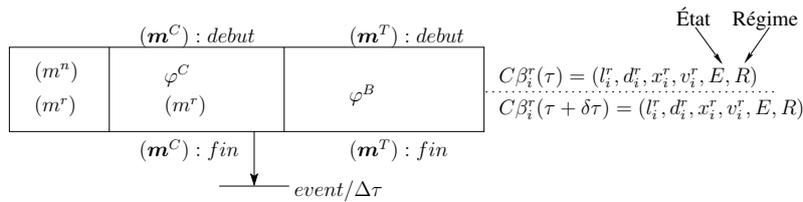


FIGURE 3.10 – Un noeud du graphe d'évolution d'un modèle RdPLots Triangulaire

La figure 3.10 présente un noeud d'un graphe d'évolution telle que :

1. la première partie représente le vecteur de marquage discret m^n et le marquage réservé m^r des places discrètes.
2. la deuxième partie représente le vecteur de flux instantané de franchissement des transitions continues φ^C , et le marquage réservé des places continues m^r . Le marquage total des places continues m^n au début et à la fin de la phase est également représenté.
3. la troisième partie représente le vecteur de flux instantané de franchissement des transitions lots φ^B et le marquage des places lots triangulaires m^T au début et à la fin de cette phase de calcul de marquage. Les caractéristiques des lots contrôlables ($C\beta_i^r(\tau) = (l_i^r(\tau), d_i^r(\tau), x_i^r(\tau), v_i^r(\tau))$) ainsi que leurs états (libre ou congestionné) et régimes (libre, congestion ou décongestion) sont décrits sur le côté droit du noeud et définis comme suit :
 - (a) l'ensemble des états d'un lot contrôlable est noté par $E = \{L, C\}$ (L : pour état libre et C pour état congestionné) ;
 - (b) l'ensemble des régimes d'un lot contrôlable est noté par $R = \{L, C, D\}$ (L : pour régime libre, C pour régime de congestion et D pour régime de décongestion).

Ces noeuds sont reliés entre eux par des arcs avec une transition marquée qui détermine les événements survenus et le délai passé entre deux états invariants consécutifs. L'étiquette associée à cette transition est notée comme suit : $\{\text{événements produit} / \Delta\tau\}$ où $\Delta\tau$ est le délai passé entre les événements.

Exemple 3.2. La figure 3.11 représente le graphe d'évolution du modèle RdPLots triangulaire présenté dans la figure 3.4.

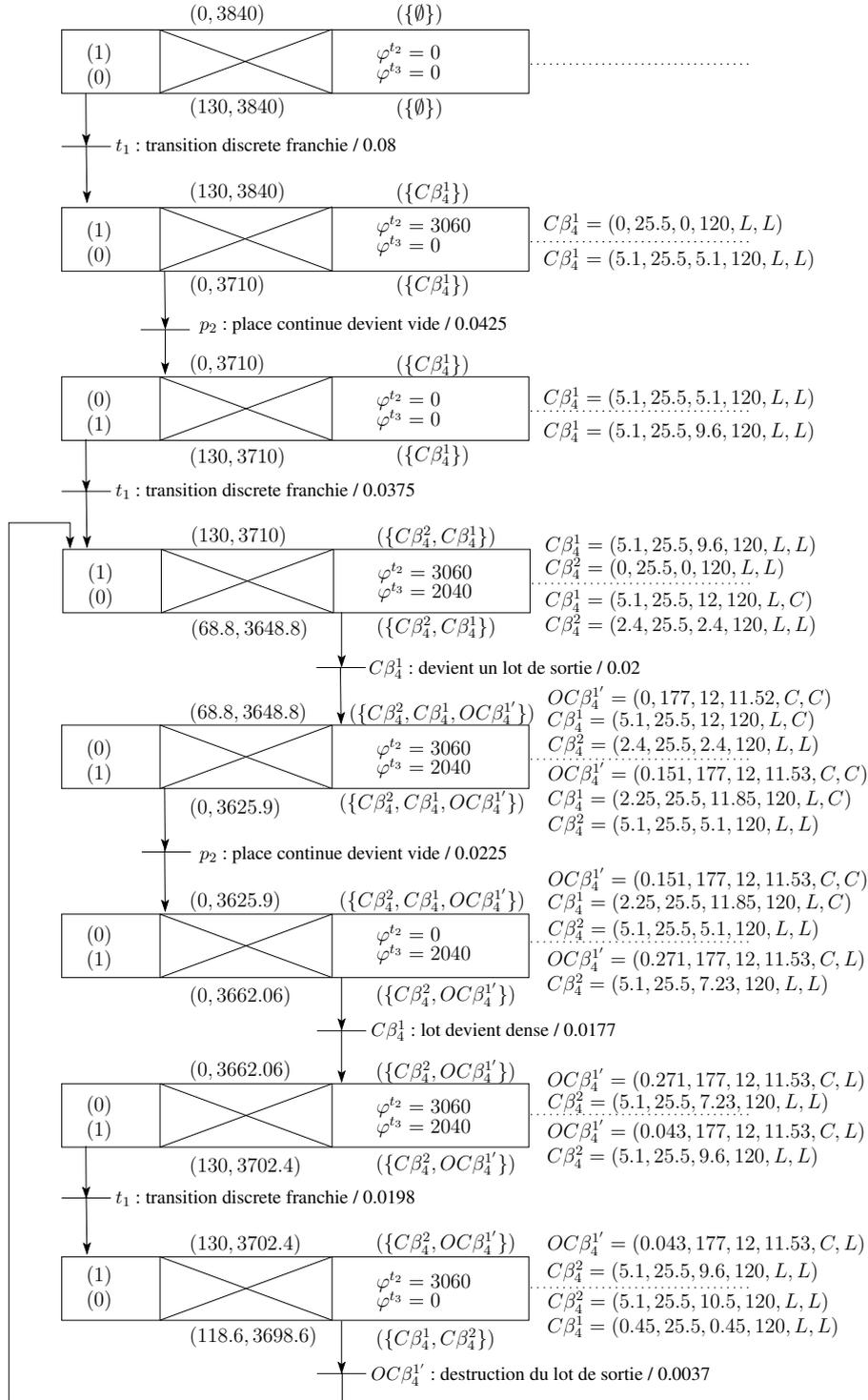


FIGURE 3.11 – Graphe d'évolution du modèle RdPLots triangulaire de la figure 3.4.

Dans ce chapitre, j'ai présenté le formalisme de RdPLots Triangulaires et sa nouvelle relation flux/densité. Les phénomènes de congestion et décongestion, engendrés par cette relation flux/densité triangulaire, ont multiplié le nombre des cas possibles dans l'évolution des lots. Avec des lots qui peuvent maintenant évoluer avec des vitesses différentes, ces phénomènes de congestion et décongestion peuvent aussi être multiples à l'intérieur des places lots. Cela nous permet de représenter assez fidèlement les comportements observés dans le trafic routier. Dans le chapitre suivant, j'introduis quelques concepts du trafic routier que j'applique aux RdPLots Triangulaires dans la modélisation dans un cas d'étude avec des données réelles.

Chapitre 4

Modélisation du trafic routier par RdPLots Triangulaires

Une des applications directes des RdPLots Triangulaires est la modélisation du trafic routier. Je présente dans ce chapitre un cas d'utilisation de ce formalisme dans ce contexte de modélisation et analyse du trafic routier. Pour cela, j'introduis brièvement le contexte d'étude du trafic routier ainsi que les concepts nécessaires à la modélisation. Le lecteur pourra facilement faire la liaison entre ces concepts et les éléments des RdPLots Triangulaires définis dans le chapitre précédent.

4.1 Contexte d'étude du trafic routier

La circulation routière constitue un processus complexe impliquant divers comportements de nature continue, comme observé sur les axes autoroutiers, ou discrète, notamment sur de feux de signalisation ou encore aux carrefours équipés de panneaux à messages variables. Pour appréhender la dynamique de ces systèmes dans l'espace et le temps, l'utilisation d'outils appropriés pour prédire, évaluer et analyser les phénomènes du trafic s'avère indispensable. Les premières investigations sur la circulation remontent aux années 1930, menées par Greenshields et al. [59], qui se sont penchés sur les performances de la circulation aux intersections. Les années 1950 ont vu émerger de nouvelles études basées sur des approches théoriques variées telles que le suivi du véhicule, la théorie du débit de la circulation routière [73, 88] et la théorie des files d'attente. Dans les années 1970, une approche novatrice a été introduite, s'inspirant de l'écoulement des fluides en mécanique des fluides [84].

Un des points cruciaux dans l'étude du trafic est la modélisation et surtout le contrôle de la congestion. Un contrôle optimal de la congestion permet une maximisation du flux de véhicules [82] et donc une réduction des coûts.

L'analyse du problème de la congestion a été déjà étudiée dans une grande variété de mo-

dèles, de méthodes et d'architectures [61]. Si nous nous focalisons sur les travaux basés sur les formalismes à événements discrets et des modèles hybrides, plusieurs auteurs utilisent ces formalismes hybrides pour la modélisation et l'analyse des systèmes de circulation, soit avec une représentation macroscopique, soit microscopique : les réseaux de Petri continus à vitesse constante ont été utilisés dans [43] pour étudier la circulation dans la ville de Turin en Italie, les auteurs de [96] utilisent des réseaux de Petri continus à vitesse variable pour représenter le phénomène de congestion, tandis que dans [63], les auteurs ont étudié le contrôle optimal des feux de signalisation d'un système de transport urbain. De nombreuses recherches ont été menées pour contrôler et prédire la congestion dans les systèmes de transport routier par des techniques récentes telles que la limitation variable de la vitesse (*VSL - Variable Speed Limite*), les systèmes de régulation des rampes d'accès (RMS) [81, 30]. L'impact principal du VSL sur le flux de la circulation est censé être la réduction de la vitesse moyenne sous des densités critiques, et l'homogénéisation des vitesses, c'est-à-dire la réduction des différences de vitesse entre les véhicules.

Récemment, Demongodin [36] définit les RdPLots avec des lots contrôlables comme un modèle mésoscopique pour représenter les goulots d'étranglement de véhicules et contrôler la vitesse. Notre objectif a été de proposer une extension du formalisme de RdPLots avec lots contrôlables qui soit adaptée aux phénomènes de congestion et décongestion du trafic routier.

Nous commençons, dans la section suivante, par introduire quelques concepts du trafic routier nécessaire à sa modélisation par des RdPLots Triangulaires.

4.2 Quelques notions et concepts du trafic routier

4.2.1 Niveaux de modélisation

La modélisation du trafic routier peut être réalisée à différents niveaux. Chaque modèle cherche à décrire et à analyser le trafic avec un certain niveau de détail. Hoogendoorn et Bovy [61] ont proposé une classification en trois catégories de modèles :

Modèles macroscopiques Le trafic est décrit avec un niveau élevé d'agrégation. Il est considéré comme l'écoulement d'un fluide sans distinguer ses composantes. Il est caractérisé, de façon global, par des mesures telles que le débit q , la densité k et la vitesse V . Le niveau général de détail est très réduit. Le premier modèle macroscopique, connu sous le nom de LWR, a été initialement proposé par Lighthill et Whitham [73] et complété par Richard [88] et repose sur la loi de la conservation de la masse. Il part du principe que la masse est transportée sans perte ni accumulation.

Les éléments fondamentaux qui représentent les flux de trafic routier de LWR sont liés aux trois relations suivantes :

1. La loi constitutionnelle des milieux continus :

$$q(\tau) = k(\tau) \cdot V(\tau) \quad (4.1)$$

2. La loi de conservation : le nombre de véhicules qui entrent en un moment donné sur une section doit être égal au nombre de véhicules sortants de la section dans le même intervalle de temps.

$$\frac{\partial k(\tau)}{\partial \tau} = \frac{\partial q(\tau)}{\partial x} = 0 \quad (4.2)$$

3. Le diagramme fondamental : pour chaque élément d'infrastructure, les trois variables fondamentales de vitesse moyenne, de densité et de débit, sur lesquelles reposent les modèles macroscopiques, sont liées entre elles par une relation fondamentale telle que le débit est égal au produit de la densité par la vitesse.

$$q(\tau) = k(\tau) \cdot V(k(\tau)) \quad (4.3)$$

Plusieurs études ont été menées de façon à trouver un diagramme fondamental qui décrit avec précision cette relation flux-densité dans le trafic surtout lorsque nous observons des phénomènes de congestion et décongestion. Nous adaptons dans nos travaux le *diagramme fondamental triangulaire* proposé par Daganzo [33] sans faire un état de l'art sur tous les diagrammes proposés. Le diagramme fondamental triangulaire, représenté dans la figure 4.1, est largement adopté par la communauté de par sa simplicité dans le calcul du flux. L'équation 4.4 ci-après décrit aussi bien le flux lorsque le trafic est dans un état libre que lorsqu'il est congestionné.

$$q(k) = \begin{cases} k \cdot V_{max} & \text{si } 0 \leq k \leq k_{cri} \\ W \cdot (k_{max} - k) & \text{si } k_{cri} \leq k \leq k_{max} \end{cases} \quad (4.4)$$

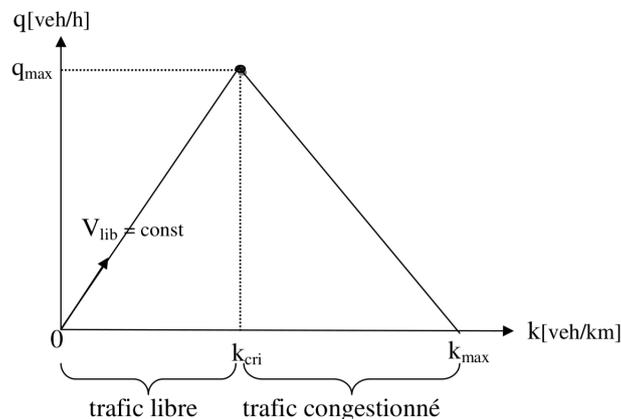


FIGURE 4.1 – Diagramme fondamental triangulaire proposé par Daganzo

Modèles microscopiques Les modèles microscopiques du trafic reposent sur une description du comportement individuel des véhicules. Cette description fine du trafic est réalisée en se basant sur des modèles représentant les interactions inter-véhicules ainsi que les interactions entre le véhicule individuel et son environnement. Un modèle microscopique décrit à la fois le comportement spatio-temporel des composants du système (véhicules et conducteurs) et leurs interactions à un niveau élevé de détail. Par exemple, pour chaque véhicule participant, le changement de voie de circulation est décrit comme une chaîne de décisions du conducteur. Cependant, en plus de la description détaillée du comportement en trafic, le comportement de contrôle du véhicule (par exemple, le changement des vitesses) est modélisé aussi selon les conditions environnementales.

Les modèles microscopiques sont appropriés pour la modélisation des interactions entre véhicules à un haut niveau de détail nécessaire pour évaluer plusieurs Systèmes de Transport Intelligents (*ITS – Intelligent Transport Systems*), mais ils sont limités à de petites zones en raison de la grande quantité de données nécessaires.

Dans le domaine de la modélisation microscopique, deux grandes classes de modèles peuvent être identifiées. La première est basée sur les *automates cellulaires* [80, 79]. La seconde est celle caractérisant les modèles *véhicule-suiveur* (car-following models) [17].

Modèles mésoscopiques Les modèles mésoscopiques se situent au niveau intermédiaire de détails entre les modèles microscopiques et macroscopiques. Les véhicules et le comportement de conducteurs ne sont pas étudiés d'une manière individuelle mais plutôt par agrégation. En effet, au lieu de s'intéresser au véhicule individuel, l'approche mésoscopique se focalise sur des paquets de véhicules caractérisés par les mêmes propriétés (c'est-à-dire même débit, densité, vitesse, etc.). Chaque paquet constitue une entité autonome en circulation dans le réseau autoroutier suivant des lois issues de la dynamique macroscopique du débit. Ils ont la possibilité de modéliser les grands réseaux sans un grand effort sur le codage et le calibrage, tout en assurant une meilleure représentation de la dynamique du trafic et les comportements individuels des participants, par rapport aux modèles macroscopiques. Les modèles mésoscopiques sont utilisés pour la planification, mais aussi pour des opérations en temps réel. Ils sont plus flexibles que les modèles macroscopiques en termes de modélisation des éléments importants, tels que le comportement de transport (par exemple, le choix d'un itinéraire, l'estimation de la longueur de congestion, l'estimation de temps de décongestion, etc.). Même si nous trouvons dans la littérature moins des modèles mésoscopiques que de modèles microscopiques et macroscopiques, nous pouvons citer, parmi d'autres, les travaux de : Taylor [93] qui modélise le réseau par des noeuds et des arcs ; Ben-Akiva et al. [16] qui utilisent un modèle de files d'attente pour représenter la congestion ; Jayakrishnan et al. [62] qui s'intéressent aux retards occasionnés aux intersections par des feux de signalisation.

4.2.2 Paramètres du trafic routier

La similitude du trafic routier, au niveau d'un tronçon d'une route, avec le comportement des fluides permet de le caractériser par la théorie de l'hydrodynamique. Nous nous intéressons à un instant τ au débit $q(\tau)$, à la densité $k(\tau)$ et à la vitesse $V(\tau)$.

Le débit routier : exprime le nombre de véhicules qui passe en un endroit en une unité de temps. Si le nombre de véhicules mesuré par un capteur dans un intervalle $\Delta\tau$ est $N(\tau)$, le débit du trafic s'exprime :

$$q(\tau) = \frac{N(\tau)}{\Delta\tau} \quad (4.5)$$

La densité routière : exprime le nombre de véhicules qui se trouvent sur un tronçon de la route. Si le tronçon de la route sur lequel la densité est mesurée est d'une longueur L et le nombre de véhicules qui se trouvent sur ce tronçon est donné par $N(\tau)$, la densité routière s'exprime par le rapport :

$$k(\tau) = \frac{N(\tau)}{L} \quad (4.6)$$

La vitesse moyenne : exprime la moyenne des vitesses des véhicules qui se trouvent sur un tronçon de la route. La vitesse moyenne peut être déterminée par le rapport entre la densité et le débit :

$$V(\tau) = \frac{q(\tau)}{k(\tau)} \quad (4.7)$$

4.2.3 Phénomènes de congestion et décongestion

Il existe deux états de circulation dans le trafic routier : fluide et congestionné. Lorsque la demande à un point donné est inférieure à la capacité de l'infrastructure, le trafic est considéré comme fluide ou libre. Lorsque la demande excède la capacité de l'infrastructure, le trafic est dit congestionné. Plusieurs diagrammes fondamentaux ont essayé de formaliser ces deux régimes de la circulation, en liant trois variables de mesure de la circulation : le débit, la densité et la vitesse du flot.

Nous expliquons le phénomène de congestion à travers la représentation débit-densité du diagramme fondamental. Lorsqu'il y a peu de véhicules sur la route (densité faible), nous supposons que le débit qui passe en un point augmente. Plus les véhicules sont nombreux sur un

segment de voie situé en amont d'un point donné (densité), plus ils sont nombreux à passer dans les minutes suivantes en ce point (débit). Cela est possible jusqu'à ce que le débit atteigne la capacité de la voie. À ce moment-là, il y a une contrainte qui s'oppose à l'écoulement libre des véhicules. C'est le point critique à partir duquel apparaît la congestion. Au-delà du point critique, plus la densité augmente, plus le débit passant en un point diminue. Le régime de congestion apparaît quand, au moins temporairement, la demande devient supérieure à l'offre. La file d'attente occasionnée par la congestion ne disparaît que lorsque la demande en amont du point de congestion baisse [29].

Plusieurs causes peuvent être à l'origine d'une congestion. Par exemple, l'augmentation de la demande aux heures de pointe peut causer des congestions récurrentes à un point fixe d'un tronçon routier. Un autre exemple est l'occurrence d'un incident sur la route, ce qui cause une réduction de l'offre. Pendant que la demande est supérieure à l'offre, le nombre de véhicules s'accumulent augmentant la longueur de la congestion. Une fois la demande redevenue inférieure à l'offre, nous observons un phénomène de décongestion.

Un point important dans la caractérisation d'un tronçon est défini à partir de quel ratio offre/demande nous passons d'un trafic fluide à un trafic congestionné. Cela est défini par la *densité critique* et caractérise aussi à quel moment le flux est maximal sur le tronçon (voir figure 4.1).

4.3 Exemple de modélisation du trafic routier

Afin de valider l'utilisation des RdPLots Triangulaires dans la modélisation du trafic routier, nous avons modélisé plusieurs cas réels de tronçons routiers. Cela nous a permis de valider les différents comportements observés dans le trafic, tels qu'une évolution en régime libre, de congestion et de décongestion et qui sont pris en compte dans le formalisme. J'ai décidé de présenter, dans cette section, une étude d'un tronçon de l'autoroute A51 qui relie Marseille et Aix en Provence et que nous avons utilisé pour valider l'évolution en régime de décongestion.

4.3.1 Modèle physique

Le tronçon d'autoroute, étudié ici, est de longueur 1.207 km. Il commence au kilomètre 2.3 jusqu'au kilomètre 3.507. Pour ce cas d'étude, il convient de préciser que le flux de ce tronçon sera modélisé uniquement pour la direction Marseille vers Aix en Provence (voir figure 4.2).

Collecte des données

Nous avons effectué le relevé des données le samedi 10 Octobre 2015 de 15h58 jusqu'à 17h34 (1 heure et 36 minutes). Sur ce tronçon, il existe 2 boucles : B1 a une longueur de 0.9 km

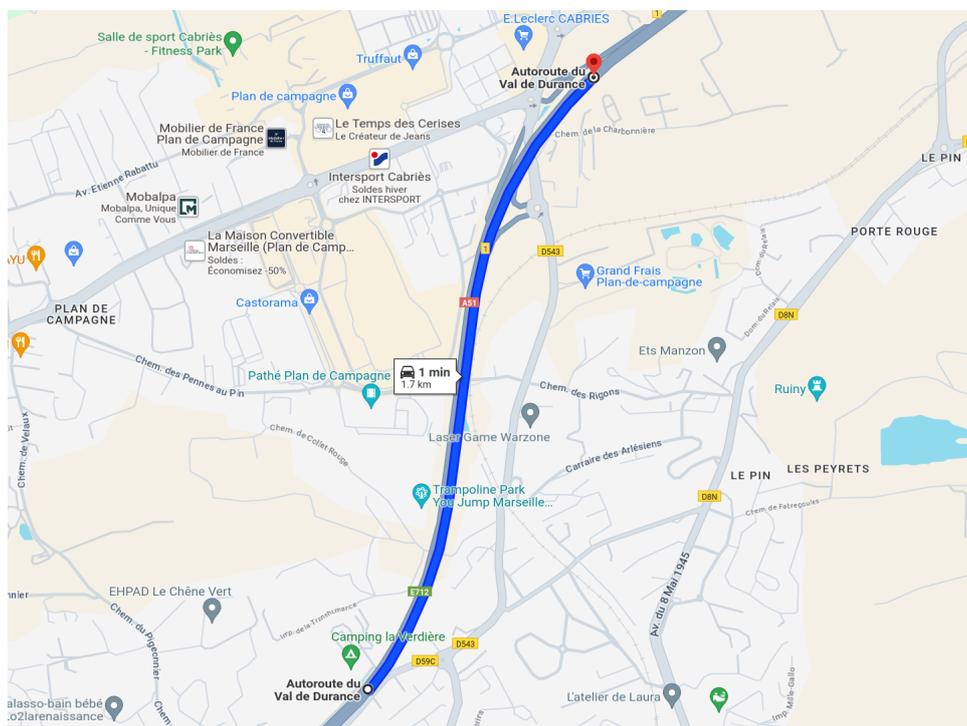


FIGURE 4.2 – Tronçon d’autoroute A51 de Marseille vers Aix-en-Provence

et B2 de longueur 0.307 km (voir figure 4.3). La boucle B3 servira à comparer les flux de sortie du tronçon et celui de la place lot triangulaire.

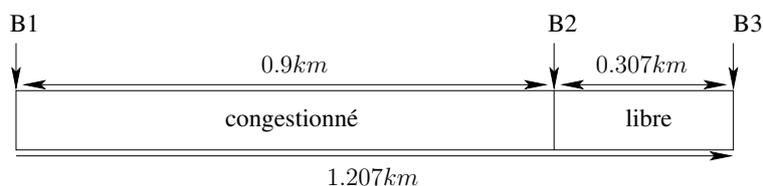


FIGURE 4.3 – Disposition des boucles de détection sur le tronçon de A51

Variation des flux d’entrée du tronçon : Le flux d’entrée de ce tronçon d’autoroute est détecté par la boucle B1 à partir de 15h58. La figure 4.4 montre la variation du flux d’entrée en cas réel entre 15h58 et 17h34. Cette variation est mesurée toutes les 6 minutes.

Dans ce cas d’étude, nous considérons que le flux maximal de ce tronçon est égal au flux maximal détecté par la boucle B3. Ce flux maximal est égal à 4450 veh/h et a été détecté à 16h04.

Détection des états du trafic routier : La boucle de détection B1 a détecté l’état bouchon (congestionné) du trafic à 15h58 jusqu’à la date 16h04. A partir de 16h10 jusqu’à 17h34, les deux boucles B1 et B2 ont détecté un état fluide (voir tableau 4.1).

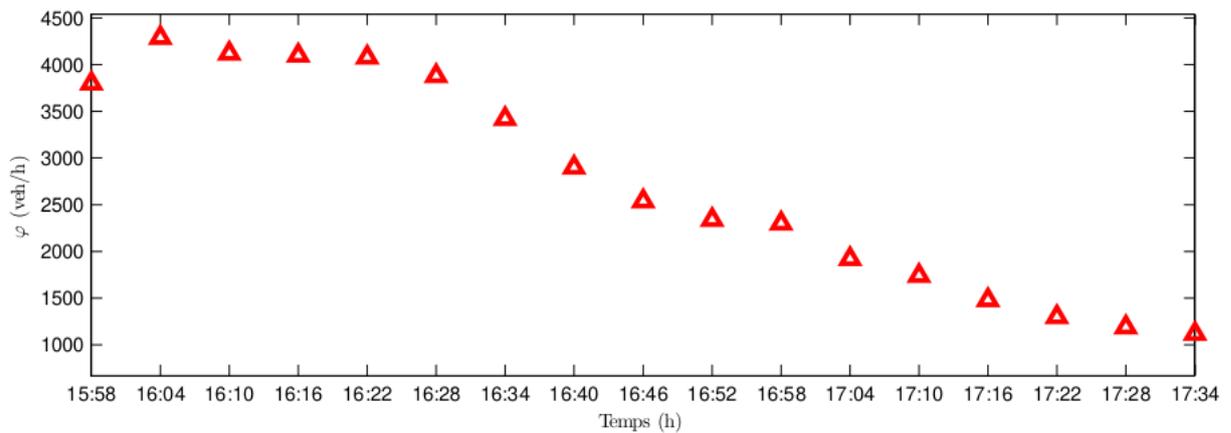


FIGURE 4.4 – Flux d'entrée en cas réel entre 15h58 et 17h34

Boucle de détection	État à 15h58	État à 16h04	État de 16h10 à 17h34
B1	bouchon	bouchon	fluide
B2	fluide	fluide	fluide

TABLE 4.1 – États du trafic de 15h58-17h34

4.3.2 Modèle RdPLots Triangulaire pour le tronçon de A51

La figure 4.5 montre le modèle RdPLots Triangulaire pour le tronçon d'autoroute étudié.

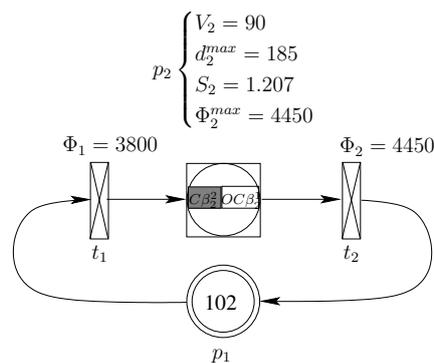


FIGURE 4.5 – RdPLots Triangulaire pour le tronçon de l'autoroute A51

Caractéristiques des transitions : Le flux maximal Φ_1 de la transition t_1 est égale au flux maximal détecté par la boucle B1 à 16h04 qui est égal à 4290 *veh/h*. Le flux maximal Φ_2 de la transition t_2 est égale au flux maximal détecté par la boucle B3 à 16h04 qui est égal à 4450 *veh/h*.

Caractéristiques de la place lot triangulaire : La vitesse maximale de la place lot est égale à 90 km/h . La longueur de cette place est égale à 1.207 km . La densité maximale de cette place est $d_2^{max} = 185 \text{ veh/km}$. Le flux maximal de la place lot p_2 est égal à son flux de sortie $\Phi_2^{max} = 4450 \text{ veh/h}$. D'où la fonction caractéristique $\gamma(p_2) = (90, 185, 1.207, 4450)$.

Diagramme fondamental triangulaire de la place lot p_2 : La relation flux-densité triangulaire pour cette place lot est représentée par le diagramme fondamental triangulaire de la figure 4.6. La densité critique d_2^{cri} de cette place est égale à : $\frac{4450}{90} = 49.44$, la vitesse de propagation de la congestion W_2 est égale à : $\frac{4450 \cdot 90}{185 \cdot 90 - 4450} = 32.82$.

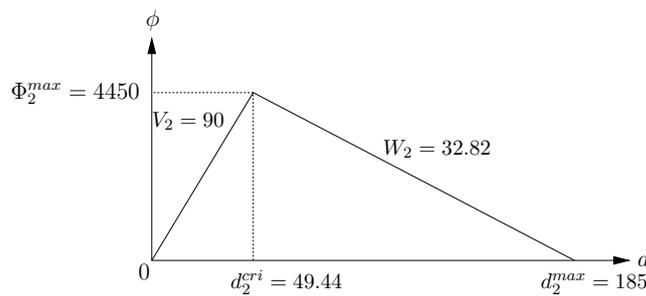


FIGURE 4.6 – Diagramme fondamental triangulaire de la place lot p_2

État initial du modèle RdPLots Triangulaire : La place lot p_2 à l'instant initial ($\tau_0 = 15\text{h}58$) contient un lot de sortie en état libre $OC\beta_2^1(\tau_0)$ en contact avec un deuxième lot contrôlable en état congestionné $C\beta_2^2(\tau_0)$.

A l'état initial, nous proposons pour notre étude l'événement contrôlé qui est défini comme suit :

- diminution du flux maximal de la transition lot t_1 à la date $\tau = 0 \text{ h}$: $\phi_2 = 3800 \text{ veh/h}$.

