Equilibrage de charge efficace et adaptatif avec contraintes temporelles pour les véhicules connectés

Jean Ibarz

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique

Présentée et soutenue par :

M. JEAN IBARZ

le mercredi 15 décembre 2021

Titre :

Équilibrage de charge efficace et adaptatif avec contraintes temporelles pour les véhicules connectés

---

Ecole doctorale :

Systèmes (EDSYS)

Unité de recherche :

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)

Directeurs de Thèse :

M. JEAN-CHARLES FABRE

M. MICHAËL LAUER

Rapporteurs :

M. JEAN-CHARLES BILLAUT, UNIVERSITE DE TOURS

M. LAURENT PAUTET, TELECOM PARISTECH

Membres du jury :

MME ALIX MUNIER, UNIVERSITE SORBONNE, Présidente

M. FRANÇOIS TAIANI, UNIVERSITE RENNES 1, Membre

M. JEAN-CHARLES FABRE, TOULOUSE INP, Membre

M. MATTHIEU ROY, LAAS TOULOUSE, Invité

MI. MICHAËL LAUER, UNIVERSITE TOULOUSE 3, Membre

M. OLIVIER FLEBUS, SOCIETE CONTINENTAL AUTOMOTIVE FRANCE SA, Invité

Remerciements

Je tiens à remercier très chaleureusement Jean-Charles FABRE, Professeur des Universités à l'Université Toulouse INP-ENSEEIHT, pour avoir brillamment assuré la direction de ma thèse. Qu'il soit également remercié pour son support conséquent, sa disponibilité, et son aide précieuse.

Je remercie également très chaleureusement Michaël LAUER, Maître de conférences à l'Université Paul Sabatier - Toulouse III, qui m'a encadré tout au long de cette thèse. Qu'il soit en particulier remercié pour son extraordinaire gentillesse, sa disponibilité permanente, et la grande qualité des échanges que j'ai pu avoir avec lui. Je suis très heureux d'avoir pu travailler aux côtés de Michaël, il représente pour moi un exemple.

Je tiens aussi à remercier Olivier FLEBUS, Data Architect et Data Community Leader chez Continental ITS, qui m'a également encadré durant cette thèse et a su m'apporter son expertise et son soutient à bon escient. Je remercie plus largement Continental ITS et l'ensemble de ses membres que j'ai eu le plaisir de côtoyer.

Je remercie sincèrement Matthieu ROY, Chargé de Recherche au LAAS-CNRS, qui a également contribué de façon importante à mes travaux. Je le remercie pour ses encouragements qui m'ont décidé à m'engager dans la réalisation de cette thèse, et je le remercie également pour son aide et les agréables moments passés à ses côtés.

Je remercie Monsieur Jean-Charles BILLAUT, Professeur à l'Ecole Polytechnique de l'Université de Tours (EPU), et Monsieur Laurent PAUTET, Professeur à Telecom Paris - Institut Polytechnique de Paris, de l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse.

J'exprime ma gratitude à Madame Alix MUNIER, Maître de conférences à Sorbonne Université - LIP6, et Monsieur François TAIANI, Professeur à l'Université de Rennes 1 - Campus Beaulieu, qui ont bien voulu être examinateurs.

Je tiens aussi à remercier l'équipe pédagogique de l'Université Paul Sabatier - Toulouse III avec qui j'ai eu le plaisir de travailler dans le cadre de mes vacations. Je remercie particulièrement Michel COMBACAU, Professeur des Universités à l'Université Paul Sabatier - Toulouse III, Pauline RIBOT et Euriell LE CORRONC, Maîtres de Conférences à l'Université Paul Sabatier - Toulouse III, pour m'avoir accueilli et m'avoir aidé avec bienveillance dans mon intégration.

Un grand merci aussi à mon grand frère et mes parents pour leur soutien indéfectible.

Ce n'est pas l'intelligence qui
nous distingue, mais la bêtise.

Bernard Arcand

Résumé

Pour améliorer l'expérience de la mobilité, une piste proposée par Kramer dès 1989 est aujourd'hui envisagée. Cette piste consiste à éliminer les manques ou déficiences d'informations et de capacités et possibilités de communication, qui seraient à l'origine de 90% de tous les accidents de trafic. À cette fin, il est envisagé que des millions de véhicules connectés agissent en tant que mineurs d'informations dans un système distribué massif. Dans ce système, chaque véhicule est équipé de plusieurs capteurs pour capter les informations locales de l'environnement. Les informations capturées par l'ensemble des véhicules sont transférées vers le Cloud, où elles sont exploitées par des services pour générer une connaissance globale et fréquemment mise à jour de l'environnement. La connaissance globale de l'environnement ainsi générée rend possible une meilleure anticipation des situations futures, dans un horizon électronique qui s'étend au-delà de la perception des capteurs embarqués.

L'importante quantité de données générées par la flotte de véhicules ainsi que le caractère fortement dynamique de l'environnement impose une optimisation efficace et adaptative du flux de données transférées des véhicules vers le Cloud. Dans cette thèse, nous proposons des éléments de solution à ce problème.

Dans la partie I, nous présentons la motivation, le contexte, et la problématique de notre travail. Nous posons le cadre du problème initial et des hypothèses que nous nous sommes imposés pour rendre l'étude du problème abordable. Nous proposons un modèle multi-échelle-temporelle du problème avec 4 niveaux hiérarchiques, un type de modèle usuellement rencontré en recherche opérationnelle. Les différentes échelles de granularité temporelle considérées sur les différents niveaux permettent une réactivité adaptée pour répondre avantageusement aux variations imprévues dans un horizon futur plus ou moins grand.

Dans la partie II, nous étudions le problème avec la granularité temporelle la plus fine, au niveau des véhicules. Nous formulons le problème comme un problème d'ordonnancement basé-valeur en nous appuyant sur des concepts de temps-réel souple. Nous réalisons une évaluation expérimentale comparative d'un ensemble d'algorithmes gloutons en-ligne, judicieusement choisis pour leur forte capacité adaptative et leur faible complexité. Nous proposons d'étendre une méthode de génération aléatoire de scénarios, ce qui permet d'une part de générer un ensemble de scénarios plus variés pour une meilleure couverture de l'évaluation, et d'autre part d'obtenir des résultats plus riches et donc potentiellement plus transparents. Nos résultats indiquent qu'un algorithme simple permet de résoudre efficacement le problème.

Dans la partie III, nous étudions le problème avec une granularité temporelle moins fine, au niveau du Cloud. Il faut concilier les besoins des différents services, sachant que leurs besoins peuvent être difficiles à estimer, qu'il ne sera pas possible de tous les satisfaire, et que certains services sont plus importants que d'autres. Nous cherchons un moyen d'abstraire l'expression concrète du besoin, et de contrôler l'influence de chaque service sur le flux de données des véhicules vers le Cloud. Nous envisageons une solution de contrôle basé-marché pour résoudre ce problème. Le pouvoir d'influence est matérialisé par du numéraire, et distribué périodiquement aux services. Les services ont la liberté et la responsabilité d'utiliser au mieux le numéraire pour satisfaire leurs besoins. L'influence des services sur le flux de données se réalise au travers d'intéractions avec un mécanisme de provision qui spécifie et dirige ces intéractions. Pour trouver un mécanisme adapté à notre problème, nous construisons une multitude de mécanismes basés sur des concepts similaires à des mécanismes existants dans la

littérature. Ensuite, nous tirons profit des techniques d'apprentissage par renforcement pour analyser les effets produits par chaque mécanisme. Nos résultats indiquent qu'un mécanisme simple et peu coercitif est un bon candidat pour résoudre efficacement notre problème.

Abstract

To improve the experience of mobility, a track proposed by Kramer in 1989 is currently being considered. This approach consists in eliminating the lack or deficiencies of information and of communication capacities and possibilities, which would be at the origin of 90% of all traffic accidents. To this end, millions of connected vehicles are envisioned to act as crowdsourcers in a massive distributed system. In this system, each vehicle is equipped with several sensors to capture local information from the environment. The information captured by all vehicles is transferred to the Cloud, where it is used by eHorizon services to generate a global and frequently updated knowledge of the environment. The global awareness of the environment thus generated makes possible a better anticipation of future situations, in an electronic horizon that extends beyond the perception of on-board sensors.

The huge amount of data generated by the vehicle fleet as well as the highly dynamic nature of the environment requires efficient and adaptive optimization of the data flow transferred from vehicles to the Cloud. In this thesis, we propose elements of solution to this problem.

In part I, we present the motivation, the context, and the problematic of our work. We set out the framework of the initial problem and the assumptions that we have imposed on ourselves to make the study of the problem tractable. We propose a multi-time-scale model of the problem with 4 hierarchical levels, a kind of model usually encountered in operations research. The different scales of temporal granularity considered on the different levels allow appropriate adaptation to respond advantageously to unforeseen variations in a more or less large future horizon.

In part II, we study the problem with the finest temporal granularity, at the vehicle level. We formulate the problem as a value-based scheduling problem using soft real-time concepts. We carry out a comparative experimental evaluation of a set of greedy on-line algorithms, judiciously chosen for their high adaptive capacity and low complexity. We propose to extend a method of random generation of scenarios, which makes it possible, on the one hand, to generate a set of more varied scenarios for a better coverage of the evaluation, and on the other hand to obtain richer and more transparent results. Our results indicate that a simple algorithm efficiently solves the problem.

In part III, we study the problem with a less fine temporal granularity, at the Cloud level. The needs of different services must be reconciled, knowing that their needs may be difficult to estimate, that it will not be possible to meet them all, and that some services are more important than others. We are looking for a way to abstract the concrete expression of the need, and to control the influence of each service on the data flow from vehicles to the Cloud. We consider a market-based control solution to solve this problem. In this solution, the power of influence is materialized by some numeraire, which it is practical to imagine as money. The numeraire is distributed periodically to the services. The services have the freedom and responsibility to make good use of the numeraire to acquire the data they want, and can accumulate numeraire to dynamically adapt to changing needs. The influence of services on the flow of data is achieved through interactions with a provision mechanism. The provision mechanism is defined by a set of rules that precisely specify how interactions take place. To find a mechanism adapted to our problem, we propose to evaluate several of them and select the one that produces the most desirable effects. We build a family of mechanisms that are based on concepts similar to mechanisms already existing in the literature, and we then evaluate these mechanisms by taking advantage of reinforcement learning techniques. Our results indicate that a simple and little coercitive mechanism is a good candidate to effectively solve our problem.

Acronymes

1HB 1st-Highest-Bid. ix, 58, 61, 64, 83, 89, 95

ACE Agents-based Computational Economics. 12, 50, 63, 66, 93, 95

ANOVA ANalysis Of VAriance. 112

C2V Cloud-to-Vehicle. 7

CDSF Continental Digital Services France. 3

CIFRE Convention Industrielle de Formation par la REcherche. 3

DTD1 Dynamic Timeliness Deadline. ix, 17, 29, 30, 31, 33

DTD2 Dynamic Timeliness Deadline Squared. 29, 30, 33, 34, 35, 37

DVD1 Dynamic Value Density. ix, 29, 30, 31, 33, 34, 37, 112, 113

DVD2 Dynamic Value Density Squared. 29, 30, 33, 34, 35, 37, 112, 113

ECSMP Equal Cost Sharing with Maximal Participation. 65, 75, 90, 92

EDF Earliest Deadline First. 24, 25, 37

EDF-VD Earliest Deadline First with Virtual Deadlines. 37

FAB FABrication. 53

FPSBA First Price Sealed-Bid Auction. 47, 48, 66

FPSBARP First-Price Sealed-Bid Auctions with a Reservation Price. 64, 75

FR Full-Rebate. 65, 83, 85, 89, 92, 95

GIGO Garbage In Garbage Out. 67

GLPK Gnu Linear Programming Kit. 24, 25

HDF Highest Density First. 37

HVR Hit Value Ratio. ix, x, 23, 24, 31, 33, 37, 112, 113

ID IDentité. 60, 61, 64, 65

INC INClusion. 53, 64

LAAS-CNRS Laboratoire d'Analyse et d'Architecture des Systèmes - Centre National de la Recherche Scientifique. 3

LSTM Long-Short-Term-Memory. 72, 73

MAC Mécanisme d'Allocation Centralisé. ix, 9, 10, 40, 41, 42, 45, 66, 98

MCT Minimal Common Threshold. ix, 43, 57, 58, 61, 65, 83, 85, 86, 89, 92, 93, 95, 96

MILP Mixed Integer Linear Problem. 24, 54

NE No-Exclusion. ix, 43, 58, 60, 61, 65, 83, 85, 93, 95, 96

NR No-Rebate. 58, 64, 65, 83, 89, 92, 95, 96

PAY PAYment. 53, 64

PMA Paiement Maximal Acceptable. 52, 53, 54, 57, 60, 61, 65, 67, 68, 71, 89, 100

PPM Provision-Point-Mechanism. 65, 75

PRG Profit Révélé du Groupe. 54, 64

PRG-MAX MAXimisation du Profit Révélé du Groupe. 54, 60, 61, 62, 64, 65

QoS Quality of Service. 19

RED REDistribution. 53

RLlib Reinforcement Learning library. xi, 72

RNN Recurrent Neural Network. 72

SDVD Semi Dynamic Value Density. 29, 30, 31, 32, 37

SP500 Standard and Poor's 500. 48

SVD Static Value Density. 29, 30, 31

V2C Vehicle-to-Cloud. 4, 6, 7, 13, 16, 17, 19, 38, 40, 41, 45, 52

VCG Vickrey-Clarke-Groves. 93

WR Weighted-Rebate. 43, 83, 85, 89, 95, 96

Liste des figures

Liste des tableaux

Table des matières

Première partie

Introduction générale

Chapitre 1

Motivation, contexte, et problématique

1.1 Comment améliorer les systèmes de transport ?

Dans le domaine automobile, tous les constructeurs et les équipementiers souhaitent améliorer les capacités et les performances des systèmes relatifs à la mobilité et à l'autonomie des véhicules. C'est l'un des défis scientifiques et techniques majeurs du moment. Pour cela, Kramer suggère dès 1989 la mise en place de fonctions et systèmes dans les véhicules pour éliminer les manques ou déficiences d'informations et de capacités et possibilités de communication, à l'origine de 90% de tous les accidents de trafic.

> *"the various deficiencies and lacks of drivers' information and communication capabilities and possibilities would cause more than 90 per cent of all road traffic accidents and, therefore, functions and systems in future vehicles would be needed which would have to rule out those "human imperfections" by replacing or complementing drivers"*
>
> - Extrait de [Kramer and Reichart, 1989], abstract.,

Un malheureux raccourci est de conclure que l'humain est l'élément le moins fiable du système, et qu'il est préférable de remplacer le conducteur humain plutôt que de l'assister. Ce raccourci est par exemple fait dans [Fleetwood, 2017] :

> *"Autonomous vehicles, which could reduce traffic fatalities by up to 90% by eliminating accidents caused by human error - estimated to be 94% of fatalities - could save more than 29000 lives per year in the United States alone [Singh, 2015]."*
>
> - Extrait de [Fleetwood, 2017], page 532.,

Si l'étude [Singh, 2015] indique effectivement que 94% des *causes critiques* d'accident qui causent un évènement critique de pré-crash sont d'origine humaine, l'étude précise aussi et surtout que ce chiffre n'est pas prévu pour être interprété comme une assignation de la faute du conducteur, du véhicule, ou de l'environnement :

> *"Although the critical reason is an important part of the description of events leading up to the crash, it is not intended to be interpreted as the cause of the crash nor as the assignment of the fault to the driver, vehicle, or environment."*
>
> - Extrait de [Singh, 2015], page 1.,

L'opinion qui consiste à croire que l'humain est peu fiable est d'ailleurs remise en question dans [Reichart, 1993]. Au contraire, les éléments apportés par Reichart indiquent qu'il est remarquablement difficile pour un équipement technique d'excéder la fiabilité humaine, ce qui suggère d'assister le conducteur humain plutôt que de le remplacer. Bien que quelques véhicules autonomes semblent pouvoir rouler depuis peu de nos jours, nous pouvons affirmer sans détour que cette analyse était pertinente

à la date de son écriture ; nous pouvons même raisonnablement affirmer que cette analyse est toujours d'actualité.

> *"Some reasons are given for questioning the prevailing opinion that the human element is the most unreliable part of the road traffic system. Individual accident risk is low on a trip, although there is considerable risk in the whole population, due to the large number of vehicles and journeys. Various models and procedures have been proposed for analysing and assessing human reliability. Human error is basically understood as a human output, outside the tolerances established by the system requirements within which a person operates. Explanations of human error in road accidents are still very incomplete, though much can be gained by observing and analysing traffic conflicts. The author argues that it is remarkably difficult for the reliability of technical equipment to exceed human reliability. This strongly suggests that it is better, not to replace the driver, but to use intelligent technical methods to assist his driving task. Such assistance includes improved driver training, as well as in-vehicle and infrastructure measures. The author concludes that a human-centred automation, cooperating with the human element, will optimise overall transport system performance."*
>
> - Extrait de [Reichart, 1993].,

1.2 Projet eHorizon

En 1999, Venhovens donne une application concrète pour assister le conducteur, qu'il soit humain ou non, dans sa tâche de conduite. Il s'agit d'étendre l'horizon visuel du conducteur à un horizon électronique, beaucoup plus grand, pour aider le conducteur à mieux anticiper les situations futures.

> *"the knowledge of the current position combined with the geometry and attributes (e.g. number of lanes and traffic signs) of the road in the immediate area of the vehicle enables the driver to extend his/her visual horizon to an electronic horizon with a much larger range."*
>
> - Extrait de [Venhovens et al., 1999], page 937.,

Ce concept d'extension de la capacité de perception de l'environnement est à l'origine de l'ambitieux projet d'innovation et de recherche eHorizon, pour "electronic Horizon" en anglais, un projet porté par Continental Automotive qui a crée à cette occasion en 2016 la filiale Continental Digital Services France (CDSF) située à Toulouse.

Ma thèse s'inscrit dans ce projet, et est financée via le dispositif de Convention Industrielle de Formation par la REcherche (CIFRE). Ce dispositif CIFRE associe Continental, le laboratoire de recherche LAAS-CNRS qui assure l'encadrement de la thèse, et moi-même.

Pour améliorer l'expérience de la mobilité, il est envisagé de générer une connaissance globale et fiable de l'environnement. Pour cela, les véhicules sont dotés d'une capacité de perception de leur environnement via de nombreux capteurs qu'ils embarquent. Ainsi dotés de cette capacité de perception, l'ensemble des véhicules va pouvoir capturer des informations locales de l'environnement dans lequel ils évoluent. Ces informations, capturées de façon distribuée dans l'espace et le temps, sont ensuite communiquées et centralisées dans le Cloud. Les données ainsi accumulées dans le Cloud sont ensuite exploitées par des services eHorizon pour générer une connaissance globale de l'environnement qui

permettra d'assister le conducteur, humain ou non, dans sa tâche de conduite. Ce principe est illustré dans la Figure 1.

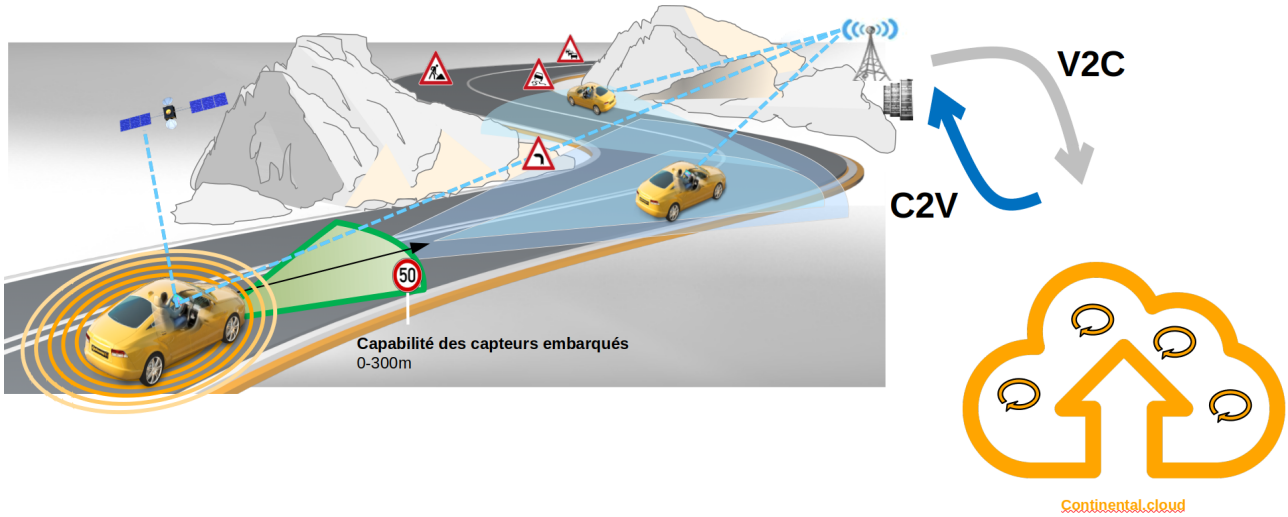


Figure 1 – *Principe du projet eHorizon.*

La flotte de véhicules capture des informations locales de l'environnement, qui sont transférées vers le Cloud. Les données ainsi agrégées dans le Cloud sont utilisées par des services eHorizon pour générer une connaissance globale de l'environnement, qui est envoyée aux utilisateurs pour améliorer l'expérience de la mobilité.

1.3 Nécessité d'optimiser le flux de données des véhicules vers le Cloud

Un véhicule connecté peut générer plus de 300TB de données par an [Dmitriev, 2017]. Pour des raisons de faisabilité technique et économique, la totalité des données générées par plusieurs millions de véhicules connectés ne pourra pas être transmis vers le Cloud, ce qui impose une optimisation critique du flux de données Vehicle-to-Cloud (V2C).

1.4 Problématique

Dans le contexte que nous venons de décrire, comment équilibrer efficacement et dynamiquement le flux de données V2C en tenant compte des propriétés spatio-temporelles des informations ? La réponse à cette question est problématique pour au moins 8 raisons :

1. l'hétérogénéité des technologies réseau est grande (3G, 4G, 5G, LTE, LTE+, MIMO, WiFi, WiMAX, LiFi, VLC, Infrared, Bluetooth, GSM, GPRS, UMTS HSPA, mmWave, etc.),
2. l'hétérogénéité des matériels est grande (capteurs de différentes technologies, différents constructeurs, etc.),
3. les ressources sont variables dans l'espace et le temps (la capacité du réseau sans-fil, la densité de véhicules dans l'espace, la forte mobilité des véhicules...),
4. le business modèle des données n'est pas bien défini,

5. la confiance que l'on peut placer dans les données est difficile à évaluer,
6. l'envergure potentiellement mondiale du système,
7. la valeur ajoutée aux données par les séquences de transformations réalisées par les services eHorizon dans le Cloud est difficile à évaluer (non-linéaire, combinatoire, connaissance imprécise des processus, qui de surcroît sont en constante évolution...),
8. l'aspect multi-objectif du problème d'optimisation (maximisation de l'utilité générée et minimisation des coûts opérationnels).

Chapitre 2

Objectifs de la recherche et décomposition du problème

Le but de notre recherche est de découvrir comment on peut équilibrer efficacement et dynamiquement le flux de données V2C en tenant compte des propriétés spatio-temporelles de l'information, et de proposer des solutions. Nous posons d'abord quelques hypothèses initiales qui délimitent le cadre du problème étudié. Ensuite, nous proposons une décomposition du problème pour en faciliter sa résolution.

2.1 Hypothèses initiales

Dans les systèmes de transports intelligents, de nombreux canaux de communication sont envisagés entre les véhicules et leur environnement. La Figure 2 illustre quelques uns de ces différents canaux de communications. Pour limiter l'étendue du problème étudié, nous considérons que les véhicules ne peuvent pas communiquer et intéragir entre eux (absence de communication V2V). Les véhicules sont toutefois dotés d'une interface de communication sans-fil qui leur permet de communiquer avec le Cloud.

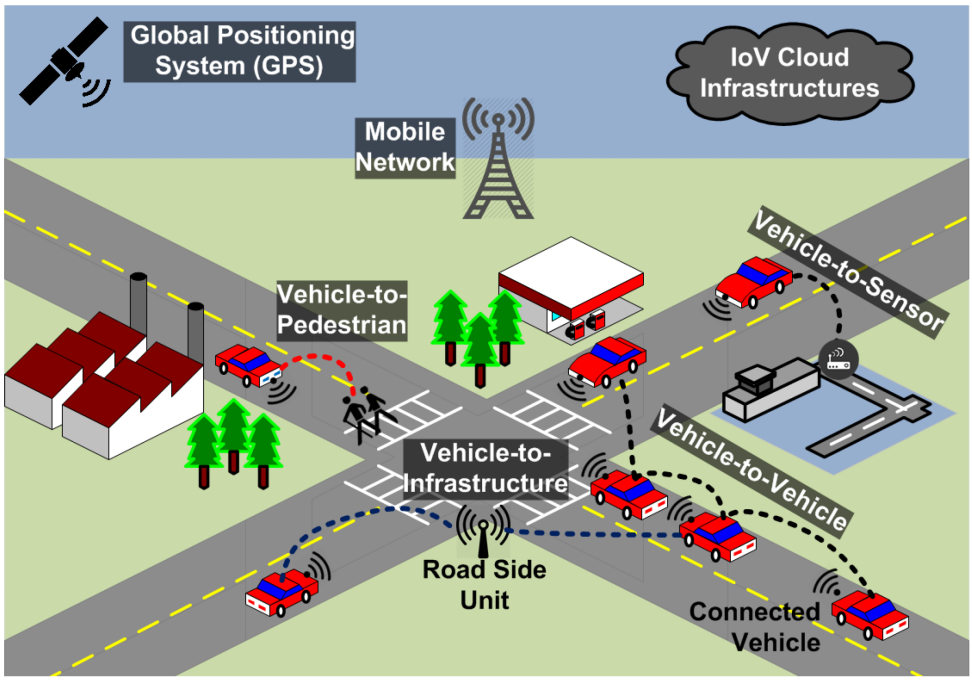


Figure 2 – *Différents canaux de communication envisagés dans les systèmes de transports intelligents. Image sourcée de [Javed et al., 2016].*

La transmission peut s'effectuer dans les deux sens : des véhicules vers le Cloud (communication V2C), ou du Cloud vers les véhicules (communication C2V). Pour pouvoir tirer profit du transfert d'information C2V, il faut déjà disposer dans le Cloud d'informations utiles pour les véhicules. Par exemple des informations concernant la présence d'un embouteillage, d'un accident, de travaux sur la chaussée, de nids de poule, de verglas, etc. Pour cette raison, nous avons considéré priorisé l'optimisation du trafic V2C. Ce travail a été conséquent et nous avons donc laissé de côté le problème de l'optimisation du trafic C2V.

Nous cherchons à optimiser le flux V2C en décidant des données qui doivent être transmises ou non, et quand. La compression du flux de données, via des techniques d'encodage ou via des techniques d'échantillonnage adaptatif, n'est pas considérée dans le champ de nos travaux.

La communication V2C est abordée avec un niveau élevé d'abstraction, c'est-à-dire que nous ne nous intéressons pas aux détails de communication trop bas niveau. Par exemple, les différents protocoles réseaux de communication ne sont pas considérés dans nos travaux.

Déterminer si une information est pertinente (utile) ou superflue (inutile) pour l'utilisateur nécessite de tenir compte de l'état cognitif de l'utilisateur. En effet, une même information peut être utile ou inutile selon le contexte. Par exemple, la présence d'une place de parking libre peut être utile ou inutile à un conducteur selon qu'il souhaite se garer ou non. Le problème d'évaluation de la pertinence (utilité, ou richesse) d'une information est complexe, ne constitue pas l'objectif principal et nos travaux, et n'est pas investigué en détail.

La communication sans-fil entre les véhicules et le Cloud est particulièrement vulnérable aux attaques et aux perturbations de fonctionnement en tous genres en raison du type de medium utilisé, l'air. Nous considérons donc que la bande passante V2C disponible peut-être nulle à tout moment, c'est-à-dire qu'il n'est pas possible de garantir une bande passante minimale disponible non-nulle. De plus, la forte mobilité des véhicules et la forte variabilité de densité des véhicules dans l'espace peuvent entraîner de fortes variations de charges sur le réseau sans-fil. L'optimisation du flux V2C que nous cherchons à proposer devra donc être adaptée/robuste à de fortes fluctuations de bande passante.

Le problème d'optimisation du flux d'informations entre les véhicules et le Cloud est un problème d'optimisation multi-objectifs, dans lequel on cherche à maximiser le flux d'informations pertinentes pour le système et, dans le même temps, diminuer les coûts opérationnels engendrés par ces flux. Ce problème est similaire à celui d'un problème de planification et de pilotage de la production de ressources. En effet, la *transmission* d'un évènement vers le Cloud peut être vue comme la *production* d'un évènement par le véhicule. Filtrer/ordonnancer les évènements transmis des véhicules vers le Cloud revient donc à piloter la production des évènements par les véhicules.

Le flux de données V2C peut être vu comme une chaîne de production constituée de trois composants :

— les véhicules, qui sont des producteurs de données (V2C),
— le Cloud, qui est constitué de services eHorizon qui cherchent à transformer les données stockées dans le Cloud pour générer de nouvelles données à valeur ajoutée,
— le "marché", constitué d'utilisateurs/consommateurs de ces nouvelles données à valeur ajoutée.

Dans cette chaîne, chacun des trois composants pourrait constituer un goulot d'étranglement et limiter le flux de données (ou déséquilibrer la chaîne). Néanmoins, nous supposons que les solutions Cloud offertes aujourd'hui par les grandes entreprises, telles qu'Amazon ou Google pour n'en citer que deux, permettent une extensibilité suffisante, et que donc la capacité de transformation des données ne devrait pas être à l'origine d'un déséquilibre de la chaîne. Au contraire, bien que les véhicules génèrent

une énorme quantité de données, certains types d'évènements peuvent être rares. Par exemple, l'observation d'un obstacle présent sur la route dans un virage sans visibilité. De plus, les coûts opérationnels engendrés par le flux de données V2C va certainement amener à limiter les données effectivement transmises vers le Cloud. Ceci nous amène à penser qu'il est raisonnable que la capacité de fabrication de données des véhicules puisse être à l'origine d'un déséquilibre de la chaîne, et que cela doit donc être considéré. Enfin, l'incertitude sur la capacité du marché à consommer les données générées par le Cloud est grande, ce qui nous oblige à considérer que la capacité du marché à consommer les données est également une origine de déséquilibre à prendre en compte dans notre problème d'optimisation.

En résumé, nous supposons donc que la capacité de transformation des données par les services eHorizon n'est pas à l'origine d'un déséquilibre de la chaîne. Au contraire, nous supposons que la capacité de fabrication et de consommation des données peut être à l'origine d'un déséquilibre, ce qui fera donc l'objet d'une attention particulière.

Les données produites, transformées, puis consommées, sont des ressources. On peut différencier deux catégories de ressources, les *ressources contraintes* et les *ressources non-contraintes*.

Ressources contraintes. Au niveau des véhicules, les ressources contraintes peuvent être les données générées de façon non-contrôlée, sporadiquement, selon les observations qui sont faites de l'environnement. Par exemple la détection d'un panneau, d'un nid de poule, d'un obstacle, d'un piéton, etc. Le plus souvent, il s'agit d'évènements qui apparaissent de façon ponctuelle.