Caractéristiques des lots contrôlables :

1. pour le lot contrôlable de sortie $OC\beta_2^1(\tau_0) = (l_2^1, d_2^1, x_2^1, v_2^1)$ en état libre : sa position est égale à la longueur de la place lot triangulaire p_2 $x_2^1 = 1.207 \text{ km}$. Sa longueur est égale à $l_2^1 = 0.3 \text{ km}$. Le flux instantané de ce lot de sortie est détecté par la boucle B2 ($\varphi_1 = 3090 \text{ veh/h}$). La vitesse de ce lot est égale à 90 km/h . Sa densité est $d_2^1 = 34.33 \text{ veh/km}$. D'où, les caractéristiques de ce lot contrôlable de sortie sont comme suit : $OC\beta_2^1(\tau_0) = (0.3, 34.33, 1.207, 90)$.
2. pour le lot contrôlable $C\beta_2^2(\tau_0)(l_2^2, d_2^2, x_2^2, v_2^2)$ en état congestionné : sa longueur est égale $l_2^2 = 0.9 \text{ km}$. Ce lot est en contact avec le lot de sortie, d'où sa position qui est calculée comme suit : $x_2^2 = x_2^1 - l_2^1 = 1.207 - 0.3 = 0.907 \text{ km}$. Le flux instantané de ce lot contrôlable est $\varphi_2 = 3800 \text{ veh/h}$. Sa vitesse est égale à 31 km/h . Sa densité est égale à

$d_2^2 = 122.58 \text{ veh/km}$. D'où les caractéristiques de ce lot contrôlable sont comme suit :
 $C\beta_2^2(\tau_0) = (0.9, 122.58, 0.907, 31)$.

La place continue p_1 limite la capacité maximale C_2^{max} de la place lot triangulaire, et son marquage est calculé comme suit : $C_2^{max} = d_2^{max} \cdot S_2 - (l_2^1 \cdot d_2^1 + l_2^2 \cdot d_2^2) = 185 \cdot 1.207 - (0.3 \cdot 34.33 + 0.9 \cdot 122.58) = 102$.

4.3.3 Analyse des résultats de simulation

Nous avons simulé le modèle de la figure 4.5 pour 1 heure et 36 minutes (de 15h58 à 17h34). La date initiale est 15h58. Le flux d'entrée à la date initiale est de 3800 *veh/h*.

Afin de respecter les données réelles fournies, nous proposons pour le changement du flux de la transition t_1 (voir figure 4.4), les événements contrôlés qui sont définis dans le tableau 4.2.

Événement	Nœud	Date	Valeur
augmentation du flux maximal de la transition	t_1	0.1	$\phi_1 = 4290$
diminution du flux maximal de la transition	t_1	0.2	$\phi_1 = 4120$
diminution du flux maximal de la transition	t_1	0.3	$\phi_1 = 4100$
diminution du flux maximal de la transition	t_1	0.4	$\phi_1 = 4080$
diminution du flux maximal de la transition	t_1	0.5	$\phi_1 = 3880$
diminution du flux maximal de la transition	t_1	0.6	$\phi_1 = 3420$
diminution du flux maximal de la transition	t_1	0.7	$\phi_1 = 2900$
diminution du flux maximal de la transition	t_1	0.8	$\phi_1 = 2540$
diminution du flux maximal de la transition	t_1	0.9	$\phi_1 = 2340$
diminution du flux maximal de la transition	t_1	1	$\phi_1 = 2300$
diminution du flux maximal de la transition	t_1	1.1	$\phi_1 = 1920$
diminution du flux maximal de la transition	t_1	1.2	$\phi_1 = 1740$
diminution du flux maximal de la transition	t_1	1.3	$\phi_1 = 1480$
diminution du flux maximal de la transition	t_1	1.4	$\phi_1 = 1300$
diminution du flux maximal de la transition	t_1	1.5	$\phi_1 = 1190$
diminution du flux maximal de la transition	t_1	1.6	$\phi_1 = 1120$

TABLE 4.2 – Liste des événements contrôlés

Comparaison du flux de sortie : Ici nous allons comparer la variation du flux de sortie de p_2 (c'est-à-dire le flux de son lot contrôlable de sortie) et le flux de sortie du tronçon détecté par la boucle B3 entre 15h58 à 17h34.

D'après les écarts calculés entre les valeurs réelles du flux de sortie du tronçon (détecté par B3) et celles fournies par simulation (flux du lot de sortie de la place p_2), nous avons remarqué que l'ensemble des écarts pour cette durée de simulation varie entre 0% et 3.2%.

En conclusion, la faiblesse des écarts trouvés entre les valeurs de simulation et les valeurs réelles (voir figure 4.7) fournit une idée sur le bon fonctionnement du modèle en ce qui concerne sa faculté à représenter un tronçon d'autoroute par un modèle RdPLots Triangulaire. D'autre

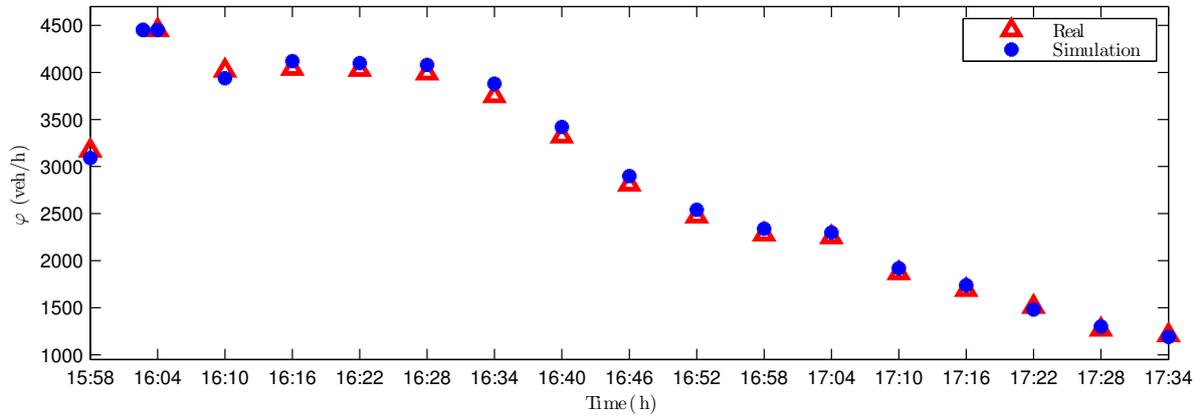


FIGURE 4.7 – Variation du flux de sortie par simulation (lot de sortie) et en cas réel (détecté par B3)

part, le modèle est capable de représenter l'évolution d'un lot congestionné en régime de décongestion, c'est-à-dire la variation de la longueur de congestion et la détermination de la date de décongestion d'un lot.

Les aspects théoriques développés sur les RdPLots et plus particulièrement sur les RdPLots Triangulaires ont été implémentés dans un logiciel appelé Simuleau et qui permet la modélisation, la simulation et l'analyse de systèmes décrits par des Réseaux de Petri. Je présente les principales caractéristiques de ce logiciel dans le chapitre suivant.

Chapitre 5

Simuleau : un simulateur pour les Réseaux de Petri Hybrides et Lots

Le développement logiciel a toujours été présent dans mes recherches. Je peux citer :

- MQNA (Markovian Queueing Networks Analyser) [27] développé au cours de mon master et dédié aux Réseaux de files d'attente ;
- PEPS (Performance Evaluation of Parallel Systems) [18, 26] dédié aux Réseaux d'Automates Stochastiques et développé pendant mon doctorat.

Le dernier en date s'intitule Simuleau. Un outil écrit en C++, qui permet de modéliser, de simuler et d'analyser des systèmes décrits à l'aide de Réseaux de Petri. Il est dédié aux RdP discrets temporisés, aux RdP continus avec une sémantique mono serveur, aux RdP hybrides et aux RdPLots. En tant qu'outil académique et de recherche, Simuleau a été développé pour être open-source, simple à utiliser (l'interface, les fichiers d'entrée et de sortie sont faciles à écrire et à lire), facile à étendre (l'interface textuelle permet d'ajouter facilement de nouveaux outils graphiques).

Le logiciel est mis à la disposition de la communauté et peut être téléchargé à partir de l'adresse <https://gitlab.lis-lab.fr/leonardo.brenner/simuleau>. Ce développement logiciel a donné lieu à une publication [22].

5.1 Historique

La première version de Simuleau a été proposée en 1993 pour implémenter les concepts de la classe RdPLots "originale" [37]. Le logiciel a été utilisé à l'époque pour modéliser le système d'embouteillage d'eau chez Perrier où les convoyeurs étaient modélisés à l'aide de places de lots.

En 1998, les concepts de capteurs et de contrôle développés en [14] ont été intégrés à l'outil.

Initialement, Simuleau a été développé sous la version Borland de C++ pour le système d'exploitation Windows. Cependant cet outil de développement en C++ a été abandonné au début des années 2000 et n'est plus compatible avec les systèmes d'exploitation récents. Quelques interfaces avec d'autres logiciels comme Syrphico et Hyphen ont été proposés au cours des années 2000. Malheureusement ces interfaces ont été perdues.

En 2016, la version actuelle de Simuleau a été proposée et est toujours en cours de développement de façon à intégrer des nouvelles théories des Réseaux de Petri. L'objectif de cette nouvelle version est de réécrire l'ancien code en utilisant des techniques de programmation modernes, d'intégrer les différentes extensions de RdPLots qui ont été développées ces dernières années, et de restructurer l'outil pour faciliter le développement de futures extensions.

5.2 Architecture logicielle

L'architecture de Simuleau est organisée en quatre modules principaux : le module simuleau, le module d'interface, le module BPN et le module d'échéancier. La figure 5.1 montre le diagramme de classes avec les principaux composants de Simuleau.

5.2.1 Module simuleau

Le module simuleau gère les trois autres modules et met en oeuvre la boucle de simulation principale. Cette boucle représente l'algorithme d'évolution, illustré dans la figure 3.9 qui dirige la dynamique des modèles RdPLots.

5.2.2 Module interface

Pour rendre l'outil plus convivial, le module d'interface de Simuleau met en oeuvre des menus textuels.

L'interface du menu textuel présente différentes options à l'utilisateur, comme nous pouvons constater dans la figure 5.2. L'option 1 permet, par exemple, à l'utilisateur de compiler le fichier d'entrée et de vérifier le modèle. Un exemple de la syntaxe de description d'un modèle est présenté dans la section 5.3.

L'option 2 du menu principal conduit l'utilisateur au menu de simulation (voir Figure 5.3). L'utilisateur peut maintenant sélectionner une option parmi différents algorithmes de simulation. L'option 2 simule l'évolution du système sans événements contrôlés (autonome). L'option 1 simule l'évolution du système à l'aide d'événements contrôlés externes, qui peuvent changer

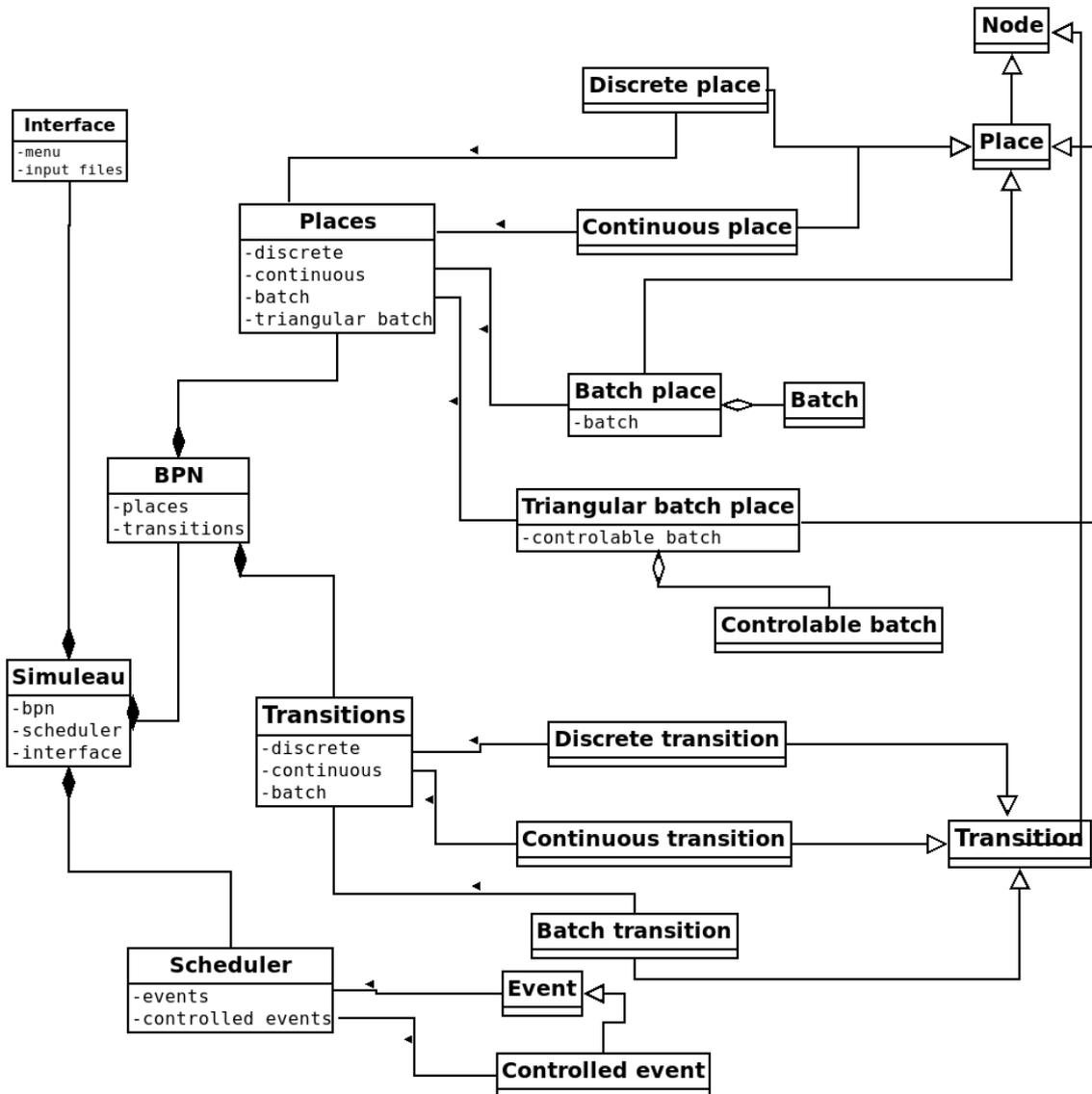


FIGURE 5.1 – Simuleau tool class diagram

le flux maximal des transitions, ainsi que la vitesse maximale des places lots. D'autres méthodes contrôlées (options 3 et 4) conduisent l'évolution du système vers un état stable.

Les autres options du menu principal sont dédiées aux paramètres ou à l'information sur Simuleau et ne sont pas décrites dans ce document.

5.2.3 Module BPN

Le module BPN implémente toutes les structures et méthodes nécessaires au stockage et à la simulation d'un modèle RdPLots. Chaque classe de ce module décrit un type de place et

```

+-----+
|           This is Simuleau - the BPN tool           |
+-----+

1) Compile a BPN model                                4) Inspect data structures
2) Simulate a compiled BPN model                      5) Dynamics facilities
3) Preferences                                        6) About this version

0) Exit Simuleau (Option 0 always exits the current menu)

```

FIGURE 5.2 – Menu principal de Simuleau

```

***** Simulating a BPN model *****

1) With controlled events                            3) SF-On/Off control method
2) Without controlled events                        4) MF-On/Off control method

```

FIGURE 5.3 – Menu de simulation de Simuleau

de transition pris en compte dans Simuleau. D'autres types de noeuds (places et transitions) peuvent être facilement implémentés pour étendre l'outil. Chaque classe de places et de transitions est une version spécialisée de la classe générale et héritent donc de ses comportements et méthodes. Les méthodes implémentées dans chaque classe permettent de calculer l'évolution du système entre les événements datés.

Le calcul de l'IFF fait partie du module BPN et constitue l'une des principales caractéristiques de Simuleau. Dans la version actuelle de l'algorithme, le calcul de l'IFF est basé sur des méthodes de programmation linéaire, en particulier la bibliothèque GLPK [1] est utilisée pour décrire les contraintes et calculer l'IFF. Selon les lois de contrôle choisies pour la simulation, le système linéaire et sa fonction objective diffèrent. Par exemple, il peut s'agir d'une maximisation des flux de franchissement, ou d'une fonction plus complexe pour mener le système dans un état stable.

5.2.4 Module échéancier

Le module de l'échéancier calcule, à partir d'un état invariant, les dates futures de tous les événements possibles qui peuvent se produire dans le modèle. Un ensemble de treize événements est pris en compte (certains sont mentionnés dans la section 3.3.1, d'autres ne sont pris en compte que dans des extensions spécifiques de RdPLots).

5.3 Description d'un modèle

Une autre caractéristique importante de Simuleau est la description du modèle d'entrée. Les modèles sont décrits dans un format d'entrée spécifique à Simuleau qui favorise la lisibilité humaine. Le fichier d'entrée est structuré en trois blocs.

Le premier bloc donne un nom au modèle et définit les unités de longueur et unités de temps.

```
// double "/" for comments
model example; // the given to this model is "example"
length unity=km;
time unity=h;
```

Listing 5.1 – Définition du nom et des unités de longueur et de temps

Le deuxième bloc est dédié à la description du modèle. Les mots clés "network description" définissent le début de ce bloc.

La première partie de ce bloc, qui débute avec le mot clé "places" est dédiée à la description des places. Quatre types de places sont pris en compte dans la version actuelle de Simuleau. Elles sont : discrete, continuous, batch, and triangular. Chacune possède certains paramètres spécifiques et d'autres optionnels. C'est le cas, par exemple, du paramètre steady marking qui sont utilisés uniquement dans la simulation avec des flux contrôlés. Le poids des arcs est aussi un des paramètres optionnels, si omis sa valeur par défaut est égale à 1.

La deuxième partie de ce bloc commence par le mot clé "transitions". Trois types de transitions sont considérés : discrete, continuous, batch. À nouveau, chaque transition a un certain nombre de paramètres spécifiques et d'autres optionnels.

Pour chaque place et/ou transition, uniquement les arcs de sortie du noeud sont mentionnés.

L'exemple ci-après illustre les descriptions pour chaque type de place et transitions.

```
network description
places
  place DPlace (discrete)
    initial marking {2} // integer value
    steady marking {1} // optional parameter
    output arc DTransition (2) // 2 is the weight of the arc

  place CPlace (continuous)
    initial marking {2.3} // real value
    steady marking {1.1} // optional parameter
    output arc CTransition // if no specified weight is 1

  place BPlace (batch)
    function (110, 200, 10) // speed, max density, and length
    initial marking {(1.2, 150, 9.0)}
    // list of initial batches
    // {(length, density, position), ..}
    steady marking {} // optional parameter
```

```

    output arc BTransition

place TBPlace (triangular)
    function (110, 200, 10, 3600)
    // speed, max density, length, and max flow
    initial marking {(1.2, 150, 9.0, 35)}
    // list of initial batches
    // {(length, density, position, speed), ..}
    steady marking {} // optional parameter
    output arc BTransition

transitions
    transition DTransition (discrete)
        timing (4) // real value
        output arc CPlace (3)

    transition CTransition (continuous)
        flow (2.0) // real value
        steady flow (2) // option parameter
        output arc BPlace
        output arc TBPlace

    transition BTransition (batch)
        flow (200.0) // real value
        steady flow (150) // option parameter

```

Listing 5.2 – Description du modèle

Il est possible d'ajouter deux blocs optionnels au fichier d'entrée. Le premier concerne les événements contrôlés. Ce bloc commence par les mots clés "controlled events" et permet d'insérer une série d'événements pour changer la vitesse maximale d'une place batch ou triangular ou changer le flux maximal d'une transition batch.

```

// the section of controlled events are option
controlled events
    max_flow_change=(flow, BTransition, 3000, 1.15);
    // event type, concerned transition, new flow, and time
    max_speed_change=(speed, TBPlace, 100, 0.4);
    // event type, concerned place, new speed, and time

```

Listing 5.3 – Événements contrôlés

Le deuxième bloc optionnel concerne des *breakpoints*. Cela permet d'insérer des points d'inspection sur une ou plusieurs dates, même s'il n'y a pas l'occurrence d'un événement à la date demandée.

```

time inspection // print the places/transitions informations at a required date
    res1 = @ 1.001 // one time at date 1.001
    res2 = # 1.02 // multiple times every 1.02 time units

```

Listing 5.4 – Breakpoints

Les événements contrôlés ainsi que les points d'inspection sont ajoutés à l'échéancier au moment de la simulation.

5.4 Sortie de la simulation

Pendant la simulation, chaque date où il y a l'occurrence d'un événement ou d'un point d'inspection, Simuleau écrit dans un fichier l'état du modèle.

Pour chaque date d'événement dans l'évolution du modèle, Simuleau écrit : la date courante, l'état de chaque place et transition, la liste de tous les événements à venir avec leurs dates respectives et la liste des événements qui seront traités dans le prochain pas de simulation.

Un exemple d'un fichier de sortie pour la date 0 est présenté ci-après.

```

=====
Current date : 0
=====
=====
At the beginning of the step
=====
Name:                DPlace
  Type:              Discrete
  Marks:             2
  Reserved Marks:   0
Name:                CPlace
  Type:              Continuous
  Marks:             5
  Reserved Marks:   0
Name:                BPlace
  Type:              Batch
  Density:           200
  Speed:             110
  Length:            10
  Batches:           1
                    (1.2, 150, 2)
Name:                CTransition
  Type:              Continuous
  Current Flow:     2
Name:                BTransition
  Type:              Batch
  Current Flow:     0

Events
Discrete transition is fired: 1
Date 1   node DTransition
Continuous place becomes empty: 1
Date 2.5   node CPlace
Batch becomes an output batch: 1
Date 0.07272727273   node BPlace

```

```
Controlled Events
Transition maximum flow changes: 1
Date 1.15  node BTransition  name max_flow_change

Next events to proceed
Event date 0.07272727273  in node BPlace  type Batch becomes an output batch
```

À chaque simulation, l'utilisateur doit définir le temps maximum à simuler. La simulation s'arrête lorsque ce temps est atteint. Cependant, pour éviter des calculs inutiles, Simuleau s'arrête s'il ne reste plus aucun événement dans l'échéancier ou, dans le cas des simulations avec des méthodes de contrôle, lorsque l'état stationnaire est atteint.

Pour finaliser ma présentation sur mes travaux sur les formalismes déterministes, je dresse dans le chapitre suivant un bilan de mes contributions.

Chapitre 6

Bilan des travaux sur les formalismes déterministes

Mes travaux sur les formalismes déterministes ont commencé après mon recrutement à Aix-Marseille Université dans le cadre de thèse de Radhia Gaddouri (2012-2015) que j'ai co-encadrée avec Isabel Demongodin sur les RdPLots Triangulaires.

Mes travaux sur les RdPLots Triangulaires ont été essentiellement développés entre 2012 et 2017 et que j'ai synthétisés dans les chapitres [3](#), [4](#) et [5](#). Ils ont donné lieu à plusieurs publications pendant et après la soutenance de la thèse. Cela constitue mes plus anciens travaux sur les Réseaux de Petri et ça a représenté un défi pour moi de m'approprier un nouveau formalisme, les RdPLots, d'intégrer les concepts d'un autre domaine de recherche, celui du trafic routier afin de proposer un modèle qui soit fidèle aux comportements réels du trafic. L'obtention des données réelles du trafic a été aussi un défi à relever. C'est un sujet de recherche qui me passionne particulièrement car pour moi ça présente un bon équilibre entre un développement théorique et la mise en pratique quasiment directe sur un cas réel. Des perspectives pour la suite de ces travaux sont discutées dans le chapitre [10](#).

À la date de ce document, Radhia Gaddouri a un poste de *Responsable des Formations par Alternance* à ESIEA (École d'Ingénieur.e.s d'un Numérique Utile) à Paris. Au total ces travaux ont donné lieu à six publications dans des conférences nationales et internationales.

En 2018, j'ai eu l'opportunité de co-encadrer le stage de master 2 de Ruotian Liu. À la fin de son stage, l'étudiant a continué en thèse que j'ai co-encadrée avec Isabel Demongodin et Rabah Ammour. Toujours dans le domaine de Réseaux de Petri, nous avons exploré le contrôle de systèmes hybrides. Malgré l'existence de méthodes de contrôle pour des systèmes hybrides, l'approche événementielle développée dans sa thèse a été innovante et a donné lieu à deux publications majeures pendant sa thèse. Ruotian Liu a soutenu sa thèse en 2021 en 39 mois malgré un contexte COVID très difficile. Dans la continuité de sa thèse, une autre publication a eu lieu en 2023 et une quatrième vient d'être soumise dans une revue internationale, où nous appliquons les méthodes de contrôle à un cas réel. Il s'agit d'une chaîne d'embouteillage de

Perrier. Nous montrons que malgré un nombre plus élevé de changement de flux transitions, nous arrivons à mener le système à un état stationnaire plus rapidement.

Après sa soutenance, l'étudiant a fait trois ans de post-doctorat à l'Université de Bari et a été recruté par la suite en tant qu'*assistant professor* à la même université.

À partir de 2020, une nouvelle thématique commence à être explorée. Il s'agit de la sécurité des systèmes cyber-physiques. Cette thématique plus récente a donné lieu à plusieurs publications, un stage de master avec une poursuite en thèse, et plus important, plusieurs projets collaboratifs tels que CP Sécurité I et II, ANR-APRENTIS et le dernier en date, l'ANR-MENACES.

Les premiers travaux nous ont permis de poser une base solide pour la modélisation, la simulation et l'analyse des systèmes cyber-physiques. Ces travaux sont toujours en cours de développement dans le cadre du projet ANR-MENACES, qui intègre dans son projet, pas uniquement un développement théorique mais aussi le développement applicatif d'un jumeau numérique pour une plateforme réelle au LURPA (Laboratoire Universitaire de Recherche en Production Automatisée). Au total, ces travaux ont été publiés dans quatre conférences internationales ainsi que dans une revue internationale.

Si dans cette première partie de ce document, j'ai présenté mes contributions à la modélisation, simulation et analyse des SED par des formalismes déterministes, dans la deuxième partie, j'expose mes travaux sur les formalismes stochastiques, toujours dans le cadre des SED.

Publications majeures

1. Leonardo BRENNER, Rabah AMMOUR, Isabel DEMONGODIN. *Simuleau : a tool for hybrid and batches Petri nets*. **17th IFAC Workshop on Discrete Event Systems (WODES)**, pp. 1-6, 2024.
2. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Robust stealthy attacks based on uncertain costs and labeled finite automata with inputs*. **IEEE Robotics and Automation Letters (RA-L)**, vol.8, issue 5, pp. 2732-2739, 2023. [hal-04202802](https://hal.archives-ouvertes.fr/hal-04202802)
3. Khalid HAMADA, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *Attack Synchronizing Sequence Computation for Output Synchronized Petri Nets with Multiple Deadlocks*. **20th IEEE International Conference on Networking, Sensing and Control (ICNSC)**, 2023. [hal-04437199](https://hal.archives-ouvertes.fr/hal-04437199)
4. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *ON/OFF control for reaching a steady state attractive region in batches Petri nets*. **IFAC World Congress 2023**, vol. 56, n. 2, pp. 9618-9623, 2023. [hal-04437189](https://hal.archives-ouvertes.fr/hal-04437189)
5. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Observer Design for Labeled Finite Automata with Inputs under Stealthy Actuators Attacks*. **16th IFAC Workshop on Discrete Event Systems (WODES)**, pp. 1-6, 2022. [hal-03879792](https://hal.archives-ouvertes.fr/hal-03879792)

-
6. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Costs analysis of stealthy attacks with bounded output synchronized Petri nets*. **IEEE 17th International Conference on Automation Science and Engineering (CASE)**, pp. 799-804, 2021. [hal-03517596](#)
 7. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Observer Design for Bounded Output Synchronized Petri Nets*. **European Control Conference (ECC)**, pp. 746-751, 2021. [hal-03517554](#)
 8. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *Event-Driven Control for Reaching a Steady State in Controlled Generalized Batches Petri Nets*. **15th IFAC Workshop on Discrete Event Systems (WODES)**, IFAC-PapersOnLine, vol. 53, issue 4, pp. 180-186, 2020. [hal-03171923](#)
 9. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *ON/OFF Control Trajectory Computation for Steady State Reaching in Batches Petri Nets*. **14th International Conference on Verification and Evaluation of Computer and Communication Systems (VECOS)**, Lecture Notes in Computer Science, vol 12519, pp. 84-99, 2020. [hal-03171901](#)
 10. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Controlled Triangular Batches Petri Nets for hybrid mesoscopic modeling of traffic road networks under VSL control*. **IEEE 12th International Conference on Automation Science and Engineering (CASE)**, pp. 427-432, 2016. [hal-01997510](#)
 11. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Extension of Batches Petri Nets by Bi-parts Batch Places*. **Int. Workshop on Petri Nets for Adaptive Discrete-Event Control Systems (ADECS)**, pp.83-102, 2014.
 12. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Mesoscopic event model of highway traffic by Batches Petri nets*. **6th International Conference on Management and Control of Production and Logistics (MCPL)**, pp.317-324, 2013.

Deuxième partie

Modélisation de systèmes stochastiques

Chapitre 7

Synthèse des travaux sur les modèles stochastiques

Mes travaux sur les formalismes stochastiques reposent essentiellement sur les formalismes de Réseaux d'Automates Stochastiques (RAS) et les Chaînes de Markov. Les modèles stochastiques sont un héritage de mes travaux de thèse [21] mais qui reviennent dans certains des travaux récents [6, 7, 28, 102, 103, 104].

Les travaux les plus récents sont sur les modèles de propagation de l'influence dans les réseaux sociaux. Cette thématique est détaillée dans le chapitre 8.

Même si les travaux purement théoriques sur les formalismes stochastiques sont moins présents dans mes recherches récentes, le côté applicatif est toujours d'actualité. Les applications plus récentes dans ces thématiques explorent la modélisation d'un algorithme d'ordonnancement par vol de tâches, bien connu dans le domaine du calcul parallèle et distribué, et de prédiction de l'état du trafic routier par patron de déplacement. Ces deux sujets sont présentés, respectivement, dans les sections 7.2 et 7.3.

Commençons par une brève introduction sur ce formalisme RAS avant de passer à ses applications.