Ressources non-contraintes. Au niveau des véhicules, les ressources non-contraintes peuvent être les données générées de façon contrôlée et à volonté. Par exemple toute information relative à l'état de mobilité du véhicule (position, vitesse, orientation...), à un état plus général du véhicule (présence/absence de dysfonctionnement), à la "vision" du radar ou des caméras, etc. Le plus souvent, il s'agit d'évènements dont l'existence est continue dans le temps et/ou qu'il est possible d'échantillonner à une fréquence aussi élevée que nécessaire.

Modéliser le caractère spatio-temporel de l'information contenue dans les données est un problème extrêmement complexe qui pourrait faire l'objet d'une thèse entière (exemple [Plumejeaud, 2011]). Il ne nous semble pas raisonnable de nous aventurer trop en détail dans ce problème de modélisation, ni de considérer qu'un unique modèle bien précis pourrait satisfaire tous les besoins futurs liés aux transports intelligents et au projet eHorizon. Nous retiendrons essentiellement que les phénomènes observés par les véhicules peuvent être repérés dans l'espace-temps, que la pertinence d'un évènement pour le système global peut être évaluée, dépend du temps écoulé depuis la génération de l'évènement et du type d'évènement.

2.2 Décomposition du problème

Optimiser le flux de données entre les véhicules et le Cloud est un problème qui présente de nombreuses difficultés :

— il faut prendre en compte les besoins des services eHorizon présents dans le Cloud ; besoins qui sont hétérogènes et variables dans le temps.
— il faut prendre en compte des contraintes de capacité. En effet, il n'est probablement pas envisageable de satisfaire entièrement les besoins de tous les services, parce que la flotte de véhicules ne génère pas assez certains évènements, ou parce que le besoin d'un service en données est trop important et engendrerait des coûts opérationnels trop importants.

— déterminer la relation entre la valeur générée par un service pour le système global et les données remontées vers le Cloud nécessite un modèle détaillé de l'ensemble des processus de transformation au sein du service. De plus, cette relation nécessiterait d'être fréquemment réévaluée pour tenir compte de l'évolution des services eHorizon.
— l'optimisation doit permettre une forte extensibilité.
— l'étendue de granularité des prises de décisions est grand.

Herrera argumente dans [Herrera, 2011] pour la mise en place d'un système de planification logistique dans un contexte de décisions hybride, à la fois centralisé et distribué. L'argumentaire est de combiner les avantages des deux mondes : la stabilité d'un système centralisé, et l'agilité/la réactivité/la nervosité d'un système distribué. En décomposant notre problème, nous donnons la possibilité d'envisager une telle solution hybride.

De plus, décomposer le problème en plusieurs sous-problèmes peut permettre de résoudre les difficultés par partie, plutôt que de chercher à toutes les résoudre dans un unique problème. Une décomposition du problème d'optimisation semble préférable, voire nécessaire.

Pour optimiser le flux V2C, nous proposons une modélisation multi-échelle-temporelle ("multi-time-scale" en anglais) du problème, comme cela est rencontré en recherche opérationnelle [Wongthatsanekorn et al., 2010]. Notre modèle comporte 4 niveaux hiérarchiques de prise de décision, où chaque niveau correspond à une échelle temporelle différente. Le modèle est illustré Figure 3.

Au niveau L3, les décisions sont prises par des décideurs/visionnaires. Le résultat est un budget alloué sur le long terme, et stratégiquement réparti entre les différents services. Le budget alloué correspond à la dépense que l'entreprise prévoit pour couvrir le coût engendré par le flux V2C, i.e. le coût engendré par la remontée vers le Cloud des données générées par les véhicules sur un horizon à "long" terme.

Au niveau L2, un Mécanisme d'Allocation Centralisé doit définir une quantité optimale de chaque type de données V2C à remonter par les véhicules, pour satisfaire les besoins des différents services eHorizon sur un horizon à "moyen" terme.

Au niveau L1, un mécanisme est chargé de contrôler le flux V2C via des décisions à "court" terme pour satisfaire la quantité de chaque type de données V2C à remonter, quantités qui sont déterminées au niveau supérieur sur un horizon à "moyen" terme.

Au niveau L0, il faut décider quelles données V2C générées par les véhicules doivent être remontées vers le Cloud avec des décisions sur un horizon à "très court" terme.

Dans cette thèse, nous traitons les problèmes de décision aux niveaux L0 et L2 uniquement. Le niveau L3 n'est pas étudié car il concerne un problème fréquemment rencontré par les investisseurs et qui sort du cadre de notre travail. Le niveau L1 est un problème certainement complexe et très intéressant, mais sur lequel nous n'avons pas pu nous pencher par manque de temps. Un travail supplémentaire est requis pour résoudre le problème d'optimisation à ce niveau.

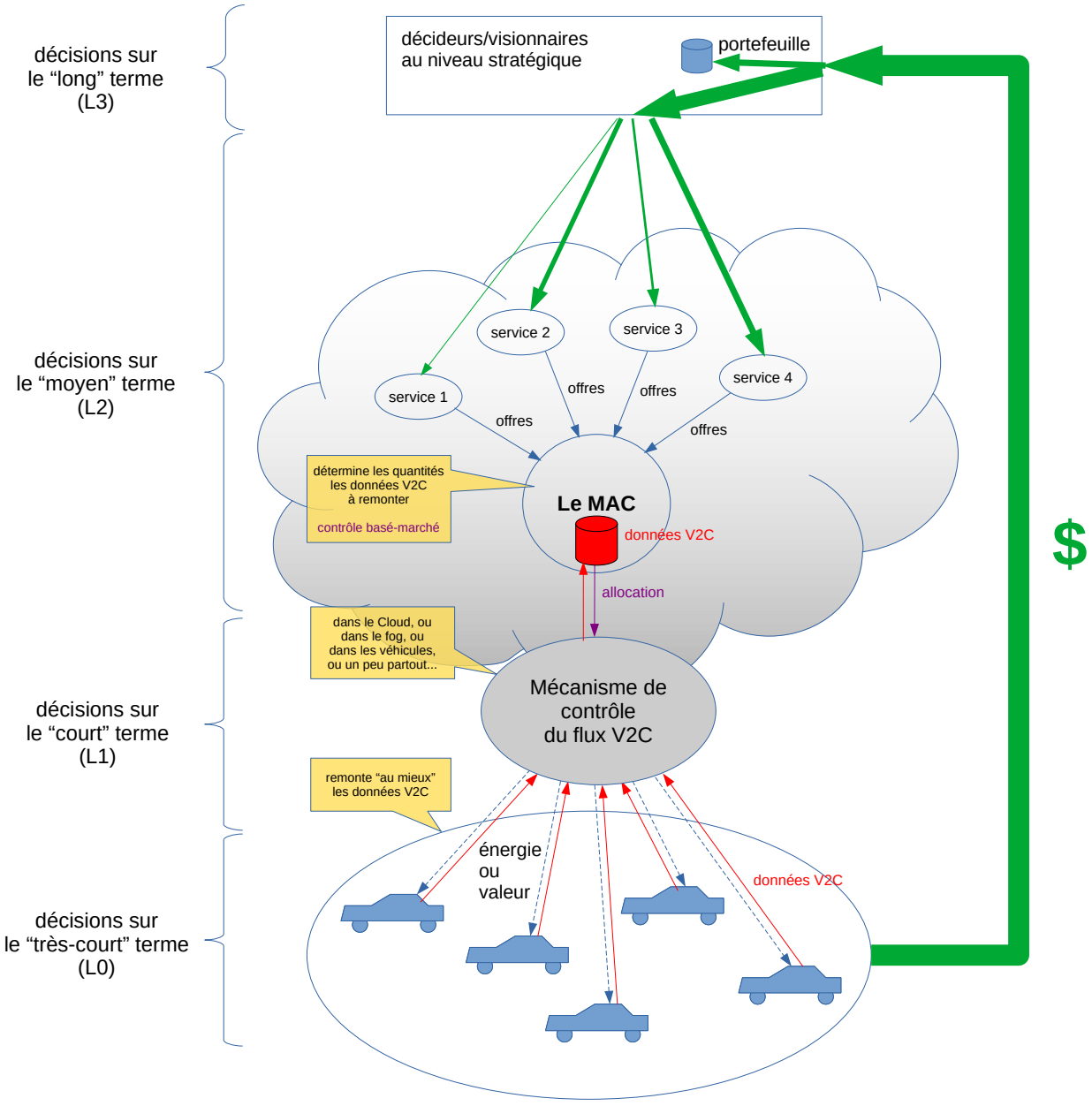


Figure 3 – *Décomposition proposée du problème d'optimisation de flux V2C dans le système de véhicules connectés. MAC est l'acronyme du Mécanisme d'Allocation Centralisé.*

Chapitre 3

Contributions

Dans ce travail de recherche nous apportons plusieurs éléments de compréhension du problème, et nous faisons un rapprochement de ce problème d'optimisation avec de multiples domaines de la littérature scientifique. Plus précisément, nous nous appuyons sur la théorie des systèmes temps-réel souple, la théorie de la conception de mécanismes d'incitation (ou des mécanismes de marché), et des techniques d'apprentissage par renforcement multi-agents appliquées dans le paradigme de l'économie computationnelle basée sur des agents multiples hétérogènes.

Nous proposons une décomposition du problème en 4 niveaux décisionnels hiérarchiques, et proposons deux éléments de solution pour contrôler le flux de données V2C :

— un contrôle via un algorithme d'ordonnancement en-ligne embarqué dans les véhicules, qui vise à optimiser la valeur des données transmises par chaque véhicule indépendamment,

— un contrôle basé-marché via un mécanisme centralisé dans le Cloud, qui vise à déterminer périodiquement la quantité optimale de chaque type de données à remonter par la flotte de véhicules.

3.1 Algorithme d'ordonnancement en-ligne embarqué dans les véhicules

Pour résoudre le problème de décision au niveau des véhicules, nous nous basons sur la théorie du temps-réel souple, et nous proposons une évaluation de la performance d'algorithmes existants dans un contexte de surcharge très élevée. Nous proposons une extension d'une méthode de génération de scénarios aléatoire existante, et nous proposons également un ajustement des scénarios pour réduire un biais d'évaluation qui pourrait être introduit et affecter la performance de certains algorithmes. Nous obtenons des résultats différents de ceux présentés dans la littérature, et proposons une explication à cela. Ces travaux ont fait l'objet d'une publication [Ibarz et al., 2020] pour la conférence RTNS 2020 [1].

3.2 Mécanisme d'Allocation Centralisé dans le Cloud

Pour le mécanisme centralisé, nous proposons une approche basée sur la conception de mécanisme, dans laquelle nous cherchons à définir les règles d'un mécanisme tel que les besoins de service eHorizon, exprimés individuellement, permette d'obtenir un résultat satisfaisant.

Nous proposons une extension d'un modèle générique de mécanisme et une redéfinition de mécanismes existants dans ce modèle étendu.

Nous étudions une famille de 8 mécanismes, dont 3 sont déjà définis et étudiés dans la littérature mais dans un contexte plus restrictif et que nous proposons donc d'étendre à notre contexte qui est plus général.

1. International Conference on Real-Time Networks and Systems, accessible à l'url https://rtns2020.inria.fr/program/

Nous utilisons une approche économique computationnelle basée agents, ou Agents-based Computational Economics (ACE) en anglais, pour dégager une compréhension du comportement individuel (niveau micro-économique) et collectif (niveau macro-économique) des services eHorizon qui intéragissent avec les mécanismes étudiés.

Nous proposons également une nouvelle métrique qui permet de mieux apprécier la qualité des mécanismes étudiés.

Chapitre 4

Structure du manuscrit

La suite du manuscrit se scinde en 2 parties.

Dans la partie II, nous traitons le problème d'optimisation au niveau L0, au plus proche des véhicules. Il faut décider quelles données générées par les véhicules doivent être transmises, et quand, pour maximiser la richesse générée pour le système. La solution doit être très réactive et robuste aux aléas du futur. Nous utilisons des concepts de la théorie du temps-réel souple pour aborder ce problème comme un problème d'ordonnancement basé-valeur. Nous réalisons une étude expérimentale comparative de différents algorithmes déjà existants et judicieusement choisis, et montrons que l'un d'eux est un bon candidat pour résoudre efficacement notre problème.

Dans la partie III, nous traitons le problème au niveau L2. Pour réaliser un équilibrage efficace et adaptatif de la capacité du flux V2C en fonction des besoins des services eHorizon, nous choisissons d'aborder le problème avec une approche de contrôle basé-marché. Nous proposons un mécanisme centralisé dans le Cloud qui planifie périodiquement la quantité de chaque type de données que la flotte de véhicules doit transmettre vers le Cloud.

Deuxième partie

Algorithme d'ordonnancement en-ligne embarqué dans les véhicules

Introduction

Dans la partie I, nous avons proposé une décomposition du problème d'optimisation du flux V2C en 4 niveaux. Dans cette partie, nous nous situons au niveau L0, au niveau des véhicules.

Les véhicules connectés génèrent des *évènements* à partir des informations brutes obtenues des capteurs sur l'environnement. Par exemple, un véhicule connecté peut générer un évènement qui indique la présence (ou l'absence) d'un panneau de signalisation, un évènement qui indique l'occurrence d'un freinage d'urgence, etc. Pour pouvoir facilement faire référence aux évènements générés par un véhicule et destinés à être transmis vers le Cloud, nous choisissons de les appeler *évènements V2C*. La Figure 4 illustre des évènements qui pourraient être générés par un système de reconnaissance d'image pour l'aide à la conduite.

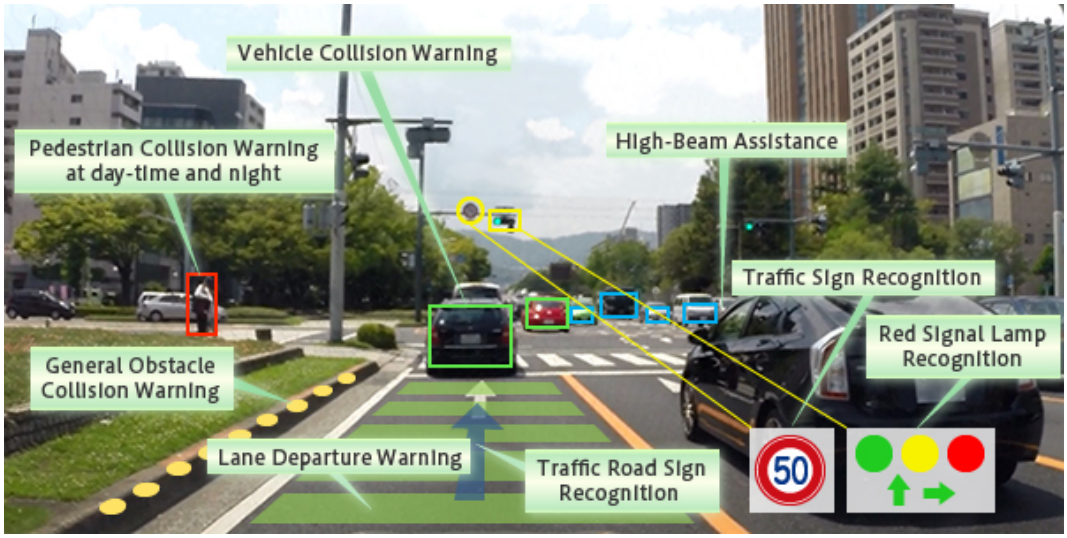


Figure 4 – *Des évènements que pourraient générer les véhicules à partir des observations locales de leur environnement.*

Source : https ://www.greencarcongress.com/2014/11/20141114-toshibaimage.html

La pertinence d'un évènement V2C varie avec le temps, parce qu'un évènement V2C caractérise l'environnement et l'environnement varie avec le temps. Par exemple, imaginons qu'un véhicule génère un évènement qui indique une place de parking libre dans un parking très prisé. L'évènement va rapidement perdre de l'utilité avec son âge, puisque la place de parking libre peut être prise par quelqu'un à tout instant. Au contraire, un évènement qui indique la présence d'un panneau de limitation de vitesse sera probablement valide plusieurs jours. Cet exemple est illustré Figure 5.

Puisque tous les évènements V2C ne peuvent pas être transmis vers le Cloud et puisque la pertinence des évènements varie avec le temps, maximiser l'utilité des évènements transmis requiert une priorisation adéquate pour décider lesquels doivent être transmis et quand.

En raison de la nature fortement dynamique des véhicules et de l'environnement dans l'espace-temps, il est difficile de prévoir les évènements qui seront générés dans l'horizon futur, même lorsque l'horizon considéré est court. La solution proposée doit donc permettre une priorisation de la transmission des évènements V2C qui soit dynamique et fortement réactive.

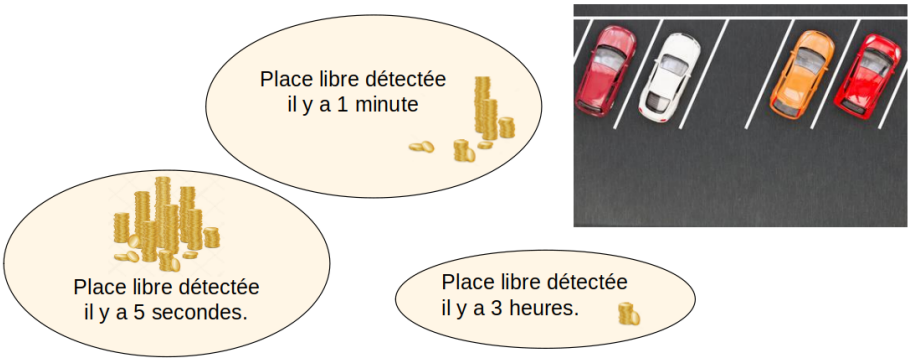


Figure 5 – *Le vieillissement d'un évènement V2C.*

Pour commencer, nous montrons comment le problème peut être défini comme un problème d'ordonnancement basé-valeur dans la théorie des systèmes temps-réel souple.

Ensuite, nous proposons une évaluation expérimentale de différents algorithmes d'ordonnancement afin d'identifier celui qui résout le plus efficacement notre problème.

Comme pour toute méthode expérimentale, il est nécessaire de s'assurer que les résultats obtenus sont représentatifs de cas réels. Comme les cas réels ne sont pas disponibles aujourd'hui, nous ne pouvons pas garantir que les résultats obtenus sont représentatifs de la réalité future.

Le choix arbitraire des scénarios générés peut nous amener à introduire un biais de sélection et, in-fine, à des analyses et conclusions peu pertinentes, voire erronées. Pour limiter ce problème, nous étendons une méthode d'évaluation expérimentale d'algorithmes d'ordonnancement déjà existante pour améliorer la diversité des scénarios générés. La méthode que nous proposons permet d'investiguer, si besoin, des paramètres qui pourraient influencer la performance des algorithmes, ce qui confère une plus grande transparence de nos résultats.

Les algorithmes évalués sont confrontés à deux instances d'un algorithme clairvoyant en-ligne qui fournissent des bornes pour l'évaluation et une juste référence pour la comparaison [Baruah et al., 1992, Gan and Suel, 2009, Attiya et al., 2010].

Les résultats que nous obtenons indiquent que l'algorithme DTD1 est le plus efficace en terme d'utilité cumulée, et est un candidat prometteur pour gérer le problème d'optimisation du transfert du flux de données V2C.

Chapitre 1

Définition du problème

Au niveau logique, les évènements sont contenus dans des *messages* ; un *message* peut être composé de plusieurs évènements.

Au niveau réseau, les messages sont divisés en *paquet(s)* de taille fixe. Nous supposons que l'envoie d'un paquet est une action atomique, c'est-à-dire que cette action ne peut être interrompue. En revanche, il est possible de suspendre (resp. reprendre) la transmission d'un message par la suspension (resp. la reprise) de la transmission des paquets restants. La préemption de la transmission d'un message est donc possible ; nous supposons de plus que cette préemption est instantanée et n'engendre pas de sur-coûts.

Les messages sont transmis vers le Cloud individuellement par chaque véhicule via un unique canal de communication de façon séquentielle, i.e. un message à la fois.

La bande passante de communication disponible est connue et constante durant l'entièreté de la simulation.

Malgré le fait que ces hypothèses sont très fortes pour notre contexte, où en réalité la bande passante disponible peut fortement varier et être difficile à prévoir, la pertinence de nos résultats expérimentaux est justifiée par le choix approprié des algorithmes évalués. En effet, nous avons argumenté en introduction que la priorisation du transfert des messages doit être fortement dynamique en raison de la forte dynamique des véhicules et de leur environnement. Les algorithmes que nous avons choisi d'évaluer possèdent deux caractéristiques pour répondre à ce besoin :

— les algorithmes prennent des décisions très fréquentes, avant le transfert de chaque paquet, ce qui confère la capacité d'adapter très rapidement la priorité des messages.
— les algorithmes considèrent un horizon futur relativement court, ce qui leur confère une certaine robustesse aux aléas du futur, par exemple une variation de la charge dû à l'apparition de nouveaux évènements, à une variation de la bande passante du réseau, etc.

Quand un message est reçu dans le Cloud, le message génère une certaine *utilité* au système global. Une définition exacte de l'*utilité* est hors du champ de ce chapitre ; mais de façon très grossière, l'utilité représente la contribution que chaque message apporte à la qualité de service fournie à l'utilisateur final.

La valeur d'utilité d'un message est liée aux évènements qu'il contient, et donc à leurs propriétés spatio-temporelles. Prenons par exemple le cas d'un évènement qui contient de l'information liée à la mobilité d'un véhicule. Après quelques secondes, le véhicule peut s'être déplacé de plusieurs centaines de mètres dans différentes directions, ou avoir modifié sa vitesse, rendant alors l'évènement peu pertinent pour des services critiques de sûreté, comme l'évitement de collision. Au contraire, un évènement indiquant la présence d'un panneau peut rester pertinent pour le système plusieurs heures voire plusieurs jours après sa génération.

La Quality of Service (QoS) du trafic dans les réseaux de communication sans-fil a été sujet à des efforts de recherche considérable dans les années récentes. Satisfaire des spécifications de QoS dans un environnement à ressources contraintes, comme un réseau de capteurs, est exceptionnellement difficile [Younis et al., 2010, Chen and Varshney, 2004, Alanazi and Elleithy, 2015]. Dans un réseau de capteur fortement mobile comme celui des véhicules, les difficultés posées par la QoS sont encore plus importantes. Il est également peu pertinent de borner le flux du trafic V2C, car la dynamique du réseau peut causer des ravages dans un transfert de données urgent [Ameen et al., 2008]. Tenant compte de ces considérations, nous supposons que les évènements transmis ne sont pas de nature critique, c'est-à-dire que de ne pas réussir à transmettre un évènement en temps voulu ne doit pas avoir une conséquence catastrophique sur le système.

On prévoit qu'une très grande quantité de données sera générée par des millions de véhicules. Envoyer tous les évènements générés par chaque véhicule n'est pas envisageable, étant donné que l'infrastructure cellulaire a déjà atteint sa capacité limite aujourd'hui. Et même dans l'éventualité où cette solution serait techniquement possible, la quantité de données transférées puis agrégées dans le Cloud serait telle que les coûts engendrés ne seraient pas tenables. Pour que le projet soit viable, il est nécessaire de trouver un compromis entre la quantité de données transmises et l'utilité globale générée. Nous supposons ici que la quantité de ressources disponibles pour la transmission des données est finie et déterminée au(x) niveau(x) supérieur(s) L1 et/ou L2, selon la décomposition du problème global que nous avons proposé et détaillé dans la partie I. Ici, nous devons donc décider quel message doit être transmit (en premier) et quand, en tenant compte de contraintes temporelles, dans l'objectif de maximiser l'utilité générée par le système global, sous une contrainte de ressources finies.

Nous supposons une isolation entre les véhicules, c'est-à-dire que les véhicules ne coopèrent pas entre eux. Ainsi, le problème de maximiser l'utilité générée par le système globale se réduit à maximiser l'utilité générée par chaque véhicule considéré indépendamment. Cette hypothèse permet une réduction considérable de la complexité du problème pour un meilleur passage à l'échelle.

Dans ces travaux, nous ne considérons pas comment les messages sont construits à partir des évènements, et nous supposons que la date d'arrivée des futurs messages n'est pas connue à l'avance. En conséquence, nous nous concentrons sur l'étude d'algorithmes d'ordonnancement *dynamique* (= *en-ligne*, *à la volée*) en situation inconnue, et de possible surcharge.

Le problème que nous résolvons dans notre travail ici consiste à trouver un algorithme d'ordonnancement préemptif, dynamique, et efficace pour envoyer un ensemble de messages (qui ont des propriétés d'utilité et temporelles), de sorte à ce que l'utilité soit maximisée pour chaque véhicule considéré individuellement. Dans la prochaine section, nous montrons que ce problème peut être traduit comme un problème d'ordonnancement à temps-réel souple.

Chapitre 2

Modélisation

2.1 Problème à temps-réel souple

Considérons un problème d'ordonnancement temps-réel où un ensemble de N jobs *indépendants* doivent être exécutés sur un unique processeur. Chaque job a une date d'arrivée, avant laquelle il ne peut pas être traité. Un job nécessite de traiter un certain nombre d'*unités d'exécution* pour être accompli. A chaque *unité de temps*, le processeur peut être au repos, ou au travail. La vitesse du processeur réfère au nombre d'unité d'exécution qu'il peut traiter par unité de temps (lorsqu'il est au travail). Au temps k, si $s[k]$ dénote la vitesse du processeur et $\overline{c}_j[k]$ dénote le nombre d'unités d'exécution restant pour qu'un job soit traité, $min(\overline{c}_j[k], s[k])$ unités d'exécution sont traitées durant cette unité de temps.

Dans notre problème, transmettre des messages vers le Cloud est similaire à traiter des jobs avec un processeur. Dans cette analogie, le nombre de paquets pour transmettre un message correspond au nombre d'unités d'exécution pour traiter un job. L'analogie entre notre problème d'ordonnancement et le problème d'ordonnancement temps-réel est résumé dans la Table 1.

Table 1 – *Analogie entre notre problème d'ordonnancement et un problème d'ordonnancement temps-réel*

Notre problème d'ordonnancement	Un problème d'ordonnancement temps-réel
Un message	Un job temps-réel
Paquets d'un message	Unités d'exécution d'un job
Interface de communication	Processeur
Bande passante disponible	Vitesse du processeur
Transmission d'un paquet	Traitement d'une unité d'exécution
Transmission d'un message	Traitement d'un job

Selon la terminologie usuelle [Buttazzo, 2011b], un job temps-réel est :
- *souple*, si accomplir le job en retard (= après sa deadline) peut malgré tout procurer de l'utilité au système, bien que l'utilité procurée soit dévaluée,
- *ferme*, si accomplir le job en retard est inutile pour le système, mais n'engendre pas de dommage,
- *dur*, si accomplir le job en retard peut engendrer des conséquences catastrophiques pour le système.

Si l'utilité que procure un job est indépendante de sa date d'accomplissement, le job est dit *non temps-réel*.

Comme déjà discuté dans la Section 1, la transmission d'un message n'est pas critique d'un point de vue sûreté ; un message correspond à un job temps-réel souple ou ferme, mais pas dur.

En accord avec le modèle introduit dans [Buttazzo, 2011a], nous caractérisons un job temps-réel J_i par un 5-uplet $< a_i, c_i, v_i, d_i, \delta_i >$, où :

— $a_i \in \mathbb{N}$ est la *date d'arrivée* du job. Un job n'est pas instancié avant sa date d'arrivée ; un job ne peut donc pas être traité avant sa date d'arrivée. Un algorithme non-clairvoyant n'est pas conscient d'un job avant sa date d'arrivée.

— $c_i \in \mathbb{N}, c_i > 0$ est le nombre d'unités d'exécution qu'il faut traiter pour que le job soit accompli. C'est une valeur entière, qui correspond à la transmission de paquet sur le réseau, un évènement que nous supposons être atomique.

— $v_i \in \mathbb{R}^+$ est l'*utilité de base* d'un job.

— $d_i \in \mathbb{N}$ est la *limite de temps ferme (relative)* du job.

— $\delta_i \in \mathbb{N}$ est le *retard limite* du job.

En suivant le concept introduit par Douglas Jensen dans [Jensen et al., 1985], une *fonction d'utilité* $\phi_i(k)$ est associée à chaque job. La fonction d'utilité exprime la valeur qui serait fournie au système au moment de l'accomplissement du job associé, valeur qui dépend de la date d'accomplissement du job. $\phi_i(k)$ est illustré dans la Figure 6 et est définie comme suit :

$$\begin{cases} \phi_i(k) = v_i & \text{si } k \in [a_i; a_i + d_i] \\ \phi_i(k) = (a_i + d_i + \delta_i - k)\,\frac{v_i}{\delta_i} & \text{si } k \in]a_i + d_i; a_i + d_i + \delta_i] \\ \phi_i(k) = 0 & \text{sinon.} \end{cases}$$

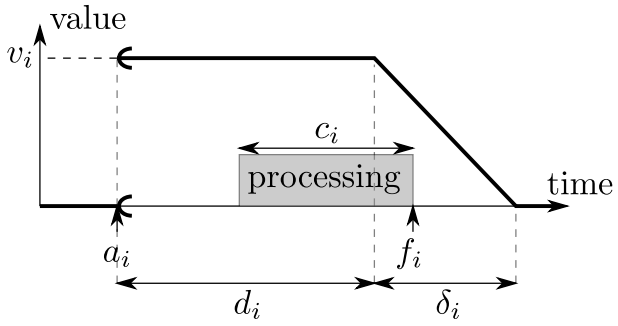


Figure 6 – *Fonction d'utilité d'un job temps-réel souple.*

Remarquons que, comme $c_i > 0$, l'utilité d'un job à sa date d'activation $\phi_i(a_i)$ vaut l'utilité de base v_i. Remarquons également que la restriction de (a_i, c_i) à $\mathbb{N}^2$, combiné avec notre hypothèse de vitesse d'exécution de processeur constant, nous permet de lancer des simulations à temps discret avec un pas-de-temps fixe. Ceci permet de simplifier grandement l'implémentation du simulateur, et réduit par conséquent le risque d'erreur d'implémentation. L'hypothèse que $(d_i, \delta_i) \in \mathbb{N}^2$ simplifie le raisonnement sans sacrifier l'expressivité, puisque l'utilisation de nombres réels n'augmenterait pas le pouvoir de notre modèle.

De cette définition formelle d'un job temps-réel souple et de la fonction d'utilité, nous pouvons faire les observations suivantes :

— $a_i + d_i$ est la *limite de temps (en valeur absolue)* du job i.

— $a_i + d_i + \delta_i$ est la *limite souple de temps (en valeur absolue)* du job i.

— La différence entre un job temps-réel ferme et un job temps-réel souple est la valeur du paramètre de retard limite δ_i : $\delta_i > 0$ pour un job temps-réel souple, et $\delta_i = 0$ pour un job temps-réel ferme.

— Par convention, nous choisissons que $d_i = \delta_i = +\infty$ pour un job non temps-réel.

Comme ce modèle de job temps-réel souple requiert seulement 5 paramètres, il est efficace en mémoire et donc approprié pour être utilisé dans des systèmes embarqués.