7.1 Réseaux d'automates stochastiques

Le formalisme des Réseaux d'Automates Stochastiques (RAS) introduit par Plateau [85] constitue un outil de modélisation efficace pour des systèmes complexes à grand espace d'état et également la résolution de ces modèles. Le principe de base de ce formalisme est de décrire un système comme un ensemble de sous-systèmes qui interagissent. Chacun de ces sous-systèmes est vu comme un automate stochastique. L'interaction entre les sous-systèmes est modélisée par des règles établies entre les états internes de chaque automate [45]. Le modèle RAS peut

être représenté par un seul automate global qui comporte tous les états possibles du système modélisé.

7.1.1 Automates stochastiques

Un automate stochastique est un modèle mathématique d'un système qui possède des entrées et sorties discrètes. On peut décrire un automate stochastique comme un *ensemble d'états* et un *ensemble de transitions* entre eux [86].

Graphiquement, les automates sont décrits par des graphes orientés où les noeuds représentent les états internes et les arcs représentent les transitions entre les états.

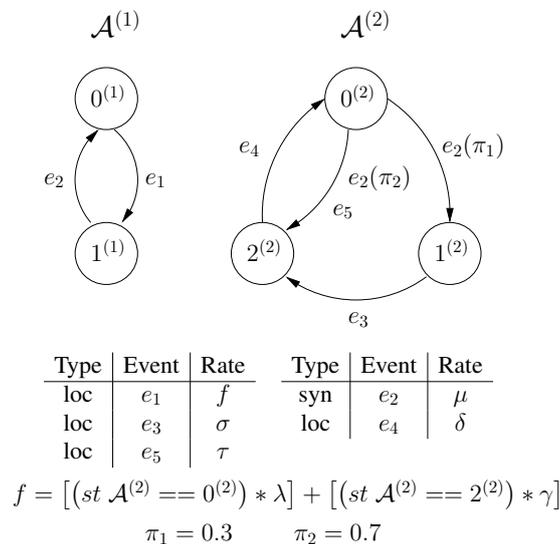


FIGURE 7.1 – Modèle RAS avec 2 automates

La figure 7.1 présente un modèle RAS avec 2 automates.

7.1.2 Événements

L'état d'un modèle RAS, appelé *état global*, est la combinaison de tous les *états locaux* de tous les automates.

Le changement de l'état global du système est le résultat de l'occurrence d'un événement. Nous avons deux types d'événements : *local* et *synchronisant*.

- **événement local** : il apparaît dans un seul automate et représente un comportement indépendant de ce sous-système ;

- **événement synchronisant** : un événement synchronisant apparaît obligatoirement dans plusieurs automates et implique un changement simultané de tous les automates concernés.

Exemple 7.1. Dans le modèle de la figure 7.1, les événements e_1, e_3, e_4 et e_5 sont des événements locaux. L'événement e_2 est un événement synchronisant et va provoquer le changement des états internes des deux automates. L'occurrence de l'événement e_2 pourra avoir lieu uniquement lorsque l'automate $\mathcal{A}^{(1)}$ est dans l'état $1^{(1)}$ et l'automate $\mathcal{A}^{(2)}$ dans l'état $0^{(2)}$.

Si un événement est associé à deux transitions, ou plus, partant d'un même état, une probabilité de routage doit être associée au couple événement-transition. L'événement e_2 dans l'automate $\mathcal{A}^{(2)}$ de la figure 7.1 présente ce cas. Alors, nous avons associé une probabilité $\pi_1 = 0.3$ pour la transition de l'état $0^{(2)}$ vers $1^{(2)}$ et $\pi_2 = 0.7$ pour la transition de l'état $0^{(2)}$ vers $2^{(2)}$. Ça représente un choix aléatoire déclenché par le même événement. ■

À chaque événement, nous associons un nom et un taux d'occurrence. Le taux d'occurrence définit la fréquence avec laquelle l'événement aura lieu. Le taux d'occurrence peut avoir une valeur constante ou fonctionnelle.

7.1.3 Taux et probabilités fonctionnels

Les taux et les probabilités d'occurrence associés à un événement dans un automate peuvent être une fonction de l'état interne d'autres automates. Dans ce cas, nous avons des taux et/ou probabilités dites *fonctionnels*. Une fonction permet d'associer à un même événement différentes valeurs déterminant son taux d'occurrence en fonction de l'état local des autres automates du modèle. D'une manière similaire, les probabilités de routage d'un événement peuvent être exprimées par des fonctions. Le modèle dans la figure 7.1 possède un taux fonctionnel associé à l'événement e_1 .

La fonction¹ f associée à l'événement e_1 dans l'automate $\mathcal{A}^{(1)}$ est évaluée à λ si l'automate $\mathcal{A}^{(2)}$ est dans l'état $0^{(2)}$, à γ si il est dans l'état $2^{(2)}$ et zéro si il est dans l'état $1^{(2)}$.

7.1.4 Équivalence avec les Chaînes de Markov

Bien qu'un modèle RAS soit représenté par un ensemble d'automates, il peut aussi être représenté par un seul automate qui contient tous les états globaux possibles du modèle. Cet automate est une représentation de la chaîne de Markov du système modélisé [46]. Dans cet automate global, il n'y a plus d'événements synchronisants, mais seulement des événements locaux. Les arcs de cet automate global sont étiquetés par les taux d'occurrence des événements et les probabilités de routage. Les durées de résidence dans chaque état sont des variables aléatoires de distribution exponentielle. Donc, à un instant de temps, le changement vers un état suivant ne dépend que de l'état courant et pas du temps écoulé dans cet état.

1. La notation utilisée dans les fonctions est celle du logiciel PEPS [26] où st indique l'état d'un automate.

La représentation graphique de la chaîne de Markov équivalente est un graphe à états-transitions qui représente l'automate global du système. Les états de la chaîne de Markov sont formés à partir du produit cartésien des états locaux de chaque automate du modèle SAN. La figure 7.2 représente la chaîne de Markov équivalente au modèle présenté dans la figure 7.1.

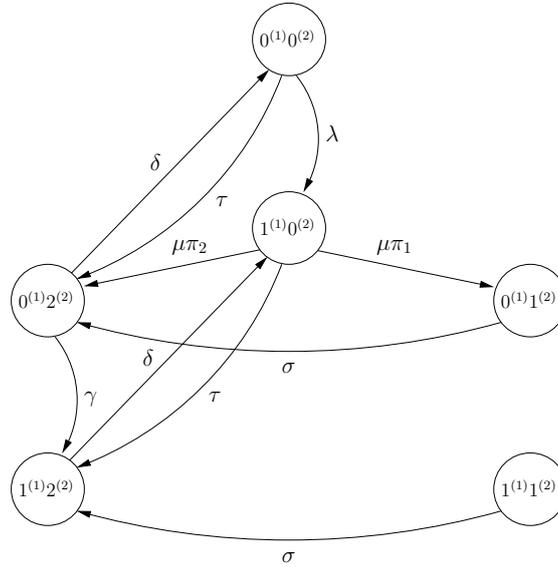


FIGURE 7.2 – Chaîne de Markov équivalente au modèle RAS de la figure 7.1.

7.2 Modélisation d'un algorithme de vol de tâches

Le vol de travail est un paradigme d'ordonnancement de tâches pour les calculs parallèles. Ce paradigme a été proposé initialement par Blumofe dans [19].

Il s'agit d'un algorithme d'ordonnancement décentralisé [3], [94] [95], dont l'objectif est d'ordonner efficacement une application sur un nombre des machines non défini. Cette méthode de parallélisation est utilisée dans plusieurs bibliothèques de programmation parallèle, tel que, Cilk [47], Kaapi [54], Satin [97].

Le principe de fonctionnement de l'ordonnancement par vol de travail est le suivant [20] :

- chaque processeur possède une liste qui contient l'ensemble de tâches qu'il doit effectuer. Si un processeur devient inactif (plus de tâches dans sa liste), il cherche à récupérer du travail dans la liste de tâches d'un autre processeur et continue cela tant qu'il n'arrive pas à en trouver.
- lors de l'exécution d'une tâche, le processeur exécute les instructions associées et cela peut générer d'autres tâches. Ces tâches seront placées dans sa liste pour la suite de l'exécution. L'ensemble des tâches à exécuter et leurs données décrivent entièrement l'application qui peut être représentée sous forme d'un graphe.

Notre objectif est d'utiliser le formalisme des Réseaux d'Automates Stochastiques pour tester les performances de cet algorithme.

Nous avons proposé un modèle RAS générique pour la modélisation de cet algorithme de vol de travail.

Ce modèle est constitué de trois types d'automates :

- les processeurs de l'architecture du support d'exécution ;
- les listes de tâches de chaque processeur ;
- la liste de tâches restantes à exécuter.

Les états de ces automates décrivent l'architecture (processeur et mémoire) de la machine alors que les transitions et les événements associés décrivent l'algorithme de vol de travail.

Exemple 7.2. La figure 7.3 présente un modèle pour une architecture constituée de trois processeurs et une centaine de tâches à exécuter. Dans ce modèle, on observe quatre types d'événements associés aux transitions :

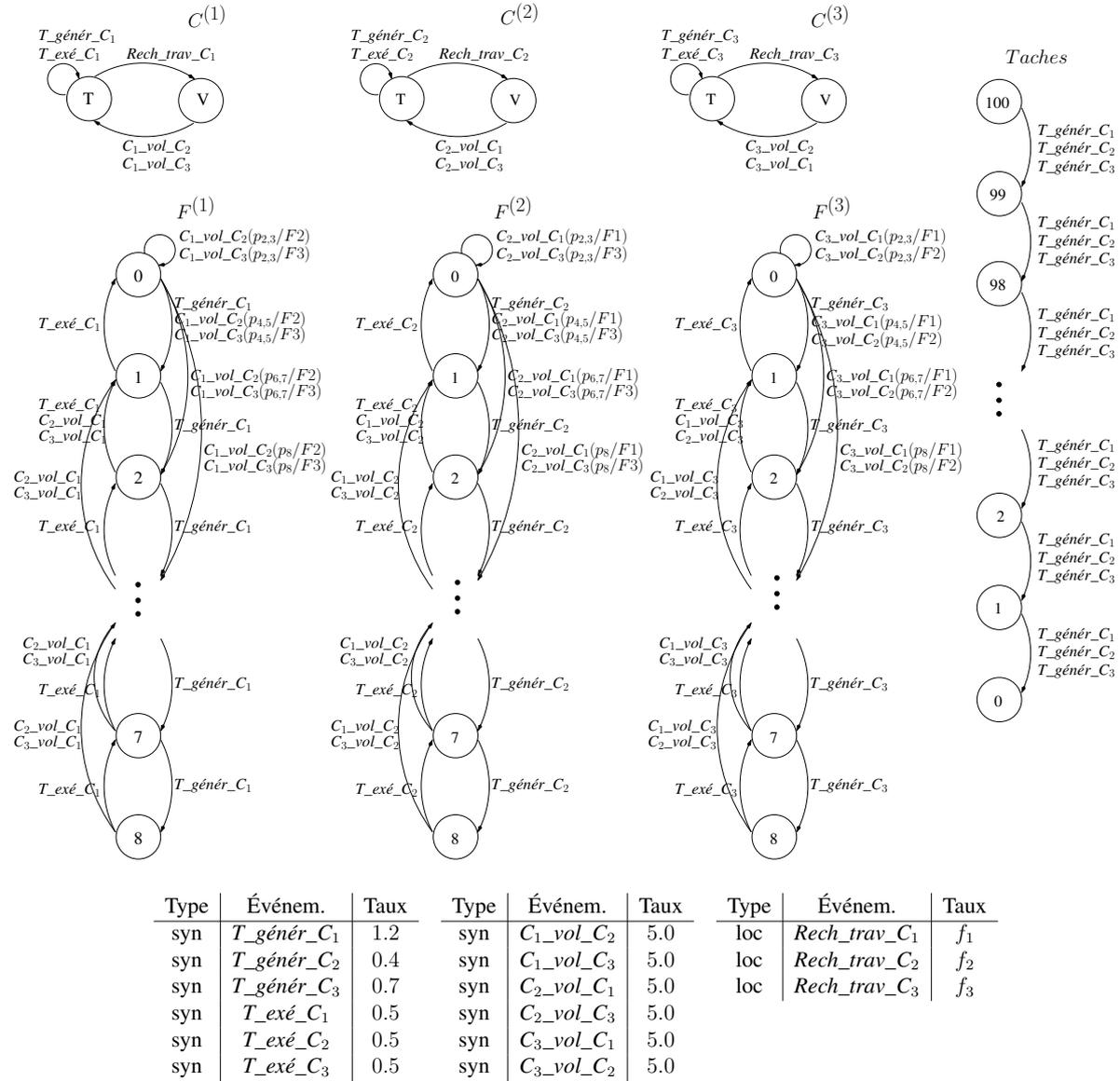
- $T_génér_C_i$: représentent la génération des tâches supplémentaires au niveau d'un processeur i ;
- $T_exé_C_i$: représentent l'exécution d'une tâche par un processeur i ;
- $C_i_vol_C_j$: modélisent le vol des tâches d'un processeur j par un processeur i ;
- $Rech_trav_C_i$: représentent le passage du processeur d'un état d'exécution de tâches à un état de voleur, où il va chercher des tâches à voler.

■

Nous nous sommes intéressés à l'évaluation transitoire du modèle proposé. Plus particulièrement à des indices tels que : le taux d'utilisation des processeurs, le nombre moyen de tâches dans la liste de chaque processeur, la probabilité que toutes les tâches soient terminées dans un temps t ainsi que le nombre moyen de tâches restantes.

Dans l'algorithme de vol de tâches, la quantité de tâches volées joue un rôle important au niveau de la performance. Nous avons donc modélisé trois quantités de vol : uniquement 1 tâche, 50% des tâches et 100% des tâches. La figure 7.4 montre la probabilité que le calcul soit terminé pour chaque temps t (avec une variation de 10 unités de temps) observé dans la résolution du modèle. Nous avons également choisi d'avoir des taux de générations de tâches différents pour chaque processeur pour forcer une disparité entre la liste des tâches de chaque processeur et s'approcher des cas réels.

Nous pouvons constater que plus rapidement la probabilité monte, plus des tâches ont été générées et exécutées. La croissance de la probabilité est plus accentuée pour le vol d'une et de 50% tâches. Cependant, nous pouvons observer un rattrapage dans le cas du vol de 100%



$$f_1 = (st F^{(1)} == 0) \&\& ((st C^{(2)} == T) \parallel (st C^{(3)} == T) \parallel (st Taches == 0))$$

$$f_2 = (st F^{(2)} == 0) \&\& ((st C^{(1)} == T) \parallel (st C^{(3)} == T) \parallel (st Taches == 0))$$

$$f_3 = (st F^{(3)} == 0) \&\& ((st C^{(1)} == T) \parallel (st C^{(2)} == T) \parallel (st Taches == 0))$$

$$p_{2,3}/F1 = (st F^{(1)} == 2) \&\& (st F^{(1)} == 3)$$

$$p_{4,5}/F1 = (st F^{(1)} == 4) \&\& (st F^{(1)} == 5)$$

$$p_{6,7}/F1 = (st F^{(1)} == 6) \&\& (st F^{(1)} == 7)$$

$$p_8/F1 = (st F^{(1)} == 8)$$

$$p_{2,3}/F2 = (st F^{(2)} == 2) \&\& (st F^{(2)} == 3)$$

$$p_{4,5}/F2 = (st F^{(2)} == 4) \&\& (st F^{(2)} == 5)$$

$$p_{6,7}/F2 = (st F^{(2)} == 6) \&\& (st F^{(2)} == 7)$$

$$p_8/F2 = (st F^{(2)} == 8)$$

$$p_{2,3}/F3 = (st F^{(3)} == 2) \&\& (st F^{(3)} == 3)$$

$$p_{4,5}/F3 = (st F^{(3)} == 4) \&\& (st F^{(3)} == 5)$$

$$p_{6,7}/F3 = (st F^{(3)} == 6) \&\& (st F^{(3)} == 7)$$

$$p_8/F3 = (st F^{(3)} == 8)$$

FIGURE 7.3 – Exemple de modélisation : architecture à 3 processeurs

des tâches à la fin de l'exécution. Cela peut être le reflet d'un équilibrage tardif des charges. Toutefois, une analyse plus fine est nécessaire pour vérifier cette hypothèse.

Ces travaux ont été commencés dans le cadre d'un stage de master de Sarah Nemmour et

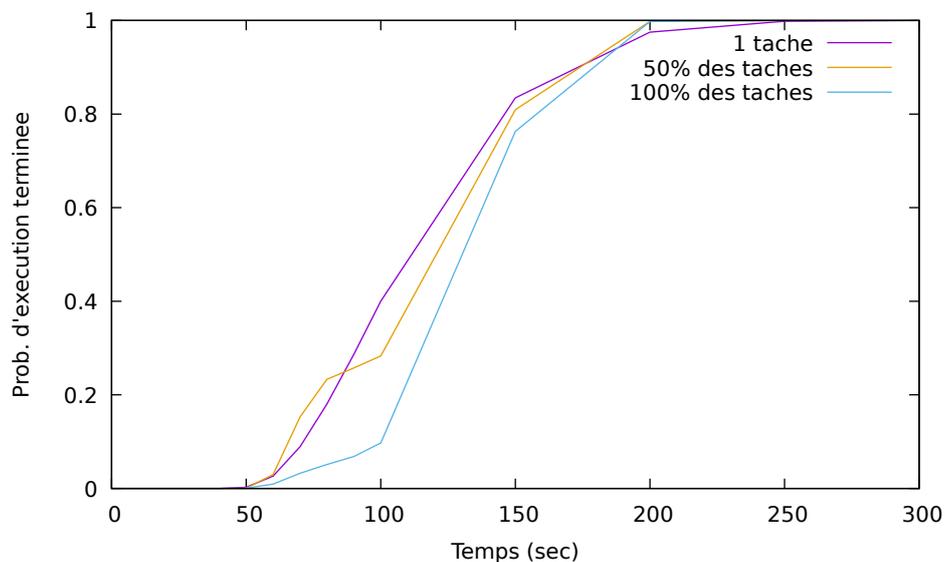


FIGURE 7.4 – Temps d'exécution

ont été publiés dans [28].

7.3 Prédiction de l'état du trafic routier basée sur des motifs et les chaînes de Markov

Plus récemment, nous avons utilisé les Chaînes de Markov pour la modélisation et la prédiction du trafic routier en explorant les patrons de déplacement, appelés aussi des motifs. Ces travaux ont été réalisés en collaboration avec Feda AlMuhisen dans le cadre de sa thèse [5] qui était encadré par Nicolas Durand et Mohamed Quafafou. Deux publications y sont associées [6, 7].

Cette approche de prédiction de l'état du trafic routier consiste à extraire des patrons de déplacement à partir de données réelles et de définir quels sont les patrons émergents dans une fenêtre temporelle. L'objectif était de prédire, pour une ville quelconque découpée en secteurs, l'état du trafic dans un horizon proche (demi-journée, journée ou semaine) pour chaque secteur de la ville.

La méthode de prédiction est divisée en trois étapes : extraction de motifs, construction du modèle markovien et prédiction du trafic.

7.3.1 Extraction de motifs

La première étape est la recherche et la construction de motifs à partir de trajectoires GPS de taxis. Dans notre approche, un motif est une séquence datée de positions GPS qui se répètent dans les trajectoires de plusieurs taxis. Cela permet la définition des patrons de déplacement. Chaque patron peut être classé comme : *Latent*, *Emerging*, *Decreasing*, *Jumping*, *Lost* ou *Nothing*.

Cette étape est détaillée dans [8]. La zone étudiée est divisée en régions de taille égale selon une valeur de granularité spatiale, puis chaque trajectoire est transformée et représentée dans cette zone comme un ensemble de régions. Ensuite, cette représentation discrète des trajectoires est utilisée pour construire un treillis de concepts formels fréquents pour encoder la correspondance maximale entre les motifs fréquents découverts et les trajectoires dans une certaine fenêtre temporelle. Chaque élément de ce treillis, appelé concept formel, est composé d'un motif fermé (d'attributs) et d'un ensemble fermé d'objets. Le motif fermé est l'ensemble maximal d'attributs partagé par l'ensemble d'objets. Ce dernier est l'ensemble maximal d'objets contenant le motif. Dans notre contexte, un concept formel contient un ensemble de régions (le motif) et l'ensemble des trajectoires (les objets) qui traversent ces régions. Un concept formel est dit fréquent lorsque le nombre d'objets dépasse un certain seuil fixé. Nous détectons l'état du trafic du motif fermé pour chaque concept formel fréquent en étudiant son évolution à travers une paire de fenêtres temporelles (t_i, t_{i+1}) . L'état du trafic appartient à un ensemble prédéfini de labels : $\{Latent (LA), Emerging (E), Decreasing (D), Jumping (J), Lost (LO), Nothing (\emptyset)\}$. L'état *Emerging* signifie que la présence du motif a augmenté dans t_{i+1} par rapport à t_i . L'état *Decreasing* signifie que la présence du motif a diminué dans t_{i+1} par rapport à t_i . L'état *Latent* signifie que la présence du motif est assez similaire dans t_{i+1} et t_i . L'état *Jumping* signifie que le motif qui était absent dans t_i est apparu dans t_{i+1} . L'état *Lost* signifie que le motif a disparu dans t_{i+1} . *Nothing* signifie qu'il n'y a pas de trafic détecté dans la fenêtre temporelle.

7.3.2 Modèle markovien

Dans la deuxième étape, pour chaque région, nous construisons une chaîne de Markov [92] où chaque état de la chaîne représente un état prédéfini du trafic. Les transitions représentent un changement de l'état du trafic d'un intervalle de temps à un autre. À chaque transition, nous associons une probabilité qui est le ratio du nombre de changements d'un état i à un état j sur le nombre total de changements à partir de l'état i . Par exemple, dans la figure 7.5, λ_{dLa} indique la probabilité de transition de l'état D à l'état LA .

7.3.3 Prédiction du trafic

Dans cette troisième étape, nous nous intéressons à prédire l'état du trafic dans une fenêtre temporelle. À partir du modèle markovien, deux types d'analyses peuvent être obtenues : stationnaire et transitoire. Une analyse stationnaire peut prédire l'état moyen du trafic. Cela re-

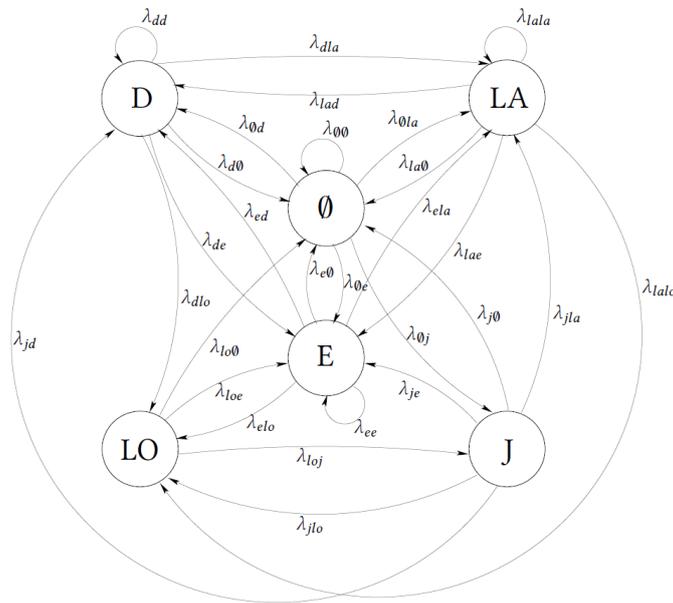


FIGURE 7.5 – Modèle markovien de chaque région

présente un étude à long terme et peut servir à une planification des voies publiques. Dans notre étude, nous sommes particulièrement intéressés par une analyse transitoire où, avec la connaissance de l'état réel du trafic à un instant t , nous pouvons prédire l'état à l'instant $t + n$ pour $n > 0$.

Le principe de la prédiction transitoire est assez simple. À partir d'un vecteur de probabilité initial (où la probabilité de 1.0 est attribuée à l'état du vecteur qui représente l'état courant du trafic à l'instant t), nous appliquons la méthode d'uniformisation [92] pour obtenir le vecteur de probabilité à $t + n$. Nous considérons que l'état avec la probabilité la plus grande correspond à la prédiction du trafic.

7.3.4 Expérimentations

Nous avons appliqué la méthode proposée sur deux ensembles de données réelles. Le premier ensemble de données concerne les déplacements de taxis dans la ville de Pékin (Chine) et le deuxième dans la ville de San Francisco (États-Unis d'Amérique).

Le tableau 7.1 résume les caractéristiques de ces deux ensembles de données.

Nous avons testé notre méthode avec des granularités spatiales de 15, 30, 60 et 120 mètres et une granularité temporelle de 12 heures (AM/PM) pour une prédiction de la période suivante. Nous avons comparé notre méthode à deux autres méthodes que nous appelons *Baseline* et *Itemsets*. La méthode *Baseline* considère uniquement des singletons (pas de motifs) et la méthode *Itemsets* considère un ensemble d'éléments sans prendre en compte l'ordre des éléments.

Caractéristiques	Pékin	San Francisco
Nombre de mesures	15 millions	11 millions
Durée	1 semaine	24 jours
Nombre de taxis	10357	500

TABLE 7.1 – Ensemble de données de la mobilité des taxis

Le tableau 7.2 ci-après montre une meilleure précision globale de notre méthode sur toutes les granularités spatiales testées.

Gran. Spatiale	Baseline	Itemsets	Motifs
15 m	0.747	0.921	0.962
30 m	0.785	0.906	0.942
60 m	0.807	0.902	0.929
120 m	0.831	0.900	0.920

Pékin

Gran. Spatiale	Baseline	Itemsets	Motifs
15 m	0.8984	0.922	0.924
30 m	0.901	0.909	0.910
60 m	0.899	0.897	0.899
120 m	0.903	0.902	0.909

San Francisco

TABLE 7.2 – Précision moyenne globale

Lors d'une comparaison de la précision par état spécifique du trafic (Tableau 7.3), nous pouvons constater une meilleure précision dans la plupart des états, sauf pour l'état *Nothing*. Cependant, cet état en particulier n'apporte pas beaucoup d'informations à l'utilisateur.

État	Baseline	Itemsets	Motifs
Emerging	0.653	0.775	0.849
Decreasing	0.754	0.936	0.945
Latent	0.784	0.969	0.987
Lost	0.880	1	0.999
Jumping	0.754	0.987	0.986
Nothing	0.929	0.778	0.863

Pékin

Status	Baseline	Itemsets	Motifs
Emerging	0.732	0.827	0.836
Decreasing	0.729	0.790	0.797
Latent	0.944	0.990	0.988
Lost	0.997	1	1
Jumping	0.997	1	1
Nothing	0.999	0.837	0.841

San Francisco

TABLE 7.3 – Précision moyenne par état du trafic

Le détail des protocoles de tests ainsi qu'une analyse plus détaillée des résultats sont dans [5, 7].

Dans ce chapitre, j'ai résumé quelques travaux, essentiellement applicatifs, sur les formalismes stochastiques tels que les RAS et les Chaînes de Markov. Dans le chapitre suivant, je détaille mes contributions, plus méthodologiques, sur le modèle de propagation de l'influence par cascade indépendante.

Chapitre 8

Propagation de l'influence par cascade indépendante

Dans ce chapitre je présente les travaux sur la propagation de l'influence dans les réseaux sociaux. Les dernières années, un intérêt croissant est porté sur ce sujet vu le grand nombre de réseaux sociaux tels que Facebook, Instagram, etc. apparu pour mettre en relation des individus ou de groupes de personnes. Ces réseaux sont personnels, professionnels, sur des thèmes et sur des formats très variés et se sont révélés un bon outil pour obtenir des informations mais aussi communiquer et propager des idées. Cela devient plus intéressant lorsqu'on utilise ces réseaux comme outils de marketing.

Nous avons développé sur des modèles de propagation dans le cadre de la thèse de Wenjing Yang (2017 – 2019) et sont basés sur le modèle de propagation par cascade indépendante.

Nous avons abordé ce modèle avec l'objectif de proposer des méthodes de calcul qui manquaient dans la littérature. Je vais par la suite présenter dans la section 8.1 les principes de bases du modèle de cascade indépendante que nous avons utilisé. Je détaille dans la section 8.2 plusieurs approches pour le calcul d'estimation de la propagation de l'influence. La section 8.3 présente le problème de maximisation de l'influence vue d'un point d'activation des liens entre les individus.

8.1 Principes de base du modèle de cascade indépendante

Un réseau social peut être considéré comme un graphe orienté composé d'un ensemble d'acteurs sociaux (tels que des individus ou des organisations) et un ensemble de liens orientés qui représentent les interactions sociales entre les acteurs. Ce concept est utilisé en sciences sociales pour étudier les relations entre les individus, les groupes, les organisations, voire des sociétés entières. Par exemple, un groupe va s'influencer les uns aux autres lorsqu'ils partagent leurs expériences sur l'achat d'un produit quelconque. Cette influence est particulièrement in-

téressante à étudier lorsqu'on parle des réseaux sociaux sur internet. L'apparition plus ou moins récente de sites de réseaux sociaux, tels que Facebook, Instagram et Twitter, relie des milliers de millions de personnes et de groupes et leur offrent une plateforme pour communiquer et s'influencer mutuellement.

8.1.1 Les réseaux sociaux et les modèles de propagation de l'influence

Les études sur la propagation de l'information à travers les réseaux sociaux ont commencé au milieu du 20ème siècle [32, 89]. Un réseau social est un graphe d'interactions et de relations entre des individus et des groupes. De nombreux modèles mathématiques de réseaux sociaux ont été proposés, parmi lesquels le modèle à seuil linéaire et le modèle à cascade indépendante [56] sont ceux qui ont reçu le plus d'attention.

Un réseau social est représenté par un graphe orienté $G = (V, E)$, dans lequel V est l'ensemble de noeuds représentant les individus du réseau et E est l'ensemble des arcs orientés entre les noeuds. Un arc $(i, j) \in E$ indique que le noeud i influence directement le noeud j . Nous utilisons indifféremment les termes individu ou noeud et arc ou lien de manière interchangeable.

Notons

- N_j^{in} l'ensemble des voisins d'entrée du noeud j tel que $N_j^{in} = \{i \in V | (i, j) \in E\}$;
- N_j^{out} l'ensemble des voisins de sortie du noeud j tel que $N_j^{out} = \{i \in V | (j, i) \in E\}$;
- $N = |V|$ le nombre de noeuds dans le réseau G .

Les ensembles N_j^{in} et N_j^{out} sont utilisés pour décrire tous les individus ayant une influence directe sur le noeud j .