Définition 1 (Job accompli)**.** *On dit qu'un job est* accompli *si et seulement si toutes les unités d'exécution requises par le job ont été traitées. Nous notons f_i la date d'accomplissement du job i.*

Définition 2 (Job en retard)**.** *On dit d'un job accompli qu'il est* en retard *si et seulement si sa valeur de retard est (strictement) plus grande que* 0*. On dit qu'un système temps-réel est* surchargé *si et seulement si tous les jobs ne peuvent pas être accomplis à temps (= sans retard).*

Dans notre contexte, les évènements sont générés apériodiquement et d'une manière imprédictible et incontrôlable. De plus, la bande passante disponible peut être aussi basse que zéro. Dans de telles conditions, le système sera probablement surchargé et ce sera le *cas nominal*.

2.2 Méthode d'évaluation

Une façon d'évaluer la garantie de performance d'un ordonnanceur en-ligne est de le comparer a un ordonnanceur *clairvoyant* [Locke, 1986], i.e. qui connaît *a priori* tous les paramètres de tous les jobs du problème d'ordonnancement.

Les approches analytiques pour évaluer l'efficacité d'un algorithme d'ordonnancement impliquent généralement la métrique bien connue du *taux de compétitivité*, introduite par Sleator et Tarjan dans [Sleator and Tarjan, 1985].

Définition 3 (Taux de compétitivité d'un algorithme)**.** *Le* taux de compétitivité *est défini comme*

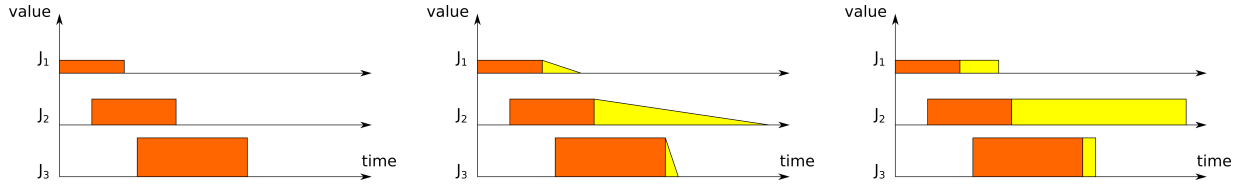
$$min\left(\left\{\frac{\Gamma_A(P)}{\Gamma_{OPT}(P)}|P \in \Omega\right\}\right)$$

où

- $\Gamma_A(P)$ *est la valeur cumulée obtenue par un algorithme A sur le problème d'ordonnancement P,*
- $\Gamma_{OPT}(P)$ *est la valeur cumulée obtenue par un algorithme clairvoyant optimal sur le même problème P,*
- Ω *est l'ensemble de tous les algorithmes d'ordonnancement existant [Buttazzo et al., 1995].*

Baruah et al. ont démontré dans [Baruah et al., 1992] que le taux de compétitivité d'un algorithme d'ordonnancement en-ligne est borné par le haut par 1/4. Cependant, ce résultat est seulement représentatif du pire scénario, dans lequel les caractéristiques des jobs sont très restrictives, et qui peut ne jamais être rencontré en pratique. Cette borne n'a donc qu'une validité théorique [Buttazzo, 2011a, Stankovic et al., 1995].

> *Before concluding the discussion on the competitive analysis, it is worth pointing out that all the above bounds are derived under very restrictive assumptions, such as all tasks have zero laxity, the overload can have an arbitrary (but finite) duration, and task's execution time can be arbitrarily small. In most real-world applications, however, tasks characteristics are much less restrictive ; therefore, the 0.25 bound has only a theoretical validity, and more work is needed to derive other bounds based on more realistic assumptions on the actual load conditions.*
>
> - Extrait de [Buttazzo, 2011a], page 305.,



(a) *Problème pessimiste* **(b)** *Problème original* **(c)** *Problème optimiste*

Figure 7 – *Problème d'ordonnancement temps-réel original, borné par un problème pessimiste et un problème optimiste.*

Un avantage de l'approche pire-cas est de fournir une garantie sur la performance de l'algorithme, mais cette garantie n'est pas requise dans notre contexte où le système est supposé non-critique. Ainsi, une approche expérimentale pour évaluer statistiquement la performance des algorithmes nous paraît plus appropriée pour pouvoir coller de façon plus stricte à des conditions réelles. Autrement dit nous sommes davantage intéressé par la performance moyenne d'un algorithme que par sa performance pire-cas. Par exemple, la table de hachage est une structure de donnée qui a une mauvaise performance pire-cas, mais cette structure est malgré tout très populaire parce que sa performance moyenne est excellente et qu'il est extrêmement improbable de rencontrer le pire-cas si les paramètres de la table sont bien choisis.

Notre objectif est d'ordonnancer un ensemble de jobs dans l'objectif de maximiser la valeur totale du système, un paradigme d'ordonnancement connu sous le nom *d'ordonnancement basé-valeur* [Burns et al., 2000]. Pour cet objectif, la métrique la plus pertinente pour évaluer la performance d'un algorithme A est probablement le *rendement de valeur cumulée* Γ_A/Γ_{OPT}, où :

— Γ_A est la valeur cumulée obtenue par l'algorithme A, et
— Γ_{OPT} est la valeur cumulée obtenue par un algorithme clairvoyant optimal OPT.

Cependant, le problème de trouver un ordonnancement optimal est généralement un problème qui ne peut pas être traité en temps raisonnable. Trouver un ordonnancement préemptif basé-valeur optimal pour un ensemble de jobs temps-réel ferme dans un système mono-processeur est similaire à trouver un sous-ensemble *ordonnançable* de jobs tel que la valeur cumulée des jobs accomplis est maximale. Un ensemble de jobs temps-réel ferme est dit *ordonnançable* si et seulement si il est existe un ordonnancement tel que tous les jobs peuvent être accomplis à temps (= sans retard). Ce problème peut se traduire comme un problème du sac-à-dos [Garey and Johnson, 1979], et est donc de complexité NP-Dur [Baruah et al., 1991].

Définition 4 (Ratio de valeur obtenu)**.** *Le Hit Value Ratio (HVR) est défini comme le ratio* Γ_A/Γ *entre la* valeur cumulée $\Gamma_A \triangleq \sum_i \phi_i(f_i)$ *obtenue par un algorithme* A *et la* valeur totale $\Gamma \triangleq \sum_i v_i$ *de l'ensemble des jobs [Buttazzo et al., 1995].*

Le HVR peut être vu comme une heuristique de la métrique de Γ_A/Γ_{OPT}, qui évite le calcul d'un ordonnancement optimal mais qui permet néanmoins une comparaison des différents algorithmes sur une dimension qui a du sens.

Nous n'avons pas choisi de considérer d'autres métriques, comme le nombre de préemption de jobs, le retard de job, le taux de succès de job, etc., car ces métriques ne sont pas censées apporter d'information importante dans notre contexte.

Pour fournir une meilleure intuition sur l'efficacité des algorithmes évalués, nous proposons d'aller plus loin en estimant la performance d'un algorithme clairvoyant optimal.

Lorsque nous considérons des jobs à temps-réel souple, l'utilité générée par le système est la valeur retournée par l'utilité de fonction non-linéaire du job accompli. Il en résulte que la valeur cumulée générée par le système est aussi non-linéaire, et que la date de complétion de chaque job doit être connue. Par conséquence, le problème de trouver un ordonnancement optimal est plus difficile que dans le cas où les jobs sont tous à temps-réel ferme, qui est déjà un problème de complexité NP-Dur.

Plutôt que de chercher un ordonnancement optimal, nous proposons d'encadrer sa performance. Pour cela, nous dérivons un problème pessimiste et un problème optimiste, où les jobs temps-réels souples sont transformés en jobs temps-réels fermes. Comme illustré dans la Figure 7, dans le problème pessimiste, la limite de retard devient nulle, c'est-à-dire qu'un job perd immédiatement toute sa valeur s'il n'est pas accompli à temps. Dans le problème optimiste, un job génère toute sa valeur (de base) s'il est accompli avant le dépassement de la limite de temps souple, et aucune valeur si cette limite est dépassée.

Comme dit précédemment, malgré le fait que ces problèmes sont plus faciles à traiter que le problème original, ils restent malgré tout NP-Dur.

On formule chacun des deux problèmes comme un Mixed Integer Linear Problem (MILP) dans l'intérêt de tirer bénéfice des puissants solveurs qui sont disponibles de nos jours. Pour nos expérimentations, nous utilisons le solveur libre d'accès Gnu Linear Programming Kit (GLPK) [1].

Puisque tous les jobs sont à temps-réel ferme, soit un j est accompli à temps et génère une utilité v_j au système global, soit le job ne génère aucune valeur pour le système car il ne termine pas du tout ou termine en retard. Le problème de trouver un ordonnancement qui maximise la valeur cumulée totale est donc équivalent au problème de trouver un sous-ensemble ordonnançable de jobs qui maximise cette même valeur.

Par ailleurs, il est bien connu que pour tout ensemble de jobs ordonnançable, l'algorithme EDF est optimal. Ceci implique qu'un ensemble de jobs est ordonnançable si et seulement si le test exact d'ordonnançabilité par EDF [Chen, 2016] est satisfait :

Theorem 1. *Un ensemble de jobs apériodique est ordonnançable (par EDF) si et seulement si*

$$\forall\,(i,k)\,, a_i < d_k, \sum_{j:a_i \le a_j \ \textit{and}\ d_j \le d_k} c_j \le d_k - a_i$$

A partir de ce critère, nous dérivons directement une formulation MILP :
maximiser

$$\sum_i y_i v_i$$

sujet à

$$\forall\,(i,k)\,, a_i < d_k, \sum_{j:a_i \le a_j \text{ and } d_j \le d_k} (y_j c_j \le d_k - a_i)$$

où y_i est une variable de décision binaire, qui indique si le job i appartient au sous-ensemble de job ordonnançable ou non, et où a_i, d_i, c_i, v_i sont les paramètres du job tel que défini dans 2.1.

Lorsque l'on a trouvé une solution à ce problème, cela signifie que l'on a trouvé un sous-ensemble de job qui maximise le HVR. Pour obtenir un ordonnancement valide, il suffirait alors d'appliquer un

1. https://www.gnu.org/software/glpk/

algorithme EDF qui ignore les jobs qui ne font pas parti du sous-ensemble trouvé. C'est une façon pratique d'utiliser le simulateur pour générer les résultats à partir de cet ordonnancement.

Les résultats obtenus par simulation peuvent aussi être avantageusement utilisé pour vérifier d'éventuelles erreurs d'implémentation. Par exemple, on peut vérifier que la valeur cumulée totale obtenue dans la simulation est égale à la valeur d'utilité de base de chaque job agrégé par somme sur le sous-ensemble de jobs ordonnaçables obtenu, ou vérifier que tous les jobs de ce sous-ensemble sont accomplis à temps.

En dépit de ces simplifications, le problème est toujours trop complexe à résoudre dans certains cas. Trouver un ordonnancement optimal dans une quantité de temps raisonnable n'est pas toujours possible. Le solveur a donc été utilisé avec les paramètres suivants pour rechercher une solution sous-optimale :

1. "−−pcost" : branche utilisant l'heuristique hybride pseudo-coût,
2. "−−mipgap 0.02" : défini une tolérance d'écart relative de 2%, c'est-à-dire que l'on s'arrête lorsque une solution sous-optimale a été trouvée et qui est distante de moins de 2% de la solution optimale.

En utilisant ces paramètres, une solution a été trouvée en quelques secondes la plupart du temps, et cela a prit jusqu'à 15 minutes pour les instances les plus difficiles [2].

2.3 Génération de scénarios

Nous voulons générer un ensemble de scénarios pour produire des résultats expérimentaux qui permettront certaines analyses statistiques. Nous avons choisi de générer un ensemble de scénarios avec des jobs aux caractéristiques aléatoires en utilisant la méthode proposée dans [Aldarmi and Burns, 1999], méthode que nous avons légèrement adaptée pour permettre une augmentation de la couverture d'analyse et une meilleure transparence de nos travaux. En effet, dans [Aldarmi and Burns, 1999], les caractéristiques des jobs sont générées pseudo-aléatoirement selon certaines lois de distribution arbitraires. Il en résulte que les scénarios générés peuvent ne pas être représentatif des cas d'utilisation réels. Comme il nous est possible d'introduire un biais de sélection dans les données expérimentales que nous utilisons, et qu'il est tout à fait raisonnable de remettre en cause notre bienveillance, nous proposons d'étendre la méthode de génération aléatoire des scénarios.

Dans la méthode que nous proposons, la loi de distribution utilisée est choisie aléatoirement parmi un ensemble de lois possibles. Ce concept, illustré Figure 8, permet d'augmenter la diversité des scénarios générés, et donc la couverture d'évaluation. La diversité des scénarios générés, beaucoup plus riche, rend possible des analyses statistiques plus poussées pour investiguer la présence de biais de sélection que nous aurions pu introduire à notre insu dans la représentativité des scénarios générés.

2. En utilisant le solveur GLPK v4.65 sur un ordinateur équipé d'un processeur Intel Core i5-6440HQ et 2x4 Go de SDRAM DDR4-2133 fonctionnant sous Lubuntu 64 bits.

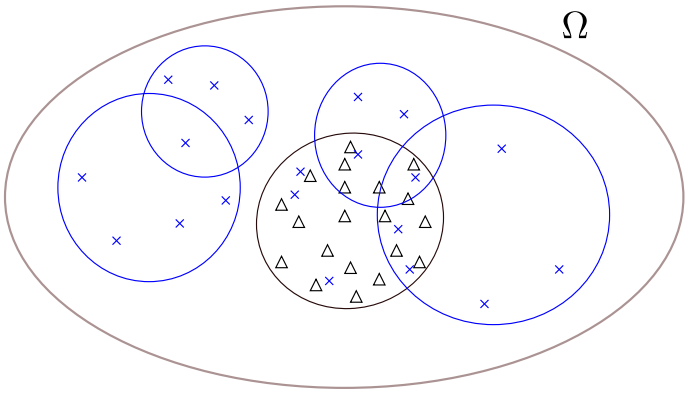


Figure 8 – *Extension proposée de l'algorithme de génération des scénarios utilisé dans [Aldarmi and Burns, 1999].*

Ω représente l'ensemble de tous les scénarios possibles. Les disques représentent les classes de scénarios. L'état de l'art est en noir [Aldarmi and Burns, 1999], notre contribution est en bleu. Les triangles noirs représentent une possible exécution de l'algorithme de l'état de l'art. Les croix bleues représentent une possible exécution de l'algorithme que nous proposons.

input :
- $N \in \mathbb{N}^*$: le nombre de jobs,
- σ : le taux de charge nominal.

output :
- J : un ensemble de N jobs temps-réel souple.

$J \leftarrow \emptyset$
$f_a \leftarrow X_a \sim EXP\left(\frac{\sigma}{E(X_c)}\right)$
Choix d'une classe de scénario
$f_c \leftarrow X_c \sim U(1,100)$ or $X_c \sim LU(1,100)$
$f_v \leftarrow Id(f_c)$ or $Inv(f_c)$ or $X_v \sim U(1,100)$ or $X_v \sim LU(1,100)$
$f_d \leftarrow X_d \sim U(1,10)$ or $X_d \sim U(1,200)$ or $X_d \sim U(100,200)$ or $X_d \sim LU(1,10)$ or $X_d \sim LU(1,200)$ or $X_d \sim LU(100,200)$
$f_\delta \leftarrow X_\delta \sim U(1,10)$ or $X_\delta \sim U(1,200)$ or $X_\delta \sim U(100,200)$ or $X_\delta \sim LU(1,10)$ or $X_\delta \sim LU(1,200)$ or $X_\delta \sim LU(100,200)$
Génère N jobs dans la classe choisie
$a_0 \leftarrow 0$
for $i \leftarrow 1$ **to** N **do**
  $a_i \leftarrow Round\left(\sum_{j=1}^{j=i} f_a(j)\right)$
  $c_i \leftarrow Round(f_c(i))$
  $v_i \leftarrow f_v(i)$
  $d_i \leftarrow Round(c_i + f_d(i))$
  $\delta_i \leftarrow Round(f_\delta(i))$
  $J_i \leftarrow < a_i, c_i, v_i, d_i, \delta_i >$
  $J \leftarrow J \cup J_i$
end

Algorithme 1 : Génération d'un scénario avec N jobs temps-réel souple et un taux de charge nominal visé λ.

Un scénario est un ensemble de jobs qui doivent être traités. Pour chaque scénario, les caractéristiques des jobs sont choisies pseudo-aléatoirement conformément à l'Algorithme 1.

L'algorithme contient deux parties principales :

1. lignes 4 à 7 (classe de scénario) : une classe de scénario est choisie aléatoirement, en choisissant les fonctions qui déterminent comment les caractéristiques des jobs seront générées. Ce bloc constitue la majeure partie des modifications introduites, comparé à la méthode utilisée dans [Aldarmi and Burns, 1999].
2. lignes 9 à 17 (génération des jobs) : un ensemble de N jobs est généré, conformément à la classe de scénario choisi, ce qui constituera un échantillon pour l'évaluation.

Dans l'Algorithme 1,
— $Id()$: la fonction *identité*,
— $Inv()$: la fonction *inverse*,
— $E(X)$: la *moyenne* (ou *la valeur espérée*) d'une variable aléatoire X,
— $CDF(X)$: la *fonction de distribution cumulée* d'une variable aléatoire X,
— $X \sim U(min, max)$: une variable aléatoire *uniforme* X, i.e. une variable aléatoire X telle que

$$CDF(X) = \begin{cases} 0 & \text{if } X < min \\ \frac{X-min}{max-min} & \text{if } min \leq X \leq max \\ 1 & \text{if } X > max \end{cases}$$

— $X \sim LU(min, max)$: une variable aléatoire *log-uniforme* X, i.e. une variable aléatoire X telle que

$$CDF(X) = \frac{ln(X) - ln(min)}{ln(max) - ln(min)}$$

— $X \sim EXP(\lambda)$: une variable aléatoire *exponentielle* X, i.e. une variable aléatoire X telle que

$$CDF(X) = 1 - e^{-\lambda X}$$

Nous utilisons la technique *d'échantillonnage par transformée inverse* pour générer des variables aléatoires uniformes, log-uniformes, et exponentielles
— $U(min, max) \sim (max - min) \times U(0, 1) + min$,
— $LU(min, max) \sim e^{U(ln(min), ln(max))}$,
— $EXP(\lambda) \sim -ln\left(U(0, 1)\right)/\lambda$.

Conformément aux hypothèses classiques [Aldarmi and Burns, 1999], les dates d'inter-arrivée d'une séquence donnée de jobs sont générées selon une variable aléatoire X_a qui suit une distribution exponentielle $EXP(\lambda), \lambda \triangleq \frac{\sigma}{E(X_c)}$, σ étant le taux de charge désiré (ligne 2 dans l'Algorithme 1). La date d'activation a_i est calculée à partir de la valeur arrondie (à l'entier le plus proche) des dates d'inter-arrivée des dates des jobs qui précèdent, job actuel inclus également (ligne 11).

Remarquons que la variance d'une variable aléatoire exponentielle est $E^2(X)$, tandis que la variance d'une variable uniforme est $\frac{(b-a)^2}{12} = \frac{(2(E(X)-a))^2}{12}$. Dans notre cas, $a > 0$, donc la variance d'une variable uniforme est strictement inférieure à $\frac{1}{3}E^2(X)$, i.e. strictement inférieure à $1/3$ de la variance d'une variable aléatoire exponentielle. Générer les dates d'inter-arrivée avec une variable aléatoire exponentielle a une probabilité plus élevée de générer des *pics* d'arrivées, i.e. un grand nombre de jobs arrivant dans une courte période de temps.

2.4 Modèle de simulation

Nous avons implémenté le simulateur utilisé dans ce travail en langage Python. La simulation consiste à exécuter les étapes suivantes en boucle, jusqu'à ce qu'une date de fin arbitraire soit atteinte :

1. Activer tous les jobs qui arrivent au cycle courant k.
2. Eliminer les jobs inutiles, c'est-à-dire les jobs dont l'utilité au cycle courant n'est pas strictement positive.
3. Obtenir une décision d'ordonnancement de l'algorithme et l'exécuter : si un job est planifié et qu'il ne s'agit pas du job en cours de traitement, alors on arrête le traitement du job courant (s'il y en a un), et on démarre (ou reprend) l'exécution du job planifié.
4. On laisse passer le temps jusqu'au prochain cycle $k + 1$, puis on met à jour les caractéristiques de tous les jobs en conséquence (unités d'exécution restantes, jobs accomplis, etc.).

Chapitre 3

Résultats et analyses

Les simulations ont été réalisées avec 1000 exécutions pour chaque valeur de taux de charge dans l'ensemble $\{25\%, 100\%, 400\%, 1600\%\}$. Chaque exécution consiste en 100 jobs générés avec l'Algorithme 1 et les paramètres suivants :

- $f_c()$, associée à la quantité d'unités d'exécution des jobs, qui est choisie avec équiprobabilité pour être soit une variable aléatoire uniforme dans $[1; 100]$, soit une variable aléatoire log-uniforme dans $[1; 100]$.
- $f_v()$, associée à la valeur des jobs, est choisie avec équiprobabilité pour être :
  - $Id(f_c)$, i.e. la valeur du job est égale à son nombre d'unités d'exécution. Dans ce cas particulier, la densité de valeur du meilleur cas possible $v_i/c_i = 1$ est la même pour tous les jobs.
  - $Inv(f_c)$, i.e. la valeur d'un job est égale à l'inverse de son nombre d'unités d'exécution. Contrairement au cas précédent, où la densité de valeur du meilleur cas possible était la même pour tous les jobs, ici la densité de valeur du meilleur cas possible $v_i/c_i = 1/c_i^2$ des jobs est exponentiellement distribuée.
  - Une variable aléatoire aléatoire uniforme dans l'intervalle $[1; 100]$.
  - Une variable aléatoire log-uniforme dans l'intervalle $[1; 100]$.
- $f_d()$, associée à la laxité du meilleur cas $d_i - c_i$ des jobs, est choisie avec équiprobabilité pour être une variable aléatoire uniforme, ou une variable aléatoire log-uniforme. Indépendamment à cela, l'intervalle de la variable aléatoire est choisi avec équiprobabilité pour être soit $[1; 10]$ pour des deadlines serrées uniquement, $[100; 200]$ pour des deadlines souples uniquement, ou $[1; 200]$ pour des deadlines souples à serrées.
- $f_\delta()$, associée au retard limite des jobs, est choisi de la même façon que $f_d()$, mais indépendamment.

Les algorithmes évalués sont Static Value Density (SVD), Semi Dynamic Value Density (SDVD), Dynamic Value Density (DVD1), Dynamic Value Density Squared (DVD2), Dynamic Timeliness Deadline (DTD1), et Dynamic Timeliness Deadline Squared (DTD2). Tous ces algorithmes sont de type glouton et fonctionnent comme suit. Au début de chaque pas de temps, l'algorithme décide du job à traiter jusqu'au prochain pas de temps. Pour cela, une fonction d'heuristique permet d'associer à chaque job actif un score d'heuristique, qui définit la priorité du job associé. Le job ayant la priorité (= score d'heuristique) la plus élevée sera traité jusqu'au prochain pas de temps. Si le job courant a le meilleur score d'heuristique, alors son traitement continu pour le prochain pas de temps. Sinon, le job courant est préempté par le job ayant le meilleur score d'heuristique (qui est strictement plus élevé que le job courant). Les algorithmes se différencient uniquement par la fonction d'heuristique qu'ils utilisent pour définir la priorité des jobs. La Table 2 décrit l'heuristique utilisée par chacun des algorithmes. Quatre algorithmes (Semi Dynamic Value Density (SDVD), Dynamic Value Density (DVD1), Dynamic Value Density Squared (DVD2), et Dynamic Timeliness Deadline Squared (DTD2)) proviennent directement de la littérature et ont été évalués dans [Aldarmi and Burns, 1999]. L'algorithme SVD a été considéré en supplément car, n'utilisant que des paramètres statiques, il n'a

besoin d'évaluer le score d'heuristique d'un job qu'une seule fois, à sa date d'activation. L'algorithme DTD1 a également été considéré en supplément car il est pour DVD1 ce que DTD2 est à DVD2, une amélioration de l'heuristique par une estimation plus précise de la valeur espérée générée par le job (par une estimation plus précise de la date espérée d'accomplissement du job).

Table 2 – *Algorithmes d'ordonnancement en-ligne évalués dans ce papier. Code couleur : paramètre statique, paramètre dynamique, paramètre dynamique avec anticipation sur le futur.*

Algorithme	**Heuristique (priorité $\propto$ valeur d'heuristique)**
SVD (Static Value Density)	v_i / c_i
SDVD (Semi Dynamic Value Density)	$\phi_i(t)$ / c_i
DVD1 (Dynamic Value Density)	$\phi_i(t)$ / $\overline{c}_i(t)$
DVD2 (Dynamic Value Density Squared)	$\phi_i(t)$ / $\overline{c}_i^2(t)$
DTD1 (Dynamic Timeliness Deadline)	$\phi_i\left(t+\overline{c}_i(t)\right)$ / $\overline{c}_i(t)$
DTD2 (Dynamic Timeliness Deadline Squared)	$\phi_i\left(t+\overline{c}_i(t)\right)$ / $\overline{c}_i^2(t)$

Notons qu'en pratique, la vitesse réseau varie et est difficilement prévisible. Les algorithmes gloutons ont l'avantage d'avoir une "vision très courte", car les décisions qu'ils prennent n'impliquent qu'un seul job dans l'horizon de temps, et donc il est attendu que ces algorithmes soient moins sensibles aux perturbations externes, comme par exemple une variation de la vitesse d'exécution ou des arrivées de jobs. Dans [Buttazzo et al., 1995], SDVD est dit afficher une très gracieuse dégradation pendant des surcharges et bénéficie d'une faible sensibilité à l'ensemble des paramètres des jobs. SVD, DVD1 et DTD1 ont été choisis pour leur ressemblance à SDVD. De plus, les algorithmes DVD2 et DTD2 ont été choisis car, dans [Aldarmi and Burns, 1999], il est montré que DVD2 est meilleur que DVD1 tandis qu'il a été envisagé que DTD2 est meilleur que DTD1.

Une attention particulière a été placée pour décider comment terminer un scénario de simulation. Bien que fastidieux et ne faisant pas partie de notre principale contribution, déterminer correctement la date de fin de la simulation dans l'objectif de maximiser la représentativité des résultats générés est obligatoire et n'est pas trivial. Une solution pourrait être d'exécuter la simulation jusqu'à la date de la dernière deadline souple. L'inconvénient de cette solution est que le taux de charge obtenu peut sensiblement différer du taux de charge visé.

Exemple 1. *Considérons un scénario avec $n = 100$ jobs, un taux de charge visé $\sigma = 1600\%$, et tous les jobs ayant les mêmes caractéristiques excepté pour la date d'activation. Tout job requiert le traitement de 10 unités d'exécution pour sa complétion et a une deadline souple relative égale à 90. La date d'activation des jobs est déterminé par une variable aléatoire exponentielle X_a avec un taux de tir $\lambda = \frac{\sigma}{E(X_c)} = 16/10 = 1.6$. L'espérance de la dernière date activation de job vaut $n\lambda = 160$ et, puisque la deadline souple d'un job vaut 90 pour tout job, l'espérance de la durée de simulation est de 250 unités de temps. Cependant, le nombre total d'unités d'exécution à traiter vaut $10n = 1000$, ce qui résulte en un taux de charge effectif de $1000/250 = 400\%$ qui est clairement différent de 1600%, le taux de charge visé.*

Une autre solution pourrait être d'exécuter la simulation jusqu'à la date finale $F = n\lambda$. Dans ce cas, le taux de charge obtenu serait exactement le taux de chargé visé, puisqu'en moyenne la dernière date d'activation vaut $F = n\lambda$. Malheureusement, certains jobs pourraient être infaisable, parce que leur date d'activation dépasserait tout simplement la date finale F, ou parce qu'ils seraient activés trop tard pour pouvoir être accompli avant la fin de la simulation.

Pour résoudre ce problème, nous proposons de supprimer les jobs infaisables, i.e. les jobs tels que la somme de leur date d'activation et du nombre d'unités d'exécution requis excède $n\lambda$. La conséquence de cela est que nous obtenons un taux de charge inférieur, comme dans le cas précédent, mais dont l'effet est beaucoup plus faible : la date finale serait égale à $100 \times 1.6 = 160$, et les jobs infaisables seraient les jobs activés après la date $160 - 10 = 150$, i.e. approximativement $10 \times 1.6 = 16$ jobs en moyenne. Le taux de charge effectif serait donc diminué de 16% du taux de charge initialement visé.

Un autre problème soulevé par la fin de simulation est que l'algorithme d'ordonnancement n'est pas conscient de l'interruption de la simulation, et peuvent prendre des mauvaises décisions à cause de cela. Des algorithmes d'ordonnancement peuvent être moins affectés que d'autre, comme les algorithmes qui priorisent le traitement de jobs plus courts. Pour remédier à cela, toutes les deadlines qui excèdent la date de fin de simulation F sont modifiées à F. En faisant cela, nous nous assurons que les algorithmes sont conscients du montant réel de temps disponible avant que les jobs perdent toute leur valeur, prévenant l'introduction d'un biais dans les résultats obtenus.

La Figure 10 affiche la performance en terme de HVR des algorithmes choisis pour chaque taux de charge dans l'ensemble $\{25\%, 100\%, 400\%, 1600\%\}$. Pour faciliter l'interprétation des résultats, nous avons également affiché le HVR moyen pour chaque algorithme et pour chaque taux de charge dans la Figure 9.

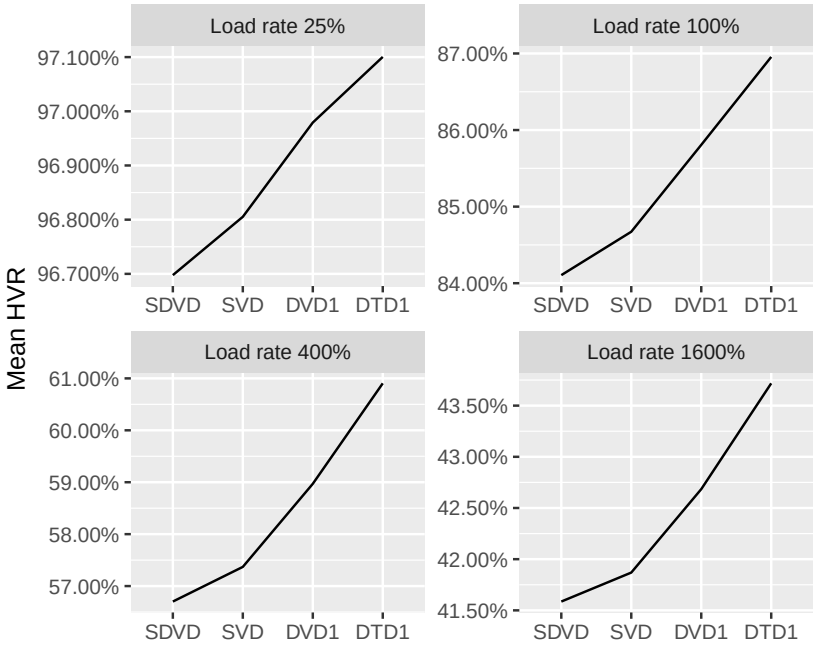


Figure 9 – *HVR moyen obtenu par différents algorithmes en-ligne sur* 1000 *scénarios pour chaque taux de charge.*

On observe que les algorithmes peuvent être ordonnés par efficacité, en regard du HVR moyen comme critère de performance, avec SVD < DVD1 < DTD1. De plus, cet ordre est valide indépendamment du taux de charge. SDVD, malgré le fait que cette heuristique utilise l'information dynamique (la valeur courante d'un job), est moins efficace sous tous les taux de charge que l'algorithme SVD qui utilise uniquement des informations statiques. Nous expliquons ce résultat par le fait que lorsqu'un job devient de plus en plus tardif, sa valeur décroît également de plus en plus et avec elle le score

d'heuristique de l'algorithme SDVD, augmentant le risque que le job devienne préempté par un autre, sans tenir compte du fait que sa valeur de densité courante soit très élevée ou non.

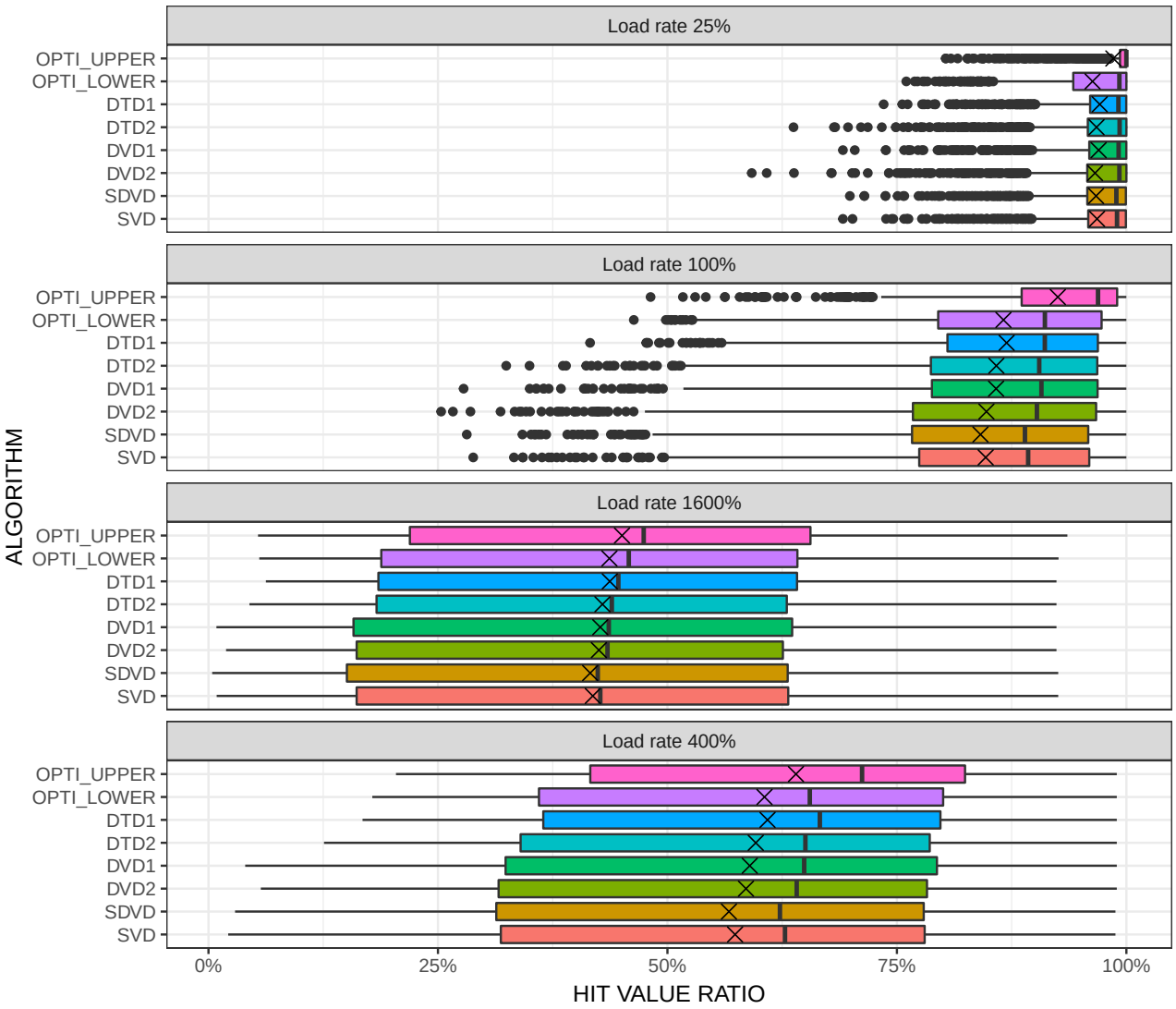


Figure 10 – *Hit Value Ratio mesuré sur des algorithmes en-ligne et clairvoyant, pour différents taux de charge.*

Pour chaque taux de charge, 1000 *scénarios sont utilisés. OPTI_LOWER (resp. OPTI_UPPER) correspond à l'algorithme clairvoyant optimal, estimé avec une erreur relative maximale de 2%, obtenu à partir du problème d'ordonnancement temps-réel ferme où les deadlines des jobs sont égales aux deadlines fermes (resp. deadline souple) du problème original. La valeur moyenne est indiquée par une croix. La moustache basse et haute (resp. la barre gauche et la barre droite d'une boîte) correspond au premier (resp. troisième) quartile. La moustache haute s'étend de la moustache jusqu'à la valeur la plus large qui n'est pas plus grande que* $1.5 \times IQR$ *de la moustache, l'Inter-Quartile Range (IQR) étant la distance entre le premier et le troisième quartile. La moustache basse s'étend de la moustache jusqu'à la valeur la plus faible qui est au plus* $1.5 \times IQR$ *de la moustache. Les valeurs aberrantes sont affichées individuellement, représentées par des disques noirs.*

Les heuristiques qui sont à la base de DVD1 et DTD1, respectivement $\phi_i(t)/\bar{c}_i$ and $\phi_i(t+\bar{c}_i)/\bar{c}_i$, où $\bar{c}_i$ représente le nombre d'unités d'exécution restantes $c_i - c_i(t)$ au temps courant, diffère uniquement dans la façon dont la valeur d'un job est calculée. Dans le cas où tous les jobs sont non temps-réel, $\forall i, d_i = +\infty$, $\phi_i()$ est une fonction constante et les deux heuristiques sont équivalentes. La conséquence de cela est que les deux algorithmes sont équivalents et doivent avoir la même efficacité. Dans un autre cas où les jobs doivent respecter des deadlines, plus les contraintes de temps sont "rigides", plus la différence de score entre les deux heuristiques est élevée. On peut donc s'attendre à ce que la différence de performance entre les deux algorithmes augmente avec la rigidité des contraintes.

Pour vérifier cela, nous avons évalué la performance des algorithmes DVD1 et DTD1 en faisant varier la limite de retard. Les résultats sont affichés Figure 11. Dans chaque scénario, 100 jobs à temps-réel ferme sont générés et tous ont la même valeur de retard limite. Un ensemble de 200 scénarios a été généré et simulé pour chaque valeur de retard limite 0, 10, 20, 40, 80, 160, 320, 640, 1280, 2560.

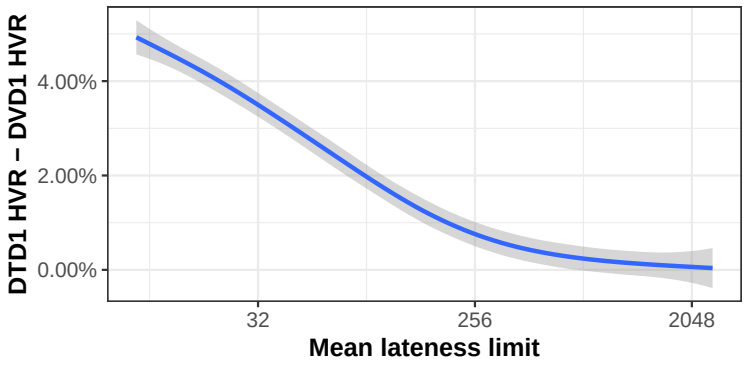


Figure 11 – *Ecart de performance entre les algorithmes DTD1 et DVD1 en fonction du retard limite moyen des jobs.*

HVR de l'algorithme DTD1 moins HVR de l'algorithme DVD1, lissé en utilisant la méthode LOESS, en fonction du retard limite moyen des jobs. 200 *scénarios ont été générés pour chaque valeur de retard limite* 0, 10, 20, 40, 80, 160, 320, 640, 1280, 2560, *avec un taux de charge de* 400% *et avec l'algorithme 1. Dans chaque scénario, les paramètres des jobs sont modifiés tel que pour tout i,* $d_i = c_i$ *et* δ_i *vaut la valeur limite de retard choisie. Finalement, la solution discutée dans la Section 3 est appliquée pour déterminer la date de fin des scénarios, supprimer les jobs infaisables (sans retard), et pour ajuster (lorsque nécessaire) les paramètres des jobs tel que la valeur absolue de deadline ferme et souple ne dépasse pas la fin de la simulation.*

Comme prévu, les résultats montrent que plus faible est la valeur de retard limite des jobs, plus élevée est la différence de performance entre les deux algorithmes. Comme dans les résultats précédents, DTD1 est plus efficace que DVD1, et nos résultats indiquent que cela est vrai indépendamment de la valeur du retard limite.

Dans [Aldarmi and Burns, 1999], il est montré que DVD2 est plus efficace que DVD1. Cela ne correspond pas à nos résultats où, en moyenne, DVD2 est moins efficace que DVD1, et cela pour tous les taux de charges. La même observation peut être faite en comparant DTD2 à DTD1. De plus, nous montrons dans l'Annexe que la différence de performance observée est statistiquement significative.

Notre explication à ce résultat inattendu est que l'analyse réalisée dans [Aldarmi and Burns, 1999] souffrirait du biais des coûts irrécupérables. Le biais des coûts irrécupérables correspond à une

tendance plus élevée à continuer un effort une fois qu'un investissement (en argent, en temps, etc.) irrécupérable a été réalisé [Arkes and Blumer, 1985]. Nous illustrons ce biais cognitif par un exemple concret directement cité de [Arkes and Blumer, 1985] :

> *A man wins a contest sponsored by a local radio station. He is given a free ticket to a football game. Since he does not want to go alone, he persuades a friend to buy a ticket and go with him. As they prepare to go to the game, a terrible blizzard begins. The contest winner peers out his window over the arctic scene and announces that he is not going, because the pain of enduring the snowstorm would be greater than the enjoyment he would derive from watching the game. However, his friend protests, "I don't want to waste the twelve dollars I paid for the ticket! I want to go!" The friend who purchased the ticket is not behaving rationally according to traditional economic theory. Only incremental costs should influence decisions, not sunk costs. If the agony of sitting in a blinding snowstorm for 3 h is greater than the enjoyment one would derive from trying to see the game, then one should not go. The $12 has been paid whether one goes or not. It is a sunk cost. It should in no way influence the decision to go.*
>
> - Extrait de[Arkes and Blumer, 1985], page 125.,

En effet, dans [Aldarmi and Burns, 1999], les auteurs expliquent que l'heuristique de DVD2 vise à "intensifier" l'effet de l'heuristique de DVD1, qui est que la priorité d'une tâche augmente lorsque le coût restant diminue, de façon à ce qu'un job qui s'exécute ait plus de chances de continuer jusqu'au bout, minimisant ainsi le "gâchis" des ressources systèmes (préalablement investies) :

> *Such behavior gives even higher preference, than DVD as described in function (2.2), to the tasks that have executed over the newly arriving tasks. Thus, preemption should be further lowered and resumption becomes more likely to happen before a task is aborted. In addition, an executing tardy-task is more likely to continue executing in order to minimize wasting the system's resources.*
>
> - Extrait de [Aldarmi and Burns, 1999], page 6.,

Cet objectif souffre précisément du biais des coûts irrécupérables. Nous pensons au contraire que l'intensification de l'effet de l'heuristique de DVD1 peut engendrer une perte d'efficacité dans un cas général, qui se reflète dans nos résultats.

Pour mieux montrer à quel point l'intensification de l'effet de DVD1 par l'élévation au carré de $\overline{c}_i(t)$ peut être néfaste, imaginons le cas concret suivant. Soit deux types de jobs : des jobs courts et des jobs longs. Supposons qu'un job court a les caractéristiques suivantes à sa date d'arrivée : $< v_s, c_s >$, et un job long a les caractéristiques suivantes : $< QKv_s, Kc_s >$, où $(Q, K) \in \mathbb{N}^2, Q > 1, K >> 1$. C'est-à-dire qu'un job long est K fois plus long à traiter qu'un job court, mais que la densité de valeur à la date d'arrivée d'un job long est Q fois plus grand que celle d'un job court. Un job court arrive pour chaque date dans l'ensemble $\{n \times c_s | n \in N\}$, et un job long arrive pour chaque date dans l'ensemble $\{n \times c_l | n \in N\}$. Notons h_s (resp. h_l) le score d'heuristique évalué à la date d'arrivée d'un job court (resp. d'un job long) pour l'algorithme DVD2 (ou DTD2). Clairement, le traitement de K jobs court requiert la même quantité de ressources que le traitement d'un job long, car $c_l = K \times c_s$. Néanmoins, le traitement d'un job long produit une utilité Q fois supérieure au traitement de K jobs court, puisque $v_l = Q \times K \times v_s$. Nous avons $h_s = v_s/c_s^2, h_l = (Q/K) \times (v_s/c_s^2)$. Dans ce cas particulier, quand $Q/K < 1$, c'est-à-dire quand K est suffisamment grand relativement à Q, l'algorithme DVD2 (ou DTD2) traite systématiquement les jobs courts au lieu des jobs longs.

Par exemple, supposons que les jobs courts ont pour caractéristiques $v_s = 4, c_s = 2$, et les jobs longs ont pour caractéristiques $v_l = 9 \times 10 \times v_s = 360, c_l = 10 \times c_s = 20$. 20 unités d'exécution sont requises pour traiter 1 job long ou 10 jobs courts, mais 1 job long génère une utilité égale à 360 alors que 10 jobs courts génèrent une utilité égale à 40. A quantité de ressources investies équivalentes, il est donc ici presque 10 fois plus rentable de traiter les jobs longs que de traiter les jobs courts. Pourtant, les algorithmes DVD2 (ou DTD2) prendront toujours la mauvaise décision de traiter les jobs courts, car $h_s = 4/2^2 = 1 > h_l = 360/20^2 = 0.9$.

Chapitre 4

Travaux associés

Douglas Jensen a proposé d'associer à chaque job une valeur exprimée en fonction du temps [Jensen et al., 1985], résultant en des fonctions de temps/d'utilité qui définissent précisément les sémantiques des systèmes à temps-réel souple. Les fonctions d'utilité utilisées dans nos travaux sont similaire à d'autres propositions [Buttazzo, 2011b] et se concentrent sur la simplicité pour des raisons de performance.

Dans ce papier, nous avons comparé la performance d'algorithmes d'ordonnancement en-ligne, en utilisant un algorithme optimal clairvoyant comme référence. Cette méthodologie a été utilisée dans plusieurs travaux de recherche antérieurs. [Baruah et al., 1992] prouve qu'aucun algorithme d'ordonnancement en-ligne ne peut avoir un taux de compétitivité strictement supérieur à 1/4 lorsque le taux de charge n'est pas borné. Les auteurs prouvent que 1/2 est une borne supérieure serrée pour des jobs avec aucune laxité dans un système à 2-processeurs.

[Koren and Shasha, 1995] décrit D^{over}, un algorithme d'ordonnancement en-ligne pour des systèmes uni-processeur surchargés. D^{over} fournit le meilleur taux de compétitivité qui peut être réalisé pour des algorithmes non-clairvoyants. Pour prouver des bornes supérieures et inférieurs sur les taux de compétitivité réalisables par un algorithme en-ligne, plusieurs techniques sont discutées dans [Karp, 1992].

De plus, [Karp, 1992] prouve qu'utiliser un taux de compétitivité comme métrique de performance est grossièrement équivalent à considérer un adversaire omniscient qui a une connaissance parfaite de l'algorithme et des ressources de calcul infinies.

Dans l'objectif d'identifier des algorithmes en-ligne qui ont une bonne performance en pratique, un affaiblissement de cet adversaire et/ou un renforcement de l'algorithme non clairvoyant évalué est suggéré. Il a été réalisé dans [Borodin et al., 1992, Fiat et al., 1991, Raghavan and Snir, 1989] que l'aléa pouvait (relativement) diminuer la puissance de l'adversaire puisque les décisions de l'algorithme en-ligne ne sont plus certaines. En accord avec cette idée, les algorithmes en-ligne aléatoires peuvent être considérés plus efficaces que les algorithmes déterministes, à l'encontre de différents types d'adversaires.

Dans [Kalyanasundaram and Pruhs, 2000], il est montré qu'augmenter modérément la vitesse du processeur utilisé par un algorithme non-clairvoyant donne effectivement à cet algorithme le pouvoir de clairvoyance. De façon plus intéressante, il est montré qu'il existe des algorithmes d'ordonnancement en-ligne avec des taux de compétitivité bornés pour toutes entrées, et qui ne sont pas vraiment corrélés avec la vitesse de processeur.

Dans [Lam et al., 2004], un renforcement de l'algorithme en-ligne est considéré. Un algorithme d'ordonnancement en-ligne est dit être *speed-s optimal* si l'algorithme peut correspondre à la performance d'un algorithme clairvoyant optimal avec le même nombre de processeurs mais en ayant une vitesse d'exécution s fois supérieure. A travers l'utilisation d'une approche analytique, il est démon-

tré que EDF-ac [1] atteint speed-2 (resp. speed-3) optimalité dans des systèmes uni-processeur (resp. multi-processeur) en condition de surcharge.

Dans [Baruah et al., 2012], le même facteur de vitesse est utilisé. L'algorithme d'ordonnancement Earliest Deadline First with Virtual Deadlines (EDF-VD) est démontré être *optimal* en respect de cette métrique. Comparé à notre étude, ces résultats sont basés sur une *Mixed-Criticality implicit-deadline sporadic tasks*, i.e. des jobs temps-réel fermes qui ont des contraintes spécifiques sur les dates d'inter-arrivée ainsi que des valeurs de deadline spécifiques.

Une étude comparative entre des algorithmes qui utilisent différentes affectations de priorité est présentée dans [Buttazzo et al., 1995]. L'analyse est basée sur les valeurs des jobs et les deadlines des jobs, mais également sur différents mécanismes de garanties, qui améliorent la performance d'un système temps-réel durant des conditions de surcharge. Les algorithmes sont comparés en utilisant la métrique HVR, pour deux ensembles de jobs différents où les caractéristiques des jobs diffèrent dans l'intérêt de voir à quel point un algorithme est sensible en respect des paramètres des jobs. Les auteurs observent que l'algorithme Highest Density First (HDF), basé sur la densité de valeur et correspondant à l'algorithme DVD1 dans notre papier, est le plus effectif dans des conditions de surcharge, affiche une dégradation très gracieuse pendant les surcharges, et n'est pas très sensible aux paramètres des jobs.

Dans [Aldarmi and Burns, 1999], une autre évaluation comparative entre les algorithmes d'ordonnancement en-ligne dans des conditions de surcharge est effectuée. Dans cette étude, les jobs sont à temps-réel souple et un ensemble unique de jobs avec des caractéristiques de jobs arbitraires a été utilisé pour l'évaluation de performance. Il est conclut que l'algorithme DVD2 est meilleur, et il est envisagé que DTD2 est un schéma efficace qui est plus adapté pour opérer sous toutes conditions de surcharge que SDVD et EDF.

1. algorithme EDF supplémenté par une simple forme de contrôle d'admission.

Conclusion

Pour optimiser le flux de données V2C au niveau des véhicules, nous avons d'abord montré que ce problème est analogue à un problème d'ordonnancement temps-réel souple. Dans notre contexte où toutes les données ne peuvent être transmises et sont générées spontanément et de façon imprévisible par les véhicules, il ne peut exister d'algorithme optimal. En effet, prendre des décisions optimales dans ces circonstances nécessite de connaître le futur, ce qui n'est pas possible en pratique. De nombreux algorithmes d'ordonnancement (non-optimaux) existent, mais leur performance dépend du ou des scénarios utilisés pour les évaluer. Pour choisir un algorithme efficace dans des conditions réelles méconnues, nous avons d'abord proposé un ensemble d'algorithmes adaptés. Il s'agit d'algorithmes simples donc adaptés pour être embarqués dans des véhicules, et robustes aux aléas du futur car prenant fréquemment des décisions qui concernent un horizon futur très court. Ensuite, nous avons proposé d'étendre une méthode d'évaluation expérimentale utilisée dans la littérature afin d'augmenter la diversité des scénarios générés, ce qui peut permettre d'augmenter la transparence des résultats et la couverture de l'évaluation. Nous avons également estimé la performance d'un algorithme optimal, qui connaît le futur et n'est donc pas implémentable en pratique, pour disposer d'une performance de référence à laquelle les algorithmes évalués peuvent être comparés. Notre évaluation expérimentale couvre des conditions de surcharge avec des taux de surcharges très élevés (jusqu'à 1600%) que l'on s'attend à rencontrer dans des cas réalistes. Nos résultats montrent qu'un algorithme est un bon candidat et sur-performe les autres algorithmes implémentables étudiés indépendamment du taux de surcharge.

Troisième partie

Mécanisme d'Allocation Centralisé dans le Cloud

Introduction

Dans cette partie, nous nous situons au niveau L2 par rapport à la décomposition du problème proposée partie I et illustrée Figure 3. Au niveau supérieur L3, des décisions sont prises périodiquement sur le long-terme. Le résultat de chaque décision est un budget alloué au coût de la remontée des données vers le Cloud sur une période de temps. Sur des périodes de temps plus courtes, il faut au niveau L2 déterminer la quantité optimale de chaque type de données à transmettre vers le Cloud, en veillant à ce que le coût engendré par le flux V2C n'excède pas le budget alloué au niveau L3.

Dans le Chapitre 1, nous précisons le problème que nous cherchons à résoudre. Il s'agit de réaliser la provision de données pour les services eHorizon, en conciliant les différents besoins qu'il ne sera pas possible de tous satisfaire entièrement en raison des contraintes économiques (viabilité du projet eHorizon) et techniques sur la fabrication des données (capacité de la flotte de véhicules à générer certains types de données et capacité du réseau cellulaire à les transmettre vers le Cloud). De plus, certains services peuvent être plus importants que d'autres, leurs besoins inconnus, difficiles à estimer, variables dans le temps, et nous souhaitons que la satisfaction des besoins individuels (= propres à chaque service) soit prioritaire face à la satisfaction des besoins globaux pour nous prémunir de situations de famines. Nous cherchons un moyen d'abstraire l'expression concrète des besoins, et de contrôler l'influence de chaque service sur le flux de données V2C. Le problème est illustré dans la Figure 1.

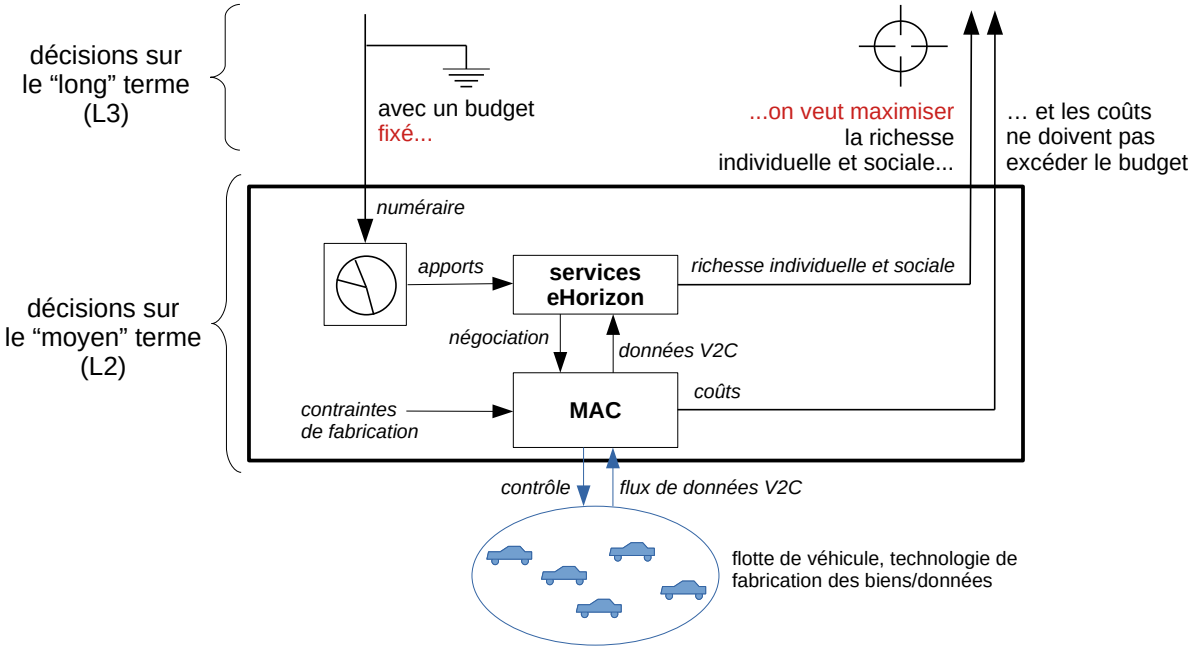


Figure 12 – *Problème de provision de données V2C.*

Un budget alloué à long-terme doit être respecté. Nous cherchons à maximiser simultanément la richesse générée individuellement par chaque service eHorizon, et la richesse générée pour l'ensemble du système (= richesse sociale). Les services négocient les données qu'ils convoitent au travers d'un fournisseur de données, le MAC. Le MAC alloue la quantité de données V2C que la flotte de véhicules doit remonter sur des périodes à moyen-terme.

Pour réaliser la négociation entre les services et le MAC, nous nous orientons vers une approche de contrôle basé-marché. Ce paradigme est en effet adapté pour réaliser le contrôle de systèmes complexes qui seraient autrement très difficiles à contrôler, maintenir, ou étendre.

> *"Market-Based Control is a paradigm for controlling complex systems that would otherwise be very difficult to control, maintain, or expand."*
>
> - Extrait de [Clearwater and Yeh, 1996], Preface.,

De plus, un marché permet de restreindre la consommation de ressources à la quantité de numéraire fourni [Clearwater and Yeh, 1996], ce qui nous intéresse pour restreindre les coûts engendrés par le flux de données V2C au budget alloué.

Un aspect remarquable d'un marché est qu'un effet désirable peut être obtenu au travers de simples règles d'intéractions. Le MAC défini précisément ces règles d'intéractions entre les services eHorizon et lui-même. Notre travail s'inscrit finalement dans le champ de la conception de mécanismes, puisqu'il s'agit de rechercher les règles du marché (= caractérisées par le MAC) qui permettent d'atteindre un effet désirable.

> *"A very abstract definition of a market is a system with locally interacting components that achieve some overall coherent global behavior. The fascinating aspect of a market is that through the simple interactions of trading, i.e. buying and selling, among individual agents a desirable global effect can be achieved, such as stable prices or fair allocation or resources. Markets don't guarantee an optimal solution but they often achieve satisfactory results. Their ability to facilitate resource allocation with very little information, i.e. price, makes them an attractive solution for many complex problems."*
>
> - Extrait de [Clearwater and Yeh, 1996], Preface.,

Dans la Section 1.2, nous faisons remarquer que les données sont des biens dits de clubs, qui possèdent en particulier la propriété de non-soustractibilité à la consommation. Cette propriété implique concrètement pour nous que la consommation de données stockées dans le Cloud par un service eHorizon ne réduit pas la quantité (de ces données) disponible pour les autres services eHorizon. Nous argumentons dans cette section de l'importance et la nécessité de prendre en compte cette propriété.

Un point crucial est de définir correctement ce qu'est l'effet désiré, car il existe une ambiguïté entre la richesse générée par chaque service et la richesse sociale générée par le système global, et un compromis est à faire entre la maximisation des deux. Nous proposerons 4 critères pour juger de la qualité d'un mécanisme de provision.

De très nombreux mécanismes ont été étudiés dans le champ de la conception de mécanisme, de l'économie, ou encore de la théorie des jeux. Cependant, en raison des nombreuses particularités de notre problème, il n'est pas possible de prédire l'effet obtenu par des mécanismes existants lorsqu'ils sont appliqués à notre problème. Par exemple, nous avons besoin de tenir compte des contraintes de capacité de production des données V2C de la flotte de véhicules, une contrainte rarement considérée dans la littérature.

Pour rechercher un mécanisme qui résout efficacement notre problème, nous proposons de construire une famille de variantes de mécanismes basés sur des principes similaires à plusieurs mécanismes existants.

Nous choisissons comme base du MAC les règles qui régissent les enchères simultanées à ballots-scellés, que nous introduisons Section 2, parce que ce type d'enchères est simple et permet une prise de décision rapide.

Nous nous basons sur un modèle générique utilisé par Norman dans [Norman, 2004], où un mécanisme est modélisé par un ensemble de règles. Nous proposons d'étendre ce modèle par l'ajout d'une règle supplémentaire, et introduisons le formalisme nécessaire à la construction d'une famille de 8 mécanismes, dont 3 sont des mécanismes similaires à ceux déjà existants et étudiés dans la littérature (mais dans un contexte différent).

Ensuite, nous recherchons un comportement décisionnel efficace des services eHorizon via l'utilisation de techniques d'apprentissage par renforcement. Les effets obtenus sont évalués au travers de 2 études de sensibilité judicieusement choisies pour dégager une connaissance qui permet de guider le choix du meilleur mécanisme.

Nous proposons de matérialiser le pouvoir d'influence par du numéraire, qu'il est pratique d'imaginer comme de la monnaie, distribué périodiquement aux services. Il est ensuite de la responsabilité de chaque service d'utiliser au mieux le numéraire possédé pour acquérir les données convoitées. Cette liberté d'action permet à un service d'implémenter des stratégies pour adapter dynamiquement son influence. Par exemple, un service peut économiser du numéraire quand ses besoins sont faibles, pour disposer d'une plus grande capacité d'influence quand ses besoins sont plus importants.

Nous supposons tout d'abord que le projet eHorizon va permettre d'améliorer les transports en générant une certaine richesse pour la société. Nous supposons également que la richesse générée pour la société est positivement corrélée avec les profits réalisés par Continental, puisqu'un utilisateur sera d'autant plus enclin à contribuer financièrement au projet eHorizon que la richesse qui est générée pour l'utilisateur est grande.

De nombreuses approches existent pour définir les prix dans les marchés : les offres à ballots-scellés, "reservation-style resource options", et "priority pricing". Dans [Waldspurger et al., 1992], un système dénommé Spawn utilise les enchères à ballots-scellés au second prix pour distribuer des ressources. Dans ce système, des tâches concourrent pour l'acquisition de ressources en soumettant des offres au propriétaire des ressources. Dans le système WALRAS, du nom de l'économiste français Léon Walras [Wellman, 1993], les agents peuvent soumettre des fonctions de demande qui spécifient la quantité demandée pour tout prix possible du bien [Wellman, 1996].

> *"Bids are demand functions, specifying the quantity demanded for any possible price of the good"*
>
> - Extrait de [Wellman, 1996], section 2.,

Dans nos travaux, nous utiliserons plutôt des fonctions de demande qui spécifient le prix pour toute quantité possible de bien.