Exemple 8.1. La figure 8.1 présente un exemple d'un graphe d'un réseau social G où :

- $V = \{1, 2, 3, 4, 5\}$;
- $E = \{(1, 2)(2, 4)(3, 1)(3, 2)(4, 2)(5, 3)\}$.

et les ensembles de voisins d'entrée et de sorties sont :

- $N_1^{in} = \{3\}$; $N_1^{out} = \{2\}$;
- $N_2^{in} = \{1, 3, 4\}$; $N_2^{out} = \{4\}$;
- $N_3^{in} = \{5\}$; $N_3^{out} = \{1, 2\}$;
- $N_4^{in} = \{2\}$; $N_4^{out} = \{2\}$;
- $N_5^{in} = \{\}$; $N_5^{out} = \{3\}$;



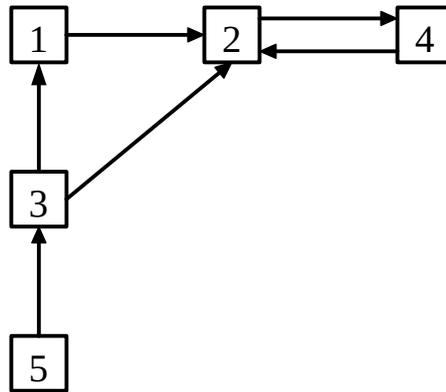


FIGURE 8.1 – Exemple d'un réseau social représenté par un graphe.

Lorsque nous nous intéressons à des réseaux sociaux avec l'objectif de diffuser des nouvelles idées ou des innovations, nous considérons que chaque nœud du réseau peut être *actif* ou *inactif*. Un nœud est dit actif aussitôt qu'il adopte l'innovation qui se diffuse sur le réseau. Les nœuds qui n'ont pas adopté l'innovation sont donc des nœuds dit inactifs. Une fois l'innovation adoptée et le nœud est devenu actif, il ne pourra pas revenir à inactif. C'est-à-dire, l'adoption d'une innovation est permanente et progressive.

Du point de vue d'un nœud j , initialement inactif, le processus de diffusion se présentera comme suit [68] :

- au fil du temps, de plus en plus des voisins d'entrée du nœud j deviennent actifs ;
- à un moment donné, cela peut amener le nœud j à devenir actif ;
- devenu actif, le nœud j déclenche la prise de décision de ses voisins de sortie ;
- les voisins de sortie vont décider à leurs tours s'ils adoptent l'innovation ou pas.

La condition d'activation d'un nœud dépend du modèle de diffusion. Plusieurs dynamiques de diffusion ont été proposées au cours des dernières décennies. Les deux modèles de diffusion les plus exploités sont : *le modèle de seuil linéaire* [58, 66, 91] et *le modèle de cascade indépendante* [13, 56, 66]. Dans nos travaux, nous nous sommes intéressés au modèle de cascade indépendante.

Initialement, un ensemble de nœuds (diffuseurs) est sélectionné pour propager l'influence à partir de l'étape $t = 0$. L'ensemble des diffuseurs initiaux, aussi appelés des *graines*, est noté par ϕ_0 . Une innovation est alors propagée à partir de l'ensemble initial étape par étape.

Notons

ϕ_t l'ensemble des nœuds activés (qui ont adopté l'innovation) à l'étape t ;

$\Phi_T = \cup_{s=0}^t \phi_s$ l'ensemble des nœuds activés dans l'étape t ou dans les étapes précédentes ;

t_{max} le nombre maximal d'étapes où plus aucun noeud adopte l'innovation ;
 $\Phi^* = \bigcup_{s=0}^{t_{max}} \phi_s$ l'ensemble final de tous les noeuds activés dans le processus de diffusion.

La section suivante présente le processus d'activation pour le modèle de cascade indépendante.

8.1.2 Modèle de cascade indépendante

Le modèle de cascade indépendante permet de décrire la diffusion des innovations comme la propagation d'un virus. Comme tous les modèles épidémiques [13, 56, 66], il repose sur l'hypothèse qu'un noeud peut adopter une innovation lorsque l'un de ses voisins a adopté l'innovation en question.

Dans un modèle de cascade indépendante, chaque arc $(i, j) \in E$ est associée à une probabilité de propagation $p : (V \times V) \rightarrow (0, 1]$, où $p_{i,j}$ représente la probabilité que le noeud j soit influencé par le noeud i à travers l'arc (i, j) à l'étape t lorsque le noeud i est activé à l'étape $t - 1$. De manière informelle, si le noeud i est activé à l'étape $t - 1$, il essaiera d'activer chacun de ses voisins de sortie à l'étape t . Notez que pour chacun de ses voisins de sortie, il n'a qu'une seule chance de l'activer. Si le noeud i ne parvient pas à activer l'un de ses voisins de sortie, il ne tentera pas d'activer le même noeud lors des étapes suivantes. Si de nombreux voisins d'entrée du noeud inactif j sont activés à l'étape $t - 1$, l'ordre dans lequel ils essaient d'influencer le noeud j à l'étape t n'affecte pas la probabilité d'activation du noeud j . C'est ce que nous appelons l'indépendance par rapport à l'ordre. Finalement, le modèle de cascade indépendante est désigné par un triple $G_{CI} = (V, E, p)$, où $G = (V, E)$ est le graphe social et $p : (V \times V) \rightarrow (0, 1]$ sont les probabilités de propagation.

8.1.3 Calcul de la propagation de l'influence par simulation

Les premières méthodes de calcul de la propagation de l'influence d'un noeud à partir d'un diffuseur initial étaient basées sur des méthodes de simulation.

Nous pouvons résumer la méthode de calcul de la propagation par simulation en quelques pas. Étant donné un ensemble de diffuseurs initiaux ϕ_0 , à l'étape t chaque noeud $i \in \phi_{t-1}$ essaie d'influencer ses voisins de sortie inactifs, c'est-à-dire, $j \in N_j^{out} \setminus \Phi_{t-1}$ avec une probabilité $p_{i,j}$. Si le noeud j est activé à l'étape t , alors il est ajouté à ϕ_t , c'est-à-dire, $\phi_t = \phi_{t-1} \cup \{j\}$. À partir de l'étape t_{max} où plus aucun noeud est activé, l'ensemble des noeuds actifs sera constant et nous considérons que la diffusion se termine avec $\Phi^* = \bigcup_{s=0}^{t_{max}} \phi_s$.

Pour simuler le processus d'évolution stochastique dans le modèle de cascade indépendante, un nombre aléatoire $r_{i,j}$ uniformément distribué dans l'intervalle $[0, 1]$ est généré, et nous supposons que le noeud i influence avec succès le noeud j lorsque $r_{i,j} \leq p_{i,j}$. Le processus d'évolution

détaillé est décrit dans l'algorithme 8.1.

ALG. 8.1 Calcul de la propagation de l'influence par simulation

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$ et un ensemble de diffuseurs initiaux $\phi_0 \subset V$

Output : L'ensemble final des noeuds actifs Φ^*

```

1  $t = 0, \Phi_t = \phi_t = \phi_0;$ 
2 while  $\phi_t \neq \emptyset$  do
3    $t = t + 1, \phi_t = \emptyset;$ 
4   for  $i \in \phi_{t-1}$  do
5     for  $j \in N_i^{out} \setminus \Phi_{t-1}$  do
6       Générer uniformément  $r_{i,j}$  de façon aléatoire entre  $[0, 1];$ 
7       if  $r_{i,j} \leq p_{i,j}$  then
8          $\phi_t = \phi_t \cup \{j\};$ 
9    $\Phi_t = \Phi_{t-1} \cup \phi_t;$ 
10  $\Phi^* = \Phi_t;$ 
  
```

Exemple 8.2. Pour exemplifier le calcul de la propagation de l'influence par simulation, reprenons le réseau social de la figure 8.1 en ajoutant quelques probabilités pour les arcs (Figure 8.2).

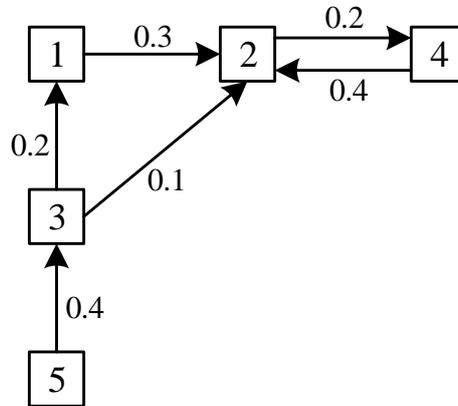


FIGURE 8.2 – Exemple d'un modèle de cascade indépendante

Considérons l'ensemble des diffuseurs initiaux $\phi_0 = \{4, 5\}$. La table 8.1 présente des probabilités aléatoires pour chaque arc et chaque étape de la simulation.

À l'étape 1 et à partir des noeuds initiaux $\{4, 5\}$ deux arcs de sortie existent $\{(4, 2), (5, 3)\}$. En vue des probabilités aléatoires tirées pour chaque arc, uniquement le noeud 3 adopte la nouveauté à cette étape, devient actif et est ajouté à $\phi_1 = \{3\}$. À l'étape 2, les deux arcs qui partent du noeud 3 sont prise en compte. À nouveau, uniquement un noeud (2) a adopté la nouveauté et est ajouté à $\phi_2 = \{2\}$. À partir du noeud 2, aucun voisin de sortie inactif existe et la diffusion s'arrête.

t	arc	$p_{i,j}$	$r_{i,j}$	adopte	ϕ_t	Φ_t
0					{4, 5}	{4, 5}
1	(4, 2)	0.4	0.654	non		
	(5, 3)	0.4	0.325	oui	{3}	{3, 4, 5}
2	(3, 1)	0.2	0.278	non		
	(3, 2)	0.1	0.089	oui	{2}	{2, 3, 4, 5}

TABLE 8.1 – Résultats d'une simulation du modèle de cascade indépendante de la figure 8.2.

Il faut noter que le processus d'évolution de ce modèle est stochastique et le même modèle peut avoir des évolutions différentes à partir du même ensemble de diffuseurs initiaux. À noter aussi que la probabilité de propagation $p_{i,j}$ représente la probabilité que le noeud i puisse influencer le noeud j et plus la valeur de $p_{i,j}$ est élevée plus forte est l'influence du noeud i sur le noeud j .

8.2 Calcul de la probabilité d'activation

Un indice clé qui permet d'évaluer l'influence globale d'un individu dans les réseaux sociaux est la probabilité d'activation. Basiquement, la probabilité d'activation est la somme des probabilités qu'un noeud soit actif lorsqu'on considère la diffusion à partir d'un individu.

Nous pouvons définir mathématiquement la probabilité d'activation $\sigma(\phi_0)$ par :

Définition 8.1. (*Probabilité d'activation*)

Soit un modèle de cascade indépendante $G_{CI} = (V, E, p)$ et un ensemble de diffuseurs initiaux ϕ_0 , la probabilité qu'un noeud $j \in V$ soit activé dans le processus de propagation de l'innovation est défini comme la probabilité d'activation du noeud j et noté par π_j .

La propagation de l'influence $\sigma(\phi_0)$ est donnée par

$$\sigma(\phi_0) = \sum_{j \in V} \pi_j$$

Au fil des années, plusieurs méthodes ont été proposées pour calculer la probabilité d'activation à partir d'un individu quelconque. Dans cette section, trois méthodes sont présentées : la méthode "classique" par simulation, une méthode de calcul exact de la probabilité d'activation ainsi que des heuristiques pour des approximations plus ou moins bonnes. La deuxième et troisième méthodes présentées font partie des apports de la thèse de Wenjing Yang [101].

8.2.1 Méthode par simulation

Le calcul de la probabilité d'activation par simulation utilise le principe de simulation de Monte Carlo. Nous la retrouvons dans plusieurs études [57, 66, 67, 72]. Ce principe consiste à simuler à des multiples reprises la propagation de l'innovation en utilisant l'algorithme 8.1. À chaque simulation, nous notons quels noeuds ont été activés. La probabilité d'activation de chaque noeud correspond au pourcentage des fois où le noeud a été activé par rapport au nombre total de simulation. Kempe et al. [66, 67] ont démontré que 10000 simulations à partir du même diffuseur sont suffisantes pour évaluer la probabilité d'activation.

Cependant, cette méthode n'est pas efficace sur le plan computationnel et d'autres travaux [31] ont été proposés pour améliorer d'efficacité computationnelle de cette méthode en limitant l'exploration de certaines branches du réseau ou en arrêtant le calcul lorsque la probabilité est inférieure à un certain seuil d'influence préfixé.

Ces améliorations réduisent le coût computationnel mais également la précision du résultat. Nous avons donc proposé une méthode de calcul exact de la probabilité d'activation.

8.2.2 Méthode de calcul exact de l'influence

Un de nos premiers apports sur la thématique de propagation d'influence par cascade indépendante a été la proposition d'un algorithme pour le calcul exact de la probabilité d'activation. Cette méthode appelée *path method* est décrite par la suite.

L'objectif de cette méthode est de prendre en compte toutes les évolutions possibles de la propagation de l'innovation d'un modèle, de sorte qu'elle puisse offrir une vision précise de la propagation de l'influence finale.

Pour cela, nous allons créer un graphe de l'évolution du modèle. Chaque élément de ce graphe est une cellule comme celle illustrée dans la figure 8.3.

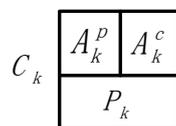


FIGURE 8.3 – Une cellule du graphe d'évolution de la *path method*.

Chaque cellule C_k est constituée de trois éléments :

- un ensemble A_k^p qui contient tous les noeuds activés avant l'étape actuelle ;
- un ensemble A_k^c qui contient tous les noeuds actifs dans l'étape courante ;
- P_k est la probabilité que l'évolution décrite par A_k^p et A_k^c se produise.

Une cellule dont les noeuds actifs est vide est appelée *cellule terminale*. Chaque cellule non terminale $C_i = (A_i^p, A_i^c, P_i)$ est reliée à chacune de ses cellules successeurs $C_k = (A_k^p, A_k^c, P_k)$ par un arc dirigé auquel est associée la probabilité de l'arc $P_a^{(i,k)}$. Elle indique la probabilité qu'une évolution atteigne la cellule C_i avant d'atteindre C_k . En additionnant toutes les probabilités des cellules terminales dont les ensembles de noeuds actifs antérieurs contiennent le noeud j , on peut calculer la probabilité d'activation exacte du noeud j .

L'algorithme 8.2 implémente la méthode pour la création du graphe d'évolution et le calcul de la probabilité d'activation.

L'algorithme définit comme cellule initiale $C_1 = (\emptyset, \phi_0, 1)$. Cela représente le fait qu'aucun noeud n'a pas été exploré ($A_a^p = \emptyset$), que l'ensemble des noeuds actifs est l'ensemble de diffuseurs initiaux ($A_1^c = \phi_0$) et la probabilité que ce noeud soit atteint est 1.0 ($P_1 = 1.0$). C_1 est également ajouté à l'ensemble des nouveaux noeuds (*new*) à explorer.

Tant que l'ensemble des nouveaux noeuds (*new*) n'est pas vide, une cellule $i \in \text{new}$ est sélectionnée et ses cellules filles sont calculées de la façon suivante. De la cellule C_i , l'innovation peut être propagée à n'importe quel sous-ensemble de

$$D = N_{A_i^c}^{\text{out}} - (A_i^p \cup A_i^c),$$

qui contient les voisins de sorties de A_i^c qui n'ont pas encore adopté l'innovation.

Deux cellules du graphe sont considérées équivalentes si les ensembles des noeuds actifs (A_i^p) et des noeuds actifs (A_i^c) à l'étape courante sont égaux. Pour chaque sous-ensemble $D_k \subseteq D$, une nouvelle cellule $C_k = (A_k^p, A_k^c, P_k)$ est créée s'il n'existe pas une autre cellule équivalente avec $A_k^p = A_i^p \cup A_i^c$ et $A_k^c = D_k$. La probabilité que la cellule C_k soit atteinte à partir de la cellule C_i est :

$$P_a^{(i,k)} = \prod_{q \in A_i^c} \prod_{r \in D - A_k^c} (1 - p(q, r)) \cdot (1 - \prod_{r' \in A_k^c} \prod_{q' \in A_i^c \cap N_r^{in}} (1 - p(q', r')))$$

Nous avons donc

$$P_k = P_i \cdot P_a^{(i,k)}$$

Si l'ensemble $A_k^c \neq \emptyset$ de la nouvelle cellule C_k n'est pas vide, alors C_k est ajoutée à l'ensemble *new*. Dans le cas contraire, C_k est une cellule terminale et ne sera pas explorée dans les étapes suivantes. Si la cellule C_k est équivalente à une autre cellule $C_{k'}$ déjà présente dans le graphe, nous ajoutons un arc de C_i à $C_{k'}$ avec $P_a^{(i,k')} = P_a^{(i,k)}$ et en ajoutant à la valeur de $P_{k'}$ le montant de $P_i \cdot \dots \cdot P_a^{(i,k')}$. Une fois la cellule explorée, elle est enlevée de l'ensemble *new*.

Après avoir construit le graphe d'évolution, nous pouvons calculer la probabilité π_j de chaque noeud $j \in V$ en ajoutant les probabilités de toutes les cellules terminales où A_i^p contient j .

ALG. 8.2 Calcul de la probabilité d'activation par la *path method*

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$ et un ensemble de diffuseurs initiaux $\phi_0 \subset V$

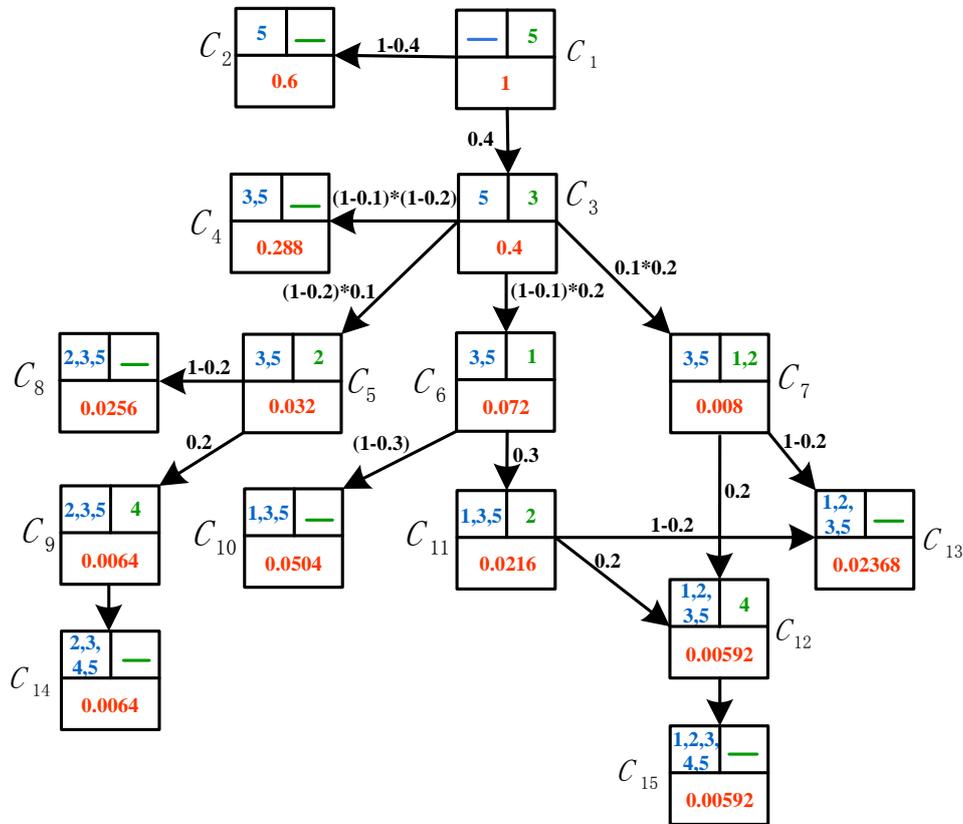
Output : La probabilité d'activation π_j^p pour chaque noeud $j \in V$

```

1 /* construction du graphe d'évolution */
2  $A_1^p = \emptyset, A_1^c = \phi_0, P_1 = 1;$ 
3 ajouter une cellule  $C_1 = (A_1^p, A_1^c, P_1)$  au graphe;
4  $new = \{C_1\}, k = 1;$ 
5 while  $new \neq \emptyset$  do
6   prendre  $C_i = (A_i^p, A_i^c, P_i) \in new;$ 
7   supprimer  $C_i$  de  $new;$ 
8    $D = N_{A_i^c}^{out} - (A_i^p \cup A_i^c);$ 
9   forall  $D' \subseteq D$  do
10     $k = k + 1;$ 
11     $D_k = D', A_k^c = D_k;$ 
12     $A_k^p = A_i^p \cup A_i^c;$ 
13     $P_a^{(i,k)} = \prod_{q \in A_i^c} \prod_{r \in D - A_k^c} (1 - p(q, r)) \cdot (1 - \prod_{r' \in A_k^c} \prod_{q' \in A_i^c \cap N_r^{in}} (1 - p(q', r')));$ 
14    if  $\exists C_{k'}, s.t. A_{k'}^c = A_k^c, A_{k'}^p = A_k^p$  then
15       $P_a^{(i,k')} = P_a^{(i,k)};$ 
16      ajouter au graphe un arc  $C_i \rightarrow C_{k'}$  avec probabilité  $P_a^{(i,k')}$ ;
17       $P_{k'} = P_{k'} + P_i \cdot P_a^{(i,k)};$ 
18       $k = k - 1;$ 
19    else
20       $P_k = P_i \cdot P_a^{(i,k)};$ 
21      ajouter une cellule  $C_k = (A_k^p, A_k^c, P_k)$  au graphe;
22      ajouter au graphe un arc  $C_i \rightarrow C_k$  avec probabilité  $P_a^{(i,k)};$ 
23      if  $A_k^c \neq \emptyset$  then
24         $new = new \cup \{C_k\};$ 
25  $n = k;$ 
26 /* calcul de la probabilité d'activation */
27 for  $j \in V$  do
28    $\pi_j = 0;$ 
29   for  $k = 1$  to  $n$  do
30     if  $A_k^c \neq \emptyset, j \in A_k^p$  then
31        $\pi_j = \pi_j + P_k;$ 
32 return  $\pi_j / \forall j \in V;$ 

```

Exemple 8.3. Le graphe d'évolution obtenu par path method du modèle de cascade indépendante de la figure 8.2 est présenté dans la figure 8.4 ci-après.

FIGURE 8.4 – Graphe d'évolution du modèle de la figure 8.2 obtenu par *path method*.

Cependant, cette méthode a une complexité en temps de $O(6^N)$ [104] où N est le nombre de noeuds du réseau. Cela, dans la pratique, empêche son utilisation sur de grands réseaux.

Une solution à ce problème de temps est l'utilisation des résultats approximatifs. Je présente dans la suite un ensemble de méthodes basées sur la méthode de point-fixe.

8.2.3 Calcul approximatif de l'influence par les méthodes de point-fixe

Aggarwal et al [2] ont proposé un algorithme appelé *Steady State Spread* (SSS) pour évaluer la propagation de l'influence dans un réseau. Cette méthode itérative basée sur le calcul de point-fixe détermine une valeur approximative de la probabilité d'activation du noeud en résolvant un système d'équations non-linéaires. Bien qu'Aggarwal et al [2] aient également fourni un algorithme itératif, ils n'ont pas prouvé que leur méthode itérative pouvait converger vers un seul résultat final et ont ignoré l'imprécision causée par certaines sous-structures spécifiques des réseaux, comme mentionné dans un travail ultérieur [105].

Commençons par donner la définition du point-fixe.

Définition 8.2. (*Point-fixe [65]*)

Soit une fonction réelle d'une variable réelle $f : \mathbb{R} \rightarrow \mathbb{R}$, un nombre réel x est un point-fixe de f s'il satisfait

$$x = f(x)$$

Soit un point $x(0)$ dans le domaine de définition de f , l'itération du point-fixe est

$$x(s+1) = f(x(s)), s = 0, 1, 2, \dots$$

qui génère la séquence $x(0), x(1), x(2), \dots$. Si la séquence converge au point x et f est continue, alors nous pouvons prouver que x est un point-fixe de f . ■

L'algorithme Steady State Spread (SSS)

Nous notons la probabilité d'activation d'un noeud $j \in V$ calculée par l'algorithme SSS par π_j^s .

De façon à pouvoir appliquer la théorie du point-fixe au calcul itératif de la propagation de l'influence, il est nécessaire de définir la fonction qui permet de calculer π_j^s . Dans le modèle de la cascade indépendante, le noeud j peut être activé par n'importe lequel de ses voisins d'entrée. De manière équivalente, pour que le noeud j ne soit pas activé, il ne doit être activé par aucun de ses voisins d'entrée. En supposant que l'activation des voisins d'entrée immédiats est un événement indépendant et qu'elle ne dépend pas de l'activation du noeud j , la probabilité de cette activation peut être notée par $\prod_{i \in N_j^{in}} (1 - \pi_i^s \cdot p_{i,j})$. La fonction de calcul peut donc être construite de la manière suivante :

$$\pi_i^s(t+1) = \begin{cases} 1 & \text{si } j \in \phi_0 \\ 1 - \prod_{i \in N_j^{in}} (1 - \pi_i^s(t) \cdot p_{i,j}) & \text{si } j \notin \phi_0 \end{cases} \quad (8.1)$$

La preuve de convergence de l'équation 8.1 est basée dans le théorème suivant :

Théorème 8.1. (*[90]*) Si une séquence de nombres réels est croissante et bornée, alors son suprême est la limite. ■

À partir de cela, nous avons fait la proposition suivante qui montre que l'équation 8.1 mène à une solution unique.

Proposition 8.1. La séquence $\{\pi_j^s(t)\}$ générée par l'équation 8.1 converge vers un point-fixe unique. ■

Le lecteur peut trouver la preuve de cette proposition dans [104].

Du fait de la convergence du point-fixe être asymptotique, nous avons besoin de définir un critère d'arrêt. Nous considérons que les itérations peuvent être arrêtées lorsque la différence absolue des résultats calculés entre deux itérations adjacentes est inférieur à $\varepsilon^* > 0$, c'est-à-dire, $\sum_{j \notin \phi_0} \pi_j^s(t+1) - \pi_j^s(t) \leq \varepsilon^*$.

Cette méthode de calcul est présentée dans l'algorithme 8.3 ci-après.

ALG. 8.3 Calcul de la probabilité d'activation par la méthode *Steady State Spread*

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un ensemble de diffuseurs initiaux $\phi_0 \subset V$ et un critère d'arrêt $\varepsilon^* > 0$

Output : La probabilité d'activation π_j^s pour chaque noeud $j \in V$

```

1  $t = 0$ ;
2  $\varepsilon = \varepsilon^* + 1$ ;
3  $\pi_j^s(0) = 1, \forall j \in \phi_0$ ;
4  $\pi_j^s(0) = 0, \forall j \in V \setminus \phi_0$ ;
5 while  $\varepsilon \geq \varepsilon^*$  do
6   for  $j \in V$  do
7     if  $j \in \phi_0$  then
8        $\pi_j^s(t+1) = 1$ ;
9     else
10       $\pi_j^s(t+1) = 1 - \prod_{i \in N_j^{in}} (1 - p_{i,j} \cdot \pi_i^s(t))$ ;
11    $\varepsilon = \sum_{j \notin \phi_0} |\pi_j^s(t+1) - \pi_j^s(t)|$ ;
12    $t = t + 1$ ;
13 return  $\pi_j^s = \pi_j^s(t-1)$ ;
```

Les problèmes de précision de l'algorithme SSS

Yang et al. [105] ont montré que l'algorithme SSS présente des problèmes de précision sur certaines structures de réseaux. Ces problèmes de précision viennent du fait que l'algorithme considère que tous les événements d'activation des voisins d'entrée d'un noeud j sont indépendants du noeud j lui-même. L'algorithme ne peut donc fournir un résultat exact lorsque le réseau a des arcs bidirectionnels ou des sous-structures avec des dépendances circulaires. Une définition formelle de ce que les acteurs appellent un réseau avec des *défauts structurels* peut être trouvée dans [105].

Dans [105], les auteurs ont également proposé une première amélioration à l'algorithme SSS qu'ils ont appelée *SSSbyStep*. Cet algorithme limite le nombre d'itérations par calculer la probabilité d'activation d'un noeud j quelconque. Cette amélioration réduit l'impact des arcs bidirectionnels dans les résultats mais n'est pas efficace dans le cas de sous-structures dépendances circulaires.

L'algorithme SSS-Noself

Afin d'améliorer la précision de l'algorithme SSS, nous proposons l'approche appelé *SSS-Noself* qui tient compte de certaines sous-structures spécifiques. L'objectif est d'assurer que le processus itératif de calcul de la probabilité d'activation d'un noeud j ne soit pas influencé par lui-même.

Pour implémenter cela, nous définissons de nouveaux réseaux où les probabilités d'influence d'un noeud q quelconque ne sont pas considérées. Autrement dit, pour tous les voisins d'entrée et de sortie du noeud q les probabilités d'influence sont mises à zéro. Nous avons appelé ce nouveau réseau *modèle de cascade indépendante restreint*.

Définition 8.3. (*Modèle de cascade indépendante restreint*)

Soit un modèle de cascade indépendante $G_{IC} = (V, E, p)$, un modèle de cascade indépendante restreint $G^{[q]} = (V, E^{[q]}, p^{[q]})$ est obtenu à partir de G par la suppression des arcs d'entrée et de sortie du noeud q . Le vecteur des probabilités d'activation $\vec{p}^{[q]}$ du réseau $G^{[q]}$ est défini par :

$$\pi_{i,j}^{[q]} = \begin{cases} 0 & \text{si } q = i \text{ ou } q = j \\ p_{i,j} & \text{autrement} \end{cases} \quad (8.2)$$

Pour permettre de calculer les probabilités d'influence sans que chaque noeud soit influencé par lui-même, $N' = N - |\phi_0|$ nouveaux réseaux sont créés où N est le nombre de noeuds dans le G et $|\phi_0|$ est le nombre de noeuds dans l'ensemble de diffuseurs initiaux.

Nous pouvons par la suite redéfinir la fonction de calcul du point-fixe par :

$$\pi_i^{[q]}(t+1) = \begin{cases} 1 & \text{si } j \in \phi_0 \\ 1 - \prod_{i \in N_j^{in}} (1 - \pi_i^{[q]}(t) \cdot p_{i,j}^{[q]}) & \text{si } j \notin \phi_0 \end{cases} \quad (8.3)$$

L'algorithme 8.4 ci-après modifie l'algorithme SSS pour calculer les probabilités d'influence où l'influence de chaque noeud j sur lui-même a été supprimée. Nous notons la probabilité d'activation d'un noeud $j \in V$ calculée par l'algorithme *SSS-Noself* par π_j^n .