> *"There have been several major approaches to setting prices in computational markets : sealed-bid auctions, reservation-style resource options, and priority pricing. Spawn [WHH+92] is an example of a system using second-price sealed-bid auctions to distribute resources. In Spawn, tasks compete for resources by submitting bids to the resource's owner. Bids can be expressed in a more complex manner than a simple price in systems like WALRAS [Wel96], where agents submit demand functions expressing the quantity desired at given prices."*

- Extrait de [Clearwater and Yeh, 1996], related work, page 198.,

Dans la plupart des études de mécanismes de provision de multiples biens de clubs, chaque agent a pour objectif de maximiser la richesse générée individuellement additionnée au numéraire qu'il cumule. Les agents prennent les *meilleures décisions* en vue de remplir cet objectif, et on parle alors d'agents (individuellement) rationnels. Le sens capturé ici par *meilleures décisions* est relativement ambigu, mais il n'est pas nécessaire de le définir très précisément pour la compréhension de nos travaux.

Dans notre contexte, le numéraire ne peut être utilisé par les agents que pour effectuer des transactions directes avec le mécanisme. Comme un agent ne peut pas l'utiliser pour autre chose que pour l'échanger avec le mécanisme pour générer de la richesse, un agent n'a aucun intérêt à cumuler le numéraire indéfiniment. Pour nous, il est plus pertinent que chaque agent ai pour objectif de maximiser la richesse générée individuellement, mais indépendamment du numéraire qu'il cumule. Cette différence d'objectif peut paraître anodine, mais peut pourtant conduire à un comportement radicalement différent.

Dans cette partie, nous proposons donc d'étudier des mécanismes de provision dans le cadre où chaque agent a pour objectif de maximiser la richesse générée individuellement, et indépendamment du numéraire qu'il cumule. Nous utilisons l'apprentissage multi-agents par renforcement pour simuler le comportement d'agents rationnels complexes. Un avantage est que les agents simulés ne seront pas affectés par les multiples biais cognitifs humains et qui rendent parfois leurs décisions irrationnelles, comme par exemple le biais des coûts irrécupérables, qui correspond à avoir une tendance plus élevée à continuer un effort une fois qu'un investissement (en argent, en temps, etc.) a été réalisé [Arkes and Blumer, 1985]. Notons néanmoins que les agents peuvent en contre-partie être affectés par d'autres biais. Les résultats obtenus sont discutés pour tenter de dégager des connaissances sur les mécanismes étudiés. Nous proposons également une nouvelle métrique, le rendement d'influence, qui rend compte de la relation entre la capacité d'influence potentielle donnée à un agent (= apport de numéraire de l'agent) et le degré d'influence que l'agent arrive effectivement à obtenir sur le mécanisme.

Les principaux résultats que nous dégageons de nos travaux sont listés ci-après :

— la redistribution de l'entièreté des surplus de paiements individuels peut inciter à un comportement non-sincère des agents et une allocation très déséquilibrée dans certains cas, alors que ce type de redistribution est classiquement connu pour l'effet inverse ; ne pas inciter à un comportement non-sincère des agents.

— un bénéfice social très intéressant est obtenu avec un mécanisme qui n'exclut jamais aucun agent à la consommation, c'est-à-dire où les biens sont considérés comme des biens publics. Avec un objectif classique, les agents adopteraient un comportement problématique de free-riding entraînant un bénéfice social médiocre. Le comportement de free-riding n'est d'ailleurs observé dans aucun des mécanismes étudiés, ce qui indique que la motivation à cumuler le numéraire indéfiniment est une source majeure du problème de free-riders.

— les deux mécanismes qui génèrent le plus grand bénéfice social sont, selon les cas, MCT-WR et NE-WR. Le mécanisme NE-WR peut être utilisé dans le cadre de la provision de biens publics car il ne requiert aucune exclusion à la consommation, contrairement à MCT-WR. De plus, MCT-WR impose un seuil de contribution minimum, variable, et incertain, aux agents qui souhaitent consommer le bien, ce qui peut engendrer une forme de regret et impose un degré de conformité à la contribution des agents qui n'existe pas avec NE-WR. Autrement dit, le mécanisme MCT-WR impose une coercition plus forte sur les agents, ce qui justifie une meilleure acceptabilité du mécanisme NE-WR.

Les résultats obtenus permettent d'établir une base de connaissance pour les mécanismes étudiés

lorsque l'objectif de chaque agent est de maximiser la richesse générée individuellement, et indépendamment du numéraire qu'il cumule.

Ce travail ouvre de nombreuses perspectives. Par exemple, cette base de connaissance peut être consolidée par la reproductibilité aisée des résultats, la possibilité de rechercher des agents cognitifs simulés plus performants, ou encore l'étude de nouveaux mécanismes.

La formalisation proposée permet une modélisation et une mise en oeuvre des expérimentations générique, qui rend possible une étude comparative systématique de nombreux mécanismes différents.

Nous proposons également un nouveau critère de comparaison, le rendement de l'influence, offrant une appréciation additionnelle de la qualité des mécanismes.

Chapitre 1

Définition du problème

1.1 Modélisation des acteurs et de leurs intéractions

Des services eHorizon consomment des données V2C centralisées dans le Cloud pour générer des services. Les services ainsi générés sont alors consommés par des clients.

Un mécanisme d'allocation centralisé dans le Cloud, appelé MAC, doit déterminer la quantité de chaque type de données V2C que la flotte de véhicules doit remonter vers le Cloud (= fabriquée) pour chaque période. Le MAC doit s'assurer que, sur le long-terme, le coût engendré par la fabrication des données n'excède pas le budget prévu.

Nous faisons ici abstraction du contrôle du flux V2C, que nous considérons parfait. Autrement dit, nous supposons que la quantité exacte de données V2C à remonter sur une période est effectivement remontée au cours de la période. En réalité, ce contrôle sera à réaliser au niveau L1, et doit faire l'objet de travaux supplémentaires.

Ainsi, le MAC doit réaliser plusieurs choses : déterminer 1) en quelle quantité doit être fabriqué chaque bien, 2) qui est autorisé à utiliser les biens fabriqués, et 3) qui paie combien.

Deux catégories d'acteurs sont impliqués dans ce processus. D'une part, des acteurs internes, qui sont :

1. le Mécanisme d'Allocation Centralisé (MAC), capable de fabriquer des biens, les données V2C, pour un certain coût, et
2. des *agents* (= les services eHorizon), qui utilisent les données V2C du Cloud fournies par le mécanisme pour fabriquer d'autres biens, les *services*. Par exemple une estimation du trafic ou une estimation de la qualité de la route.

D'autre part, des acteurs externes, qui sont des *clients*. Par exemple un transporteur qui utiliserait les informations du trafic pour optimiser ses livraisons. Les acteurs internes constituent ce que l'on appelle un *groupe*. Chaque client contribue financièrement et volontairement au groupe, en échange de quoi le groupe fournit des services aux clients. Nous supposons que le degré de contribution de l'ensemble des clients dépend de la *satisfaction*, ou *richesse*, que les services eHorizon leurs apportent.

La relation entre les différents acteurs du système est illustrée Figure 13.

Le mécanisme et les agents ont un objectif commun : celui de maximiser le *profit* du groupe auquel ils appartiennent. La difficulté est que le profit du groupe est lié à la quantité de matières premières fabriquées par une chaîne de relations complexes, chaîne qui est illustrée Figure 14.

Maximiser le profit du groupe passe par deux objectifs conflictuels, maximiser le revenu du groupe et minimiser le coût de fabrication, qui dépendent tous deux des matières premières fabriquées. D'un côté, la relation avec le coût de fabrication paraît facile à modéliser. Par exemple, il est possible de

considérer un coût marginal constant, comme dans [Clarke, 1971, Moulin, 1994], c'est-à-dire que le coût est proportionnel à la quantité de bien fabriqué. Mais d'un autre côté, la relation avec le revenu du groupe est complexe à modéliser.

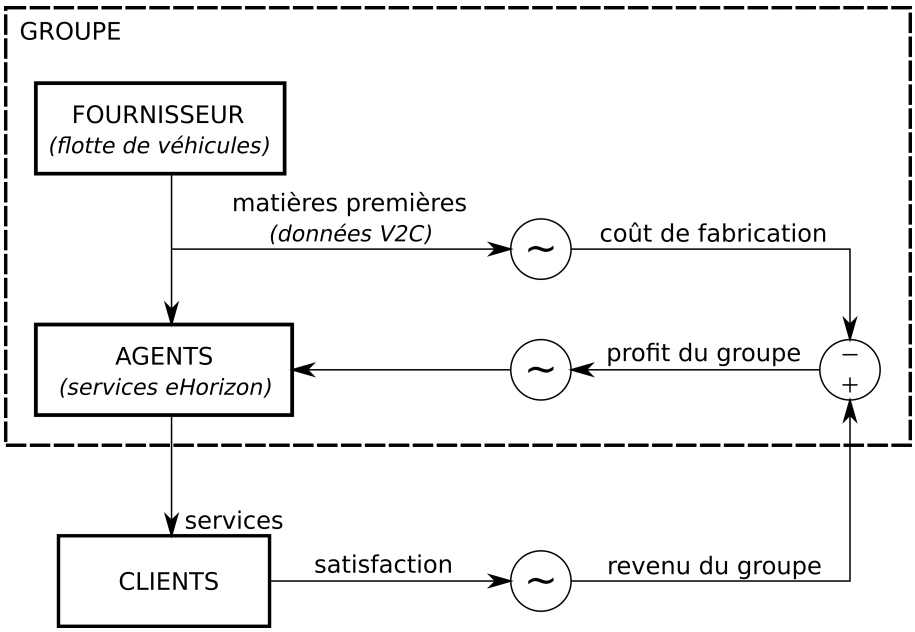


Figure 13 – *Relation entre les différents acteurs.*

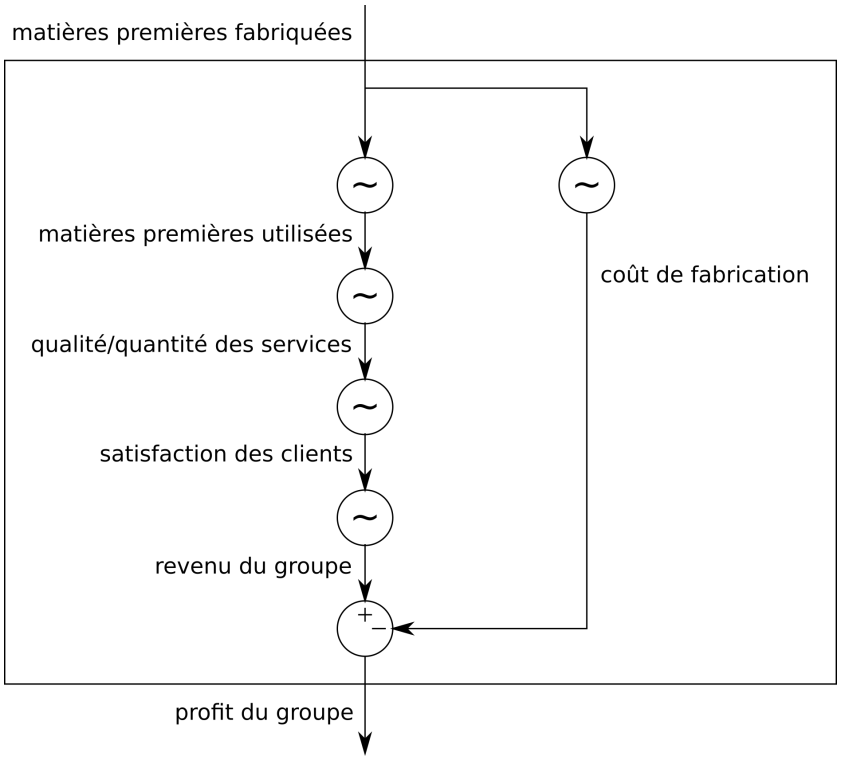


Figure 14 – *Chaîne de relations entre les matières premières fabriquées et le profit du groupe.*

Pour simplifier le problème, nous supposons que la part de contribution de chaque service eHorizon au revenu d'eHorizon est connue. Nous faisons également une hypothèse d'isolation, c'est-à-dire que la part de contribution de chaque service au revenu du groupe est indépendante des autres services.

Enfin, on suppose que la richesse totale est égale à la somme de la richesse individuellement générée par chaque service (principe de superposition). Ces simplifications nous amènent à la chaîne de relations représentée Figure 15.

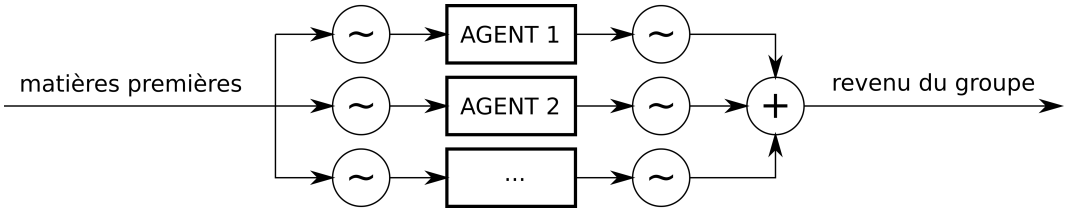


Figure 15 – *Chaîne de relations simplifiée entre les matières premières fabriquées et le revenu du groupe.*

A priori, les agents sont les mieux placés pour déterminer le type et la quantité de matières premières dont ils ont besoin pour générer des services. Nous considérons donc que le processus de décision de production des matières premières doit être, au moins en partie, décentralisé sur les agents. On fait le choix de conférer aux agents la liberté d'influencer la fabrication des matières premières par le mécanisme en les dotant d'une capacité d'influence. Nous associons la capacité d'influence d'un agent à une quantité de numéraire pour deux raisons. Premièrement, la relation entre l'argent et la capacité d'influence, largement répandue dans notre société, nous est familière et nous confère donc une certaine *pensée intuitive*. Deuxièmement, cela nous permet de nous rapprocher de nombreuses études qui concernent les mécanismes de provision de biens en théorie de l'économie.

Sous cet angle, nous considérons un marché dans lequel les agents sont des acheteurs qui rivalisent entre eux et négocient avec un unique vendeur, le mécanisme, dans le but d'acquérir des matières premières. L'idée sous-jacente est que le revenu du groupe pourra être maximisé en distribuant stratégiquement une quantité de numéraire à chaque service, et en choisissant un ensemble approprié de règles qui définit précisément l'interaction entre les services et le mécanisme.

1.2 Non-soustractibilité des données

Il est important de préciser que, selon la classification introduite par Ostrom et al. [Ostrom et al., 1977] illustrée Figure 16, les données font partie de la catégorie des biens de clubs car elles satisfont les deux propriétés suivantes :

1. non-soustractibilité (ou faible soustractibilité) à la consommation : la consommation d'une quantité de bien par un agent n'affecte pas (ou affecte peu) la quantité disponible à la consommation pour les autres agents,
2. (forte) excluabilité à la consommation : il est facile d'exclure des agents à la consommation du bien.

Un type d'enchères à la fois très populaire et très simple est celui des enchères à ballots-scellés au premier prix, First Price Sealed-Bid Auction (FPSBA), dans lequel le bien est remporté et payé par l'agent qui propose la meilleure offre. Ce type d'enchères, qui est pertinent dans le cas d'un bien

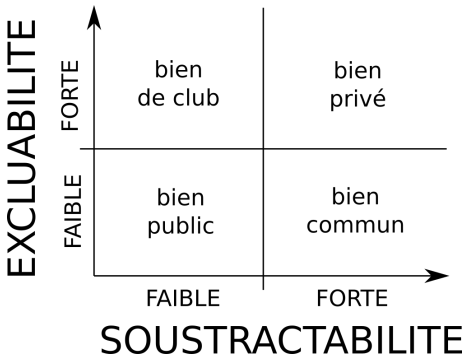


Figure 16 – *Différents types de bien selon la classification proposée par Ostrom et al. [Ostrom et al., 1977].*

soustractible à la consommation, ne l'est pas vraiment pour un bien non-soustractible. En effet, avec FPSBA, un seul agent est inclus à la consommation du bien tandis que tous les autres en sont exclus. Mais, puisque les matières premières sont non-soustractibles à la consommation, pourquoi donc exclure des agents qui pourraient en bénéficier gratuitement, sans aucun coût supplémentaire pour le groupe ? Cette question est légitime et a déjà été posée, au moins depuis 1958 par Samuelson :

> *"For what, after all, are the true marginal costs of having one extra family tune in on the program ? They are literally zero. Why then prevent any family which would receive positive pleasure from tuning in on the program from doing so ?"*
>
> - Extrait de [Samuelson, 1958], page 335.,

Prenons un exemple concret pour souligner l'importance de la propriété de non-soustractibilité. La conception du clip vidéo "Luis Fonsi - Despacito ft. Daddy Yankee" a engendré un certain coût. Mais ensuite, après avoir été publié sur la plateforme Youtube en 2017, le clip a été vu (ou consommé) plus de 7 milliards de fois. La propriété de non-soustractibilité d'un bien permet la démultiplication de la richesse générée par le bien par la démultiplication de la consommation du bien.

Nous pouvons aussi utiliser l'économie actuelle pour témoigner de l'importance de cette propriété. Standard and Poor's 500 (SP500) est un indice boursier basé sur 500 grandes sociétés américaines cotées en bourses, et 73 des entreprises qui constituent cet indice font partie du secteur des technologies d'information [SPGlobal, 2021]. Ce secteur se compose d'entreprises qui produisent des biens soustractibles à la consommation, comme du matériel ou des équipements semi-conducteurs, mais aussi et surtout d'entreprises qui produisent des biens non-soustractibles à la consommation, comme des logiciels et des services liés à l'information [ValuePenguin, 2021]. La Figure 17 illustre l'évolution, sur les 10 dernières années, de la valeur de l'indice S&P500, et la valeur de l'indice S&P500 Information Technology (Sector) du secteur des technologies d'information. On peut constater que l'indice S&P500 Information Technology (Sector), associé aux 73 entreprises du secteur des technologies d'information, présente une croissance significativement plus importante (environ 2 fois supérieure) que l'indice S&P500, associé aux 500 plus grandes entreprises de l'économie américaine. Ceci nous semble révélateur de la richesse importante que peuvent générer les biens non-soustractibles à la consommation.

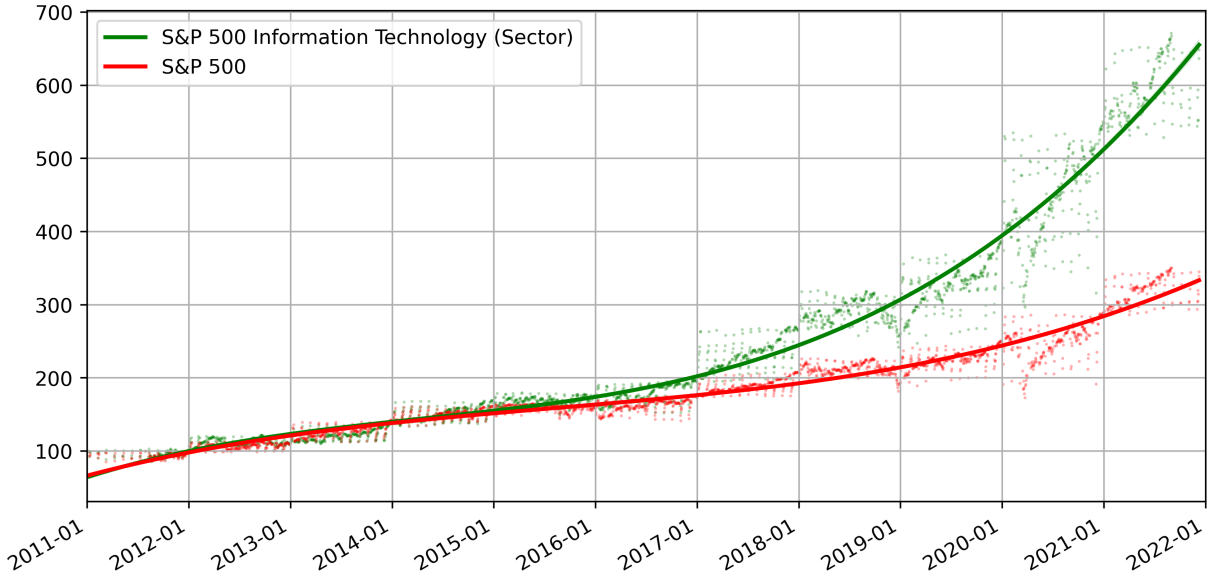


Figure 17 – *Évolution historique sur les 10 dernières années des indices boursiers S&P500 et S&P500 Information Technology (Sector)*

L'indice S&P500 est constitué des 500 plus grandes entreprises basées aux USA. L'indice S&P500 Information Technology (Sector) est composé des 73 entreprises constituant l'indice S&P500 et qui appartiennent au secteur des technologies d'information. La valeur des deux indices est normalisée à 100 à la date la plus ancienne. Une régression polynomiale d'ordre 3 a été utilisée pour montrer la tendance d'évolution des deux indices. Données utilisées sourcées de https://www.spglobal.com le 1 Septembre 2021.

1.3 Exclusion à la consommation

Dans la littérature, une justification à l'exclusion d'un bien de club est que l'exclusion peut limiter le problème bien connu de *free-riders* [Moulin, 1994] [1]. Le problème de free-riders est un type de défaillance du marché qui se produit lorsque la plupart des agents deviennent des free-riders, reconnaissant et exploitant l'opportunité de sous-payer un bien dont ils bénéficient [Rittenberg, 2008].

En fait, une des racines du problème de free-riders se trouve dans la définition de l'objectif des agents. Très souvent, un agent cherche à maximiser la valeur générée par les biens consommés, additionnée à la quantité de numéraire cumulée par l'agent. Cependant, dans notre contexte, l'objectif de l'agent est uniquement de maximiser la valeur générée par les biens consommés, indépendamment du numéraire qu'il cumule. En conséquence, un agent n'a pas d'intérêt à maximiser le numéraire qu'il cumule. Bien au contraire, un agent a plutôt intérêt à échanger le numéraire avec le mécanisme pour contribuer à la fabrication des biens qui lui bénéficieraient. Cette différence d'objectif, minime mais pourtant fondamentale, implique que les effets obtenus par des agents en intéraction avec un mécanisme de provision dans de nombreux travaux de la littérature sont susceptibles d'être radicalement différents des effets obtenus dans notre cas.

1. Ce concept a été initialement nommé *free rider problem*, et malgré que les ouvrages spécialisés de ce problème utilisent la traduction "problème du passager clandestin" [Gérard-Varet, 1998, Milgrom and Roberts, 2016, Sève, 2013], cette traduction ne permet pas, seule, de faire clairement référence au concept de free-rider. Nous choisissons donc de ne pas traduire la dénomination originale de ce problème.

Conclusion

Dans ce chapitre, nous avons précisé le cadre du problème que nous cherchons à résoudre en posant un certain nombres d'hypothèses. Nous avons identifié deux caractéristiques majeures de notre problème qui, ensemble, rendent notre problème singulier au regard de la majorité des travaux existants dans la littérature.

Il s'agit d'une part de la propriété de non-soustractibilité des données, qui veut que la consommation de données ne réduit pas la quantité de données disponible pour les autres. Nous avons justifié l'importance critique de cette propriété, qui permet la multiplication de la richesse générée par des données fabriquées par la multiplication de leur consommation, pour un coût de consommation très faible devant le coût de fabrication.

D'autre part, l'objectif des services eHorizon diffère de l'objectif usuel qui consiste à maximiser la richesse générée par les biens consommés additionnée au numéraire cumulé. Ici, l'unique intérêt des services eHorizon est de maximiser la richesse qu'ils génèrent individuellement. Cette différence d'objectif implique que l'exclusion à la consommation des biens fabriqués, qui peut trouver du sens pour palier au problème de free-riders, est difficile à justifier dans notre cas.

De ce fait, il est difficile de prédire la performance des mécanismes de provision étudiés dans la littérature, ce qui nécessite donc de rechercher un mécanisme approprié à notre problème. Pour cela, nous proposons de construire une multitude de mécanismes, que nous évaluerons pour sélectionner le meilleur.

Dans le Chapitre 2, nous introduisons les enchères simultanées à ballots-scellés, un type d'enchères très populaire qui constituera la base des mécanismes que nous étudierons. Nous introduisons ensuite le formalisme nécessaire à la construction d'une famille de variantes de mécanismes basés sur des concepts similaires à des mécanismes existants.

Dans le Chapitre 3, nous proposons un moyen pour analyser les effets obtenus avec les différents mécanismes. En raison de la difficulté de l'analyse, nous nous orientons vers une approche expérimentale qui gagne en popularité, l'approche Agents-based Computational Economics (ACE). Dans notre cas, cette approche consiste à tirer profit des techniques d'apprentissage par renforcement pour apprendre aux services eHorizon à révéler stratégiquement et efficacement leurs besoins aux mécanismes. Les résultats obtenus avec des stratégies stables et efficaces sont ensuite discutés et analysés pour guider le choix d'un mécanisme parmi plusieurs étudiés.

Chapitre 2

Construction d'une famille de mécanismes à étudier

Introduction

Nous devons optimiser la provision itérée (ou répétée) de multiples biens non-binaires (dont la quantité s'exprime dans $\mathbb{N}_+$), en prenant aussi en compte de potentielles contraintes sur la capacité de fabrication des biens. Dans la littérature, les mécanismes sont souvent définis dans un cadre plus restrictif que le nôtre, par exemple pour la provision d'un unique bien non-binaire [Clarke, 1971] ou pour la provision de multiples biens binaires [Guo and Conitzer, 2010] (dont la quantité possible s'exprime dans $\{0, 1\}$, c'est-à-dire que le bien n'a que deux états possibles "fourni" ou "non-fourni"). Nous proposons donc de définir et étudier des mécanismes qui sont basés sur des principes similaires à des mécanismes déjà étudiés dans la littérature, mais qui peuvent s'appliquer dans notre cadre plus général.

Contributions. Nous définissons un nouveau formalisme qui permet de rapprocher des mécanismes de provision déjà existants et différents, mais qui présentent des concepts en commun. Dans la Section 2.1, nous introduisons les bases du formalisme permettant de définir des règles qui représentent les concepts. Les mécanismes peuvent alors être construits comme un ensemble de règles, règles qui sont introduites Section 2.2 et Section 2.3.

Enfin, nous montrons dans la Section 2.4 que 3 mécanismes populaires s'inscrivent dans notre formalisme. D'autres mécanismes connus pourraient également facilement s'inscrire dans notre formalisme par la définition de nouvelles règles.

2.1 Formalisme préliminaire

Soit AGENTS un ensemble fini d'agents, et BIENS un ensemble fini de biens de clubs. Un unique mécanisme MECH est doté d'une technologie qui lui permet de fabriquer chaque bien, pour un coût de fabrication proportionnel à la quantité de bien fabriqué et indépendant du bien, c'est-à-dire que le coût marginal de fabrication est constant et identique pour tous les biens.

Les agents sont dotés d'un bien privé, le numéraire, qu'il est commode d'imaginer comme une monnaie. Les agents peuvent échanger du numéraire avec MECH pour acquérir des biens. Une propriété classique et souvent considérée en conception de mécanisme est la propriété de participation volontaire. Le sens capturé par "volontaire" (ou "participation volontaire") est parfois discuté :

> *"We argue that previous treatments fail to recognize the full meaning of "voluntary"."*
>
> - Extrait de [Dixit and Olson, 2000], abstract.,

Pour définir précisément le sens capturé par la propriété de participation volontaire, nous introduisons la notion de Paiement Maximal Acceptable (PMA).

Définition 5 (Paiement Maximal Acceptable (PMA))**.** *Le Paiement Maximal Acceptable (PMA) est une fonction associée à un bien qui donne la quantité de numéraire maximale qu'un agent consent à payer au mécanisme en échange d'une quantité de ce bien, c'est-à-dire en échange d'une quantité de bien fabriquée par MECH et que l'agent est autorisé à consommer.* $Domaine(PMA) = \mathbb{N}_+$, $Codomaine(PMA) = \mathbb{R}_+$.

Définition 6 (Participation volontaire)**.** *Un mécanisme satisfait la propriété de participation volontaire si et seulement si le paiement réalisé par un agent n'est jamais supérieur au Paiement Maximal Acceptable révélé par l'agent au mécanisme.*

La propriété de participation volontaire n'est pas nécessaire dans notre cadre. En effet, nous pourrions choisir la facilité de faire preuve d'autoritarisme en imposant la participation des services eHorizon. Mais cette solution n'est pas toujours possible ou désirable. Pour augmenter la portée de nos travaux et l'intérêt des mécanismes étudiés, nous nous limitons à étudier des mécanismes qui satisfont la propriété de participation volontaire.

Les agents ne sont pas nécessairement sincères, c'est-à-dire qu'ils sont libres de révéler stratégiquement au mécanisme le Paiement Maximal Acceptable (PMA) pour chaque bien. Par exemple, un agent est libre de révéler un PMA nul pour un bien, malgré le fait que ledit bien puisse lui bénéficier de façon strictement positive. Un agent sincère, au contraire, révélerait un PMA strictement positif et justement proportionné au regard du bénéfice généré.

Les agents communiquent simultanément au mécanisme le PMA qu'ils consentent de payer en échange d'une certaine quantité de données V2C, et cela pour chaque type de données. Cette information est privée aux agents et révélée uniquement au mécanisme. Ce processus est similaire à celui des enchères simultanées à ballots-scellés, un type d'enchères très étudié dans la littérature et qui constitue donc une bonne base de travail pour nous.

De plus, pour éviter d'avoir à gérer des cas où un agent pourrait être endetté, on considère qu'un agent ne peut pas révéler une offre qu'il ne peut pas payer. On dit dans ce cas que les agents sont à budget contraint, ce qui se traduit plus formellement par la contrainte suivante :

$$\forall agent \in AGENTS, NUMERAIRE(agent) \geq \sum_{bien \in BIENS} max(PMA_{bien}(\cdot))$$

Compte tenu des PMA révélés par les agents au mécanisme, le mécanisme doit 1) déterminer en quelle quantité chaque bien est fabriqué, 2) déterminer comment les biens fabriqués sont distribués aux agents, et 3) déterminer les transferts de numéraires entre le mécanisme et les agents, qui peuvent avoir lieu dans un sens comme dans l'autre. Inspiré par [Norman, 2004, Fang and Norman, 2010], nous modélisons un mécanisme comme un ensemble de 4 règles :

1. une règle de fabrication, notée FAB, qui détermine en quelle quantité chaque bien est fabriqué. La fabrication d'une quantité x d'un bien engendre un coût de fabrication $COUT(x)$. $Codomaine(FAB) = \mathbb{N}_+^{\#BIENS}$. Dans notre cas, le coût sera linéaire en la quantité de bien fabriquée, mais ce choix n'est pas une restriction imposée par le modèle.
2. une règle de paiement, notée PAY, qui détermine le paiement réalisé par chaque agent au mécanisme. Chaque agent possède une quantité de numéraire, qui peut être utilisé pour réaliser un paiement au mécanisme. $Codomaine(PAY) = \mathbb{R}_+^{\#AGENTS}$.
3. une règle d'inclusion, notée INC, qui détermine pour chaque paire (agent, bien) si l'agent est autorisé ou non à consommer le bien. Autrement dit, nous nous autorisons avec cette règle la possibilité que le mécanisme exclut une partie des agents à consommer des biens fabriqués. $Codomaine(INC) = \{0,1\}^{\#AGENTS \times \#BIENS}$.
4. une règle de redistribution, notée RED, qui détermine pour chaque agent le paiement réalisé par le mécanisme à l'agent. Concrètement, cette règle permet de redistribuer une partie du numéraire payé préalablement par un ou plusieurs agents au mécanisme, par exemple lorsque le paiement est considéré comme excessif. Le lecteur intéressé pourra trouver dans [Marks and Croson, 1998] une étude de trois règles de redistribution. $Codomaine(RED) = \mathbb{R}_+^{\#AGENTS}$.