Différemment de l'algorithme SSS qui calcule la probabilité d'activation pour tous les noeuds $j \in V$ du réseau à chaque étape t , l'algorithme *SSS-Noself* calcule également la probabilité d'activation pour tous les noeuds j à chaque étape t en considérant qu'un noeud $q \in V \setminus \phi_0$ reste inactif. Cela engendre une complexité de temps de $O(N^2)$ contre une complexité de $O(N)$ pour l'algorithme SSS.

ALG. 8.4 Calcul de la probabilité d'activation par la méthode *SSS-Noself*

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un ensemble de diffuseurs initiaux $\phi_0 \subset V$ et un critère d'arrêt $\varepsilon^* > 0$

Output : La probabilité d'activation π_j^n pour chaque noeud $j \in V$

```

1  $t = 0$ ;
2  $\varepsilon = \varepsilon^* + 1$ ;
3 for  $q \in V \setminus \phi_0$  do
4    $\pi_j^{[q]}(0) = 1, \forall j \in \phi_0$ ;
5    $\pi_j^{[q]}(0) = 0, \forall j \in V \setminus \phi_0$ ;
6    $\pi_j(0) = 1, \forall j \in \phi_0$ ;
7    $\pi_j(0) = 0, \forall j \in V \setminus \phi_0$ ;
8   while  $\varepsilon \geq \varepsilon^*$  do
9     for  $q \in V \setminus \phi_0$  do
10      for  $j \in V$  do
11        if  $j \in \phi_0$  then
12           $\pi_j^{[q]}(t+1) = 1$ ;
13           $\pi_j(t+1) = 1$ ;
14        else
15           $\pi_j^{[q]}(t+1) = 1 - \prod_{i \in N_j^{in}} (1 - p_{i,j}^{[q]} \cdot \pi_i^{[q]}(t))$ ;
16           $\pi_j(t+1) = 1 - \prod_{i \in N_j^{in}} (1 - p_{i,j} \cdot \pi_i^{[j]}(t))$ ;
17       $\varepsilon_1 = \sum_{j \notin \phi_0} |\pi_j(t+1) - \pi_j(t)|$ ;
18       $\varepsilon_2 = \sum_{j \notin \phi_0} |\pi_j^s(t+1) - \pi_j^s(t)| (q \in V \setminus \phi_0)$ ;
19       $\varepsilon = \max(\varepsilon_1, \varepsilon_2)$ ;
20       $t = t + 1$ ;
21 return  $\pi_j^n = \pi_j(t-1)$ ;

```

L'algorithme SSS-Bounded-Path

Une autre approche efficace, également basée sur le calcul du point-fixe, a été proposée pour calculer la probabilité qu'un noeud soit activé par un chemin de longueur minimale à partir de l'ensemble de diffuseurs. Le principe de cette approche est que, pour chaque noeud $j \in V \setminus \phi_0$, le calcul de la probabilité d'activation soit réalisé uniquement à une étape t précise du calcul.

Une première approche qui limite le nombre d'itérations pour calculer la probabilité d'activation est basée sur le calcul du chemin de longueur minimale. Nous définissons le chemin de longueur minimale d'un noeud $j \in V \setminus \phi_0$ par sp_j sous l'hypothèse que j soit accessible à partir de ϕ_0 . Kimura et Saito [69] ont proposé les méthodes SPM et SP1M où le noeud j peut être actif uniquement dans l'étape $t = sp_j$ pour SPM ou $t = sp_j + 1$ pour SP1M. Dans des approches similaires [31, 105], les chemins de longueur maximale sont utilisés pour limiter les étapes où un noeud j peut être activé.

Un problème commun à ces trois approches est la complexité importante engendrée par le calcul des chemins minimaux ou maximaux qui doit être réalisé au préalable. L'algorithme de Dijkstra est souvent utilisé pour déterminer ces chemins.

Nous avons proposé une approche basée sur la méthode de calcul du point-fixe qu'utilise une limitation de la longueur de chemin mais qui n'a pas besoin d'un calcul préalable des chemins. Dans notre procédure, nous calculons sp_j à l'étape t lorsque la probabilité π_j passe pour la première fois de zéro à une valeur non nulle. En outre, nous définissons la limite du chemin pour calculer π_j en tenant compte non seulement des chemins les plus courts, mais aussi des chemins dont la longueur n'est pas supérieure à $sp_j + b_0$, où b_0 est un nombre entier constant qui définit notre limite de longueur de chemin. Il est évident que nous obtenons les mêmes résultats que l'approche SPM lorsque $b_0 = 0$, et SP1M lorsque $b_0 = 1$. De plus, lorsque b_0 est suffisamment grand, cet algorithme est équivalent à l'algorithme SSS.

Cet algorithme, appelé *SSS-Bounded-Path* (Alg. 8.5), peut fournir donc une limite inférieure pour le calcul de la propagation de l'influence en fonction de la longueur choisie pour b_0 . Nous notons la probabilité d'activation d'un noeud $j \in V$ calculée par l'algorithme *SSS-Bound-Path* pour $b_0 \in \mathbb{N}$ par $\pi_j^{bp(b_0)}$.

ALG. 8.5 Calcul de la probabilité d'activation par la méthode *SSS-Bounded-Path*

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un ensemble de diffuseurs initiaux $\phi_0 \subset V$, limite du chemin $b_0 \in \mathbb{N}$ et un critère d'arrêt $\varepsilon^* > 0$

Output : La probabilité d'activation π_j^{bp} pour chaque noeud $j \in V$

```

1  $t = 0$ ;
2  $\varepsilon = \varepsilon^* + 1$ ;
3  $\pi_j^{bp}(0) = 1, \forall j \in \phi_0$ ;
4  $\pi_j^{bp}(0) = 0, \forall j \in V \setminus \phi_0$ ;
5  $b_j = \inf, \forall j \in V \setminus \phi_0$ ;
6  $stop = 0$ ;
7 while  $\varepsilon \geq \varepsilon^*$  do
8   while  $stop = 0$  do
9      $stop = 1$ ;
10    for  $j \in V$  do
11       $\pi_j^{bp}(t+1) = \pi_j^{bp}(t)$ ;
12      if  $j \notin \phi_0$  &&  $t \leq b_j$  then
13         $\pi_j^{bp}(t+1) = 1 - \prod_{i \in N_j^{in}} (1 - p_{i,j} \cdot \pi_i^{bp}(t))$ ;
14         $stop = 0$ ;
15        if  $\pi_j^{bp}(t+1) \neq 0$  &&  $\pi_j^{bp}(t) = 0$  then
16           $b_j = t + 1 + b_0$ ;
17       $t = t + 1$ ;
18     $\varepsilon = \sum_{j \notin \phi_0} |\pi_j^{bp}(t+1) - \pi_j^{bp}(t)|$ ;
19 return  $\pi_j^{bp} = \pi_j^{bp}(t-1)$ ;

```

L'algorithme *SSS-Bounded-Path* généralise les approches SPM et SP1M [69], en exploitant le calcul efficace du point-fixe par l'algorithme SSS. Le résultat de *SSS-Bounded-Path* ($b_0 = 0$) peut être considéré comme une limite inférieure pour la probabilité d'activation exacte. Comme pour l'algorithme SSS, la complexité en temps de la méthode *SSS-Bounded-Path* est $O(N)$. Cependant, dans la plupart des cas, *SSS-Bounded-Path* ($b_0 = 0$) arrête l'itération avant qu'elle ne converge, de sorte que *SSS-Bounded-Path* ($b_0 = 0$) prend généralement moins de temps que l'algorithme SSS.

Comparaison des résultats

Dans la suite, nous faisons une comparaison des résultats obtenus, en termes de précision, par les algorithmes proposés dans les sections précédentes. Les résultats obtenus pour le calcul de la propagation de l'influence du modèle de la figure 8.2 sont présentés dans la table 8.2. Les colonnes en gras (noeuds 2 et 4) montrent les résultats où nous avons des influences d'un noeud sur lui-même.

Méthodes	Noeuds				
	1	2	3	4	5
Path-method	0.08	0.0616	0.4	0.0123	1
SSS	0.08	0.0678	0.4	0.0132	1
SSS-Noself	0.08	0.0630	0.4	0.0126	1
SSS-Bounded-Path ($b_0 = 0$)	0.08	0.0400	0.4	0.0080	1
SSS-Bounded-Path ($b_0 = 1$)	0.08	0.0630	0.4	0.0126	1
SSS-Bounded-Path ($b_0 = 2$)	0.08	0.0660	0.4	0.0132	1
SSS-Bounded-Path ($b_0 = 3$)	0.08	0.0678	0.4	0.0132	1

TABLE 8.2 – Résultats de la propagation de l'influence du modèle de la figure 8.2

Comme déjà mentionné auparavant, *Path-method* nous permet d'obtenir le résultat exact des probabilités pour la propagation de l'influence et établir une base de comparaison. Nous pouvons constater que *SSS* donne une valeur supérieure à celle de *Path-method*. *SSS-Noself* donne un résultat plus précis que *SSS* et *SSS-Bounded-path* ($b_0 = 0$) donne une limite inférieure. Cela nous mène à la proposition suivante.

Proposition 8.2. Soit un modèle de cascade indépendante $G_{IC} = (V, E, p)$ et un ensemble de diffuseurs initiaux $\phi_0 \subset V$, la probabilité d'activation du noeud j calculée par *Path-method* (π_j), *SSS* (π_j^s), *SSS-Bounded-Path* ($b_0 = 0$) ($\pi_j^{bp(0)}$) et *SSS-Noself* (π_j^n) satisfait :

$$\pi_j^{bp(0)} \leq \pi_j \leq \pi_j^n \leq \pi_j^s \quad (8.4)$$

■

Le lecteur peut retrouver la preuve de cette proposition dans [104].

Si nous considérons uniquement les méthodes *SSS* et *SSS-Bounded-Path*, nous pouvons déduire les remarques suivantes.

Remarque 8.1. Soit un modèle de cascade indépendante $G_{IC} = (V, E, p)$ et un ensemble de diffuseurs initiaux $\phi_0 \subset V, \forall j \in V$, nous avons $\pi_j^{bp(b)} \leq \pi_j^{bp(b')}$ pour $0 \leq b < b'$.

La déduction est triviale étant donné que plus long sont les chemins évalués dans le calcul, plus le noeud a une probabilité d'être activé.

Remarque 8.2. Soit un modèle de cascade indépendante $G_{IC} = (V, E, p)$ et un ensemble de diffuseurs initiaux $\phi_0 \subset V, \forall j \in V$, nous avons $\lim_{b \rightarrow \infty} \pi_j^{bp(b)} = \pi_j^s$ pour le même critère d'arrêt ε^* .

Deux critères sont utilisés pour arrêter la méthode *SSS-Bounded-Path*, le nombre d'itérations qui est limité à b et l'erreur acceptable ε^* . Lorsque b tend l'infini, uniquement le critère ε^* sera prise en compte. Cela revient à être équivalent à la méthode *SSS*.

Une comparaison plus détaillée basée sur des évaluations expérimentales sur différents modèles se trouve également dans [104].

8.3 Maximisation de l'influence par activation des liens

Le marketing viral est l'une des principales motivations de l'étude de la propagation de l'influence. Lors d'une promotion d'un nouveau produit ou d'une nouvelle information à travers les réseaux sociaux par l'effet du bouche-à-oreille, un moyen rentable de propagande consiste à cibler un ensemble de diffuseurs initiaux dans le réseau en investissant une quantité limitée de ressources. L'objectif est que ces diffuseurs initiaux incitent d'autres membres du réseau à adopter le même produit ou la même information, entraînant ainsi une propagation potentiellement importante dans le réseau.

Le problème de la maximisation de l'influence par la sélection des diffuseurs initiaux peut être décrit pour un réseau social et un modèle de propagation de l'influence comme le choix d'un ensemble de diffuseurs initiaux comprenant jusqu'à K noeuds du réseau de sorte que la propagation finale de l'influence soit maximisée.

L'utilisation de techniques algorithmes pour la maximisation de l'influence comme un outil de marketing viral a été initialement proposée par Domingos et Richardson [44] dans un cadre probabiliste basé sur les champs aléatoires de Markov. Kempe et al [66, 67] ont d'abord formulé le problème de la maximisation de l'influence par la sélection des diffuseurs comme un problème d'optimisation stochastique discrète. Plusieurs travaux ont prouvé que le problème de maximisation de l'influence dans le modèle à seuil linéaire ainsi que le modèle de cascade indépendante sont *NP-hard* [42, 100]. Kempe et al [67] ont également proposé un algorithme basé sur la simulation de Monte Carlo qui garantit que la propagation de l'influence se rapproche de

la propagation optimale avec un facteur de $(1 - 1/e)$ où e est le logarithme naturel e . Cependant, cette approche prend beaucoup de temps, c'est pourquoi plusieurs travaux ultérieurs se sont concentrés à améliorer l'efficacité de ces approches algorithmiques.

Nous abordons dans cette section le problème de maximisation de l'influence en s'intéressant au lien entre les noeuds. Notre idée est, avec un budget limité, choisir un ensemble de liens à activer de façon à maximiser l'influence en sachant que l'activation de chaque lien a un coût différent.

Commençons par formaliser le problème.

8.3.1 Formalisation du problème

Nous considérons un modèle de cascade indépendante $G_{IC} = (V, E, p)$ où les arcs dans E sont inactifs par défaut mais peuvent être activés par un contrôleur externe. Nous considérons aussi un vecteur $c \in V \times V$ où le coût d'activation $c_{i,j}$ représente le coût pour activer le lien entre le noeud i et le noeud j . Notons $\Phi_K \subset 2^V$ l'ensemble de tous les sous-ensembles de V de cardinalité K . Supposons que l'ensemble de diffuseurs initiaux est une variable aléatoire uniformément distribuée, qui prend la valeur ϕ_0 avec une probabilité $pr(\phi_0) = \frac{1}{|\Phi_K|}$ pour tous les ensembles $\phi_0 \in \Phi_K$, où $|\Phi_K|$ est le nombre d'éléments dans Φ_K . Nous pouvons alors activer un ensemble de liens $E_a \subset E$ pour construire un graphe actif où le coût total associé à ces liens ne dépasse pas un budget \mathcal{B} . L'ensemble des noeuds connectés par les liens dans E_a est appelé *noeuds potentiellement actifs* et est représenté par V_a . Le graphe actif est donc représenté par $\hat{G}_{IC} = (V_a, E_a, p)$. L'objectif est de maximiser l'espérance de propagation de l'influence (EIP), dénotée par $\mathbb{E}[\sigma(E_a, \phi_0)]$ par l'activation de l'ensemble des liens dans E_a . Nous définissons formellement ce problème de la manière suivante :

Définition 8.4. (*Maximisation de l'influence par activation des liens*)

Soit un modèle de cascade indépendante $G_{IC} = (V, E, p)$, une constante entière K et un vecteur de coût $c \in V \times V$. Notons $\Phi_K \subset 2^V$ l'ensemble de tous les sous-ensembles de V de cardinalité K . L'ensemble des diffuseurs initiaux ϕ_0 , c'est-à-dire, l'état initial du réseau, est une variable aléatoire avec une probabilité $pr(\phi_0) = \frac{1}{|\Phi_K|}$. Le problème consiste à activer un ensemble de liens $E_a \subset E$ pour construire un graphe actif $\hat{G}_{IC} = (V_a, E_a, p)$, tel que l'espérance de propagation de l'influence (EIP) $\mathbb{E}[\sigma(E_a, \phi_0)]$ soit maximisée et le coût total de l'activation des liens $(i, j) \in E_a$ ne soit pas supérieur à un budget \mathcal{B} , c'est-à-dire,

$$\begin{aligned} \max \quad & \mathbb{E}[\sigma(E_a, \phi_0)] \\ \text{tel que} \quad & \sum_{(i,j) \in E_a} c_{i,j} \leq \mathcal{B}, \\ & E_a \subset E, \\ & \phi_0 \subset \Phi_K \subset 2^V, \\ & \mathcal{B} \in \mathbb{R}_+ \end{aligned} \tag{8.5}$$

où

$$\begin{aligned}\mathbb{E}[\sigma(E_a, \phi_0)] &= \sum_{\phi_0 \in \Phi_K} \sigma(E_a, \phi_0) \cdot pr(\phi_0) \\ &= \frac{1}{|\Phi_K|} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(E_a, \phi_0),\end{aligned}$$

et

$$\sigma(E_a, \phi_0) = \sum_{j \in V_a \setminus \phi_0} \pi_j$$

■

Nous avons démontré certaines propriétés sur ce problème telles que la monotonie et la sous-modularité ainsi que le fait d'être *NP-Hard* en ce qui concerne la couverture maximale. Toutes les preuves de ces propriétés sont disponibles dans [101] et ne seront pas abordées dans ce document.

Je présente par la suite plusieurs heuristiques pour calculer l'ensemble optimal des liens à activer.

8.3.2 Heuristiques Cost-Degree

Dans l'objectif de maximiser l'EIP et sous la condition d'activer un ensemble de liens où le coût ne dépasse pas un budget donné, nous proposons deux algorithmes qui explorent le coefficient coût-degré (*Cost-Degree*) pour calculer l'ensemble optimal de liens à activer. L'idée de ces deux algorithmes est de prendre en compte la probabilité de propagation et le coût d'activation de chaque lien. Les liens qui ont une haute probabilité de propagation avec un faible coût d'activation seront les premiers à être considérés.

L'algorithme 8.6 présente une première approche, appelée *Simplex Cost-Degree*, où le coefficient coût-degré $\Theta_{i,j}^{(1)} = \frac{p_{i,j}}{c_{i,j}}$ prend en compte uniquement la probabilité et le coût de chaque lien. À chaque itération, le lien avec la valeur de $\Theta_{i,j}^{(1)}$ maximale est ajouté à l'ensemble des liens E_a .

Malgré une faible complexité ($O(m)$) où $m = |E|$, cet algorithme peut avoir de mauvaises performances dans la maximisation de l'influence sur certaines structures de réseaux. Par exemple, lorsqu'un lien (i, j) est associé à un coefficient coût-degré $\Theta_{i,j}^{(1)}$ élevé mais les voisins de sortie du noeud j ont des coefficients faibles ou, dans des cas extrêmes, n'ont pas de voisins de sortie pour propager l'influence.

L'algorithme 8.7 intègre au calcul du coefficient coût-degré $\Theta_{i,j}^{(2)} = \frac{p_{i,j} \cdot \sum_{s \in N_j^{out}} p_{j,s}}{c_{i,j}}$ les probabilités de propagation des voisins de sortie du noeud j .

L'exemple 8.4 ci-après décrit le processus de sélection pour ces deux algorithmes.

ALG. 8.6 Simplex Cost-Degree

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un vecteur de coût c et un budget \mathcal{B}

Output : Un ensemble d'arcs actifs E_a

- 1 $E_a = \emptyset$;
- 2 calculer $\Theta_{i,j}^{(1)} = \frac{p_{i,j}}{c_{i,j}} \forall (i, j) \in E$;
- 3 **while** $\sum_{(i,j) \in E} c_{i,j} \leq \mathcal{B}$ **do**
- 4
$$E_a = E_a \cup \left\{ \underset{(i,j) \in E \setminus E_a}{\operatorname{argmax}} (\Theta_{i,j}^{(1)}) \right\}$$
- 5 **return** E_a ;

ALG. 8.7 Multiple Cost-Degree

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un vecteur de coût c et un budget \mathcal{B}

Output : Un ensemble d'arcs actifs E_a

- 1 $E_a = \emptyset$;
- 2 calculer $\Theta_{i,j}^{(2)} = \frac{p_{i,j} \cdot \sum_{s \in N_j^{\text{out}}} p_{j,s}}{c_{i,j}} \forall (i, j) \in E$;
- 3 **while** $\sum_{(i,j) \in E} c_{i,j} \leq \mathcal{B}$ **do**
- 4
$$E_a = E_a \cup \left\{ \underset{(i,j) \in E \setminus E_a}{\operatorname{argmax}} (\Theta_{i,j}^{(2)}) \right\}$$
- 5 **return** E_a ;

Exemple 8.4. La figure 8.5 présente un modèle de cascade indépendante. Nous considérons un vecteur de coût c tel que $c_{1,3} = 10, c_{2,3} = 9, c_{3,4} = 4, c_{4,3} = 8, c_{4,5} = 10, c_{4,6} = 5, c_{4,7} = 9$, la cardinalité de l'ensemble des diffuseurs initiaux est fixée à $K = 2$ et le budget $\mathcal{B} = 25$.

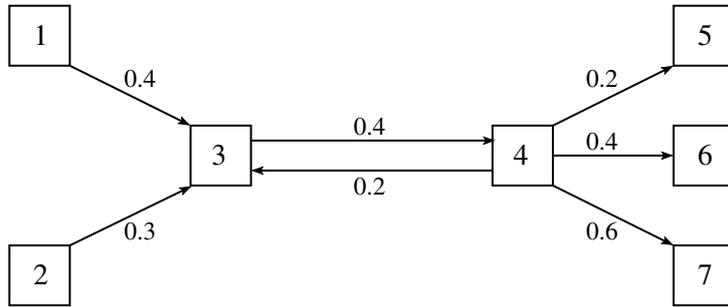


FIGURE 8.5 – Un modèle de cascade indépendante avec 7 noeuds.

Pour l'algorithme Simplex Cost-Degree, nous avons pour $\Theta_{i,j}^{(1)}$ les valeurs suivantes :

$$\begin{aligned}\Theta_{1,3}^{(1)} &= \frac{p_{1,3}}{c_{1,3}} = 0.040, & \Theta_{2,3}^{(1)} &= \frac{p_{2,3}}{c_{2,3}} = 0.033, & \Theta_{4,3}^{(1)} &= \frac{p_{4,3}}{c_{4,3}} = 0.025, \\ \Theta_{4,5}^{(1)} &= \frac{p_{4,5}}{c_{4,5}} = 0.020, & \Theta_{4,6}^{(1)} &= \frac{p_{4,6}}{c_{4,6}} = 0.080, & \Theta_{4,7}^{(1)} &= \frac{p_{4,7}}{c_{4,7}} = 0.067, \\ \Theta_{3,4}^{(1)} &= \frac{p_{3,4}}{c_{3,4}} = 0.100.\end{aligned}$$

L'ensemble des liens actifs sélectionnés, pour un budget \mathcal{B} de 25, est $E_a = \{(3, 4), (4, 6), (4, 7)\}$. La somme des coûts associée à ces trois liens est de 18, donc, inférieure à la limite de 25 mais aucun autre lien ne peut être ajouté. L'EIP associé est $\mathbb{E}[\sigma(E_a, \phi_0)] = 0.410$.

Dans le cas de l'algorithme *Multiple Cost-Degree*, nous avons pour $\Theta_{i,j}^{(2)}$ les valeurs suivantes :

$$\begin{aligned}\Theta_{1,3}^{(2)} &= \frac{p_{1,3} \cdot p_{3,4}}{c_{1,3}} = 0.016, & \Theta_{2,3}^{(2)} &= \frac{p_{2,3} \cdot p_{3,4}}{c_{2,3}} = 0.013, \\ \Theta_{4,3}^{(2)} &= \frac{p_{4,3} \cdot p_{3,4}}{c_{4,3}} = 0.010, & \Theta_{4,5}^{(2)} &= \frac{p_{4,5}}{c_{4,5}} = 0.0, \\ \Theta_{4,6}^{(2)} &= \frac{p_{4,6}}{c_{4,6}} = 0.0, & \Theta_{4,7}^{(2)} &= \frac{p_{4,7}}{c_{4,7}} = 0.0, \\ \Theta_{3,4}^{(2)} &= \frac{p_{3,4} \cdot (p_{4,3} + p_{4,5} + p_{4,6} + p_{4,7})}{c_{3,4}} = 0.120.\end{aligned}$$

L'ensemble des liens actifs sélectionnés est $E_a = \{(3, 4), (1, 3), (2, 3)\}$ avec une somme des coûts de 23. L'EIP associé est $\mathbb{E}[\sigma(E_a, \phi_0)] = 0.307$.

Il faut noter que malgré une EIP plus élevée pour l'algorithme *Simplex Cost-Degree*, les noeuds 1 et 2 ne seront jamais activés mais les noeuds activés auront une probabilité plus forte. Dans le cas de l'algorithme *Multiple Cost-Degree*, nous avons une couverture plus importante des noeuds mais avec une EIP plus faible.

Il est évident que, dans ces deux algorithmes, uniquement les probabilités d'activation et les coûts d'activation sont pris en compte pour la sélection des liens. Les liens choisis ne sont pas forcément le meilleur choix car ils n'ont pas de voisins de sortie ou menant à d'autres diffuseurs initiaux. Autrement dit, la propagation de l'influence n'est pas prise en compte dans la sélection. Les algorithmes suivants combinent ces fautes avec une nouvelle heuristique qui prends en compte la propagation de l'influence sur le réseau.

8.3.3 Heuristiques Influence-Degree

Nous proposons deux algorithmes basés dans des coefficients influence-degré (*Inf-Degree*) où la propagation de l'influence est prise en compte à chaque itération de façon à calculer l'impact de l'ajout de chaque lien à l'ensemble E_a .

L'algorithme 8.8 introduit une première approche, appelée *Simplex Inf-Degree*, qui intègre le calcul du coefficient influence-degré $\zeta_{i,j}^{(1)} = \frac{\mathbb{E}[\sigma(E_a \cup \{(i,j)\}, \phi_0)] - \mathbb{E}[\sigma(E_a, \phi_0)]}{c_{i,j}}$ à la sélection des liens

à activer. Pour cela, notons Φ_K l'ensemble de tous les sous-ensembles de V de cardinalité K tel que $\Phi_K = \phi_{0,1}, \phi_{0,2}, \dots, \phi_{0,r}, \dots$ et $\Delta_r = \sigma(E_a \cup (i, j), \phi_{0,r}) - \sigma(E_a, \phi_{0,r})$, alors nous avons :

$$\begin{aligned} \zeta_{i,j}^{(1)} &= \frac{\mathbb{E}[\sigma(E_a \cup \{(i,j)\}, \phi_0)] - \mathbb{E}[\sigma(E_a, \phi_0)]}{c_{i,j}} \\ &= \frac{\frac{1}{\Phi_K} \cdot \left[\sum_{\phi_0 \in \Phi_K} \sigma(E_a \cup \{(i,j)\}, \phi_0) - \sum_{\phi_0 \in \Phi_K} \sigma(E_a, \phi_0) \right]}{c_{i,j}} \\ &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_{0,r} \in \Phi_K} \Delta_r}{c_{i,j}} \end{aligned}$$

ALG. 8.8 Simplex Inf-Degree

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un vecteur de coût \mathbf{c} , un budget \mathcal{B} et un entier K

Output : Un ensemble d'arcs actifs E_a

- 1 Notons Φ_K l'ensemble de tous les sous-ensembles de V de cardinalité K ;
 - 2 $E_a = \emptyset$;
 - 3 **while** $\sum_{(i,j) \in E} c_{i,j} \leq \mathcal{B}$ **do**
 - 4 calculer $\zeta_{i,j}^{(1)} = \frac{\mathbb{E}[\sigma(E_a \cup \{(i,j)\}, \phi_0)] - \mathbb{E}[\sigma(E_a, \phi_0)]}{c_{i,j}} \forall (i, j) \in E \setminus E_a$;
 - 5 $E_a = E_a \cup \{ \underset{(i,j) \in E \setminus E_a}{\operatorname{argmax}} (\zeta_{i,j}^{(1)}) \}$
 - 6 **return** E_a ;
-

La complexité associée à cet algorithme est $O(m^2 |\Phi_K| T)$ où T est le temps requis pour calculer la propagation de l'influence $\sigma(\cdot)$, $m = |E|$ et $|\Phi_K| = \frac{N!}{K!(N-K)!}$ où $N = |V|$.

Malheureusement, ces deux approches basées sur les coefficients influence-degré ne fournissent qu'une approximation de la solution optimale. Nous avons pu observer que si l'EIP de E_a sélectionné n'était pas inférieure à la variation incrémentale de l'EIP causée par l'ajout d'un lien non sélectionné, alors nous pouvons approximer la solution optimale de $\frac{1}{2}(1 - \frac{1}{e})$.

Nous avons pu proposer un théorème avec une garantie d'approximation de la solution optimale.

Théorème 8.2. Soit un modèle de cascade indépendante $G_{IC} = (V, E, p)$, une constante entière K , un vecteur de coût $\mathbf{c} \in V \times V$ et un entier K tel que la cardinalité de l'ensemble des diffuseurs initiaux $|\phi_0| = K$. Soit $E_a^* = \underset{\sum_{(i,j) \in E_a} c_{i,j} \leq \mathcal{B}}{\operatorname{argmax}} \mathbb{E}[\sigma(E_a, \phi_0)]$ l'ensemble optimal de liens

qui maximise l'EIP avec un budget \mathcal{B} . Calculer l'ensemble des liens actifs $E_a \subset E$ par les algorithmes Simplex Inf-Degree avec le même budget. Soit $(u, v) \in E \setminus E_a$ n'importe quel lien non sélectionné dans l'ensemble E_a . Si l'EIP de E_a n'est pas inférieure à la variation incrémentale de l'EIP engendrée par l'addition de l'importe quel lien non sélectionné, c-à-d, $\mathbb{E}[\sigma(E_a, \phi_0)] \geq \mathbb{E}[\sigma(E_a \cup \{(u, v)\}, \phi_0)] - \mathbb{E}[\sigma(E_a, \phi_0)] \forall (u, v) \in E \setminus E_a$, nous avons

$$\mathbb{E}[\sigma(E_a, \phi_0)] \geq \frac{1}{2} \left(1 - \frac{1}{e}\right) \cdot \mathbb{E}[\sigma(E_a^*, \phi_0)].$$



La preuve de ce théorème se trouve dans [101] (Chapitre 5) et ne sera pas présentée dans ce document.

L'algorithme *Simplex Inf-Degree* considère uniquement le coût d'activation du lien en question pour le calcul du coefficient. Il nous semble plus raisonnable de considérer le coût total des liens dans l'ensemble E_a . Dans l'algorithme *Multiple Inf-Degree* (Alg. 8.9) nous considérons la somme des coûts de tous les liens déjà dans l'ensemble E_a plus le lien en question pour le calcul du coefficient $\zeta_{i,j}^{(2)} = \frac{\mathbb{E}[\sigma(E_a \cup \{(i,j)\}, \phi_0)] - \mathbb{E}[\sigma(E_a, \phi_0)]}{\sum_{(u,v) \in E_a \cup \{(i,j)\}} c_{u,v}}$.