Remarque 1. *Le domaine de chaque type de règle n'est pas défini ici car il peut dépendre de la règle utilisée. Par exemple, le paiement d'un agent peut dépendre des PMA révélées par l'ensemble des agents, ou non.*

Ce modèle a l'avantage d'être très modulaire et de permettre l'expression de nombreux mécanismes, en particulier trois mécanismes bien connus, ce que nous montrerons par la suite.

L'intérêt d'ajouter une règle de redistribution au modèle proposé dans[Norman, 2004, Fang and Norman, 2010] est que cela permet de simplifier l'expression des règles de paiement, et nous justifions par la suite ce que cela apporte.

En effet, un mécanisme dont la règle de paiement est particulière peut être exprimé de façon équivalente avec une règle de redistribution particulière et une règle de paiement plus simple. Prenons pour exemple des enchères de type Vickrey, où un unique bien binaire est attribué à l'agent le plus offrant, mais au prix du deuxième plus offrant. Soit deux agents $agent1$,$agent2$, qui révèlent au mécanisme $PMA_{agent1}(0) = PMA_{agent2}(0) = 0$, et $PMA_{agent1}(1) = 3$, $PMA_{agent2}(1) = 7$. 7 est l'offre la plus élevée, $agent2$ remporte donc le bien. On peut considérer deux modèles équivalents :

1. règle de paiement particulière sans redistribution : $agent2$ paie la deuxième offre la plus élevée $PMA_{agent1}(1) = 3$.
2. règle de paiement simple avec redistribution particulière : $agent2$ paie son offre $PMA_{agent2}(1) = 7$, et le mécanisme redistribue à l'agent la différence entre l'offre la plus élevée et la deuxième offre la plus élevée, $PMA_{agent2}(1) - PMA_{agent1}(1) = 7 - 3 = 4$. Au final, comme pour le premier modèle, $agent2$ obtient le bien en échange de $7 - 4 = 3$ unités de numéraire.

L'inconvénient du premier modèle est que, comme la règle de paiement est particulière, la satisfaction de la propriété de participation volontaire du mécanisme dépend de la règle de paiement.

Au contraire, lorsqu'un agent paie exactement le paiement maximal acceptable qu'il a révélé au mécanisme, comme c'est le cas dans le deuxième modèle, la propriété de participation volontaire est nécessairement satisfaite, indépendamment de la règle de redistribution utilisée. Un deuxième avantage de l'utilisation d'une règle de paiement simple est que cela facilite la détermination de la quantité de matières premières à fabriquer lorsque cette décision dépend de la règle de paiement, mais ne dépend pas de la règle de redistribution. Ceci justifie l'intérêt de l'extension du modèle par l'ajout d'une règle de redistribution que nous proposons.

Pour les raisons que nous venons d'évoquer, nous choisissons de définir une règle de paiement très simple et intuitive, et qui sera à la base de tous les mécanismes que nous étudierons.

Définition 7 (Règle de paiement ID (= IDentité))**.** *Règle de paiement où le paiement d'un agent pour une quantité de bien x est donné par $Id(PMA(x))$. $Domaine(ID) = \mathbb{R}_+$, $Codomaine(ID) = \mathbb{R}_+$. Deux propriétés de cette règle découlent directement de la définition du PMA :*

- *cette règle garantit la satisfaction de la propriété de participation volontaire.*
- *cette règle maximise, pour une quantité de chaque bien consommé par chaque agent, les paiements réalisés par les agents au mécanisme sous la contrainte de la propriété de participation volontaire.*

Nous souhaitons que le mécanisme détermine les quantités de biens fabriqués qui maximisent le profit du groupe. On fait l'hypothèse que, lorsque les agents sont sincères, le PMA total est égal au revenu du groupe. Le PMA total moins coût de fabrication correspond donc au profit du groupe. Cependant, lorsque les agents ne sont pas sincères, la correspondance entre le PMA total et le revenu du groupe est "brouillée", et la correspondance entre le PMA total et le profit du groupe est "brouillée" également. On prend donc la précaution de faire correspondre le PMA total au *profit révélé du groupe*, plutôt qu'au profit du groupe.

Définition 8 (Profit Révélé du Groupe (PRG))**.** *Le Profit Révélé du Groupe (PRG) vaut le PMA total moins le coût de fabrication total.*

Nous devons nous assurer que, sur le long-terme, les coûts engendrés par la fabrication des biens n'excèdent pas le budget alloué. Un moyen de garantir cela est de s'assurer que le coût de fabrication est (entièrement) couvert par le paiement des agents pour chaque manche, c'est-à-dire que le mécanisme est à budget non-déficitaire. Cette solution est suffisante mais non-nécessaire, donc elle est restrictive. Nous avons choisi cette solution malgré tout car elle a le mérite d'être simple.

En imposant que $PMA(0) = 0$ pour tout agent et pour tout bien, on impose que le profit révélé du groupe (= PMA total moins coût de fabrication) est toujours nul lorsque la quantité de bien fabriquée est nulle. Cette contrainte est peu restrictive en pratique, mais permet de garantir que le mécanisme est à budget non-déficitaire lorsque le profit révélé du groupe est maximisé pour chaque manche, ce qui fait l'objet de notre prochaine définition.

Définition 9 (Règle de fabrication PRG-MAX)**.** *Notons $q = (q_1, q_2, ..., q_m)$, où q_i est la quantité de bien i à fabriquer. La règle de fabrication PRG-MAX détermine le vecteur q de quantités de biens à fabriquer qui maximise le paiement total moins coût de fabrication total, compte-tenu des PMA révélés au mécanisme, de la règle de paiement, de la règle d'inclusion, et de possibles contraintes additionnelles, par exemple une contrainte liée à une capacité limitée de fabrication des biens.*

Remarque 2. *Quand l'espace de solutions est petit, le vecteur q optimal peut être trouvé par énumération. Sinon, une solution peut être trouvée plus efficacement avec une formulation Mixed Integer Linear Problem (MILP) du problème et l'utilisation d'un solveur approprié.*

Dans le cas dégénéré où plusieurs solutions sont possibles, on cherche en second à maximiser la quantité totale de bien fabriqué. Si malgré ce second critère plusieurs solutions sont encore possibles, la solution retenue est choisie aléatoirement.

Exemple 2. *Si la fabrication d'une unité de bien A génère un PRG égal à -1, et la fabrication d'une unité de bien B génère un PRG égal à 0, alors on préfère générer une unité de bien B plutôt que de ne rien fabriquer.*

Exemple 3. *Si la fabrication d'une unité de bien A génère un PRG égal à 0, et la fabrication d'une unité de bien B génère un PRG égal à 0, alors on choisira aléatoirement de fabriquer une unité de bien A ou une unité de bien B.*

Bien sûr, la pertinence de la règle PRG-MAX dépend de la corrélation qui existe entre le profit du groupe et le profit révélé du groupe. Si cette corrélation est positive et suffisamment élevée, alors la règle PRG-MAX peut conduire à remplir de façon satisfaisante l'objectif initial ; celui de maximiser le profit du groupe. Tous les mécanismes que nous étudions dans la suite utilisent la règle de paiement ID et la règle de fabrication PRG-MAX.

2.2 Règles d'inclusion

La règle d'inclusion détermine si un agent est autorisé ou non à consommer un bien. Quand un agent est autorisé à consommer un bien, on dit que l'agent est inclus (à la consommation du bien). Lorsqu'un agent ne peut pas consommer un bien, on dit que l'agent est exclu (à la consommation du bien).

Par définition de la propriété de non-excluabilité d'un bien, un mécanisme pour biens non-excluables ne peut exclure aucun agent à la consommation des biens. Nous introduisons donc une règle d'inclusion qui permet de prendre en compte ce cas.

Définition 10 (Règle d'inclusion No-Exclusion (NE))**.** *Règle d'inclusion qui n'exclut jamais aucun agent à la consommation des biens fabriqués, ou de façon équivalente qui inclut toujours tous les agents à la consommation des biens fabriqués.*

La règle d'inclusion NE équivaut à l'absence de règle d'inclusion. Cette règle permet de caractériser un mécanisme pour biens non-excluables (= biens dont il n'est pas possible d'empêcher les agents de les consommer), mais permet également de caractériser un mécanisme pour biens excluables pour lequel tous les agents sont toujours inclus par choix.

Lorsqu'un bien est indivisible et ne peut être consommé que par un unique agent, un choix classique est d'attribuer le bien à l'agent qui propose la meilleure offre. C'est le cas par exemple pour les enchères de type First-Price, Vickrey, English, Dutch, qui sont décrites succinctement dans [Alvarez and Nojoumian, 2020]. Cette règle est l'objet de la prochaine définition.

Définition 11 (Règle d'inclusion 1st-Highest-Bid (1HB))**.** *Règle d'inclusion qui inclut un unique agent du premier rang de l'ensemble des agents partiellement ordonnés dans l'ordre décroissant de leur PMA, évalué pour la quantité x de bien fabriquée par le mécanisme, lorsque $x > 0$.*

La règle d'inclusion 1HB n'est pas pertinente pour un bien excluable et non-soustractible à la consommation. La raison est que cette règle exclut tous les agents à la consommation du bien sauf un, alors que le bien pourrait être consommé par tous les agents sans engendrer de coût supplémentaire. Comme le fait remarquer Masso dans [Massó et al., 2015], l'exclusion à la consommation d'agents

qui pourraient générer une richesse strictement positive de la consommation du bien est à l'origine d'inefficacités.

> *"Since we are interested in mechanisms satisfying individual rationality, it turns out that inefficiencies arise from the exclusion (as users) of some (or all) agents who have strictly positive valuations."*
>
> - Extrait de [Massó et al., 2015], page 32.,

A l'inverse, inclure sans condition tous les agents, comme c'est le cas avec la règle d'inclusion NE, peut conduire à un problème de free-riding. En effet, lorsqu'un agent rationnel cherche à maximiser la richesse générée par les biens consommés additionnée au numéraire que l'agent cumule, l'agent peut reconnaître l'opportunité de consommer les biens sans avoir nécessairement besoin de contribuer pour cela, ce qui peut inciter l'agent à peu ou pas contribuer du tout.

Beaucoup de solutions existent pour limiter le problème de free-riding. Une solution est d'imposer un seuil de contribution minimal qui conditionne l'inclusion (ou l'exclusion) d'un agent, ce qui peut motiver les agents à contribuer un minimum pour obtenir le droit de consommer les biens.

Dans notre problème, l'objectif des agents consiste uniquement à maximiser la richesse générée par les biens consommés. Avec un tel objectif, il est difficile d'imaginer un mécanisme pour lequel le comportement des agents conduirait à un problème de free-riders. En effet, cumuler du numéraire revient à ne pas l'échanger avec le mécanisme. Puisque le mécanisme utilise le numéraire pour fabriquer les biens, réduire la quantité de numéraire fournie au mécanisme revient à réduire la capacité du mécanisme à fabriquer des biens. Au final, si le mécanisme fabrique moins de biens, cela engendre potentiellement une réduction des biens consommés par l'agent, ce qui est contre-productif compte-tenu du fait que l'unique objectif de l'agent est de maximiser la richesse générée par les biens qu'il consomme. En d'autres termes plus simples, cumuler le numéraire indéfinitivement aura probablement un effet négatif pour l'agent (une potentielle réduction de la quantité de biens consommés), et n'aura probablement pas d'effet positif (car cumuler le numéraire ne fait plus partie de l'objectif de l'agent).

L'absence de motivation des agents à devenir un free-rider n'est toutefois pas une certitude absolue, et puisque la solution d'exclure un agent pour le démotiver à devenir un free-rider fait partie de nombreux mécanismes et études associées [Norman, 2004, Moulin, 1994, Moulin and Shenker, 1996, Fang and Norman, 2010], nous jugeons pertinent d'étudier quel est l'impact de l'exclusion d'agents dans notre contexte.

La règle d'inclusion introduite ci-après est à la base du mécanisme de valeur de Shapley [Dobzinski and Sundararajan, 2008]. Dans cette règle, un seuil de contribution commun à tous les agents et aussi petit que possible est défini, dans l'idée d'inclure un maximum d'agents tout en imposant une contribution minimale et identique à tous les agents qui permette de couvrir entièrement le coût de fabrication du bien.

Définition 12 (Règle d'inclusion Minimal Common Threshold (MCT))**.** *Règle d'inclusion où un agent est inclus si et seulement si son paiement est supérieur à un seuil. Ce seuil est commun à tous les agents et est égal au coût de fabrication du bien divisé par le nombre d'agents inclus.*

L'ensemble d'agents S inclus avec la règle d'inclusion MCT peut être déterminé par le processus à tâtonnement décrit par l'Algorithme 2.

entrée :
— $PMA(1) : (p_i)_{i \in AGENTS}$: le prix pour chaque agent i et pour une unité de bien.
— *cout* : le coût de fabrication d'une unité de bien.
sortie :
— $AGENTS\_INCLUS$: l'ensemble d'agents inclus.

1 $AGENTS\_INCLUS \leftarrow AGENTS$.
2 **faire**
3 $\quad p_{min} \leftarrow cout/\#AGENTS\_INCLUS$
4 $\quad AGENTS\_A\_EXCLURE \leftarrow \{p_i < p_{min} | i \in AGENTS\_INCLUS\}$.
5 $\quad AGENTS\_INCLUS \leftarrow AGENTS\_INCLUS \setminus AGENTS\_A\_EXCLURE$.
6 **tant que** $(AGENTS\_A\_EXCLURE \neq \emptyset)$ *et* $(AGENTS\_INCLUS \neq \emptyset)$
7 termine et renvoie $AGENTS\_INCLUS$.

Algorithme 2 : Processus à tâtonnement qui permet de déterminer l'ensemble d'agents inclus par la règle d'inclusion MCT en fonction du PMA révélé par chaque agent pour un bien binaire.

Exemple 4. *Soit un unique bien binaire, pour lequel la fabrication d'une unité coûte 4 unités de numéraire, et 5 agents qui révèlent comme prix maximal acceptable* $(4, 0.8, 0.9, 0, 3.5)$.

1. *entrée de l'algorithme :* $PMA(1) : (4, 0.8, 0.9, 0, 3.5)$
2. *phase d'initialisation (ligne 1) :* $AGENTS\_INCLUS \leftarrow \{1, 2, 3, 4, 5\}$
3. *itération 1 :*
   (a) $p_{min} \leftarrow 4/5 = 0.8$
   (b) $AGENTS\_A\_EXCLURE \leftarrow \{4\}$ *(car* $0 < 0.8$*)*
   (c) $AGENTS\_INCLUS \leftarrow \{1, 2, 3, 4, 5\} \setminus \{4\} = \{1, 2, 3, 5\}$
   (d) $AGENTS\_A\_EXCLURE \neq \emptyset$ *et* $AGENTS\_INCLUS \neq \emptyset$*, donc on itère à nouveau.*
4. *itération 2 :*
   (a) $p_{min} \leftarrow 4/4 = 1$
   (b) $AGENTS\_A\_EXCLURE \leftarrow \{2, 3\}$ *(car* $0.8 < 1$ *et* $0.9 < 1$*)*
   (c) $AGENTS\_INCLUS \leftarrow \{1, 2, 3, 5\} \setminus \{2, 3\} = \{1, 5\}$
   (d) $AGENTS\_A\_EXCLURE \neq \emptyset$ *et* $AGENTS\_INCLUS \neq \emptyset$*, donc on itère à nouveau.*
5. *itération 3 :*
   (a) $p_{min} \leftarrow 4/2 = 2$
   (b) $AGENTS\_A\_EXCLURE \leftarrow \emptyset$
   (c) $AGENTS\_INCLUS \leftarrow \{1, 5\} \setminus \emptyset = \{1, 5\}$
   (d) $AGENTS\_A\_EXCLURE = \emptyset$*, donc on quitte la boucle.*
6. *l'algorithme termine et renvoie* $AGENTS\_INCLUS = \{1, 5\}$.

On peut vérifier que ce résultat n'est pas aberrant : lorsque deux agents sont inclus, le seuil de contribution minimal vaut $4/2 = 2$*, et effectivement il n'y a que les deux agents 1 et 5 qui consentent à payer cette valeur ou plus.*

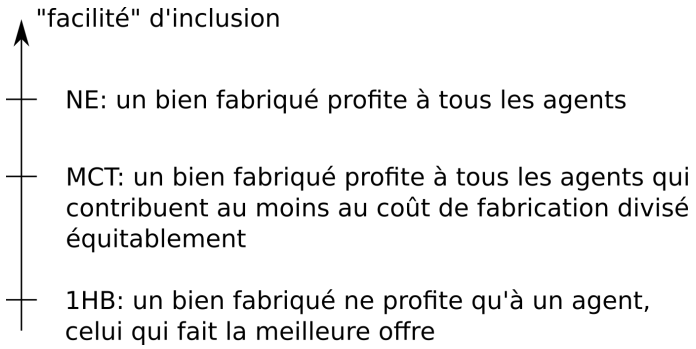


Figure 18 – *Règles d'inclusion 1HB, MCT, et NE ordonnées par facilité d'inclusion.*

Les trois règles d'inclusion 1HB, MCT, et NE peuvent être classées par "facilité" d'inclusion comme indiqué dans la Figure 18.

Finalement, il nous reste à introduire des règles de redistribution pour pouvoir caractériser entièrement un mécanisme.

2.3 Règles de redistribution

Une règle de redistribution triviale est celle qui consiste à ne jamais rien redistribuer aux agents.

Définition 13 (Règle de redistribution No-Rebate (NR))**.** *Règle de redistribution où le paiement réalisé par le mécanisme à un agent est toujours nul, c'est-à-dire que le mécanisme ne redistribue jamais de numéraire aux agents.*

La règle de redistribution NR équivaut tout simplement à l'absence de processus de redistribution.

En théorie de conception de mécanisme, les trois propriétés qui suivent, associées à un mécanisme, sont souvent considérées :

— budget non-déficitaire : le budget du mécanisme est nécessairement non-déficitaire,
— budget non-excédentaire : le budget du mécanisme est nécessairement non-excédentaire,
— budget équilibré : le budget du mécanisme est nécessairement équilibré.

Le numéraire cumulé par le mécanisme sur une manche est la différence entre la quantité de numéraire possédé par le mécanisme à la fin de la manche et la quantité de numéraire possédé par le mécanisme au début de la manche. La correspondance entre le numéraire cumulé sur une manche par le mécanisme et l'équilibre du budget du mécanisme est donnée Table 3.

Quantité de numéraire cumulé par le mécanisme	Equilibre du budget du mécanisme
> 0	excédentaire
$= 0$	équilibré
< 0	déficitaire

Table 3 – *Relation entre la quantité de numéraire cumulé par le mécanisme sur une manche et l'équilibre du budget du mécanisme sur la même manche.*

Pour définir d'autres règles de redistribution moins triviales, nous définissons deux notions de surplus de paiement, la première associée à l'ensemble d'agents, la deuxième associée à chaque agent individuellement.

Définition 14 (Surplus de paiement total)**.** *Le surplus de paiement total est défini pour un bien donné. Le surplus de paiement total vaut la différence entre le paiement total et le paiement minimal total. Le paiement total est le paiement de chaque agent agrégé par somme sur l'ensemble des agents. Le paiement minimal total est le coût de fabrication du bien.*

Tel que défini ici, le surplus de paiement total correspond tout simplement à la quantité de numéraire cumulé par le mécanisme en l'absence de règle de redistribution. Ceci permet donc de caractériser l'équilibre du budget en fonction de la quantité de surplus de paiement total redistribuée. Pour simplifier le raisonnement, considérons la provision d'un unique bien sur une manche. De plus, considérons un surplus de paiement total positif à redistribuer, puisque la redistribution d'un surplus de paiement total négatif, bien que permise par notre formalisme, est plus difficile à se représenter. Dans cette situation, lorsque la quantité totale de numéraire redistribuée est respectivement (inférieure, égale, supérieure) au surplus de paiement total, alors le budget du mécanisme est (excédentaire, équilibré, déficitaire). Des règles de redistribution basées sur cette notion de surplus de paiement total permettent ainsi un lien potentiellement direct avec l'équilibre du budget du mécanisme.

En l'absence de redistribution des surplus de paiements, sauf à garantir qu'aucun surplus de paiement ne puisse jamais avoir lieu, un mécanisme est à budget excédentaire. Lorsqu'une quantité strictement positive de surplus de paiement total n'est pas redistribuée aux agents et conservée à jamais par le mécanisme, la quantité de numéraire correspondante ne pourra jamais être utilisée par les agents pour acquérir les biens dont ils ont besoin, et donc la quantité de numéraire budgétisée ne pourra alors jamais être entièrement dépensée par les agents sur le long terme.

Pour rendre le budget du mécanisme équilibré, et ainsi permettre aux agents de dépenser sur le long terme l'entièreté du numéraire budgétisé, tout en préservant le certain équilibre souhaité entre la capacité d'influence des agents par les coefficients w_i qu'on leur a associé, il paraît pertinent de redistribuer les surplus de paiements de la même façon que le budget leur est distribué, c'est-à-dire avec les mêmes coefficients w_i. En effet, lorsque les coefficients qui correspondent à la redistribution des surplus de paiements sont identiques à ceux de la distribution du budget total, le surplus de paiement total d'une manche k peut être vu comme une augmentation du budget total de la manche suivante $k+1$. On peut donc s'attendre à ce que cela ne déséquilibre pas la capacité d'influence entre les différents agents. La Figure 19 illustre ce principe.

Bien sûr, il est tout à fait envisageable de redistribuer le surplus de paiement total avec une pondération différente de celle utilisée pour distribuer le budget total aux agents. Ceci nous amène à introduire la règle de redistribution pondérée.

Définition 15 (Règle de redistribution Weighted-Rebate (WR))**.** *Règle de redistribution où un coefficient w_i est associé à chaque agent i. Le montant redistribué à l'agent i vaut le surplus de paiement total pondéré par w_i.*

Remarque 3. *Le mécanisme est à budget (déficitaire, équilibré, excédentaire) lorsque la somme des coefficients w_i est respectivement (> 1, $= 1$, < 1). Cette propriété découle de la relation entre la quantité de numéraire totale redistribuée et le surplus de paiement total vue précédemment.*

On introduit à présent le surplus de paiement d'un agent, qui permet de définir des règles de

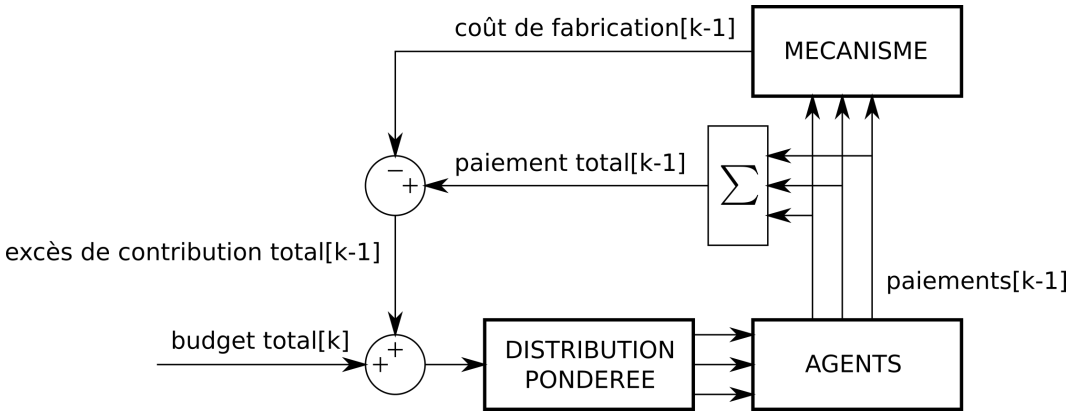


Figure 19 – *Diagramme de flux du numéraire avec la règle de redistribution pondérée.*

Le surplus de paiement total de la manche précédente vient s'ajouter au budget total de la manche courante pour être distribué de façon pondérée aux agents.

redistribution basées sur une notion de surplus de paiement similaire, mais où le montant redistribué à chaque agent peut dépendre de la contribution individuelle de l'agent.

Définition 16 (Surplus de paiement d'un agent)**.** *Le surplus de paiement d'un agent est la différence entre le paiement réalisé par l'agent et un paiement minimal. Le surplus de paiement d'un agent dépend de la définition du paiement minimal. Comme le paiement minimal associé à un agent n'est pas toujours facile à définir, on dit que le surplus de paiement n'est pas défini lorsque le paiement minimal n'est pas défini.*

Pour garder une certaine cohérence avec la définition du paiement minimal total, on souhaite que le paiement minimal associé à chaque agent et agrégé par somme soit égal au paiement minimal total. Ceci nous amène à définir le paiement minimal d'un agent selon la règle d'inclusion.

2.3.1 Paiement minimal d'un agent pour la règle d'inclusion NE

Nous choisissons de ne pas définir de paiement minimal d'un agent pour la règle d'inclusion NE, en raison d'un problème que nous allons décrire. Pour la règle d'inclusion NE, où un agent est inclus quel que soit son paiement, on pourrait vouloir définir le paiement minimal d'un agent comme le plus petit paiement de l'agent pour lequel, lorsque le paiement des autres agents est inchangé, la même quantité de bien pourrait être consommée par l'agent. C'est l'idée qui est à la base du mécanisme de Vickrey-Clarke-Groves pour bien privé ou pour bien public. Malheureusement, défini ainsi, le paiement minimal associé à chaque agent et agrégé par somme sur l'ensemble des agents ne serait pas égal, en général, au paiement minimal total.

Exemple 5. *Soit* 2 *agents, un mécanisme basé sur la règle de paiement ID, la règle de fabrication PRG-MAX, et la règle d'inclusion NE. Les deux agents révèlent un Paiement Maximal Acceptable identique* $PMA_{agent1}(1) = PMA_{agent2}(1) = 5$*. Le coût de fabrication du bien vaut* 0 *pour une quantité nulle, et* 7 *pour une unité fabriquée :* $COUT(0) = 0, COUT(1) = 7$*. Le mécanisme détermine que le profit maximal est obtenu lorsque les deux agents sont inclus et consomment une unité de bien fabriqué. Si l'offre de l'agent* 2 *est inchangée, l'agent* 1 *aurait pu consommer une unité de bien en proposant* $PMA(1) = 7 - 5 = 2$*; il en est de même pour le paiement minimal de l'agent* 2*. La somme du paiement minimal des deux agents vaut* $2 + 2 = 4$*, alors que le paiement minimal total, égal au coût*

de fabrication du bien, vaut 7.

En raison de ce problème de cohérence entre le paiement minimal des agents et le paiement minimal total, nous choisissons plutôt de considérer que le paiement minimal d'un agent n'est pas défini pour la règle d'inclusion NE.

2.3.2 Paiement minimal d'un agent pour la règle d'inclusion 1HB

Pour la règle d'inclusion 1HB, lorsque la meilleure offre couvre le coût de fabrication, un seul agent qui réalise la meilleure offre est inclus. Pour cette règle d'inclusion, on définit alors le paiement minimal d'un agent comme le coût de fabrication associé à la quantité de bien consommée par l'agent.

Exemple 6. *Comme précédemment, soit* 2 *agents, un mécanisme basé sur la règle de paiement ID, la règle de fabrication PRG-MAX, et la règle d'inclusion 1HB. Les deux agents révèlent tous deux* $PMA(0) = 0$ *et* $PMA(1) = 5$*. Le coût de fabrication du bien est* $COUT(0) = 0$*,* $COUT(1) = 2$*. Le mécanisme détermine que le profit maximal est obtenu lorsqu'un des deux agents est inclus et consomme une unité de bien fabriqué. Les deux agents sont éligibles à l'inclusion, mais la règle 1HB impose qu'un seul des deux soit inclus. Supposons que le mécanisme détermine que le premier agent est inclus. Le paiement minimal de l'agent 1 vaut* $COUT(1) = 2$*, le paiement minimal de l'agent* 2 *vaut* $COUT(0) = 0$*. Un raisonnement similaire s'applique si le mécanisme détermine que le deuxième agent est inclus. Dans tous les cas, le paiement minimal agrégé par somme* $0 + 2$ *sur l'ensemble des agents est bien égal au paiement minimal total* 2*.*

Remarque 4. *Puisque le paiement minimal d'un agent dépend de la quantité que l'agent consomme, et que la quantité de bien consommé par l'agent est conditionné par son inclusion (l'agent consomme une quantité nulle s'il est exclus, la quantité fabriquée s'il est inclus), le paiement minimal est conditionné par l'inclusion ou non de l'agent.*

2.3.3 Paiement minimal d'un agent pour la règle d'inclusion MCT

Pour la règle d'inclusion MCT, le paiement minimal d'un agent vaut :

— lorsque l'agent est inclus : le coût de fabrication du bien divisé par le nombre d'agents inclus.

— lorsque l'agent est exclus : 0.

Le paiement minimal d'un agent peut être défini de façon plus simple, et peut-être un peu plus naturelle, en imposant la contrainte que le coût d'un bien fabriqué en quantité nulle vaut 0, comme suit :

$$COUT(\text{quantité\_consommée})/\#\text{AGENTS\_INCLUS}$$

En effet, lorsque $COUT(0) > 0$, il est possible, et probable, que tous les agents choisissent de révéler $PMA(0) = 0$ au mécanisme, ce qui poserait un problème de division par zéro. En revanche, lorsque $COUT(0) = 0$, alors $PMA(0) \geq COUT(0)$ pour tout agent, quel que soit le PMA révélé par chaque agent. Ainsi, tous les agents sont nécessairement inclus lorsque le bien n'est pas fabriqué, ce qui revient à dire que tous les agents sont nécessairement autorisés à consommer un bien fabriqué en quantité nulle. En effet :

— lorsque le bien n'est pas fabriqué, tous les agents sont inclus et paient

$$COUT(\text{quantité\_consommée})/\#\text{AGENTS\_INCLUS}$$

$$COUT(0)/\#\text{AGENTS} = 0$$

— lorsque le bien est fabriqué, alors la règle de fabrication PRG-MAX garantit que le coût de fabrication est couvert par la contribution totale des agents, ce qui implique que $\#AGENTS\_INCLUS \geq 1$. En conséquence, deux cas sont possibles :
  — lorsque l'agent est inclus :

$$COUT(\text{quantité\_fabriquée})/\#\text{AGENTS\_INCLUS}$$

$$= COUT(\text{quantité\_consommée})/\#\text{AGENTS\_INCLUS}$$

  — lorsque l'agent est exclus :

$$COUT(\text{quantité\_consommée})/\#\text{AGENTS\_INCLUS}$$

$$= COUT(0)/\#\text{AGENTS\_INCLUS}$$

$$= 0/\#\text{AGENTS}$$

$$= 0$$

Nous introduisons à présent la règle de redistribution qui garantit à chaque agent de se voir redistribuer l'entièreté de son surplus de paiement. Nous verrons plus tard que cette règle permet de caractériser un mécanisme existant de la littérature.

Définition 17 (Règle de redistribution Full-Rebate (FR))**.** *Règle de redistribution où le paiement réalisé par le mécanisme à chaque agent est égal au surplus de paiement individuel dudit agent si ce surplus est positif, zéro sinon. Lorsque le surplus de paiement d'un agent n'est pas défini, cette règle n'est pas définie.*

Remarque 5. *Cette règle confère la propriété de budget équilibré au mécanisme, puisque redistribuer l'entièreté du surplus de paiement de chaque agent équivaut à redistribuer l'entièreté du surplus de paiement total.*

2.3.4 Synthèse

Les 3 règles de redistribution que nous venons d'introduire sont relativement simples, ce qui ne les rend pas pour autant inintéressantes d'après le travail réalisé dans [Cavallo, 2008], qui soulève la question suivante :

> *"do we really need a very sophisticated mechanism, optimal in the worst case, when a simple mechanism works quite well in general ?"*
>
> - Extrait de [Gujar and Narahari, 2011].,

Les règles de redistribution sont synthétisées et définies plus formellement dans la Table 4.

Règle de redistribution	**Quantité de numéraire redistribuée à un agent i pour k unités de bien j fabriquées**	
NR (No-Rebate)	0	
WR (Weighted-Rebate)	$\underbrace{w_i}_{\text{poids}} \left[\underbrace{\underbrace{\sum_{r \in N} p(r,j)}_{\text{paiement total}} - \underbrace{c(k,j)}_{\text{coût du bien}}}_{\text{surplus de paiement total}} \right]$	
FR (Full-Rebate)	NE	non-définie
	MCT	$\underbrace{\underbrace{p(i,j)}_{\text{paiement de l'agent}} - \underbrace{\frac{c(k,j)}{\text{nbr. d'agents inclus}}}_{\text{coût égalitairement partagé}}}_{\text{surplus de paiement de l'agent}}$ si i est inclus, 0 sinon.
	1HB	$\underbrace{\underbrace{p(i,j)}_{\text{paiement de l'agent}} - \underbrace{c(k,j)}_{\text{coût du bien}}}_{\text{surplus de paiement de l'agent}}$ si i est inclus, 0 sinon.

Table 4 – *Définition des 3 règles de redistribution NR, WR, et FR.*

Les 3 règles d'inclusion et les 3 règles de redistribution que nous avons introduit rendent possible la génération d'une famille de 8 variantes de mécanismes, que nous évaluerons grâce à l'approche ACE.

Maintenant que ces définitions ont été introduites, nous allons montrer comment trois mécanismes populaires s'inscrivent dans notre modèle.

2.4 Mécanismes populaires qui s'inscrivent dans notre modèle

2.4.1 First-Price Sealed-Bid Auctions with a Reservation Price (FPSBARP)

Le mécanisme <PRG-MAX, ID, 1HB, NR> correspond à une enchère de type First-Price Sealed-Bid Auctions with a Reservation Price (FPSBARP) [McAfee and McMillan, 1987]. Dans cette enchère, l'offre minimale est le coût de fabrication du bien. L'agent dont l'offre est supérieure à l'offre minimale et la plus élevée remporte le bien et paie son offre. En cas d'égalité, un seul agent parmi l'ensemble des agents qui réalisent la meilleure offre est choisi avec équiprobabilité. En effet :

- la règle de paiement ID impose qu'un agent paie son offre lorsqu'il remporte le bien ($PAIEMENT = PMA(1)$), et ne paie rien lorsqu'il ne remporte pas le bien ($PAIEMENT = PMA(0)$),
- la règle d'inclusion 1HB impose que lorsque le bien est fabriqué, l'un des agents qui a proposé la meilleure offre remporte le bien,
- la règle de fabrication PRG-MAX impose, compte tenu des deux règles précédentes, que le bien est fabriqué ssi la meilleure offre pour une unité de bien couvre le coût de fabrication ($max(PMA(1)) \geq COUT(1)$) :
  - condition nécessaire : si $max(PMA(1)) < COUT(1)$, alors comme aucun agent ne peut couvrir seul le coût de fabrication, le PRG serait négatif si le bien était fabriqué. Pour que la règle de fabrication PRG-MAX détermine que le bien est fabriqué, il est donc nécessaire que $max(PMA(1)) \geq COUT(1)$.
  - condition suffisante : si $max(PMA(1)) \geq COUT(1)$, alors 1HB implique qu'un agent consomme le bien et paie la meilleure offre, $max(PMA(1))$.

La règle d'inclusion 1HB garantit que, lorsque le bien est fabriqué, l'agent qui fait la meilleure offre remporte le bien. La règle de paiement garantit que chaque agent paie ce qu'il a proposé, i.e. l'offre pour une unité de bien lorsqu'il est fabriqué et remporté par l'agent, l'offre pour 0 unité de bien lorsque le bien n'est pas fabriqué ou lorsque l'agent ne remporte pas le bien. La règle de fabrication PRG-MAX, quant à elle, garantit que le bien est fabriqué si au moins un agent propose une offre supérieure ou égale au coût de fabrication.

Exemple 7. *Soit un bien de club binaire, dont le coût de fabrication est donné par la relation* (*quantité* $\mapsto$ *coût*) *suivante :* $\{(0 \mapsto 0), (1 \mapsto 5)\}$.

L'agent 1 révèle au mécanisme la relation (*quantité reçue* $\mapsto$ *pma*) *suivante :* $\{(0 \mapsto 0), (1 \mapsto 3)\}$, *et l'agent 2 révèle au mécanisme la relation* $\{(0 \mapsto 0), (1 \mapsto 7)\}$.

La règle de fabrication PRG-MAX détermine que, compte tenu des règles INC et PAY, 1 *unité de bien est fabriquée. En effet, PRG-MAX cherche à déterminer la quantité de bien à fabriquer pour maximiser le profit révélé du groupe. Comme le bien est binaire, il n'y a que deux possibilités :*

- *si la quantité de bien fabriquée vaut* 0*, le paiement total vaut* 0*, le coût de fabrication vaut* 0*, et alors le profit révélé du groupe vaut* 0*,*
- *si la quantité de bien fabriquée vaut* 1*, le paiement total vaut* 7 *(l'agent* 2 *remporte le bien, il paie* 7*, l'agent* 1 *ne remporte pas le bien, il paie* 0*), le coût de fabrication vaut* 5*, et alors le profit révélé du groupe vaut* $7 - 5 = 2$.

Enfin, comme la règle de redistribution est NR, le mécanisme ne redistribue rien aux agents.

2.4.2 Provision-Point-Mechanism (PPM)

PPM [Li et al., 2016] est un mécanisme défini pour un unique bien binaire. Dans ce mécanisme :

- aucun agent n'est jamais exclus (= règle d'inclusion NE),
- le bien est fabriqué si le paiement de tous les agents couvre le coût de fabrication du bien (= règle de fabrication PRG-MAX), et le paiement des agents est déterminé comme suit (= règle de paiement ID) :
  - si le bien est fabriqué, les agents paient leur offre ($PMA(1)$),
  - si le bien n'est pas fabriqué, les agents ne paient rien ($PMA(0)$).
- le mécanisme ne redistribue jamais rien aux agents (= règle de redistribution NR)

Autrement dit, le bien est fabriqué si et seulement si $PMA(1)$ sommé pour chaque agent est supérieur ou égal au coût de fabrication du bien $COUT(1)$. Si le bien est fabriqué, chaque agent paie son offre ($PMA(1)$) et tous les agents peuvent profiter du bien. Si le bien n'est pas fabriqué, les agents ne paient rien.

PPM correspond donc à un mécanisme <PRG-MAX, ID, NE, NR> dans notre formalisme dans le cas d'un unique bien binaire. Le mécanisme <PRG-MAX, ID, NE, NR>, applicable dans le cadre de multiples biens non-binaires, peut être considéré comme une généralisation de PPM.

2.4.3 Equal Cost Sharing with Maximal Participation (ECSMP)

ECSMP [Shichijo and Fukuda, 2016] est un mécanisme défini pour un unique bien binaire, qui lorsque le bien est fabriqué, maximise le nombre d'agents inclus par les trois conditions suivantes :

1. le paiement de chaque agent inclus est le coût de fabrication divisé par le nombre d'agents inclus,
2. le paiement d'un agent n'est pas plus élevé que le Paiement Maximal Acceptable (PMA) qu'il a révélé, et
3. tous les agents non-inclus paient 0.

Pour une règle de paiement ID, la condition 2 correspond à la règle d'inclusion MCT, où tout agent qui consent à payer au-dessus du seuil minimal commun est inclus. Toujours pour une règle de paiement ID, la condition 3 nécessite que chaque agent révèle $PMA_{agent}(0) = 0$. Notons que $PMA_{agent}(0) = 0$ n'est pas vraiment restrictif, car il est difficile d'imaginer une raison qui justifierait qu'un agent individuellement rationnel révèle autre chose.

Enfin, la condition 1 est vérifiée avec la règle de redistribution FR, qui redistribue le surplus de paiement individuel de chaque agent, de sorte qu'après redistribution la contribution d'un agent inclus est exactement égal au seuil de contribution minimal commun.

<PRG-MAX, ID, MCT, FR> est donc un mécanisme équivalent au mécanisme ECSMP dans le cadre de l'allocation d'un unique bien binaire. Comme précédemment, <PRG-MAX, ID, MCT, FR> peut être considéré comme une extension de ECSMP à de multiples biens non-binaires.

Conclusion

Dans ce chapitre, nous avons contribué à la définition d'un nouveau formalisme, permettant de construire une famille de 8 variantes de mécanismes à étudier. De plus, nous avons montré que ce formalisme est une généralisation de 3 mécanismes déjà étudiés dans la littérature. Dans le prochain chapitre, nous proposons un moyen d'analyse pour évaluer la qualité des différents mécanismes.

Chapitre 3

Moyen d'analyse

Introduction

Nous devons trouver un moyen d'analyse pour juger de la qualité des différents mécanismes et ainsi choisir un mécanisme qui répond efficacement à notre problème.

La Section 3.1 justifie le moyen d'analyse que nous avons retenu.

Dans un premier temps, nous allons rappeler un peu plus en détail la technique d'apprentissage par renforcement multi-agents. Ensuite, nous détaillons le modèle d'environnement multi-agents implémenté, et les choix de modélisation que nous avons fait. Nous justifions ensuite le choix de l'algorithme d'apprentissage utilisé, ainsi que le choix des valeurs des différents hyperparamètres.

Dans la Section 3.3, nous décrivons le modèle de l'environnement multi-agents implémenté, modèle qui symbolise le mécanisme de provision. Enfin, la mise en oeuvre de l'algorithme d'apprentissage fait l'objet de la Section 3.4.

3.1 Choix de l'approche ACE

L'utilisation d'une approche analytique pour évaluer les qualités d'un mécanisme d'enchères sur une unique manche n'est déjà pas simple. Par exemple, les preuves expérimentales obtenues ne correspondent pas toujours aux prédictions théoriques. L'explication des différences constatées entre les prédictions théoriques et les résultats expérimentaux pour le modèle d'enchères FPSBA, qui est pourtant le plus simple, a par exemple fait coulé beaucoup d'encre [Salo and Weber, 1995]. De plus, l'approche analytique est grandement complexifiée lorsque le mécanisme est joué sur plusieurs manches successives.

Dans notre cas, nous souhaitons que la provision de biens soit répétée (ou itérée) sur plusieurs manches. Des hypothèses restrictives s'imposeraient donc probablement pour rendre une approche analytique abordable, ce qui pourrait mettre en échec la pertinence du modèle étudié. Pour cette raison, nous nous orientons plutôt vers une approche expérimentale.

Une approche déjà existante et qui gagne en popularité [Mosavi et al., 2020] consiste à utiliser des agents dotés de capacités cognitives artificielles pour obtenir des solutions permettant l'étude et l'analyse du comportement optimal des agents [Redko and Laclau, 2019]. Cette approche s'inscrit plus généralement dans le paradigme Agents-based Computational Economics (ACE).

Pour nous, il s'agit de simuler les services eHorizon intéragissant avec le MAC par des agents intéragissant avec un environnement virtuel. La méthode d'apprentissage par renforcement est utilisée pour apprendre à chaque service eHorizon un comportement décisionnel, appelé stratégie ou politique, qui optimise la richesse générée individuellement par le service. Le comportement ainsi obtenu permet

de générer des résultats que nous discutons pour tenter d'en dégager une connaissance, qui permettra finalement de guider le choix d'un mécanisme.

Ce moyen d'analyse n'échappe pas au concept Garbage In Garbage Out (GIGO), selon lequel des données d'entrée défectueuses ou absurdes (de mauvaise qualité) produisent des sorties absurdes ou "déchets" (de mauvaise qualité). Il est important que les résultats soient qualitatifs pour que les analyses qui en découlent le soient également. Et puisque la qualité des résultats dépend de la qualité des stratégies utilisées par les agents, il est important que les agents trouvent des stratégies de bonne qualité.

Pour palier à ce problème, nous répétons plusieurs fois la même expérimentation afin d'obtenir une description statistique des solutions obtenues. Cette description statistique ne permet pas d'évaluer le degré de qualité des stratégies obtenues, mais permet néanmoins d'évaluer la consistance de la qualité des stratégies obtenues.

Dans un premier temps, nous allons rappeler un peu plus en détail la technique d'apprentissage par renforcement multi-agents. Ensuite, nous détaillons le modèle d'environnement multi-agents implémenté, et les choix de modélisation que nous avons fait. Nous justifions ensuite le choix de l'algorithme d'apprentissage utilisé, ainsi que le choix des valeurs des différents hyperparamètres.

3.2 Principe de l'apprentissage multi-agents

Les agents sont plongés dans un environnement multi-agents, qui symbolise le mécanisme. Une manche représente une allocation. Au début d'une *manche*, chaque agent décide, compte tenu de son état, une action qu'il envoie à l'environnement. La transmission de l'action d'un agent au mécanisme symbolise la révélation des PMA de l'agent au mécanisme. Après avoir obtenu les actions de tous les agents, l'environnement renvoie une *observation* et une *récompense* propre à chaque agent, ce qui termine la manche. Dans notre cas, l'observation envoyée à un agent correspond à la quantité de numéraire dont dispose le service pour la prochaine manche, et la récompense correspond à la richesse générée par les biens obtenus par le service dans la manche qui a eu lieu. Un *épisode* est constitué d'une succession de manches, qui peut être en nombre infini, lorsque l'environnement n'impose pas l'arrêt des intéractions, ou en nombre fini, le cas échéant. La Figure 20 illustre les échanges d'informations entre les agents et l'environnement.

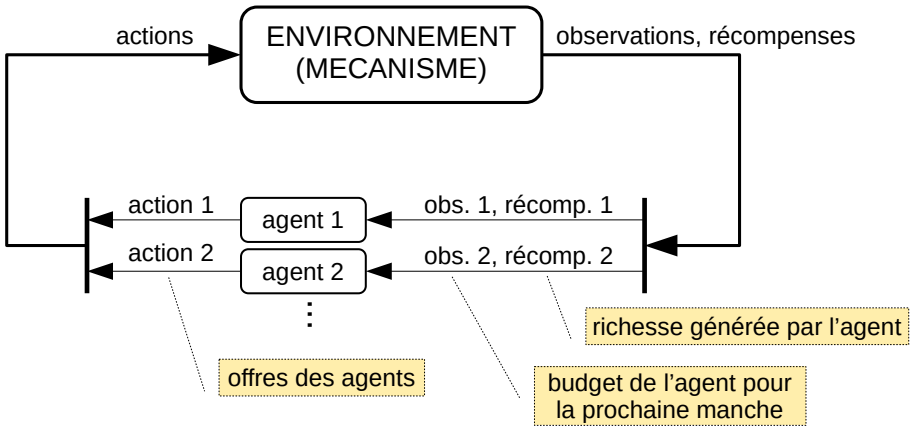


Figure 20 – *Apprentissage multi-agents par renforcement (figure inspirée de [Patkin et al., 2019]).*

Un algorithme d'apprentissage par renforcement cherche à déterminer, à partir d'expériences itérées d'un agent avec l'environnement, une politique optimale de l'agent. La *politique* d'un agent est une fonction qui, à chaque état de l'agent, préconise la décision que l'agent doit prendre. Dans notre cas, les agents sont les services eHorizon, l'environnement est le mécanisme de provision, et la politique d'un agent est une fonction qui préconise les PMA que doit révéler stratégiquement le service eHorizon au mécanisme.

La politique peut aussi être probabiliste et préconiser une distribution de probabilité de la décision à prendre. Une politique est optimale lorsqu'elle maximise l'espérance des récompenses cumulées $\mathbb{E}(\sum_k G_k)$ de l'agent sur un épisode. Puisqu'un épisode peut comporter un nombre infini de manches, $\sum_k G_k$ peut ne pas être bien définie.

Un *coefficient de dévaluation* γ, compris dans $[0, 1]$, permet de rendre cette somme bien définie et convergente. De plus, γ permet d'adapter le degré de prise en compte des récompenses obtenues sur l'horizon futur. Par exemple, lorsque γ vaut 0, l'agent est "myope" et prend uniquement en compte la récompense qui serait obtenue à l'issue de la manche courante, sans tenir compte des récompenses qui seraient obtenues plus tard. A l'opposé, lorsque γ vaut 1, l'agent tiendra compte identiquement de toutes les récompenses qui seraient obtenues dans le futur. La Figure 21 illustre la valeur du coefficient de dévaluation en fonction de la distance de la manche dans l'horizon futur. On configure alors usuellement un algorithme d'apprentissage pour rechercher une politique qui maximise la récompense cumulée dévaluée $R : \sum_k \gamma^k G_k, \gamma \in \mathbb{R}_{[0,1]}$. γ est un hyperparamètre, dont le choix est discuté plus tard dans la Section 3.4.

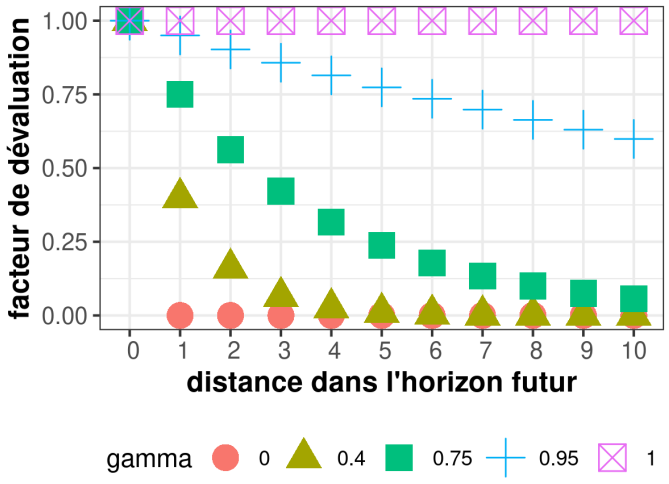


Figure 21 – *Poids associé à une récompense future pour différentes valeurs du coefficient de dévaluation γ.*

Nous allons à présent discuter des choix de modélisation que nous avons eu à faire.

3.3 Modèle de l'environnement multi-agents

Comme illustré sur la Figure 20, l'environnement doit recevoir des actions de chaque agent, puis renvoyer des observations et des récompenses qui sont propres à chaque agent. Le diagramme détaillé des flux de l'environnement multi-agents utilisé pour l'analyse des propriétés des mécanismes est représenté sur la Figure 22.

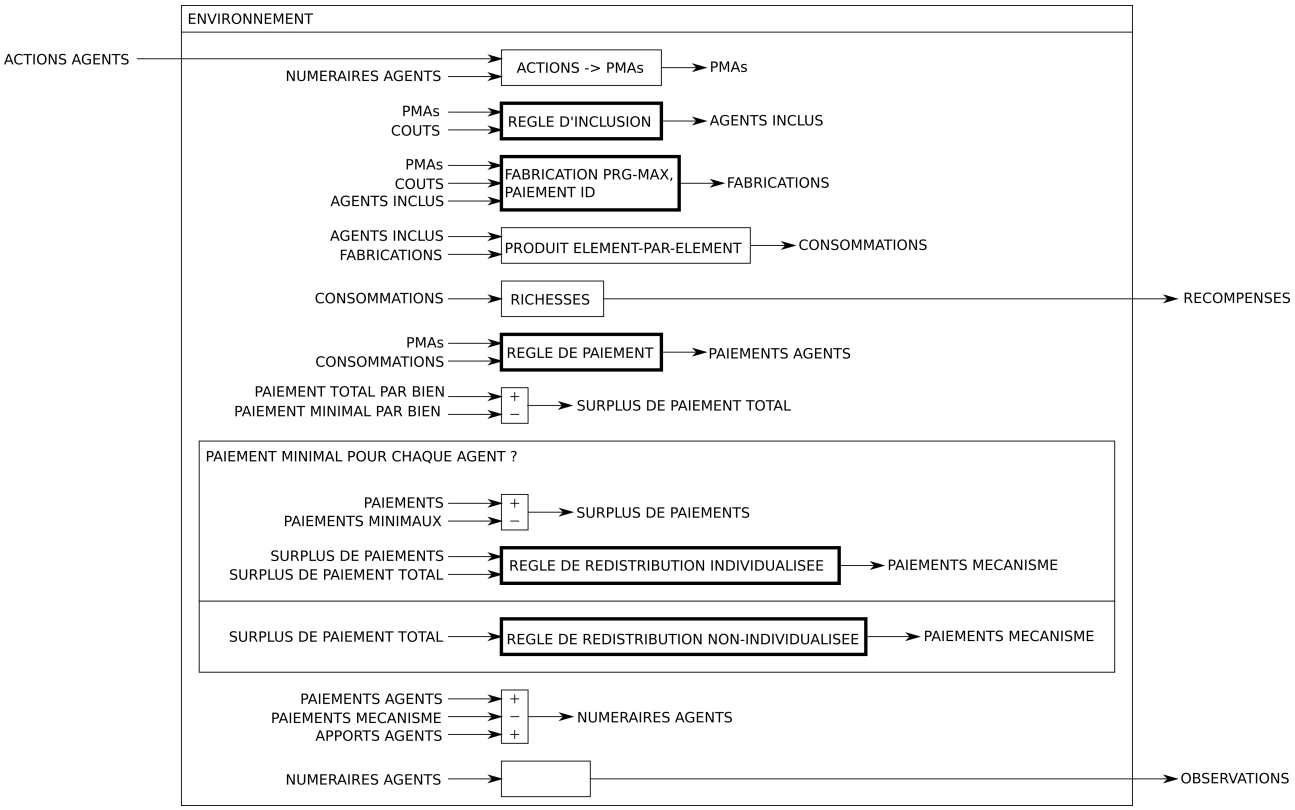


Figure 22 – *Diagramme de flux de l'environnement multi-agents implémenté.*

Nous détaillons à présent les paramètres de l'environnement, son initialisation ainsi que le déroulement d'une manche.

Paramètres de l'environnement

— Matrice d'APPORTS, qui détermine la quantité de numéraire ajoutée à chaque agent avant la première manche, puis à chaque fin de manche,

— Matrice RICHESSES telle que le produit par élément de RICHESSES et CONSOMMATIONS donne la richesse générée pour chaque paire (agent, bien).

Initialisation de l'environnement

1. apport de numéraire pour chaque agent : NUMÉRAIRE(agent) = NUMÉRAIRE(agent) + APPORTS(agent)
2. observation retournée pour chaque agent : NUMÉRAIRE(agent)

Déroulement d'une manche

1. chaque agent détermine, compte-tenu de son état et des observations qu'il a reçu de l'environnement, une action : pour chaque agent, ACTION(agent) = POLITIQUE(agent)
2. L'action de chaque agent est transmise à l'environnement. L'environnement, compte-tenu du numéraire de chaque agent, détermine le PMA, ou prix, révélé par chaque agent. Cette transformation permet de garantir que le paiement d'un agent est admissible pour toute allocation, c'est-à-dire que pour toute quantité $(x_1, x_2, ..., x_n)$ de bien fourni à un agent, $\text{PMA}(x_1) + \text{PMA}(x_2) + ... \leq$ NUMÉRAIRE(agent).
3. Compte tenu des PMAs de chaque paire (agent, bien) et des COÛTS marginaux de chaque bien, la règle d'inclusion détermine les UTILISATEURS, qui sont les agents autorisés à consommer

les biens fabriqués par le mécanisme.

4. Compte tenu des PMAs, COÛTS, UTILISATEURS, de la règle de fabrication PRG-MAX et de la règle de paiement ID, le mécanisme détermine FABRICATIONS, la quantité de chaque bien fabriqué par le mécanisme.
5. Compte tenu de FABRICATIONS et UTILISATEURS, un produit matriciel élément-par-élément détermine les CONSOMMATIONS, la quantité consommée pour chaque paire (agent, bien).
6. Compte tenu de CONSOMMATIONS et RICHESSES, un produit matriciel élément-par-élément détermine la récompense de chaque agent.
7. Compte tenu des PMAs et des CONSOMMATIONS, la règle de paiement détermine PAIEMENTS, le paiement pour chaque paire (agent, bien).
8. Compte tenu du PAIEMENT TOTAL PAR BIEN et du PAIEMENT MINIMAL PAR BIEN, le calcul de la différence détermine le SURPLUS DE PAIEMENT TOTAL de chaque bien. Le PAIEMENT MINIMAL PAR BIEN dépend de la règle d'inclusion.
   - Lorsque le paiement minimal pour chaque agent est défini :
     - PAIEMENTS et PAIEMENTS MINIMUMS permettent de déterminer les SURPLUS DE PAIEMENTS, qui est constitué du surplus de paiement pour chaque paire (agent, bien).
     - Compte tenu de SURPLUS DE PAIEMENTS et SURPLUS DE PAIEMENT TOTAL, REGLE DE REDISTRIBUTION INDIVIDUALISÉE détermine PAIEMENTS MÉCANISME, le paiement réalisé par le mécanisme pour chaque agent.
   - Lorsque le paiement minimal pour chaque agent n'est pas défini :
     - Compte tenu du SURPLUS DE PAIEMENT TOTAL, la REGLE DE REDISTRIBUTION NON INDIVIDUALISÉE détermine PAIEMENTS MÉCANISME, le paiement réalisé par le mécanisme pour chaque agent.
9. Le numéraire de chaque agent est mis à jour : NUMÉRAIRE(agent) = NUMÉRAIRE(agent) - PAIEMENT(agent) + PAIEMENT MÉCANISME(agent) + APPORT(agent).
10. L'environnement retourne à chaque agent une OBSERVATION. Chaque agent reçoit sa propre OBSERVATION : la quantité de numéraire dont dispose l'agent à la fin de la manche. Un agent n'observe pas la quantité de numéraire d'un autre agent.

Le comportement de l'environnement étant établi, nous devons à présent définir le domaine des actions, observations et récompenses. Le choix de ces domaines est important, car il peut influencer (positivement ou négativement) la performance d'apprentissage des agents.

Espace des actions Nous nous limitons à l'étude de mécanismes dans le cas de multiples biens binaires, de façon à ce que l'action d'un agent puisse correspondre à un scalaire pour chaque bien (domaine d'actions $\mathbb{R}^{\#\text{BIENS}}$), plutôt qu'à une fonction (domaine d'actions $\mathbb{N}^{R^{\#\text{BIENS}}}$). Nous avons donc fait le choix que les coefficients d'un vecteur d'action indiquent une valeur de numéraire relative à la quantité de numéraire dont dispose l'agent pour permettre aux coefficients du vecteur d'action d'un agent d'être borné dans $[0, 1]$, ce qui facilitera la performance d'apprentissage des agents. Comme nous souhaitons aussi que les agents soient à budget contraint, c'est-à-dire qu'ils ne puissent pas révéler des offres qu'ils ne pourraient pas payer, il faut nous assurer que la somme des coefficients du vecteur d'action d'un agent ne dépasse jamais 1. Nous garantissons le respect de cette contrainte par une étape de normalisation non-linéaire qui s'applique systématiquement à chaque action :

$$\text{action\_normalisée} = \text{action}/max(1, \sum_i \text{action}_i)$$

Exemple 8. *Considérons un cas avec* 3 *biens et* 1 *agent qui dispose de* 10 *unités de numéraire. L'environnement multi-agent reçoit de l'agent l'action* $(1, 0.4, 0.6)$*. Comme* $1+0.4+0.6 = 2$*, et que* $2 > 1$*, l'action vaut après normalisation* $(1/2, 0.4/2, 0.6/2) = (0.5, 0.2, 0.3)$*. Comme l'agent dispose de 10 unités de numéraire, le PMA révélé par l'agent pour chaque bien vaut simplement* $10 \times (0.5, 0.2, 0.3) = (5, 2, 3)$.

Exemple 9. *Considérons un autre cas avec* 3 *biens et* 1 *agent qui dispose de* 14 *unités de numéraire. L'environnement multi-agent reçoit de l'agent l'action* $(0.1, 0.3, 0)$*. Comme* $0.1 + 0.3 + 0 = 0.4$*, et que* $0.4 < 1$*, l'action est inchangée et vaut toujours* $(0.1, 0.3, 0)$ *après normalisation. Comme l'agent dispose de 14 unités de numéraire, le PMA révélé par l'agent pour chaque bien vaut simplement* $14 \times (0.1, 0.3, 0) = (1.4, 3.2, 0)$.

Remarque 6. *Un autre choix possible aurait été d'introduire un coefficient supplémentaire dans le vecteur d'action permettant d'indiquer la quantité de numéraire que l'agent souhaite conserver. Par exemple, l'action* $[0.3, 0.4, 1-(0.3+0.4)]$ *pourrait correspondre à des PMAs révélés de* 30% *du numéraire actuel de l'agent pour le premier bien,* 40% *du numéraire actuel de l'agent pour le second bien, et* 30% *du numéraire restant ne serait pas misé. Avec ce choix, la somme du vecteur d'action est invariablement égal à* 1*, permettant alors l'utilisation directe d'une fonction d'activation de sortie qui modélise une distribution de probabilité catégorique sur la dernière couche d'un agent. Toutefois, bien que cette solution soit pertinente, l'ambiguïté qu'elle implique sur les indices et la taille du vecteur d'action par rapport au nombre de biens nous à conduit à nous tourner plutôt vers la première solution.*

Espace des observations Nous souhaitons évaluer le comportement des agents en présence d'un minimum d'informations ; un agent observe uniquement le numéraire qu'il possède à la fin de la manche. Comme il connaît le numéraire qu'il possède au début d'une manche, il peut en déduire le numéraire cumulé sur la manche. L'espace des observations pour chaque agent est donc $\mathbb{R}_+$. Comme un agent peut cumuler du numéraire, et qu'on fixe le nombre de manches à 10 maximum, le numéraire maximal qu'un agent peut cumuler sur un épisode entier vaut $10 \times APPORT(\text{agent})$, auquel on doit ajouter éventuellement des apports générés par la redistribution des surplus de paiements. Or les apports liés à la redistribution des surplus de paiements dépendent de la règle de redistribution et des surplus de paiements réalisés par les agents dans l'épisode. Pour avoir un domaine de l'espace des observations qui soit valide pour toutes les expérimentations, nous avons choisi de borner arbitrairement la dimension des observations à l'apport total des agents * le nombre de manches. En effet, il est trivial de vérifier que le numéraire cumulé par un agent sur un épisode entier ne peut pas dépasser cette borne, quels que soient les surplus de paiements réalisés par les agents et la façon dont ils sont redistribués.