ALG. 8.9 Multiple Inf-Degree

Input : Un modèle de cascade indépendante $G_{CI} = (V, E, p)$, un vecteur de coût c , un budget \mathcal{B} et un entier K

Output : Un ensemble d'arcs actifs E_a

- 1 Notons Φ_K l'ensemble de tous les sous-ensembles de V de cardinalité K ;
 - 2 $E_a = \emptyset$;
 - 3 **while** $\sum_{(i,j) \in E} c_{i,j} \leq \mathcal{B}$ **do**
 - 4 calculer $\zeta_{i,j}^{(2)} = \frac{\mathbb{E}[\sigma(E_a \cup \{(i,j)\}, \phi_0)] - \mathbb{E}[\sigma(E_a, \phi_0)]}{\sum_{(u,v) \in E_a \cup \{(i,j)\}} c_{u,v}} \forall (i,j) \in E \setminus E_a$;
 - 5 $E_a = E_a \cup \{ \underset{(i,j) \in E \setminus E_a}{\operatorname{argmax}}(\zeta_{i,j}^{(2)}) \}$
 - 6 **return** E_a ;
-

La complexité de l'algorithme 8.9 est égale à celle de l'algorithme 8.8.

Exemple 8.5. *Considérons à nouveau le modèle de cascade indépendante de la figure 8.5, ainsi que les mêmes valeurs des coûts d'activation c , de la cardinalité de l'ensemble des diffuseurs initiaux K et du budget \mathcal{B} de l'exemple 8.4.*

Lorsque nous appliquons l'algorithme Simplex Inf-Degree, pour la première itération, nous avons :

$$\begin{aligned} \zeta_{1,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(1,3)\}, \phi_0)}{c_{1,3}} = 0.0095, & \zeta_{2,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(2,3)\}, \phi_0)}{c_{2,3}} = 0.0079, \\ \zeta_{4,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,3)\}, \phi_0)}{c_{4,3}} = 0.0060, & \zeta_{4,5}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,5)\}, \phi_0)}{c_{4,5}} = 0.0048, \\ \zeta_{4,6}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,6)\}, \phi_0)}{c_{4,6}} = 0.0190, & \zeta_{4,7}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,7)\}, \phi_0)}{c_{4,7}} = 0.0159, \\ \zeta_{3,4}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(3,4)\}, \phi_0)}{c_{3,4}} = 0.0238. \end{aligned}$$

À la fin de cette première itération, nous avons $E_a = \{(3,4)\}$, $\mathbb{E}[\sigma(\{(3,4)\}, \phi_0)] = 0.0952$ pour un coût d'activation de $c_{3,4} = 4$.

À la deuxième itération, nous avons :

$$\begin{aligned}\zeta_{1,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(1,3) \cup (3,4)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4)\}, \phi_0)]}{c_{1,3}} = 0.0126, \\ \zeta_{2,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(2,3) \cup (3,4)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4)\}, \phi_0)]}{c_{2,3}} = 0.0105, \\ \zeta_{4,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,3) \cup (3,4)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4)\}, \phi_0)]}{c_{4,3}} = 0.0060, \\ \zeta_{4,5}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,5) \cup (3,4)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4)\}, \phi_0)]}{c_{4,5}} = 0.0063, \\ \zeta_{4,6}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,6) \cup (3,4)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4)\}, \phi_0)]}{c_{4,6}} = 0.0252, \\ \zeta_{4,7}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,7) \cup (3,4)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4)\}, \phi_0)]}{c_{4,7}} = 0.0210.\end{aligned}$$

À la fin de la deuxième itération, nous avons $E_a = \{(3, 4), (4, 6)\}$, $\mathbb{E}[\sigma(\{(3, 4) \cup (4, 6)\}, \phi_0)] = 0.2210$ pour un coût d'activation de $c_{3,4} + c_{4,6} = 9$.

À la troisième itération, nous avons :

$$\begin{aligned}\zeta_{1,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(1,3) \cup (3,4) \cup (4,6)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6)\}, \phi_0)]}{c_{1,3}} = 0.0135, \\ \zeta_{2,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(2,3) \cup (3,4) \cup (4,6)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6)\}, \phi_0)]}{c_{2,3}} = 0.0133, \\ \zeta_{4,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,3) \cup (3,4) \cup (4,6)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6)\}, \phi_0)]}{c_{4,3}} = 0.0059, \\ \zeta_{4,5}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,5) \cup (3,4) \cup (4,6)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6)\}, \phi_0)]}{c_{4,5}} = 0.0063, \\ \zeta_{4,7}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,7) \cup (3,4) \cup (4,6)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6)\}, \phi_0)]}{c_{4,7}} = 0.0210.\end{aligned}$$

À la fin de la troisième itération, nous avons $E_a = \{(3, 4), (4, 6), (4, 7)\}$, $\mathbb{E}[\sigma(\{(3, 4) \cup (4, 6) \cup (4, 7)\}, \phi_0)] = 0.4100$ pour un coût d'activation de $c_{3,4} + c_{4,6} + c_{4,7} = 18$.

À la quatrième itération, nous avons :

$$\begin{aligned}\zeta_{1,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(1,3) \cup (3,4) \cup (4,6) \cup (4,7)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6) \cup (4,7)\}, \phi_0)]}{c_{1,3}} = 0.0110, \\ \zeta_{2,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(2,3) \cup (3,4) \cup (4,6) \cup (4,7)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6) \cup (4,7)\}, \phi_0)]}{c_{2,3}} = 0.0013, \\ \zeta_{4,3}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,3) \cup (3,4) \cup (4,6) \cup (4,7)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6) \cup (4,7)\}, \phi_0)]}{c_{4,3}} = 0.0, \\ \zeta_{4,5}^{(1)} &= \frac{\frac{1}{\Phi_K} \cdot \sum_{\phi_0 \in \Phi_K} \sigma(\{(4,5) \cup (3,4) \cup (4,6) \cup (4,7)\}, \phi_0) - \mathbb{E}[\sigma(\{(3,4) \cup (4,6) \cup (4,7)\}, \phi_0)]}{c_{4,5}} = 0.0062.\end{aligned}$$

À la fin de la quatrième itération, le choix se porte sur le lien (1, 3). Cependant, le coût d'activation $c_{1,3} = 10$ fait dépasser le budget $\mathcal{B} = 25$. Le lien ne sera pas ajouté à l'ensemble E_a et l'algorithme s'arrête.

Pour l'algorithme Multiple Inf-Degree, nous avons, pour la première itération, exactement les mêmes résultats que pour l'algorithme Simplex Inf-Degree. $\zeta_{i,j}^{(2)} = \zeta_{i,j}^{(1)} \forall (i, j) \in E$. Nous avons donc $E_a = \{(3, 4)\}$, $\mathbb{E}[\sigma(\{(3, 4)\}, \phi_0)] = 0.0952$ pour un coût d'activation de $c_{3,4} = 4$.

À la deuxième itération, nous avons les valeurs de $\zeta_{i,j}^{(2)}$ suivantes :

$$\zeta_{1,3}^{(2)} = 0.0090, \quad \zeta_{2,3}^{(2)} = 0.0073, \quad \zeta_{4,3}^{(2)} = 0.0040, \quad \zeta_{4,5}^{(2)} = 0.0045, \quad \zeta_{4,6}^{(2)} = 0.0140, \quad \zeta_{4,7}^{(2)} = 0.0145.$$

À la fin de la deuxième itération, nous avons $E_a = \{(3, 4), (4, 7)\}$, $\mathbb{E}[\sigma(\{(3, 4) \cup (4, 7)\}, \phi_0)] = 0.2837$ pour un coût d'activation de $c_{3,4} + c_{4,7} = 13$.

À la troisième itération, nous avons les valeurs de $\zeta_{i,j}^{(2)}$ suivantes :

$$\zeta_{1,3}^{(2)} = 0.0061, \quad \zeta_{2,3}^{(2)} = 0.0048, \quad \zeta_{4,3}^{(2)} = 0.0023, \quad \zeta_{4,5}^{(2)} = 0.0027, \quad \zeta_{4,6}^{(2)} = 0.0070.$$

À la fin de la troisième itération, nous avons $E_a = \{(3, 4), (4, 7), (4, 6)\}$, $\mathbb{E}[\sigma(\{(3, 4) \cup (4, 7) \cup (4, 6)\}, \phi_0)] = 0.4100$ pour un coût d'activation de $c_{3,4} + c_{4,7} + c_{4,6} = 18$.

À la quatrième itération, nous avons les valeurs de $\zeta_{i,j}^{(2)}$ suivantes :

$$\zeta_{1,3}^{(2)} = 0.0053, \quad \zeta_{2,3}^{(2)} = 0.0041, \quad \zeta_{4,3}^{(2)} = 0.0018, \quad \zeta_{4,5}^{(2)} = 0.0022.$$

À la fin de la quatrième itération, le choix se porte à nouveau sur le lien (1, 3), ce que fait dépasser le budget $\mathcal{B} = 25$. Le lien ne sera pas ajouté à l'ensemble E_a et l'algorithme s'arrête.

Le tableau 8.3 ci-après résume les résultats obtenus pour les quatre algorithmes.

Algorithme	E_a	$\mathbb{E}[\sigma(E_a, \phi_0)]$
Simplex Cost-Degree	$\{(3, 4), (4, 6), (4, 7)\}$	0.4100
Multiple Cost-Degree	$\{(3, 4), (1, 3), (2, 3)\}$	0.3072
Simplex Inf-Degree	$\{(3, 4), (4, 6), (4, 7)\}$	0.4100
Multiple Cost-Degree	$\{(3, 4), (4, 6), (4, 7)\}$	0.4100

TABLE 8.3 – Comparaison des ensembles E_a et des EIP pour les quatre algorithmes de sélection pour le modèle de la figure 8.5

Nous proposons par la suite une évaluation expérimentale des ces quatre algorithmes sur des données réelles.

8.3.4 Évaluation expérimentale

Nous avons réalisé une série d'expériences sur des ensembles de données réelles afin d'évaluer les heuristiques proposées ci-avant dans le but de maximiser l'EIP en activant un ensemble de liens.

Le premier ensemble de données réelles, appelé *Advise-Seek*, est issu d'une étude réalisée sur une succursale d'une banque australienne [83]. Il représente la relation de demande de conseil au sein d'un personnel de 11 personnes, c'est-à-dire, que dans le réseau, un lien dirigé de i à j indique que i demande conseil à j si un problème survient au travail. Le réseau Advise-Seek comporte donc 11 noeuds et 30 liens.

Le deuxième ensemble de données, appelé *HighTec*, a été collecté auprès des cadres d'une entreprise de haute technologie aux États-Unis [70]. Il représente la relation de hiérarchie entre 21 cadres de l'entreprise, c'est-à-dire qu'un lien dirigé de i à j indique que i fait rapport à j . Ce réseau est composé de 21 noeuds et 20 liens.

Les deux réseaux considérés sont à la fois dirigés et binaires. Pour construire des modèles de cascade indépendante à partir de ces réseaux, nous avons généré des probabilités de propagation aléatoires $p_{i,j}$ uniformément distribuées dans l'intervalle $[0.1, 0.5)$ pour chaque lien (i, j) . Le coût d'activation $c_{i,j}$ associé au lien (i, j) a été choisi aussi aléatoirement et uniformément dans l'intervalle $[1, 10]$. Le budget pour l'activation des liens \mathcal{B} varie dans l'ensemble $\{10, 20, 30, 40, 50\}$ et la cardinalité de l'ensemble de diffuseurs initiaux K est choisie parmi $\{1, 3, 5\}$. Tous les algorithmes ont été implémentés en Python et les expériences ont été réalisées sur un ordinateur équipé d'un processeur Intel Core i5 à 2.40 GHz et d'une mémoire vive de 8 Go.

Nous avons évalué les performances des quatre algorithmes, Simplex Cost-Degree (SimCD), Multiple Cost-Degree (MulCD), Simplex Inf-Degree (SimID) et Multiple Inf-Degree (MulID), en termes de propagation de l'influence et de temps de calcul pour la sélection des liens à activer. Pour valider la solution, la propagation de l'influence de chaque ensemble de diffuseurs initiaux possible est calculée par l'algorithme SSS présenté dans la section précédente. L'EIP est ensuite obtenu en calculant la moyenne de ces valeurs de propagation d'influence.

Dans les résultats obtenus pour le réseau Advise-Seek (Figure 8.6), nous avons pu observer que SimID surpasse les autres heuristiques dans la plupart des cas, sauf pour $\mathcal{B} = 20, K = 3$ et $\mathcal{B} = 20, K = 5$, où MulID produit des EIP légèrement plus importants que SimID. SimCD et MulCD ont des performances similaires pour toutes les valeurs de K et de budget \mathcal{B} et sont dans quasiment tous les cas inférieurs à SimID et MulID.

Dans le réseau HighTec (Figure 8.7), lorsque $K = 3$ et $K = 5$, SimID et MulID obtiennent des résultats presque identiques, de même que SimCD et MulCD. Dans l'ensemble, SimID peut sélectionner l'ensemble des liens actifs pour obtenir une valeur EIP plus élevée que les autres heuristiques dans la plupart des cas.

Nous avons également évalué les performances des quatre algorithmes en ce que concerne

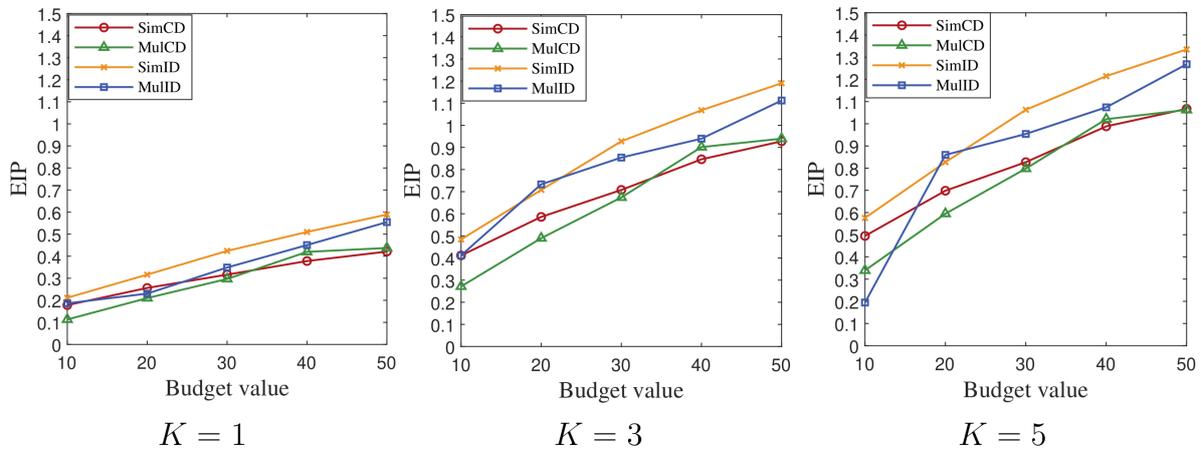


FIGURE 8.6 – EIP calculée par les quatre algorithmes de sélection pour le réseau *Advise-seek* avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$.

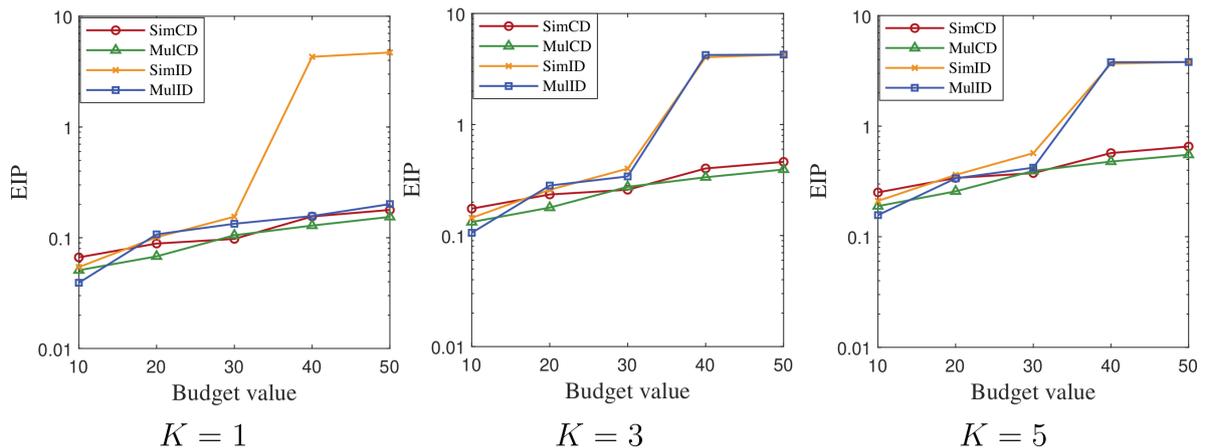


FIGURE 8.7 – EIP calculée par les quatre algorithmes de sélection pour le réseau *HighTec* avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$.

le temps de calcul. Les résultats obtenus sont présentés dans les tableaux 8.4 et 8.5 ci-après. Il faut noter que le temps de calcul pris en compte concerne uniquement le temps nécessaire pour choisir l'ensemble de liens actifs par les différentes heuristiques, cela ne prend pas en compte le temps de calcul de l'EIP. Nous avons pu observer que la sélection de liens par SimCD ou MulCD ne dépend pas de la taille de l'ensemble de diffuseurs initiaux et prend très peu de temps dans les deux réseaux. En revanche, la complexité de SimID et MulID est étroitement liée à la cardinalité de l'ensemble de diffuseurs initiaux K , de sorte que SimID et MulID prennent beaucoup plus de temps lorsque K augmente.

En outre, un budget plus important permet d'effectuer davantage d'itérations pour sélectionner plus de liens actifs. Ainsi, la durée d'exécution de SimID et MulID augmente également

avec le budget, mais la croissance de la durée d'exécution de SimID et MulID dépend toujours davantage de la cardinalité de l'ensemble de semences K que du budget \mathcal{B} .

Taille de l'ensemble de diffuseurs Temps de calcul \ Budget Algorithme	$K = 1$					$K = 3$					$K = 5$				
	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
SimCD	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
MulCD	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
SimID	0.9	1.3	1.6	2.0	2.2	12.4	18.0	21.9	28.1	30.9	38.1	54.2	76.3	70.1	76.1
MulID	0.7	0.9	1.2	1.7	2.3	11.2	22.2	19.0	26.6	27.2	28.1	50.0	50.1	65.1	62.3

TABLE 8.4 – Comparaison des temps de calcul pour la sélection des liens avec les quatre algorithmes de sélection pour le réseau *Advise-seek* avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$.

Taille de l'ensemble de diffuseurs Temps de calcul \ Budget Algorithme	$K = 1$					$K = 3$					$K = 5$				
	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
SimCD	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
MulCD	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
SimID	1.2	1.2	1.2	4.1	6.8	52.4	80.0	123.6	193.9	295.9	797.4	1386.6	2006.3	2646.0	3892.6
MulID	0.6	1.2	1.4	2.1	2.0	41.4	80.2	92.3	179.1	240.7	671.0	1029.3	3612.3	3217.9	4175.3

TABLE 8.5 – Comparaison des temps de calcul pour la sélection des liens avec les quatre algorithmes de sélection pour le réseau *HighTec* avec $K = \{1, 3, 5\}$ et un budget $\mathcal{B} = \{10, 20, 30, 40, 50\}$

Dans ce chapitre, j'ai présenté mes contributions sur le modèle de propagation de l'influence par cascade indépendante en ce qui concerne les méthodes de calcul de la propagation mais aussi sur la maximisation de l'influence sur le réseau. Dans le chapitre suivant, je dresse un bilan de mes travaux sur les formalismes stochastiques.

Chapitre 9

Bilan des travaux sur les formalismes stochastiques

Malgré l'explosion de l'espace d'état souvent associée aux modèles stochastiques, nous arrivons à concevoir des modèles simples qui permettent d'analyser et évaluer le comportement des systèmes avant même son déploiement.

Dans le chapitre de synthèse de cette partie, j'ai commencé par présenter un modèle RAS pour le paradigme d'ordonnancement par vol de tâches. Ce modèle, issue de plusieurs discussions avec des collègues de thèse au Laboratoire d'Informatique de Grenoble (LIG) est le dernier en date d'une série des travaux sur la modélisation des algorithmes parallèles. Mon intérêt pour la programmation parallèle et distribuée ne date pas d'aujourd'hui. Ma première publication [23] dans ce domaine date de 2002 et s'intéresse à la modélisation et analyse du taux d'utilisation des liens réseau dans un cluster. Toujours dans une approche théorique, d'autres publications ont suivi au cours des années. Dans [24] et [15], un modèle analytique des algorithmes parallèles du type master/slave a été proposé. Dans mes travaux de thèse [25], j'ai proposé un modèle pour représenter la structure de la grille de calcul GRID5000. Ce modèle s'intéressait à la disponibilité ponctuelle dans les machines qui la composaient en prenant en compte le taux de défaillances des machines mais aussi des défaillances du réseau de communication.

Dans le cadre de la thèse de Feda Almuhsen, encadrée par Nicolas Durand et Mohamed Quafafou, j'ai eu l'opportunité de reprendre mes recherches sur la modélisation du trafic routier dans un contexte stochastique. Dans cette collaboration, nous avons pu mettre ensemble le domaine d'extraction de connaissances et de l'évaluation des performances pour prédire l'état du trafic urbain. La particularité de ce modèle est la simplicité de la chaîne de Markov. Malgré cela, nous avons pu constater que l'extraction de connaissances, un traitement préalable et une représentation de cela par des Chaînes de Markov donne des meilleures prédictions qu'une simple analyse statistique. Feda Almuhsen est aujourd'hui ingénieure de recherche en intelligence artificielle au CEA. Ces travaux ont donné lieu à une conférence nationale et une conférence internationale.

Dans le dernier chapitre de cette partie, j'ai détaillé mes travaux sur la propagation de l'influence. Ce sujet est en vogue depuis plusieurs années et où énormément de contributions ont été faits sur ce sujet. Nous avons pu trouver deux points sur lesquels la communauté ne s'était pas encore penché.

Dans la littérature sur les modèles de propagation de l'influence, la plupart des études sur le calcul sont basées sur des méthodes de simulation sans garantie de l'exactitude des solutions. Nous avons pu, non seulement, proposer une méthode qui calcule la solution exacte de la propagation mais aussi des améliorations à d'autres méthodes déjà existantes.

Les contributions sur la maximisation de la propagation de l'influence par l'activation des liens dans les modèles de cascade indépendantes sont une approche tout à fait innovante. Nous avons décidé d'aborder le sujet de la maximisation de l'influence d'un point de vue qui se focalisait sur les liens entre chaque individu du réseau, contrairement aux approches classiques qui se focalisaient sur l'influence que chaque individu pouvait propager.

Mes recherches sur la propagation de l'influence ont été menées dans le cadre du projet PHC-Caï Yuanpei : IMEBS et de la thèse de Wenjing Yang, aujourd'hui "associate professor" à Northwest University à Xi'an en Chine. Ces travaux ont donné lieu à trois publications dont deux conférences internationales et un journal.

Malgré le grand nombre de travaux sur la propagation de l'influence présentes dans la littérature, certains aspects restent très peu explorés, comme par exemple, l'estimation de la probabilité d'influence entre deux individus. Des perspectives de recherche sur cet aspect sont présentés dans le chapitre 10.

Publications majeures

1. Feda ALMUHISEN, Nicolas DURAND, Leonardo BRENNER, Mohamed QUAFAROU. *Sequential Patterns for Spatio-Temporal Traffic Prediction*. **20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)**, pp. 595-602, 2021. [hal-03480960](#)
2. Wenjing YANG, Leonardo BRENNER, Alessandro GIUA. *Influence Maximization in Independent Cascade Networks Based on Activation Probability Computation*. **IEEE Access**, vol. 7, pp. 13745-13757, 2019. [hal-02373686](#)
3. Wenjing YANG, Leonardo BRENNER, Alessandro GIUA. *Influence Maximization by Link Activation in Social Networks*. **IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)**, pp.1248-1251, 2018. [hal-01997472](#)
4. Wenjing YANG, Leonardo BRENNER, Alessandro GIUA. *Computation of Activation Probabilities in the Independent Cascade Model*. **5th International Conference on Control, Decision and Information Technologies (CoDIT)**, pp.791-797, 2018. [hal-01997464](#)
5. Leonardo BRENNER, Sarah NEMMOUR, Ihab SBEITY. *Modélisation et évaluation de performance d'un algorithme de vol de travail sur des architectures multi-coeurs*. **10ème**

Conférence Internationale de Modélisation, Optimisation et Simulation (MOSIM),
pp.1-10, 2014. [hal-01166686](#)

Chapitre 10

Conclusion et perspectives

10.1 Conclusion générale

Dans ce document, j'ai présenté une vision globale, parfois plus, parfois moins détaillée, de l'ensemble de mes travaux de recherche depuis mon recrutement. Même si dans la structuration de ce document, j'ai séparé mes contributions dans deux catégories de formalismes, le fil conducteur de ma recherche se trouve dans la modélisation et l'analyse des systèmes à événements discrets. Analyse que je fais plus souvent par des méthodes de simulation dans le cas des formalismes déterministes ou par des méthodes numériques dans le cas des formalismes stochastiques.

Bien que ce fil conducteur soit assez générique, cela représente bien mes activités de recherche qui sont entre le domaine de l'informatique et celui de l'automatique.

Dans la première partie de ce document, j'ai présenté plusieurs travaux qui touchent aux formalismes des RdP. Ces travaux ont été développés dans le cadre de deux thèses ainsi que de plusieurs projets.

Mes premières contributions sur les formalismes déterministes ont commencé par l'extension des RdPLots de façon à intégrer dans le formalisme une relation flux/densité que nous pouvons trouver dans différents systèmes tels que ceux du trafic routier. Un des côtés intéressants des travaux que nous avons des contributions plus théoriques avec l'extension du formalisme, des contributions plus applicatives dans la modélisation du trafic et la collecte et l'analyse des données réelles et dans le développement d'un simulateur.

Le simulateur Simuleau que j'ai présenté dans le chapitre 5 est l'outil qui implémente, non seulement les contributions sur les RdPLots Triangulaires, mais aussi les méthodes de contrôle du flux qui ont été développées plus tard dans le cadre de la thèse de Ruotian Liu et que j'ai présentées brièvement dans la section 2.2 du chapitre 2.

Au niveau des projets, j'ai eu l'occasion de travailler avec des chercheurs reconnus dans

la communauté nationale et internationale. Si je me focalise sur les travaux sur la sécurité des systèmes cyber-physiques qui ont commencé en 2020 dans le cadre des projets CPsécurité (I et II) et qui sont toujours en cours dans l'ANR-PRC MENACES montrent l'intérêt de la communauté dans une configuration mondiale où les attaques à ce type de systèmes sont de plus en plus courantes.

En ce qui concerne mes recherches sur des formalismes stochastiques, j'ai exposé dans la deuxième partie de ce document, les travaux développés après ma thèse, mais mes travaux sur cette catégorie de formalismes remontent à bien plus long temps.

Cela va de même pour mes contributions sur la modélisation de systèmes parallèles où ma première publication dans le domaine date de 2002 [23]. Mes contributions sur la modélisation un algorithme d'ordonnancement par vol de tâches sont les dernières d'une série de modèles dédiés à l'évaluation de performances des systèmes parallèles.

Dans le cadre des collaborations avec l'équipe DANA au LIS, j'ai eu l'opportunité de mettre en oeuvre un modèle de prédiction du trafic urbain qui a mélangé des techniques de fouilles de données spatio-temporelles avec des modèles stochastiques, dans ce cas, les Chaînes de Markov. L'exploration des données réelles issues des taxis à Pékin et San Francisco montrent l'applicabilité et l'intérêt de la méthode.

Sur le sujet de propagation de l'influence, qui est en vogue depuis plusieurs années, nous avons trouvé une communauté très large et active. Les premières contributions sur les méthodes de calcul de la propagation exacte ainsi que les méthodes de maximisation de l'influence par analyse de liens entre les individus ont été innovantes et plusieurs fois référencées par la communauté. C'est une approche qui mérite de continuer d'être explorée.

Nous avons pu constater que mes travaux ont des apports aussi bien théoriques sur le domaine de systèmes à événements discrets ainsi que des apports plus applicatifs liés aux domaines du transport, des réseaux sociaux et plus récemment de la sécurité des systèmes cyber-physiques. Cela ouvre plusieurs perspectives qui sont brièvement exposées dans la suite.

10.2 Perspectives

Pour la suite de mes recherches, je suis particulièrement intéressé par les deux thématiques détaillées dans les chapitres 3 et 8.

En ce que concerne les RdPLots Triangulaires, mon objectif est d'étendre le formalisme afin de proposer un environnement complet pour la modélisation, la simulation et l'analyse du trafic multimodal. De mon point de vue, cela comprend plusieurs aspects pas encore entièrement explorés. Un premier aspect est d'intégrer la notion de dépassement des lots déjà proposé par Demongodin [36] mais pas encore repris dans les RdPlots Triangulaires. Un autre aspect qui m'intéresse est la modélisation du trafic piéton. Même si le trafic piéton partage plusieurs points en commun avec le trafic routier, tel qu'un diagramme fondamental similaire, d'autres

points lui sont particuliers. Nous pouvons citer comme une caractéristique particulière du trafic piéton, un flux bidirectionnel sur un même espace, c'est le cas d'un trottoir où, par exemple, un piéton peut changer de direction à n'importe quel moment. Notre objectif est toujours de proposer un modèle au niveau mésoscopique où nous modélisons le déplacement de piétons en les considérant par groupes d'individus. Une approche au niveau mésoscopique permet une simulation plus rapide que les modèles microscopiques et un modèle plus riche en détails que les modèles macroscopiques. Toujours dans l'idée des trafics multimodaux, l'intégration des voies cyclables, de plus en plus présentes, peut donner matière à étendre le formalisme. Un autre point faiblement exploré jusqu'à maintenant est la modélisation des lois de contrôle pour le trafic routier au niveau des formalismes SED. Nous pouvons nous inspirer de certains travaux réalisés au niveau macroscopique [30] en les adaptant et améliorant pour le niveau mésoscopique.

Du côté des travaux sur la propagation de l'influence, j'aimerais explorer par la suite des méthodes d'apprentissage automatique sur des données brutes pour la création des réseaux, c'est-à-dire, les connexions entre les individus et surtout pour l'estimation de la probabilité de l'influence d'un individu sur un autre. Si les méthodes de propagation de l'influence sont largement connues et utilisés dans la communauté, les travaux sur l'estimation de la probabilité de l'influence individuelle sont beaucoup moins nombreux. Nous pouvons imaginer plusieurs cas de figure où une estimation plus précise de l'influence individuelle est importante. Par exemple, nous pouvons construire un réseau (et estimer l'influence) sur un sujet précis (la marque préférée de téléphone portable au moment d'un achat). Cette influence entre les individus ne sera pas la même sur un sujet différent (la meilleure équipe de football) par exemple. J'aimerais également explorer la notion du temps dans ce type de réseau, par exemple, la probabilité d'influence peut changer au cours du temps après des désaccords sur des sujets sensibles. Un désaccord entre deux individus peut couper l'influence sur tout un sous-ensemble d'individus du réseau. Le support de diffusion utilisé, où l'information se propage à des vitesses différentes est aussi un sujet intéressant à explorer.

Appliquer des méthodes d'apprentissage automatique sur de grands volumes de données implique un grand besoin en puissance de calcul. L'idée de coupler ces méthodes aux techniques de calcul de haute performance nous permettrait de générer des réseaux à la volée et avoir des résultats plus pertinents avec une vraie relation avec le monde réel.

Étant donné mon recrutement récent en tant que Chaire Professeur Junior sur le projet OSMIUM (*End-to-end Synthesis Modelling InfrastructUre siMulation for Environmental Data Science*) pour une période de quatre ans, mon projet de recherche concerne aussi mes contributions à ce projet.

Le projet OSMIUM vise à développer un framework/environnement de modélisation qui puisse explorer des méthodes d'Intelligence Artificielle (IA) et des calculs de haute performance (HPC) pour la simulation et visualisation des informations.

Nous parlons ici des systèmes qui sont de plus en plus complexes et où c'est souvent difficile d'avoir une vision globale de leur comportement. Quand nous parlons des systèmes du type manufacturier, une conception plus réfléchie de chaque composant individuel du système permet une connaissance plus accrue, plus étudiée et le comportement global peut être estimé à

partir des comportements de chaque composant. Cependant, quand nous parlons des systèmes auto-organisés comme par exemple les réseaux sociaux où les connections entre les individus ne sont pas connues à l'avance et sont, surtout très dynamiques, l'utilisation des données peut aider à la modélisation et la formalisation de ces systèmes de façon à avoir une description plus précise d'eux.

Basé sur les objectifs du projet OSMIUM, je vois le développement de ce projet de recherche selon trois axes :

- Modélisation et formalisation des informations ;
- Utilisation de l'IA pour l'accélération de la simulation et visualisation tout en augmentant la précision ;
- Utilisation du calcul de haute performance en support à l'IA, à la simulation et à la visualisation.

Mon objectif est de proposer un environnement de modélisation, simulation et visualisation des données où l'IA, plus particulièrement l'apprentissage automatique, sera utilisé dans l'extraction des informations, les formalismes à événements discrets dans la modélisation et simulation des connaissances extraites et HPC en support à l'implémentation de ces méthodes en vue de l'accélération et la précision des résultats.

L'intérêt de l'utilisation des formalismes à événement discrets dans mon projet réside dans le fait de proposer un formalisme qui puisse structurer les connaissances et démontrer certaines propriétés du modèle en utilisant des outils bien connus de ce type de formalisme. Par exemple, des preuves de justesse, de vivacité, de blocage du modèle obtenu.

Des méthodes d'apprentissage automatique seront utiles dans la découverte des comportements (dynamique) du modèle qui n'ont pas été prévus initialement, comme par exemple, des failles non prévues. Tout ça à partir de, potentiellement, plusieurs sources de données.

Un deuxième rôle que j'aimerais exploiter à partir de l'apprentissage automatique est la découverte de bons paramètres des modèles formels déjà existants. Dans certains systèmes, la dynamique interne du système est bien connue. Cependant, les taux, les probabilités, ou de façon plus générique, le temps associé aux changements des états du modèle ne sont pas connus. L'exploitation des données, potentiellement datées, peuvent apporter des informations importantes et augmenter la précision du modèle.

Le HPC entre en jeu au tout début du processus de modélisation en support à des méthodes d'apprentissage automatique, mais aussi à la fin du processus dans la simulation et la visualisation des données et résultats obtenus. À nouveau, un traitement parallèle de ces informations peut apporter une accélération importante à toutes ces étapes. Mais cela passera par une parallélisation efficace des méthodes d'apprentissage automatique et de simulation spécifiques à nos besoins.

Bibliographie

- [1] Gnu linear programming kit. <https://www.gnu.org/software/glpk/>. Accessed : 2023-01-27.
- [2] C. C. Aggarwal, A. Khan, and X. Yan. On flow authority discovery in social networks. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 522–533. SIAM, 2011.
- [3] K. Agrawal, C. E. Leiserson, Y. He, and W. J. Hsu. Adaptive work-stealing with parallelism feedback. *ACM Trans. Comput. Syst.*, 26(3), sep 2008.
- [4] H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8 :159–188, 1998.
- [5] F. Almuhsen. *Leveraging formal concept analysis and pattern mining for moving object trajectory analysis*. PhD thesis, 2018. Thèse de doctorat dirigée par Quafafou, Mohamed et Durand, Nicolas Informatique Aix-Marseille 2018.
- [6] F. Almuhsen, N. Durand, L. Brenner, and M. Quafafou. Prédiction de l'état du trafic routier basée sur les motifs et les chaînes de markov. In *25èmes Rencontres de la Société Francophone de Classification (SFC)*, pages 1–3, 2018.
- [7] F. Almuhsen, N. Durand, L. Brenner, and M. Quafafou. Sequential patterns for spatio-temporal traffic prediction. In *20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 595–602, 2021.
- [8] F. AlMuhsen, N. Durand, and M. Quafafou. Detecting behavior types of moving object trajectories. *International Journal of Data Science and Analytics*, 5(2-3) :169–187, 2018.
- [9] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre. Observer design for bounded output synchronized Petri nets. In *2021 European Control Conference (ECC)*, pages 746–751. IEEE, 2021.
- [10] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre. Observer design for labeled finite automata with inputs under stealthy actuators attacks. *IFAC-PapersOnLine*, 55(28) :46–51, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.
- [11] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre. Robust stealthy attacks based on uncertain costs and labeled finite automata with inputs. *IEEE Robotics and Automation Letters*, 8(5) :2732–2739, 2023.
- [12] R. Ammour, L. Brenner, I. Demongodin, S. Amari, and D. Lefebvre. Costs analysis of stealthy attacks with bounded output synchronized Petri nets. In *2021 IEEE 17th*

- International Conference on Automation Science and Engineering (CASE)*, pages 799–804, 2021.
- [13] R. M. Anderson, B. Anderson, and R. M. May. *Infectious Diseases of Humans : Dynamics and Control*. Oxford University Press, 1991.
- [14] N. Audry and F. Prunet. Controlled batches Petri nets. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 1849–1854, 1994.
- [15] L. Baldo, L. Brenner, L. G. Fernandes, P. Fernandes, and A. Sales. Performance models for master/slave parallel programs. *Electronic Notes in Theoretical Computer Science*, 128(4) :101 – 121, 2005.
- [16] M. Ben-Akiva, H. Koutsopoulos, C. Antoniou, and R. Balakrishna. Traffic simulation with DynaMIT. In *Fundamentals of Traffic Simulation*, volume 145, pages 363–398. Springer New York, New York, NY, 2010.
- [17] M. Ben-Akiva, H. Koutsopoulos, R. Mishalani, and Q. Yang. Simulation laboratory for evaluating dynamic traffic management systems. *Journal of Transportation Engineering*, 123(4) :283–289, 1997.
- [18] A. Benoit, L. Brenner, P. Fernandes, B. Plateau, and W. J. Stewart. The PEPS software tool. In P. Kemper and W. H. Sanders, editors, *Computer Performance Evaluations, Modelling Techniques and Tools. 13th International Conference, TOOLS 2003, Urbana, IL, USA, September 2-5, 2003, Proceedings*, volume 2794 of *Lecture Notes in Computer Science*, pages 98–115. Springer, 2003.
- [19] R. Blumofe. *Executing multithreaded programs efficiently*. PhD thesis, Massachusetts Institute of Technology, 1995.
- [20] R. Blumofe and C. Leiserson. Scheduling multithreaded computations by work stealing. *Journal of ACM*, 46(5) :720–748, 1999.
- [21] L. Brenner. *Réseaux d’Automates Stochastiques : Analyse transitoire en temps continu et algèbre tensorielle pour une sémantique en temps discret*. Theses, Institut National Polytechnique de Grenoble - INPG, Sept. 2009.
- [22] L. Brenner, R. Ammour, and I. Demongodin. Simuleau : a tool for hybrid and batches Petri nets. In *17th IFAC Workshop on Discrete Event Systems, WODES, To appear*, 2024.
- [23] L. Brenner, C. A. F. De Rose, and P. Fernandes. An Analytical Model to Evaluate the Performance of Cluster Architectures. In *3rd LCI International Conference on Linux Clusters : The HPC Revolution*, Saint Petersburg, USA, 2002.
- [24] L. Brenner, L. G. Fernandes, P. Fernandes, and A. Sales. Performance Analysis Issues for Parallel Implementations of Propagation Algorithm. In *Proceedings of the 15th Symposium on Computer Architecture and High Performance Computing*, pages 183–190, São Paulo, November 2003.
- [25] L. Brenner, P. Fernandes, J.-M. Fourneau, and B. Plateau. Modelling grid5000 point availability with san. *Electronic Notes in Theoretical Computer Science*, 232 :165–178, 2009. Proceedings of the Third International Workshop on the Practical Application of Stochastic Modelling (PASM 2008).

- [26] L. Brenner, P. Fernandes, B. Plateau, and I. Sbeity. Peps 2007 - stochastic automata networks software tool. In *4th International Conference on the Quantitative Evaluation of SysTems (QEST) 2007*, pages 163–164, Edimbourg, UK, sep 2007.
- [27] L. Brenner, P. Fernandes, and A. Sales. MQNA - Markovian Queueing Networks Analyser. In *11th IEEE/ACM International Symposium on Modelling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS'03)*, pages 194–199, Orlando, Florida, October 2003. IEEE Press.
- [28] L. Brenner, S. Nemmour, and I. Sbeity. Modélisation et évaluation de performance d'un algorithme de vol de travail sur des architecture multi-coeurs. In *10^{ème} Conférence Internationale de Modélisation, Optimisation et Simulation (MOSIM)*, pages 1–10, 2014.
- [29] C. Buisson and J. Lesort. *Comprendre le trafic routier*. Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publique (CERTU), 2010.
- [30] C. Canudas de Wit. Best-effort highway traffic congestion control via variable speed limits. In *the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5959–5964, 2011.
- [31] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1029–1038. ACM, 2010.
- [32] J. S. Coleman, E. Katz, and H. Menzel. *Medical Innovation : A Diffusion Study*. Bobbs-Merrill, Indianapolis, 1966.
- [33] C. Daganzo. The cell transmission model : A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B : Methodological*, 28(4) :269–287, 1994.
- [34] R. David and H. Alla. *Discrete, continuous, and hybrid Petri nets*, volume 1. Springer, 2010.
- [35] I. Demongodin. Generalised batches Petri net : hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Systems*, 11(1-2) :137–162, 2001.
- [36] I. Demongodin. Modeling and analysis of transportation networks using batches Petri nets with controllable batch speed. In *Int. Conf. on Applications and Theory of Petri Nets*, pages 204–222, 2009.
- [37] I. Demongodin, N. Audry, and F. Prunet. Batches petri nets. In *Proceedings of IEEE Systems Man and Cybernetics Conference - SMC*, volume 1, pages 607–617 vol.1, 1993.
- [38] I. Demongodin and A. Giua. Linear programming techniques for analysis and control of batches Petri nets. *IFAC Proceedings Volumes*, 43(12) :1–6, 2010. 10th WODES.
- [39] I. Demongodin and A. Giua. Stationary behavior of controlled generalized batches Petri nets. *IFAC Proceedings Volumes*, 45(29) :54–60, 2012.
- [40] I. Demongodin and A. Giua. Dynamics and steady state analysis of controlled generalized batches Petri nets. *Nonlinear Analysis : Hybrid Systems*, 12 :33–49, 2014.
- [41] I. Demongodin and F. Prunet. Extension of hybrid Petri nets to accumulation systems. In *IMACS Int. Symp. on Mathematical Modelling and Scientific Computing*, 1992.

- [42] X. Deng, Y. Dou, T. Lv, and Q. V. H. Nguyen. A novel centrality cascading based edge parameter evaluation method for robust influence maximization. *IEEE Access*, 5 :22119–22131, 2017.
- [43] A. Di Febbraro, D. Giglio, and N. Sacco. Urban traffic control structure based on hybrid Petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 5(4) :224–237, 2004.
- [44] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66. ACM, 2001.
- [45] P. Fernandes. *Méthodes numériques pour la solution de systèmes markoviens à grand espace d'états*. PhD thesis, Institut National Polytechnique de Grenoble, 1998.
- [46] P. Fernandes, B. Plateau, and W. J. Stewart. Efficient descriptor - Vector multiplication in Stochastic Automata Networks. *Journal of the ACM*, 45(3) :381–414, 1998.
- [47] M. Frigo, C. Leiserson, and K. Randall. The implementation of the cilk-5 multithreaded language. In *ACM SIGPLAN conference on programming language design and implementation*, volume 33, pages 212–223, 1998.
- [48] R. Gaddouri. *Réseaux de Petri Lots Triangulaires pour la modélisation mésoscopique et l'étude de la congestion dans le trafic routier*. PhD thesis, Aix-Marseille University, France, 2015.
- [49] R. Gaddouri, L. Brenner, and I. Demongodin. Simulation of traffic road by batches Petri nets as hybrid mesoscopic models. In *6th Int Workshop on Applied Modeling and Simulation (WAMS)*, pages 1–10, 2013.
- [50] R. Gaddouri, L. Brenner, and I. Demongodin. Mesoscopic event model of highway traffic by batches Petri nets. *IFAC Proceedings Volumes*, 46(24) :317–324, 2013. 6th IFAC Conference on Management and Control of Production and Logistics (MCPL).
- [51] R. Gaddouri, L. Brenner, and I. Demongodin. Extension of batches Petri nets by bi-parts batch places. In *International Workshop on Petri Nets for Adaptive Discrete-Event Control Systems (ADECS)*, pages 83–102, 2014.
- [52] R. Gaddouri, L. Brenner, and I. Demongodin. Modélisation et simulation du trafic routier par réseaux de Petri lots triangulaires. In *10ème Conférence Internationale de Modélisation, Optimisation et Simulation (MOSIM)*, pages 1–11, Nancy, France, 2014.
- [53] R. Gaddouri, L. Brenner, and I. Demongodin. Controlled triangular batches Petri nets for hybrid mesoscopic modeling of traffic road networks under vsl control. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 427–432. IEEE, 2016.
- [54] T. Gautier, X. Besseron, and L. Pigeon. Kaapi : A thread scheduling runtime system for dataflow computations on a cluster of multi-processors. In *International Workshop on Parallel Symbolic Computation*, pages 15–23, 2007.
- [55] A. Giua and M. Silva. Modeling, analysis and control of discrete event systems : a Petri net perspective. *IFAC-PapersOnLine*, 50(1) :1772–1783, 2017.
- [56] J. Goldenberg, B. Libai, and E. Muller. Talk of the network : A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3) :211–223, 2001.

- [57] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++ : optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.
- [58] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6) :1420–1443, 1978.
- [59] B. Greenshields, J. Bibbins, W. Channing, and H. Miller. A study of traffic capacity. *Highway research board proceedings*, 1935(14) :448–477, 1935.
- [60] K. Hamada, R. Ammour, L. Brenner, and I. Demongodin. Attack synchronizing sequence computation for output synchronized Petri nets with multiple deadlocks. In *20th IEEE International Conference on Networking, Sensing and Control*, pages 1–6, 2023.
- [61] S. Hoogendoorn and P. H. L. Bovy. State-of-the-art of vehicular traffic flow modelling. In *Delft University of Technology, Delft, The*, pages 283–303, 2001.
- [62] R. Jayakrishnan, H. Mahmassani, and T. Hu. An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research Part C : Emerging Technologies*, 2(3) :129–147, 1994.
- [63] J. Júlvez and R. Boel. Modelling and controlling traffic behaviour with continuous Petri nets. In *the 16th IFAC World Congress*, pages 02454–1, 2005.
- [64] J. Julvez, L. Recalde, and M. Silva. Deadlock-freeness analysis of continuous mono-t-semiflow petri nets. *IEEE Transactions on Automatic Control*, 51(9) :1472–1481, 2006.
- [65] S. Kakutani. A generalization of brouwer’s fixed point theorem. *Duke Mathematical Journal*, 8(3) :457–459, 1941.
- [66] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146. ACM, 2003.
- [67] D. Kempe, J. Kleinberg, and E. Tardos. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*, pages 1127–1138. Springer, 2005.
- [68] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4) :105–147, 2015.
- [69] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 259–271. Springer, 2006.
- [70] D. Krackhardt. Cognitive social structures. *Social Networks*, 9(2) :109–134, 1987.
- [71] J. LeBail, H. Alla, and R. David. *Hybrid Petri Nets*, volume 11. Springer-Verlag, 2nd edition, 1991.
- [72] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429. ACM, 2007.
- [73] M. J. Lighthill and G. B. Whitham. On kinetic waves : Ii) a theory of traffic flow on long crowded roads. In *Proc. Royal Society A229*, pages 281–345, 1955.

- [74] R. Liu, R. Ammour, L. Brenner, and I. Demongodin. Algorithm for controlling the transient behavior of controlled generalized batches Petri nets. In *12ème Colloque sur la Modélisation des Systèmes Réactifs (MSR)*, pages 1–2, 2019.
- [75] R. Liu, R. Ammour, L. Brenner, and I. Demongodin. Event-driven control for reaching a steady state in controlled generalized batches Petri nets. *IFAC-PapersOnLine*, 53(4) :180–186, 2020. 15th IFAC Workshop on Discrete Event Systems WODES 2020 — Rio de Janeiro, Brazil, 11-13 November 2020.
- [76] R. Liu, R. Ammour, L. Brenner, and I. Demongodin. On/off control trajectory computation for steady state reaching in batches Petri nets. In *International Conference on Verification and Evaluation of Computer and Communication Systems*, pages 84–99. Springer, 2020.
- [77] R. Liu, R. Ammour, L. Brenner, and I. Demongodin. On/off control for reaching a steady state attractive region in batches petri nets. *IFAC-PapersOnLine*, 56(2) :9618–9623, 2023. 22nd IFAC World Congress.
- [78] M. Moalla, J. Pulou, and J. Sifakis. Synchronized Petri nets : A model for the description of non-autonomous systems. In J. Winkowski, editor, *Mathematical Foundations of Computer Science 1978*, pages 374–384. Springer Berlin Heidelberg, 1978.
- [79] K. Nagel. Particle hopping models and traffic flow theory. *Physical Review E*, 53(5) :4655–4672, 1996.
- [80] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12) :2221–2229, 1992.
- [81] K. Ng, M. Reaz, and M. Ali. A review on the applications of Petri nets in modeling, analysis, and control of urban traffic. *Intelligent Transportation Systems*, 14(2) :858–870, 2013.
- [82] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12) :2043–2067, December 2003.
- [83] P. Pattison, S. Wasserman, G. Robins, and A. M. Kanfer. Statistical evaluation of algebraic constraints for social networks. *Journal of Mathematical Psychology*, 44(4) :536–568, 2000.
- [84] H. J. Payne. Models of freeway traffic and control. In *Mathematical Models of Public Systems*, volume 1, pages 51–61, 1971.
- [85] B. Plateau. *L'évaluation du parallélisme et de la synchronisation*. PhD thesis, Paris-Sud Orsay, 1984.
- [86] B. Plateau and J.-M. Fourneau. A methodology for solving markov models of parallel systems. *Journal of Parallel and Distributed Computation*, 12(4) :370–387, 1991.
- [87] M. Poggi, I. Demongodin, N. Giambiasi, and A. Giua. Testing experiments on synchronized Petri nets. *IEEE Trans. on Automation Science and Engineering*, 11(1) :125–138, 2014.
- [88] P. Richards. Shock waves on the highway. *Operations Research*, 4(1) :42–51, 1956.
- [89] B. Ryan and N. C. Gross. The diffusion of hybrid seed corn in two iowa communities. *Rural Sociology*, 8(1) :15, 1943.

- [90] E. Schechter. *Handbook of Analysis and Its Foundations*. Academic Press, 1996.
- [91] T. C. Schelling. *Micromotives and Macrobehavior*. W.W. Norton & Company, 2006.
- [92] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.
- [93] N. Taylor. The contrans dynamic traffic assignment model. *Networks and Spatial Economics*, 3(3) :297–322, 2003.
- [94] M. Tchiboukdjian, V. Danjean, T. Gautier, F. Mentec, and B. Raffin. A work stealing scheduler for parallel loops on shared cache multicores. In *Euro-Par 2010*, volume 6586 of *Lecture Notes in Computer Science (LNCS)*, pages 99–107, 2011.
- [95] M. Tchiboukdjian, N. Gast, D. Trystam, J.-L. Roch, and J. Bernard. A tighter analysis of work stealing. In *Algorithms and Computation*, volume 6507 of *Lecture Notes in Computer Science (LNCS)*, pages 291–302, 2010.
- [96] C. Tolba, D. Lefebvre, P. Thomas, and A. El Moudni. Continuous and timed Petri nets for the macroscopic and microscopic traffic flow modelling. *Simulation Modelling Practice and Theory*, 13(5) :407–436, 2005.
- [97] R. Van Nieuwpoort, J. Maassen, T. Kielmann, and H. E. Bal. Satin : Simple and efficient java-based grid programming. *Journal for Parallel and Distributed Computing*, 65(3) :19–32, 2005.
- [98] C. R. Vázquez and M. Silva. Timing and liveness in continuous petri nets. *Automatica*, 47(2) :283–290, 2011.
- [99] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time control for structurally persistent continuous petri nets. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2771–2776, 2010.
- [100] H. Wu, J. Shang, S. Zhou, Y. Feng, B. Qiang, and W. Xie. Laim : A linear time iterative approach for efficient influence maximization in large-scale networks. *IEEE Access*, 6 :44221–44234, 2018.
- [101] W. Yang. *Influence Maximization in Social Networks*. PhD thesis, 2019. Thèse de doctorat dirigée par Leonardo Brenner et Alessandro Giua, Automatique Aix-Marseille 2019.
- [102] W. Yang, L. Brenner, and A. Giua. Computation of activation probabilities in the independent cascade model. In *5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 791–797. IEEE, 2018.
- [103] W. Yang, L. Brenner, and A. Giua. Influence maximization by link activation in social networks. In *23rd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2018, Torino, Italy, September 4-7, 2018*, pages 1248–1251. IEEE, 2018.
- [104] W. Yang, L. Brenner, and A. Giua. Influence maximization in independent cascade networks based on activation probability computation. *IEEE Access*, 7 :13745–13757, 2019.
- [105] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad. On approximation of real-world influence spread. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 548–564. Springer, 2012.

Annexe A

Curriculum Vitae

A.1 Identification

Nom	BRENNER
Prénom	Leonardo
Date de naissance	05 Octobre 1978
Adresse professionnelle	Laboratoire d'Informatique en Calcul Intensif et Image pour la Simulation (LICIIS) Université de Reims Champagne-Ardenne UFR Sciences Exactes et Naturelles Département Informatique 51100 Reims
Email	leonardo.brenner@univ-reims.fr

A.2 Parcours professionnel

09/2023 – 08/2027	Chaire Professeur Junior Laboratoire d'Informatique en Calcul Intensif et Image pour la Simulation (LICIIS) UFR Sciences Exactes et Naturelles Université de Reims Champagne-Ardenne
09/2010 – ...	Maître de Conférences Équipe Modèles et Formalismes à Évènements Discrets (MoFED) Laboratoire d'Informatique & Systèmes (LIS) - à partir de 2018 Laboratoire des Sciences de l'Information et des Systèmes (LSIS) - 2010-2017 Faculté des Sciences

	Aix-Marseille Université
09/2009 – 08/2010	ATER à temps plein Équipe Systèmes Communicants (SysCom) Centre de Recherche en STIC (CresTIC) UFR Sciences Exactes et Naturelles Université de Reims Champagne-Ardenne
09/2008 – 08/2009	ATER à mi-temps Équipe Middleware efficiently scalable (MESCAL) Laboratoire d'Informatique de Grenoble (LIG) UFR Informatique, Mathématiques et Mathématiques Appliquées Université Joseph Fourier
09/2006 – 01/2007	Enseignant vacataire Département Licence, Sciences et Technologies Université Joseph Fourier
09/2003 – 08/2004	Administrateur de systèmes Projet CASCO Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) Porto Alegre, Brésil
08/2003 – 12/2003	Enseignant vacataire Faculdade de informática Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) Porto Alegre, Brésil

A.3 Formation

09/2004 – 09/2009	Doctorat en Informatique Institut Polytechnique de Grenoble (Grenoble INP), Grenoble, France École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique Laboratoire d'Informatique de Grenoble
-------------------	--

—	<ul style="list-style-type: none"> — Directrice : Mme. Brigitte Plateau — Date de soutenance : 16 septembre 2009 — Lieu : Laboratoire d’Informatique de Grenoble — Rapporteurs : <ul style="list-style-type: none"> — Leïla Kloul (Maître de Conference avec HDR, Prism, Université de Versailles) — Patrice Moreaux (Professeur, Listic, Université de Savoie) — Membres du jury : <ul style="list-style-type: none"> — Andrzej Duda (Professeur, Grenoble INP) — Leïla Kloul (Maître de Conference avec HDR, Université de Versailles) — Patrice Moreaux (Professeur, Université de Savoie) — Brigitte Plateau (Professeur, Grenoble INP) — Jean-Marc Vincent (Maître de Conference, Université Joseph Fourier) — Titre de la thèse : “Réseaux d’Automates Stochastiques : Analyse transitoire en temps continu et Algèbre tensorielle pour une sémantique en temps discret”
02/2002 – 08/2003	<p>Master en Informatique (Msc.) Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Brésil Institut d’Informatique</p> <ul style="list-style-type: none"> — Directeur : M. Paulo Fernandes — Titre ^a : “Méthodes d’agrégation pour le formalisme de Réseaux d’Automates Stochastiques” <hr style="width: 20%; margin-left: 0;"/> <p><i>a.</i> Titre original en portugais : Agregação em Redes de Autômatos Estocásticos</p>
03/1997 – 12/2001	<p>Licence en Informatique (Bsc. / BAC+4) Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Brésil Institut d’Informatique</p>

A.4 Charges collectives

A.4.1 Responsabilité de formation

2022 – 2023	<p>Co-porteur de la licence informatique Co-porteur de la licence informatique pour l’habilitation de 2024 UFR Sciences Aix-Marseille Université</p>
-------------	--

2018 – 2023	Responsable d'année Responsable de la deuxième année de la licence informatique du site Aix-Montperrin UFR Sciences Aix-Marseille Université
-------------	--

A.4.2 Administration de la science et vie de l'unité

2020 – 2022	Membre commission de contrats doctoraux de l'École Doctorale Mathématique et Informatique Laboratoire d'Informatique et Systèmes Aix-Marseille Université
2020 et 2021	Commissions de sélection MCF à l'Aix-Marseille Université MCF section CNU 27 en 2021 (poste 966) MCF section CNU 27 en 2020 (poste 895)
2018 – 2021	Commissions Ad-hoc des Attachés Temporaires d'enseignement et Recherche Plusieurs commissions ad-hoc pour les postes ATER en section CNU 27 au Département Informatique et Interactions

Annexe B

Activités liées à l'enseignement

Dans cette annexe, je détaille mes activités pédagogiques depuis mon recrutement à Aix-Marseille Université à l'UFR Sciences. J'ai exercé mes activités dans cette université de 2010 à 2023. Je suis actuellement (depuis Septembre 2023) en détachement à l'Université de Reims Champagne-Ardenne sur un contrat de Chaire Professeur Junior avec charge d'enseignement de 64 heures EqTD. Mon nouveau poste étant trop récent, dans la suite de cette annexe, je ne décrirai pas mes activités pédagogiques au sein de l'Université de Reims Champagne-Ardenne. Les enseignements détaillés ci-après concernent principalement la période de ma carrière comprise entre 2010 à 2023.

Certains cours m'ont suivi tout au long de ma carrière, comme c'est le cas de l'UE de *Systèmes d'exploitation* et d'*Architecture des ordinateurs*. D'autres se sont ajoutés au cours des années. J'ai toujours pour objectif d'assurer non seulement les cours magistraux mais au moins un groupe de TD et de TP pour mieux suivre l'évolution des étudiants. Mais cela engendre parfois beaucoup d'heures complémentaires à mon service.

L'organisation de cette annexe commence par une brève description de mes responsabilités administratives (Section [B.1](#)) dans l'enseignement. La section [B.2](#) présente les matières enseignées (une brève description, le public visé et le volume horaire) ainsi qu'un tableau récapitulatif du volume horaire enseigné sur la période 2010-2023. La section [B.3](#) liste les encadrements de stages et de Travaux d'Étude et Recherche (TER) en licence et en première année de master.

B.1 Responsabilités administratives

La prise de responsabilités administratives est venue plus tard avec plus de maturité et de connaissance du système pédagogique de l'université.

Le tableau ci-après décrit mes responsabilités administratives au sein d'Aix-Marseille Université.

2022 – 2023	<p>Co-porteur de la licence informatique</p> <ul style="list-style-type: none"> — <i>Activités</i> : le travail de portage pour cette nouvelle habilitation consiste principalement dans la modernisation des pratiques pédagogiques en sens large, qui passe par un découpage en bloc de compétences, la mise en place d'évaluations par compétences, et plus largement une réflexion sur l'évaluation des étudiants ainsi que sur nos méthodes pédagogiques, une réévaluation des distributions horaires (découpage CM/TD/TP mais également rythmes semestriels), ... — <i>Effectif concerné</i> : \approx 500 étudiants en licence L2 et L3 informatique divisés sur 5 parcours et 4 sites d'enseignements.
2018 – 2023	<p>Responsable pédagogique de la licence L2 informatique, parcours informatique du site d'Aix-Montperrin</p> <ul style="list-style-type: none"> — <i>Activités</i> : organisation de l'équipe pédagogique, organisation de l'emploi de temps, réalisation des jurys de semestres et d'année, l'analyse des dossiers de candidatures externes telles que <i>Études en France</i> (500 – 1000 dossiers par an) et <i>eCandidat</i> (\approx 200 dossiers par an). — <i>Effectif concerné</i> : \approx 70 étudiants divisés en 2 groupes de TD/TP, \approx 20 enseignants

B.2 Matières enseignées

Je présente dans cette section une liste des matières que j'ai enseignées sur la période de 2010 à 2023.

2021 – 2023	Master 2 IOT	Développement pour systèmes embarqués 2
2018 – 2023	Master 2 IOT	Développement pour systèmes embarqués 1
2018 – 2023	Master 1 EEEA	Systèmes à événement discret et commande supervisée
2018 – 2023	L2 informatique	Programmation C et système
2018 – 2023	L2 math-info	Systèmes d'exploitation
2015 – 2018	L2 informatique	Programmation UNIX 2
2015 et 2017	L3 SPI	Modèles à évènements discrets
2014 – 2018	L2 informatique	Programmation UNIX 1
2012 – 2017	M1 informatique	Modèles à évènements discrets
2012 – 2017	M2 Image et Systèmes	Formalismes à évènements discrets
2011 – 2011	L3 informatique	Traduction et sémantique
2011 – 2011	M1 informatique	Modélisation et simulation
2010 – 2023	L3 informatique	Systèmes d'exploitation
2010 – 2023	L2 informatique	Architecture des ordinateurs

2010 – 2010 | L1 Physique-Chimie | **Introduction à l'informatique**

En plus des cours mentionnés ci-avant, un Diplôme Inter-Universitaire (DIU) a été proposé durant l'année 2018/2019 pour former des enseignants à enseigner l'informatique au lycée au moment de la mise en place de la spécialité "Numérique et sciences informatiques" au baccalauréat. Je suis intervenu dans cette formation pour un total de 47 heures et un effectif au tour de 50 étudiants divisés en deux groupes. Plus précisément, j'ai enseigné dans l'UE "Architecture matérielle et robotique, systèmes et réseaux".

À cela s'ajoute encore une formation courte sur mesure, intitulée "**Développement de logiciel embarqué avec Zéphyr**", pour la société *STID* (<https://stid.com>). Cette formation de 36 heures distribuées sur 6 jours a été partagée avec Peter Niebert.

B.2.1 Volume horaire

Le total EqTD présenté dans le tableau ci-dessous n'est pas en compte les 24 – 28 heures, selon l'année, de prime de responsabilité pédagogique (PRP) en tant que responsable d'année. Ces heures s'ajoutent alors au total EqTD à partir de 2018.

Année	Total EqTD	Heures comp.	Heures licence	Heures master
2022/2023	296	104	230	66
2021/2022	321	129	221	100
2020/2021	266	74	211	55
2019/2020	300	108	225	75
2018/2019	264	72	254	10
2017/2018	336	146	306	30
2016/2017	297	105	270	27
2015/2016	263	71	235	28
2014/2015	278	86	262	16
2013/2014	262	70	254	8
2012/2013	212	20	204	8
2011/2012	230	38	210	20
2010/2011	128 ¹	00	128	0

B.3 Encadrement de stages et TER

En plus des cours dispensés en licence et master, j'ai encadré plusieurs stages et TER dont la liste est ci-après.

— **Hassen MHENNI et Bastein FERAUD** (2022) - *Loseamin*

Le stage réalisé par ces deux étudiants propose un modèle SAN pour les pronostics sportifs. Le logiciel développé dans le stage permet la récupération des données statistiques de chaque joueur, compose deux équipes et calcule la probabilité de victoire basée sur la possession du ballon par chaque équipe.

- **Yoann ANEZIN** (2019) - Étude du trafic routier avec le modèle de transmission cellulaire.

TIPE (Travail d'Initiative Personnelle encadré) en classe préparatoire en MP au Lycée Thiers de Marseille.

- **Vincent PIERRE, Lucas LOIGNON et Rémi DEUTSCH** (2018) - *New Tile*

Il s'agit d'une application web qui serait à mi-chemin entre les réseaux sociaux bien connus et les plateformes de créations de contenus avancés.

- **Vincent HUYNH - Florent MARTIN - Marvin VAUGE** (2017) - *SIMSE*

Les étudiants ont développé une interface graphique pour un simulateur de système d'exploitation minimaliste utilisé dans l'UE de *Systèmes d'exploitation*.

- **Kandel ACHIRAFI, Ibrahima SEYE et Houmed HASSAN** (2015) - *SIMSAN*

SIMSAN est un simulateur graphique et interactif pour le formalisme de Réseaux d'Automates Stochastiques (SAN).

- **Jean-Baptiste BONARDO, Maxime LACROIX et Jérémy VAZ BORGES** (2014) - *Freeway LEGO*

Ce logiciel implémente une interface graphique pour permettre de construire des routes à partir de blocs de base. La route est convertie en RDPLOTS pour la simulation.

- **Hugo BOUNOUA, Soufiane MOUJIB, El Makki VOUNDY et Jesus ZAVARCE** (2012) - *JWIRE*

Ce logiciel implémente une interface graphique pour permettre de simuler des circuits logiques en Java.

À cela s'ajoute encore plusieurs suivis de stage en entreprise qui ne seront pas détaillés ici.

Les activités pédagogiques ont toujours été une partie importante dans ma carrière. Je trouve que les échanges réguliers avec les étudiants sont très enrichissants et m'ont permis de découvrir des nouvelles technologies, des nouvelles tendances et parfois même des nouvelles façons d'enseigner.

Annexe C

Activités liées à la recherche

Dans cette annexe je présente un résumé de mes activités liées à la recherche depuis mon arrivée au LIS. Cela comprend la période de 2010 à 2023.

Dans la suite de cette annexe (section [C.1](#)), je détaille les encadrements au niveau de doctorat et en deuxième année de master. La section [C.2](#) présente mes participations à des projets. Pour terminer, une liste de mes publications est fournie dans la section [C.3](#).

C.1 Encadrements

C.1.1 Encadrement doctoral

- **Antoine THEBAULT** (2024-...) - *Visualisation interactive de séquences temporelles de volumes à l'aide de représentations neuronales*. Co-direction : Laurent Lucas (40%), Stéphanie Prevost (30%) et Leonardo Brenner (30%).

Sujet : Ce travail de recherche s'inscrit dans les ambitions de la CPJ End-to-end Synthesis Modelling Infrastructure simulation for Environmental Data Science (OSMIUM). L'enjeu principal de ces deux développements est ainsi de construire une simulation temporelle interactive, en combinant la visualisation de l'information, volumique notamment, l'apprentissage automatique et les environnements virtuels.

Financement : ANR CPJ OSMIUM

- **Khalid HAMADA** (2022-...) - *Formalismes de réseaux de Petri pour l'analyse de la sécurité des systèmes cyber-physiques*. Co-direction : Isabel Demongodin (40%), Rabah Ammour (30%) et Leonardo Brenner (30%). Je me suis retiré de l'encadrement de thèse en Septembre 2023 après mon détachement à l'Université de Reims pour le poste de Chaire Professeur Junior.

Sujet : L'objectif de cette thèse est le développement de nouveaux outils théoriques

pour l'analyse des Réseaux de Petri synchronisés avec sortie (OutSynPNs). Deux types d'analyses seront étudiées : (1) la détection de cyber-attaques dans des systèmes cyber-physiques (CPS) modélisés par OutSynPNs et, (2) développement des stratégies de défense par l'exploration de séquences de synchronisation. Les méthodes développées seront appliquées aux CPS et particulièrement aux systèmes de contrôle en réseau pour les processus manufacturés.

Financement : École doctorale MI - Mathématiques et Informatique de Marseille

Publications : 1 conférence internationale [60].

- **Ruotian LIU** (2018-2021) - *Contributions to event driven control of batches Petri nets for steady state reaching*. Co-direction : Isabel Demongodin (40%), Rabah Ammour (30%) et Leonardo Brenner (30%).

Sujet : Le problème considéré dans cette thèse porte sur la commande de systèmes hybrides étudiés du point de vue de la théorie des systèmes à événements discrets. L'objectif était de déterminer une trajectoire transitoire pour atteindre un état donné à partir d'un état initial dans un système modélisé par un réseau de Petri.

Financement : China Scholarship Council (CSC)

Publications : 3 conférences internationales [77, 75, 76], 1 résumé en conférence nationale [74].

Poste actuel : Assistant professor à Université Polytechnique de Bari, Bari, Italie.

- **Wenjing YANG** (2016-2019) - *Influence maximization in social networks*. Co-direction : Alessandro Giua (50%) et Leonardo Brenner (50%).

Sujet : Cette thèse a exploré la notion de propagation d'influence dans le réseaux sociaux par le modèle de cascade indépendante. Ce type de modèle suppose qu'un individu adopte une innovation avec une certaine probabilité si au moins un de ses voisins l'a adoptée. La thèse aborde l'estimation de l'influence, la maximisation de l'influence par la sélection des diffuseurs initiaux, et la maximisation de l'influence par l'activation des liens.

Financement : China Scholarship Council (CSC)

Publications : 1 revue internationale [104], 2 conférences internationales [102, 103].

Poste actuel : Associate professor à Northwest University, Xi'an, Chine.

- **Radhia GADDOURI** (2012-2015) - *Réseaux de Petri Lots Triangulaires pour la modélisation mésoscopique et l'étude de la congestion dans le trafic routier*. Co-direction : Isabel Demongodin (50%) et Leonardo Brenner (50%).

Sujet : Le problème considéré dans cette thèse est la modélisation et la simulation du trafic routier au niveau mésoscopique comme un système à événements discrets. Une extension du formalisme Réseaux de Petri Lots (RdPPLots), appelée RdPLots Triangulaire, est proposée pour la représenter des retards variables au niveau des flux continus liés aux phénomènes du trafic routier tels que la congestion et décongestion.

Financement : ressources propres + ATER

Publications : 5 conférences internationales [49, 50, 51, 52, 53].

Poste actuel : Responsable des formations par alternance à ESIEA - École d'ingénieur.e.s d'un numérique utile, Paris, France.

C.1.2 Encadrement stage M2

- **Khalid HAMADA** (2022) - *Petri net synchronizing sequences for the security of cyber-physical systems*. Co-direction Rabah Ammour (50%) et Leonardo Brenner (50%). L'étudiant a continué en thèse à la fin de son stage.
- **Ruotian LIU** (2018) - *Consistency of diagnosability of composed net systems via buffer places*. Co-direction Isabel Demongodin (50%) et Leonardo Brenner (50%). L'étudiant a continué en thèse à la fin de son stage.
- **Nabil ADRAR** (2016) - *Étude sur la modélisation et la gestion de mouvements de foules par les Réseaux de Petri Lots Triangulaires*.
- **Youcef HAMDAD** (2013) - *Modélisation et simulation du trafic routier par Réseaux de Petri Lots*.
- **Sarah NEMMOUR** (2012) - *Évaluation de performances d'algorithmes de vol de travail sur des architectures multiprocesseurs multi-coeurs*.
Publication : 1 conférence internationale [28].
- **Iheb MAHMOUDI** (2011) - *Parallélisation d'une méthode de multiplication vecteur-descripteur*. Co-direction Isabel Demongodin (50%) et Leonardo Brenner (50%).

C.2 Rayonnement et diffusion scientifique

C.2.1 Participation à des projets

Pendant ces 4 dernières années, j'ai participé à plusieurs projets de recherche nationaux et internationaux :

- **ANR-PRC : MENACES** (Timed Event-Based Methods for Networked Control Systems Security) 2023-2027. Porteur : Isabel Demongodin. Partenaires : Université Le Havre Normandie, Aix-Marseille Université, Université Sorbonne Paris-Nord. Budget total : 331K euros, budget LIS : 184K euros

Les systèmes de contrôle en réseau (NCS) sont une forme particulière de systèmes cyber-physiques dans lesquels le système à contrôler et les différents composants tels que les actionneurs, les contrôleurs et les capteurs sont répartis physiquement, tandis que la communication entre ces composants est réalisée à l'aide d'un réseau de communication numérique partagé par d'autres applications et systèmes de contrôle. Le projet se concentre sur la détection des attaques furtives contre les systèmes de contrôle en réseau, dans lesquels l'intrus est supposé pouvoir modifier, supprimer ou ajouter des ordres de

contrôle ou des rapports de capteurs, dans le but d'altérer le fonctionnement du processus ou de tromper ses utilisateurs.

- **ANR-SIOMRI : APPRENTIS** (Optimisation par APPROche foRMelle Et iTelligence arTificielle des moyens mobiles de Surveillance et d'intervention sur un site à risque) 2021 - 2023. Porteur : Dimitri Lefebvre. Partenaires : Université Le Havre Normandie, IMT Lille Douai, Aix-Marseille Université, Université Sorbonne Paris-Nord. Budget total : 148K euros, budget LIS : 33K euros

Le projet APPRENTIS concerne la sûreté de sites industriels ou portuaires présentant des risques. L'objectif opérationnel est de proposer un outil logiciel d'aide à la décision pour planifier les missions de surveillance et d'intervention. Cet outil minimisera le coût de configuration et de mise en oeuvre des missions réalisées par des agents mobiles (drones ou véhicules automatisés). L'originalité de la méthode proposée pour résoudre ces problèmes repose sur une modélisation du problème et une méthodologie de résolution issues des systèmes à événements discrets et de l'intelligence artificielle. Le rôle de l'intelligence artificielle dans ce projet est la détection des situations anormales à partir des données récoltées par les agents mobiles.

- **INS2I : CPSecurity I et II** (Event-Based Methods and Formalisms for Security Analysis of Cyber-Physical Systems) 2020 et 2021. Porteur : Rabah Ammour. Partenaires : Université Le Havre Normandie, Aix-Marseille Université, Université Sorbonne Paris-Nord. Budget total : $2 \times 5K$ euros

Un Systèmes Cyber-Physique (Cyber-Physical System ou CPS) est un système complexe dans lequel des éléments informatiques collaborent pour le contrôle et la commande d'entités physiques. Le CPS traite des données issues de son environnement et agit sur le processus physique au travers d'une boucle de contrôle. L'objectif de ce projet était donc de s'intéresser à la modélisation des CPS avec des approches basées sur les Systèmes à Événements Discrets et au développement de nouvelles méthodes permettant de détecter les problèmes de sécurité auxquels fait face ce type de systèmes.

- **PHC-Cai Yuanpei : IMEBS** (Identification Methods of Event-Based Systems) 2017-2019. Porteur : Isabel Demongodin. Partenaires : Aix-Marseille Université, Xidian University. Budget LIS : 63K euros

Ce projet explore les problèmes classiques d'identification dans la théorie de systèmes. Autrement dit, il s'agit de définir un modèle où l'observation des signaux d'entrée et de sortie d'un système s'approche de celui du modèle proposé. Trois thèmes ont été abordé dans ce projet : 1) Identification des défauts basée sur des modèles d'événements discrets ; 2) Identification de l'état dans des modèles à événements discrets ; 3) Identification de modèles de diffusion pour les réseaux sociaux.

C.2.2 Membre de jurys et commissions

2022 – 2022	Membre de la commission d'experts au concours "Prix de thèse URCA 2022" Université de Reims Champagne-Ardenne (URCA)
-------------	--

2020 – 2022	Membre de la commission de contrats doctoraux de l'École Doctorale Mathématique et Informatique Laboratoire d'Informatique et Systèmes Aix-Marseille Université
2020 et 2021	Commissions de sélection MCF à l'Aix-Marseille Université MCF section CNU 27 en 2021 (poste 966) MCF section CNU 27 en 2020 (poste 895)
2018 – 2021	Commissions Ad-hoc des Attachés Temporaires d'enseignement et Recherche Plusieurs commissions ad-hoc pour les postes ATER en section CNU 27 au Département Informatique et Interactions

C.2.3 Membre des comités d'organisation

2017	MSR2017 Membre du comité d'organisation de la conférence MSR (Modélisation des Systèmes Réactifs) Novembre 2017, Marseille, France
------	--

C.3 Liste de publications

C.3.1 Chapitre de livres

1. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *Avaliação de Desempenho de Sistemas Paralelos. 4a Escola Regional de Alto Desempenho (ERAD 2004)*. Porto Alegre, 2004, v. 1, p. 97-120. (langue : portugais)

C.3.2 Revues internationales avec comité de lecture

1. Sara HSAINI, Rabah AMMOUR, Leonardo BRENNER, My El Hassan CHARAF, Isabel DEMONGODIN, Dimitri LEFEBVRE. *A Reconfiguration Method for Muti-Robot Monitoring Patrols*. **Cybernetics and Systems**, pp. 1-26, 2023. [hal-04223793](#)
2. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Robust stealthy attacks based on uncertain costs and labeled finite automata with inputs*. **IEEE Robotics and Automation Letters (RA-L)**, vol.8, issue 5, pp. 2732-2739, 2023. [hal-04202802](#)
3. Wenjing YANG, Leonardo BRENNER, Alessandro GIUA. *Influence Maximization in Independent Cascade Networks Based on Activation Probability Computation*. **IEEE Access**, vol. 7, pp. 13745-13757, 2019. [hal-02373686](#)

4. Ihab SBEITY, Leonardo BRENNER, Mohamed DBOUK. *Generating a Performance Stochastic Model from UML Specifications*. **International Journal of Computer Science Issues**, vol. 8, pp. 13-21, 2011. doi.org/10.48550/arXiv.1202.0414
5. Leonardo BRENNER, Paulo FERNANDES, Jean-Michel FOURNEAU, Brigitte PLATEAU. *Modelling Grid5000 point availability with SAN*. **Electronic Notes In Theoretical Computer Science**, vol. 232, pp. 165-178, 2009. doi.org/10.1016/j.entcs.2009.02.056
6. Ihab SBEITY, Leonardo BRENNER, William STEWART, Brigitte PLATEAU. *Phase-type distributions in Stochastic Automata Networks*. **European Journal of Operational Research**, vol. 186, issue 3, pp. 1008-1028, 2008. doi.org/10.1016/j.ejor.2007.02.019
7. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *The Need for and the Advantages of Generalized Tensor Algebra for Structured Kronecker Representations*. **International Journal Of Simulation Systems Science Technology**, vol. 6, issue 3-4, pp. 52-60, 2005.
8. Lucas BALDO, Leonardo BRENNER, Luiz Gustavo FERNANDES, Paulo FERNANDES, Afonso SALES. *Performance Models for Master/Slave Parallel Programs*. **Electronic Notes In Theoretical Computer Science**, vol. 128, issue 4, pp. 101-121, 2005. doi.org/10.1016/j.entcs.2005.01.015
9. Anne BENOIT, Leonardo BRENNER, Paulo FERNANDES, Brigitte PLATEAU. *Aggregation of Stochastic Automata Networks with Replicas*. **Linear Algebra and its Applications**, vol 386, pp. 111-136, 2004. doi.org/10.1016/j.laa.2004.02.015

C.3.3 Actes publiés de conférences internationales, congrès et colloques

1. Leonardo BRENNER, Rabah AMMOUR, Isabel DEMONGODIN. *Simuleau : a tool for hybrid and batches Petri nets*. **17th IFAC Workshop on Discrete Event Systems (WODES)**, pp. 1-6, 2024
2. Khalid HAMADA, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *Attack Synchronizing Sequence Computation for Output Synchronized Petri Nets with Multiple Deadlocks*. **20th IEEE International Conference on Networking, Sensing and Control (ICNSC)**, 2023. hal-04437199
3. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *ON/OFF control for reaching a steady state attractive region in batches Petri nets*. **IFAC World Congress 2023**, vol. 56, n. 2, pp. 9618-9623, 2023. hal-04437189
4. Sara HSAINI, Rabah AMMOUR, Leonardo BRENNER, My EL Hassan CHARAF, Isabel DEMONGODIN. *A Decentralized based Approach using Hybrid Filtered Beam Search Algorithm for Monitoring Patrols*. **9th International Conference on Control, Decision and Information Technologies (CoDIT)**, pp. 1436-1441, 2023. hal-04437212
5. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Observer Design for Labeled Finite Automata with Inputs under Stealthy Actuators Attacks*. **16th IFAC Workshop on Discrete Event Systems (WODES)**, pp. 1-6, 2022. hal-03879792

6. Feda ALMUHISEN, Nicolas DURAND, Leonardo BRENNER, Mohamed QUAFARFOU. *Sequential Patterns for Spatio-Temporal Traffic Prediction*. **20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)**, pp. 595-602, 2021. [hal-03480960](#)
7. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Costs analysis of stealthy attacks with bounded output synchronized Petri nets*. **IEEE 17th International Conference on Automation Science and Engineering (CASE)**, pp. 799-804, 2021. [hal-03517596](#)
8. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Observer Design for Bounded Output Synchronized Petri Nets*. **European Control Conference (ECC)**, pp. 746-751, 2021. [hal-03517554](#)
9. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *Event-Driven Control for Reaching a Steady State in Controlled Generalized Batches Petri Nets*. **15th IFAC Workshop on Discrete Event Systems (WODES)**, IFAC-PapersOnLine, vol. 53, issue 4, pp. 180-186, 2020. [hal-03171923](#)
10. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *ON/OFF Control Trajectory Computation for Steady State Reaching in Batches Petri Nets*. **14th International Conference on Verification and Evaluation of Computer and Communication Systems (VECOS)**, Lecture Notes in Computer Science, vol 12519, pp. 84-99, 2020. [hal-03171901](#)
11. Wenjing YANG, Leonardo BRENNER, Alessandro GIUA. *Influence Maximization by Link Activation in Social Networks*. **IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)**, pp.1248-1251, 2018. [hal-01997472](#)
12. Wenjing YANG, Leonardo BRENNER, Alessandro GIUA. *Computation of Activation Probabilities in the Independent Cascade Model*. **5th International Conference on Control, Decision and Information Technologies (CoDIT)**, pp.791-797, 2018. [hal-01997464](#)
13. Leonardo BRENNER, Isabel DEMONGODIN. *Controlled Triangular Batches Petri Nets : a Highway Case Study*. **29th European Modeling and Simulation Symposium (EMSS)**, pp. 541-545, 2017. [hal-01785943](#)
14. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Controlled Triangular Batches Petri Nets for hybrid mesoscopic modeling of traffic road networks under VSL control*. **IEEE 12th International Conference on Automation Science and Engineering (CASE)**, pp. 427-432, 2016. [hal-01997510](#)
15. Leonardo BRENNER, Sarah NEMMOUR, Ihab SBEITY. *Modélisation et évaluation de performance d'un algorithme de vol de travail sur des architecture multi-coeurs*. **10ème Conférence Internationale de Modélisation, Optimisation et Simulation (MOSIM)**, pp.1-10, 2014. [hal-01166686](#)
16. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Modélisation et simulation du trafic routier par Réseaux de Petri Lots Triangulaires*. **10ème Conférence Internationale de Modélisation, Optimisation et Simulation (MOSIM)**, pp.1-10, 2014. [hal-01166684](#)

17. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Extension of Batches Petri Nets by Bi-parts Batch Places*. **Int. Workshop on Petri Nets for Adaptive Discrete-Event Control Systems (ADECS)**, pp.83-102, 2014.
18. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Simulation of traffic road by Batches Petri nets as hybrid mesoscopic models*. **6th Int Workshop on Applied Modeling and Simulation (WAMS)**, pp.1-10, 2013.
19. Radhia GADDOURI, Leonardo BRENNER, Isabel DEMONGODIN. *Mesoscopic event model of highway traffic by Batches Petri nets*. **6th International Conference on Management and Control of Production and Logistics (MCPL)**, pp.317-324, 2013.
20. Leonardo BRENNER, Paulo FERNANDES, Brigitte PLATEAU, Ihab SBEITY. *PEPS 2007 - Stochastic Automata Networks Software Tool*. **Quantitative Evaluation of Systems (QEST 2007)**. Edinburgh. New York : IEEE Press, p. 72-74. 2007.
21. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES, Thais WEBBER. *A Framework to Decompose GSPN Models*. **26th International Conference on Applications and Theory of Petri Nets (ICATPN 2005)**, LNCS vol. 3536 Springer-Verlag. Miami, USA, p. 128-147. Juin 2005.
22. Cristiano BERTOLINI, Leonardo BRENNER, Paulo FERNANDES, Afonso SALES, Avelino F. ZORZO. *Structured Stochastic Modeling of Fault-Tolerant Systems*. **12th International Symposium on Modelling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS 2004)**, 2004, Volendam, The Netherlands. IEEE Press, p. 139-146. 2004.
23. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *The Need for and the Advantages of Generalized Tensor Algebra for Structured Kronecker Representations*. **UK Performance Engineering Workshop (UKPEW 2004)**, Bradford. Bradford : University of Bradford, 2004. p. 48-60. 2004.
24. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *MQNA - Markovian Queueing Networks Analyser*. **11th International Symposium on Modelling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS 20003)**, Orlando, FL, USA. New York : IEEE Press, p. 194-199. 2003.
25. Anne BENOIT, Leonardo BRENNER, Paulo FERNANDES, Brigitte PLATEAU. *Aggregation of Stochastic Automata Networks with Replicas*. **4th International Conference on the Numerical Solution of Markov Chains (NSMC'03)**, Urbana, IL, USA. Raleigh : North Carolina University, p. 145-166. 2003.
26. Anne BENOIT, Leonardo BRENNER, Paulo FERNANDES, Brigitte PLATEAU, William STEWART. *The PEPS Software Tool*. **13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS 2003)**. Urbana, IL, USA. LNCS 2794. p. 98-115. 2003.
27. Leonardo BRENNER, Luiz G. FERNANDES, Paulo FERNANDES, Afonso SALES. *Performance Analysis Issues for Parallel Implementations of Propagation Algorithm*. **15th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2003)**. São Paulo. New York : IEEE Press, p. 183-190. 2003.

28. Leonardo BRENNER, Cesar A. DE ROSE, Paulo FERNANDES. *An Analytical Model to Evaluate the Performance of Cluster Architectures*. In : **Third LCI International Conference on Linux Clusters : The HPC Revolution 2002**. Saint Petersburg, FL, USA p. 1-12.

C.3.4 Posters et Résumés dans des événements nationaux

1. Khalid HAMADA, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *Attack Synchronizing Sequences for Output Synchronized Petri Nets with Multiple Deadlocks*. **14ème Colloque sur la Modélisation des Systèmes Réactifs (MSR2023)**. 2023.
2. Ruotian LIU, Rabah AMMOUR, Leonardo BRENNER, Isabel DEMONGODIN. *Algorithm for Controlling the Transient Behavior of Controlled Generalized Batches Petri Nets*. **12ème Colloque sur la Modélisation des Systèmes Réactifs (MSR2019)**. 2019. [hal-02433029](#)
3. Feda ALMUHISEN, Nicolas DURAND, Leonardo BRENNER, Mohamed QUAFARFOU. *Prédiction de l'état du trafic routier basée sur les motifs et les chaînes de Markov*. **25èmes Rencontres de la Société Francophone de Classification (SFC)**, 2018. pp. 1-3 [hal-01858562](#)
4. Leonardo BRENNER, Brigitte PLATEAU. *Efficacité de méthodes de détection du régime stationnaire pour l'analyse de la disponibilité ponctuelle des grands modèles*. **Atelier d'Évaluation de Performance (AEP '9)**. pp.1 - 2. 2008.

C.3.5 Thèses et dissertations

1. Leonardo BRENNER. Réseaux d'Automates Stochastiques : *Analyse transitoire en temps continu et Algèbre tensorielle pour une sémantique en temps discret*. Thèse de doctorat. Institut Polytechnique de Grenoble. 2009
2. Leonardo BRENNER. *Aggregação em Redes de Automatos Estocásticos*. Dissertation de master. Pontifícia Universidade Católica do Rio Grande do Sul. 2003.

C.3.6 Rapports de Recherche

1. Ihab SBEITY, Leonardo BRENNER, Brigitte PLATEAU, William STEWART. *Phase-type distributions in Stochastic Automata Networks*. Rapports de Recherche NCSU RR-49, 2005.
2. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES, Thais WEBBER. *A Framework to Decompose GSPN models*. Rapport de Recherche PUCRS RR-38.2003.
3. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *Why you should care about Generalized Tensor Algebra*. Rapport de Recherche PUCRS RR-37. 2003.
4. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *Markovian Queueing Networks Analyser*. Rapport de Recherche PUCRS RR-30. 2003.

5. Leonardo BRENNER, Paulo FERNANDES, Caio G. P. ALEGRETTI. *Redes de Autômatos Estocásticos : Um Formalismo para Avaliação de Desempenho e Confiabilidade de Sistemas*. Rapport de Recherche PUCRS RR-30. 2002. (langue : portugais)

C.3.7 Exposés et séminaires invités

1. Rabah AMMOUR, Said AMARI, Leonardo BRENNER, Isabel DEMONGODIN, Dimitri LEFEBVRE. *Réseaux de Petri synchronisés avec sorties pour l'analyse de la vulnérabilité des systèmes cyber-physiques ace aux cyber attaques*. **1er Congrès Annuel de la Société Française d'Automatique de Génie Industriel et de Productique**. Marseille, France, 2023.
2. Leonardo BRENNER, Isabel DEMONGODIN. *Batches Petri Nets : Hybrid Formalisms for High Speed Production and Transportation Systems*. **Workshop on Formal Methods in Intelligent Manufacturing**. Xi'an, China, 2017.
3. Leonardo BRENNER, Isabel DEMONGODIN. *Modélisation mésoscopique du trafic routier par Réseaux de Petri Lots Triangulaires*. **GT Systèmes à Événements Discrets, JAMACS**. Lille, France, 2016.
4. Leonardo BRENNER, Paulo FERNANDES, Afonso SALES. *Avaliação de Desempenho de Sistemas Paralelos*. **4a Escola Regional de Alto Desempenho (ERAD 2004)**. Porto Alegre, Brésil, 2004. (langue : portugais)

C.3.8 Diffusion de Logiciels

SIMULEAU - Ce logiciel académique permet la modélisation et simulation de plusieurs classes de Réseaux de Petri. Il est spécialement dédié à la classe des RdPLots et nous permet de tester les nouvelles méthodes et théories développées dans notre équipe. <https://gitlab.lis-lab.fr/leonardo.brenner/simuleau>

PEPS - Logiciel développé pour faciliter la modélisation et la solution de modèles décrits dans le formalisme des Réseaux d'Automates Stochastiques. Ce logiciel est gratuitement mise en diffusion à partir de l'adresse <http://www-id.imag.fr/Logiciels/peps>

MQNA - Logiciel développé pour la modélisation et le calcul de solutions stationnaires d'une large classe de modèles de réseaux de files d'attente.