Espace des récompenses La récompense est choisie pour refléter l'utilité générée par un agent. Dans nos expérimentations, un seul bien peut être fabriqué par période, et la richesse générée par un agent pour une unité de bien ne dépasse jamais 1. Par conséquent, l'espace des récompenses est $\mathbb{R}_{[0,1]}$. Ainsi donc, l'objectif de l'agent est de maximiser l'utilité réelle générée sur un épisode, et non pas ce qui est souvent fait dans la littérature, maximiser la somme (utilité générée + numéraire) sur un épisode. Ce choix est extrêmement important.

Bien que nous faisons l'hypothèse, dans notre analyse, que l'apport de numéraire distribué à chaque agent est constant, cette quantité évolue en réalité au cours du temps et dépend indirectement du revenu du groupe, donc de la satisfaction des utilisateurs, et donc in fine de la richesse globale réellement générée par l'ensemble des agents. Il n'est donc pas rationnel sur le long terme pour un ou plusieurs agents de maximiser le numéraire cumulé lorsque cela se traduit par une diminution trop importante de la richesse que l'agent génère. Cet aspect n'est pas pris en considération dans l'objectif

usuel où un agent cherche à maximiser la richesse qu'il génère additionnée au numéraire qu'il cumule, ce qui peut engendrer des néfastes et que nous verrons plus tard. Pour tenir compte de cet aspect, mais sans modéliser précisément l'effet de cette rétroaction, nous considérons donc que les agents cherchent uniquement à maximiser la richesse qu'ils génèrent, et les épisodes sont de durée suffisamment courte pour que l'apport en numéraire à chaque agent puisse être considéré comme constant.

3.4 Mise en œuvre de l'algorithme d'apprentissage

Librairie d'apprentissage par renforcement utilisée

Les librairies d'apprentissage par renforcement les plus connues sont probablement *Tensorflow*, la librairie développée par Google, et *Pytorch*, développée par Facebook. Ces librairies permettent d'entraîner rapidement des agents dans des environnements à un seul agent. Mais l'entraînement d'agents sur des environnements multi-agents nécessitent un travail d'implémentation plutôt complexe et coûteux en temps. Nous nous sommes donc orienté sur la librairie Reinforcement Learning library (RLlib), qui fournit une solution "sur étagère" adaptée à cette spécificité d'entraînement multi-agents, que nous avons choisi pour réaliser nos expérimentations.

Choix de l'algorithme d'entraînement

La librairie RLlib inclut un grand nombre d'algorithmes d'entraînement. Nous devons choisir un algorithme aux caractéristiques suivantes :

— espace d'action continu,
— adapté au multi-agent,
— supporte une mémorisation de l'état de l'agent, par l'intermédiaire de couches Long-Short-Term-Memory (LSTM) ou Recurrent Neural Network (RNN).

La table des algorithmes candidats, compte tenu de ces contraintes, est donnée Table 5.

Algorithme d'apprentissage	Support du modèle			
	RNN	LSTM auto-wrapping	Attention	Autoreg
A2C, A3C [Mnih et al., 2016]	X	X	X	X
BC [Wang et al., 2018]	X			
IMPALA [Espeholt et al., 2018]	X	X	X	X
MARWIL [Wang et al., 2018]	X			
PG [Sutton et al., 1999]	X	X	X	X
PPO, APPO [Schulman et al., 2017]	X	X	X	X

Table 5 – *Liste des algorithmes candidats inclus dans la librairie RLlib.*

Les algorithmes candidats sont ceux qui sont adaptés au multi-agent, qui supportent un espace d'action continu, et qui supportent une mémorisation de l'état des agents via LSTM ou RNN.

Une comparaison de performance entre les différents algorithmes candidats pourrait guider le choix de l'un d'entre eux. Mais la performance d'un algorithme d'entraînement dépend des hyperparamètres utilisés. Une comparaison honnête de la performance des différents algorithmes nécessite donc de rechercher, avec un même degré d'effort, les meilleurs hyperparamètres pour chaque algorithme. Plutôt que cela, nous avons jugé plus avisé de concentrer nos efforts dans la recherche des meilleurs hyperparamètres pour un unique algorithme, en choisissant arbitrairement d'utiliser l'algorithme IMPALA.

Choix des hyperparamètres

Le choix des hyperparamètres utilisés pour l'entraînement est crucial et délicat : la qualité des résultats, et donc la pertinence des analyses qui en découlent, est fortement dépendant de la capacité des agents à apprendre/trouver une bonne stratégie.

Pour évaluer la pertinence des hyperparamètres, nous avons choisi un cas où les deux agents ont des besoins identiques et cherchent à acquérir uniquement le bien 1. La politique optimale pour chaque agent est donc de toujours révéler une offre nulle pour le bien 2, et une offre pour le bien 1 qui maximise la quantité de bien 1 acquise au cours de l'épisode. Le mécanisme de provision utilisé était le mécanisme 1HB-NR, et l'apport de 0.4 unité de numéraire par agent. Les surplus de paiement étant perdus, les agents devaient faire en sorte de limiter les surplus de paiements réalisés. Nous avons fait varier séquentiellement et un par un les paramètres suivants :

— la taille des couches,
— le nombre de couches,
— la fonction d'activation non-linéaire,
— le taux d'apprentissage,
— le coefficient d'entropie,
— la taille des lots de données,
— la taille du tampon de relecture,
— le taux d'utilisation du tampon de relecture,
— la taille et la longueur de séquence maximale de la couche LSTM, utilisée pour la mémorisation.

Pour plusieurs valeurs d'un même paramètre testé, nous avons joué plusieurs apprentissages. Nous avons ensuite observé les valeurs associées à la pire convergence (en terme de score d'épisode moyen), et les valeurs associées aux meilleures convergences, pour en déduire si certaines valeurs de certains paramètres influençaient négativement ou positivement ou pas la convergence. Nous avons répété le processus pour de nombreuses valeurs et pour de nombreuses combinaisons de paramètres, jusqu'à obtenir un ensemble d'hyperparamètres qui nous semblaient convenir.

Pour déterminer si un hyperparamètre est meilleur qu'un autre, nous avons considéré la vitesse de convergence et la répétabilité de la convergence du score d'épisode, mais surtout la stabilité de la convergence. Par exemple, on ne souhaite pas que le score d'épisode chute sensiblement après avoir convergé vers une certaine valeur. Si la stabilité de la convergence nous intéresse, c'est parce que nous devons conserver les mêmes hyperparamètres pour tous les scénarios considérés dans nos études de sensibilité pour que les résultats obtenus soient comparables. Il faut donc que les hyperparamètres choisis permettent un apprentissage robuste et efficace pour tous les scénarios que nous considérons dans nos travaux.

Les valeurs des hyperparamètres que nous avons finalement retenus sont les suivantes :
— taille et nombre de couches denses : $[32, 64, 128, 64, 32]$ (i.e. 5 couches denses).
— fonction d'activation non-linéaire : Relu.
— taux d'apprentissage : 5×10^{-5}.
— coefficient d'entropie : 0.
— taille des lots de données : 100.
— taille du tampon de relecture : 20000 épisodes complets (i.e. 200000 manches d'épisode).
— taux d'utilisation du tampon de relecture : 10%.
— LSTM (Long-Short-Term-Memory [Hochreiter and Schmidhuber, 1997]), taille [32] et longueur maximale de séquence 10 (i.e. un épisode complet).

Un autre hyperparamètre à définir est le coefficient de dévaluation γ, dont la valeur est comprise dans $[0, 1]$, que nous avons brièvement introduit précédemment et qui permet d'adapter le degré de considération des récompenses obtenues sur l'horizon futur. Par exemple, lorsque $\gamma = 0$, l'agent est myope et ne prend en compte que la récompense qui serait obtenue à la fin de la manche en cours, sans tenir compte des récompenses qui seraient obtenues plus tard. En revanche, lorsque $\gamma = 1$, l'agent prend en compte à l'identique toutes les récompenses qui seraient obtenues dans le futur. Dans nos expérimentations, nous nous trouvons dans une configuration épisodique avec un horizon temporel relativement court, nous avons donc choisi de considérer l'épisode entier dans l'horizon temporel, en fixant $\gamma = 1$.

D'autres hyperparamètres utilisés mais n'ayant pas fait l'objet d'une attention particulière sont donnés ci-dessous :

— "batch_mode" : "complete_episodes",
— "framework" : "tf" (= utilisation de tensorflow).
— "learner_queue_size" : 100.
— "learner_queue_timeout" : 1000.
— "num_envs_per_worker" : 4.
— "num_gpus" : 1.
— "num_sgd_iter" : 1.
— "num_workers" : 0.
— "rollout_fragment_length" : 100.
— "vf_loss_coeff" : 0.5.
— "vtrace" : "false".

Conclusion

Dans ce chapitre, nous avons proposé un moyen d'analyse pour juger de la qualité de différents mécanismes. Nous mettons en œuvre ce moyen d'analyse dans le prochain chapitre.

Chapitre 4

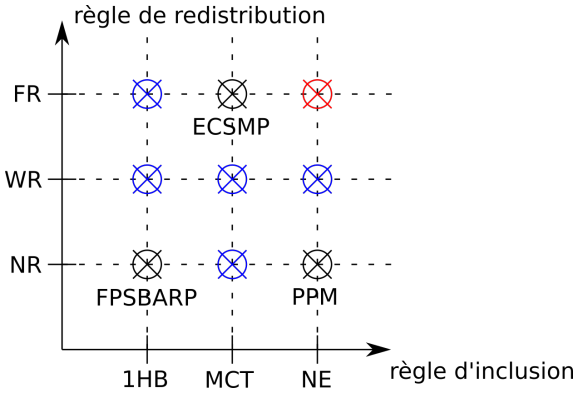
Résultats

4.1 Mécanismes de provision évalués

Puisque l'approche que nous avons choisie nous permet d'incorporer dans notre étude de nombreux mécanismes, nous proposons d'étudier une famille de mécanismes issus de toutes les combinaisons valides des 3 règles d'inclusion (1HB, MCT, NE) et des 3 règles de redistribution (NR, WR, FR), introduites respectivement dans les sections 2.2 et 2.3. La combinaison de la règle d'inclusion NE et de la règle de redistribution FR est invalide, en raison du fait que le paiement minimal associé à un agent n'est pas défini pour la règle d'inclusion NE, donc l'excès de contribution à un agent n'est pas défini non plus et la redistribution individualisée à un agent réalisée par la règle FR n'est pas possible. Pour cette raison, le mécanisme NE-FR n'est pas valide et ne sera pas étudié. Comme précisé dans la Section 2.1, tous les mécanismes que nous étudions utilisent la règle de paiement ID (l'agent paie l'offre qu'il a révélé) et la règle de fabrication PRG-MAX (quantité fabriquée qui maximise le profit révélé du groupe). Au total, nous étudions donc les 8 mécanismes représentés dans la Figure 23 sous la forme d'une table et d'un espace à deux dimensions qui permet une meilleure visualisation de la proximité entre les différents mécanismes.

Mécanisme	INC	RED
MCT-NR	MCT	NR
MCT-WR	MCT	WR
MCT-FR	MCT	FR
1HB-NR	1HB	NR
1HB-WR	1HB	WR
1HB-FR	1HB	FR
NE-NR	NE	NR
NE-WR	NE	WR
NE-FR	NE	FR

(a)



(b)

Figure 23 – *Famille de 8 mécanismes évalués, construits à partir des combinaisons des 3 règles d'inclusion et des 3 règles de redistribution.*

*En **noir**, un mécanisme connu dans la littérature. En **bleu**, un mécanisme dérivé d'un mécanisme connu de la littérature dont seule la règle d'inclusion (INC) ou la règle de redistribution (RED) est changée. En rouge, un mécanisme non étudié car non-défini. FPSBARP est l'acronyme de First-Price Sealed-Bid Auctions with a Reservation Price [McAfee and McMillan, 1987], PPM est l'acronyme de Provision-Point-Mechanism [Zubrickas, 2014], et ECSMP est l'acronyme de Equal Cost Sharing with Maximal Participation [Shichijo and Fukuda, 2016].*

4.2 Paramètres et analyses

Dans tous les cas, un épisode est constitué de 10 manches successives de provisionnement. Ce nombre de manches a été choisi pour être à la fois suffisamment faible pour permettre un apprentissage rapide par les algorithmes d'apprentissage, et suffisamment élevé pour que le comportement des agents soit représentatif du comportement qu'ils adopteraient sur le long terme.

On considère 2 agents et 2 biens binaires. De plus, la capacité de fabrication est telle qu'un seul des deux biens peut être fabriqué, au maximum, à chaque période. Cette limitation de capacité de fabrication est représentative de certains cas pouvant exister dans la réalité, par exemple lorsqu'un débit de données demandé ne peut être fourni en raison d'une limitation sur le débit disponible. Ce cas nous paraît intéressant à étudier, pour deux raisons. Premièrement, cette limitation de la capacité de fabrication impose une priorisation du bien à fabriquer par le mécanisme lorsque les agents cherchent à acquérir simultanément un bien distinct chacun. Cette priorisation peut avoir une conséquence significative sur le comportement des agents. Deuxièmement, cette contrainte implique que le coût de fabrication maximal est borné pour chaque manche. Par exemple, si le coût marginal de fabrication de chacun des deux biens binaires vaut 3, le coût de fabrication maximal est borné par 3. De ce fait, lorsque le budget total alloué à chaque manche est supérieur au coût de fabrication maximal possible, alors, selon la règle de redistribution des excès de contribution, il est possible que le budget total ne puisse jamais être écoulé. Par exemple, toujours avec un coût de fabrication marginal borné par 3, si le budget total vaut 4 et que l'excès de contribution total est toujours entièrement redistribué aux agents, le numéraire cumulé par l'ensemble des agents augmentera au minimum de 1 par manche. On qualifie ce cas particulier, où la fabrication peut être saturée pour chaque manche, de *fabrication durablement saturable*. Nous verrons qu'une saturation durable de la fabrication peut entraîner un comportement significativement différent des agents. Ces cas d'études sont représentés dans la Table 6 et seront détaillés plus tard.

Étude de sensibilité...	Matrice des besoins	Matrice d'apports	Degré de compatibilité des besoins	Degré d'équilibre des apports	Taux de charge à long-terme λ
au degré de compatibilité des besoins	$\begin{pmatrix} 1 & 0 \\ \alpha & 1-\alpha \end{pmatrix}$	$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \lambda$	$\alpha \in \{0, 0.1, ..., 1.0\}$	maximal	$\lambda \in \{80\%\}$
au degré d'équilibre des apports	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\frac{1}{1+\beta} \begin{pmatrix} 1 \\ \beta \end{pmatrix} \lambda$	nul	$\beta \in \{0.4, 0.6, ..., 1.0\}$	$\lambda \in \{100\%, 160\%, 320\%\}$

Table 6 – *Études de sensibilité réalisées.*

Dans la matrice des besoins, un coefficient dans la i-ème ligne et j-ème colonne correspond au bien-être généré par l'agent i pour la consommation d'une unité de bien j. Dans la matrice d'apports, un coefficient dans la i-ème ligne correspond au numéraire fourni à l'agent i à chaque manche. α, β et λ sont trois paramètres que l'on fait varier selon l'étude de sensibilité.

4.3 Processus de génération des résultats

Les résultats sont générés en deux temps. Premièrement, comme illustré dans la Figure 24, nous exécutons 10 instances d'entraînement qui se terminent après avoir atteint 1 000 000 manches (c'est-à-dire 100 000 épisodes de 10 manches) pour chaque scénario dans un ensemble de scénarios à étudier. Dans toutes les instances d'apprentissage, nous utilisons le même ensemble d'hyperparamètres choisis, comme décrit dans le Chapitre 3 Section 3.4. Un répertoire unique est créé pour chaque instance de formation, contenant des scalaires Tensorboard à des fins de surveillance et un fichier "result.json" qui enregistre certaines valeurs utiles.

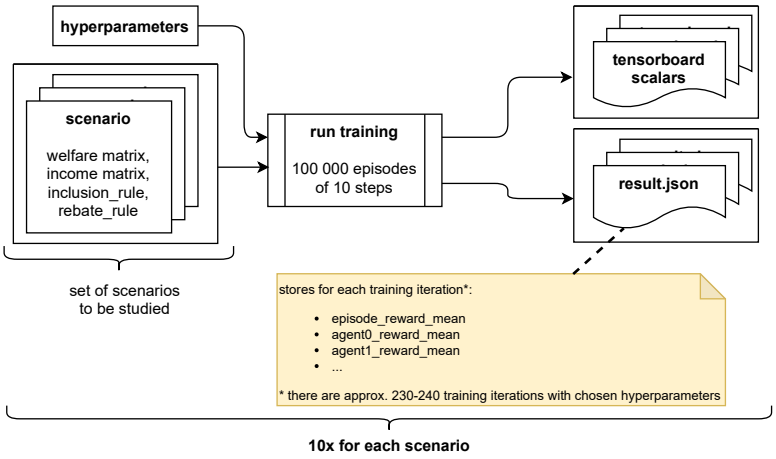
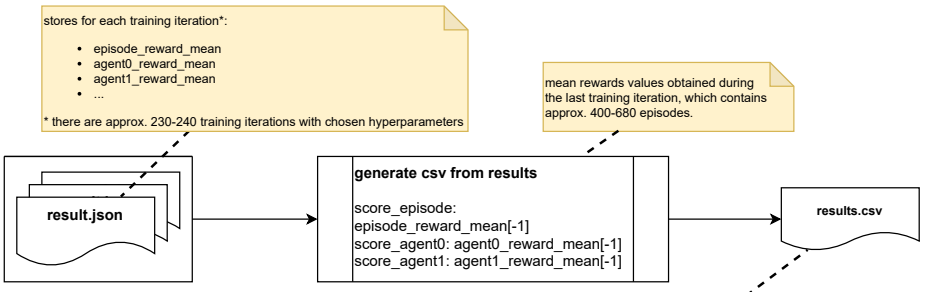


Figure 24 – *Processus de génération des résultats (partie 1/2).*

Deuxièmement, comme illustré dans la Figure 25, nous rassemblons les valeurs de tous les fichiers "result.json" générés pour générer un panda Dataframe, qui est finalement écrit dans un fichier "results.csv".



results.csv

INCLUSION_RULE	REBATE_RULE	INCOME[0]	INCOME[1]	REBATE_WEIGHT[0]	REBATE_WEIGTH[1]	WELFARE[1,0]	WELFARE[1,1]	SCORE_EPISODE	SCORE_AGENT0	SCORE_AGENT1
1HB	WR	0.4	0.4	0.5	0.5	0.2	0.8	6	1	5
...	...	...	...	...	...	...	...	...	...	...

Figure 25 – *Processus de génération des résultats (partie 2/2).*

Les valeurs montrées dans le fichier "results.csv" sont fictives et uniquement destinées à des fins d'illustration.

La Figure 26 montre une capture d'écran de l'interface Tensorboard, ce qui permet de visualiser la convergence de quelques valeurs scalaires au cours de chaque instance d'entraînement sur toutes nos expériences [1].

Figure 26 – *Capture d'écran de l'interface Tensorboard, montrant la convergence de* 3 *scalaires pour les différentes instances d'entraînement.*

Une instance d'entraînement dure environ 100000 *épisodes de* 10 *manches, c'est-à-dire environ 1 000 000 de manches. "episode_reward_mean" correspond à la richesse totale générée en moyenne au cours d'un épisode. "policy_reward_mean/agent_0" (resp. "policy_reward_mean/agent_1") correspond à la richesse individuellement générée par l'agent 1 (resp. agent 2) en moyenne au cours d'un épisode.*

1. Il y a 8800 instances d'entraînement effectuées pour la première étude de sensibilité (8 mécanismes x 11 valeurs du paramètre α x 10 répétitions), et il y a 2400 instances d'entraînement effectuées pour la deuxième étude de sensibilité (8 mécanismes x 10 valeurs du paramètre β x 3 valeurs du paramètre λ x 10 répétitions).

4.4 Critères d'évaluation

Pour nous, les critères qui caractérisent la performance d'un mécanisme sont les suivants :

— **maximisation de la richesse sociale :** nous voulons maximiser la richesse sociale générée sur le long terme, compte-tenu du budget alloué sur le long terme.

— **maximisation de la richesse individuelle :** nous voulons maximiser la richesse générée individuellement par chaque agent. Quand la maximisation de la richesse sociale au moindre coût et la maximisation de la richesse individuelle des agents sont incompatibles, nous voulons favoriser la maximisation de la richesse individuelle pour nous prémunir d'une potentielle famine de certains agents.

— **élégance :** toutes autres choses égales, nous préférons que le mécanisme de provision soit simple et général (critère inspiré de [Kelly et al., 2003]). Ce critère est pertinent car notre mécanisme de provision, comme tout système de règles, peut être "piraté". On entend ici par "piraté" la définition donnée par B. Schneier dans son récent essai [Schneier, 2021], à savoir l'utilisation du système à des fins non-intentionnées et non-anticipées par le concepteur. Hors, toujours d'après Schneier, un système complexe avec beaucoup de règles est particulièrement vulnérable, simplement parce qu'il y a plus de possibilités pour des conséquences non-intentionnées et non-anticipées.

— **rendement d'influence :** nous voulons qu'il existe un lien direct entre la répartition du budget et la capacité des agents à influencer le mécanisme. Très grossièrement, si 2 agents ont des besoins totalement incompatibles (i.e. le bien qui intéresse un agent n'intéresse pas l'autre agent, et vice versa), et que par exemple le budget est réparti entre les agents $(1, 2)$ dans les proportions $(70\%, 30\%)$, nous souhaitons que les agents puissent acquérir les biens qu'ils convoitent dans les mêmes proportions $(70\%, 30\%)$ sur le long terme (à condition bien sûr que les agents ont les mêmes capacités cognitives). Nous introduirons plus précisément cette notion dans la deuxième analyse de sensibilité.

Remarque 7. *Une propriété qui est parfois recherchée est la propriété de* budget équilibré. *Ici, cette propriété permettrait simplement que le budget alloué pour la provision des données puisse être entièrement dépensé par les agents sur le long terme (budget non-excédentaire), sans jamais être dépassé (budget non-déficitaire). Nous souhaitons que le budget soit non-déficitaire, mais la propriété de budget non-excédentaire n'est pas réellement recherchée. En réalité, à performances égales, un mécanisme à budget non-déficitaire et excédentaire nous conviendrait tout aussi bien qu'un mécanisme à budget non-déficitaire et non-excédentaire.*

4.5 Analyse de sensibilité au degré de compatibilité des besoins

Dans cette étude de sensibilité, résumé à la première ligne de la table 6, nous cherchons à évaluer comment varie le degré de coopération de deux agents en fonction du degré de compatibilité de leurs besoins. On fixe une matrice de besoins $U = [[1, 0], [\alpha, 1 - \alpha]]$, où α est un coefficient dont la valeur est choisie dans $\{0, 0.1, 0.2, ..., 1\}$. Le cas où α vaut 0 correspond au cas où le besoin des deux agents est entièrement incompatible/disjoint : chaque agent rivalise pour acquérir un bien distinct. Le cas où α vaut 1 correspond au cas où $U = [[1, 0], [1, 0]]$, i.e. le besoin de l'agent 2 est entièrement compatible avec celui de l'agent 1. Le cas où α est compris entre 0 et 1 correspond au cas intermédiaire, où la richesse générée par le bien 1 est supérieure à celle du bien 2 lorsque $\alpha < 0.5$, et réciproquement inférieure lorsque $\alpha > 0.5$. La matrice d'apport est fixée à $[0.5, 0.5] \times 0.8$, c'est-à-dire que chaque agent a le même pouvoir d'influence, et le taux de charge à long-terme 0.8 ne permet pas la fabrication d'un bien pour toute période (coût = 1), i.e. la fabrication n'est pas durablement saturable.

Comme un épisode complet comporte 10 manches, un agent reçoit, en l'absence de règle de redistribution, $0.4 \times 10 = 4$ unités de numéraire sur 10 manches. Chaque agent peut donc acquérir seul au maximum 4 unités de bien, ou 8 unités de bien en coopérant avec son rival.

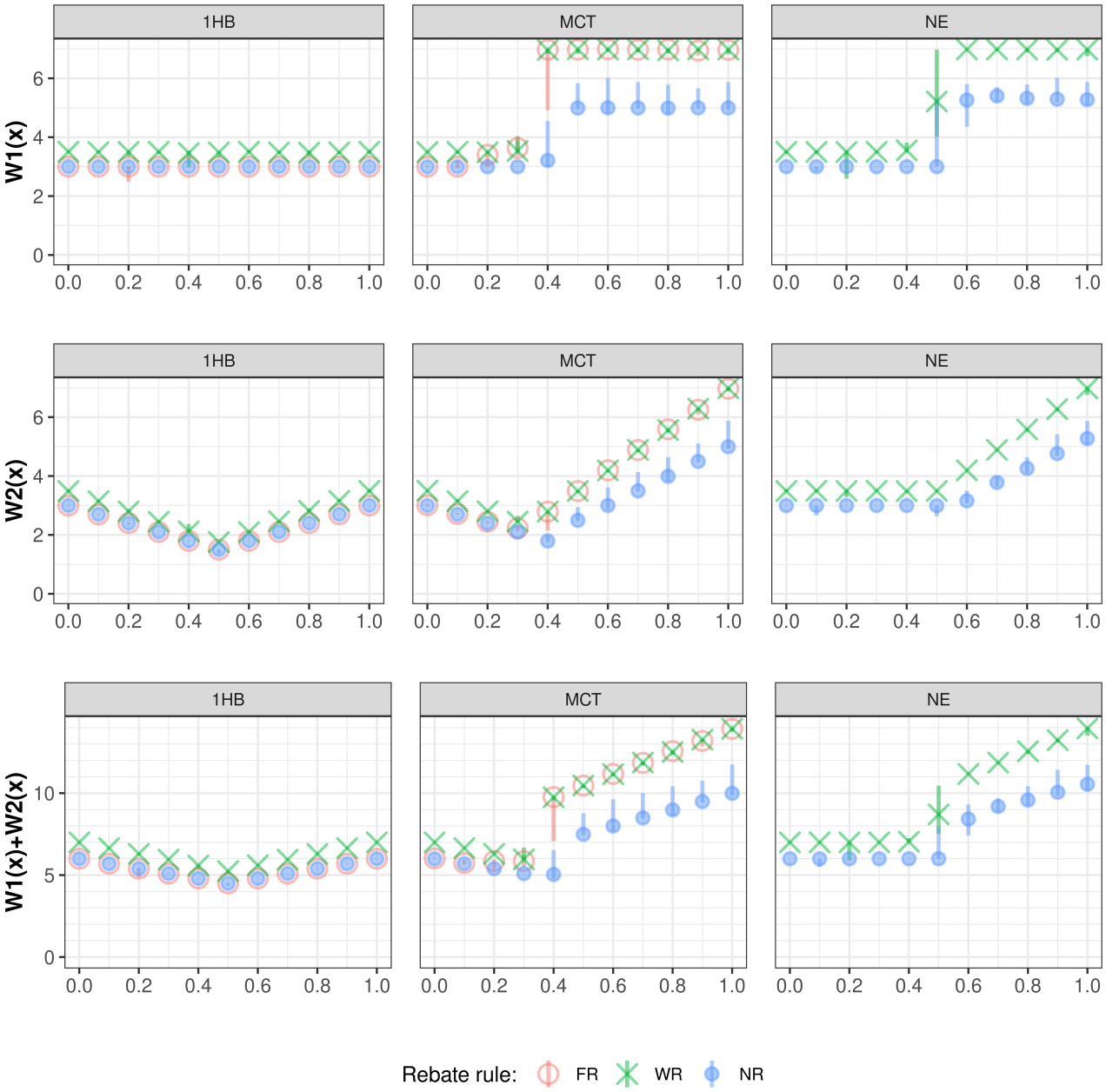


Figure 27 – *Richesse générée en moyenne au cours d'un épisode entier en fonction du degré de compatibilité α des besoins des deux agents.*

$W1(\alpha)$ et $W2(\alpha)$ correspondent respectivement à la richesse générée en moyenne au cours d'un épisode entier par l'agent 1 et par l'agent 2. Les résultats sont générés pour différentes valeurs de $\alpha \in \{0.0, 0.1, 0.2, ..., 1.0\}$. Pour chaque valeur de α, l'entraînement des agents est effectué sur 1 million de manches, soit 100000 épisodes. La richesse moyenne générée pour un entraînement est calculée à partir des 100 derniers épisodes de l'entraînement. Le processus d'entraînement est répété 10 fois, avec de nouveaux agents à chaque fois, ce qui correspond donc à une série de 10 valeurs pour chaque valeur de α. Un marqueur indique la médiane d'une série, tandis que le bas et le haut de la barre verticale indiquent respectivement le minimum et le maximum d'une série.

Les résultats de cette analyse de sensibilité sont présentés sur la page précédente, Figure 27.

Avec la règle d'inclusion 1HB, un seul des deux agents peut acquérir un bien. Puisque les deux agents ont les mêmes capacités cognitives (leurs modèles sont identiques) et le même pouvoir d'influence (car ils ont le même apport de numéraire par période), on s'attend à ce qu'ils puissent acquérir autant de biens chacun, indépendamment du mécanisme et de la valeur de α. Comme l'apport de numéraire par agent vaut 0.4 par manche, et que le bien ne peut être fabriqué que lorsqu'un agent propose une offre qui permet de couvrir seul le coût de fabrication, il faut attendre la 3ème manche pour que l'un des agents puisse acquérir une première unité de bien avec 1.2 unités de numéraire cumulé depuis le début de l'épisode. Puisque chaque agent peut cumuler 4 unités de numéraire sur les 10 manches en l'absence de redistribution d'excès de contribution, ils peuvent acquérir chacun 4 unités de numéraire sur 7 manches, de la 3ème à la 10ème manche incluse, soit 8 unités de bien au total. Mais comme il n'est pas possible de fabriquer plus d'une unité de bien par manche, 7 unités de bien au maximum peuvent être fabriquées sur ces 7 manches, ce qui provoque un conflit. En effet, pour au moins une manche, un des deux agents ne pourra pas obtenir le bien qu'il convoîte. Comme les deux agents ont la même capacité cognitive, on peut s'attendre à ce que l'un des deux remporte aléatoirement le bien, au détriment de l'autre. Dans le meilleur des cas, un agent devrait donc obtenir en moyenne 3.5 unités de bien sur un épisode entier. C'est effectivement ce que l'on observe avec le mécanisme 1HB-WR :

— la richesse générée par l'agent 1 par unité de bien 1 consommé vaut 1, indépendamment de α. Il se trouve que $W1(\alpha)$ vaut 3.5, indépendamment de α, ce qui signifie que l'agent a consommé 3.5 unités de bien 1 en moyenne sur un épisode.

— la richesse générée par l'agent 2 par unité de bien 1 (resp. 2) consommé vaut 1 lorsque $\alpha = 1$ (resp. $\alpha = 0$), et décroit linéairement pour atteindre 0.5 lorsque α tend vers 0.5. A quantité de bien consommé égale, l'agent 2 génère plus de richesse avec le bien 1 lorsque $\alpha > 0.5$, plus de richesse avec le bien 2 lorsque $\alpha < 0.5$, et la même quantité de richesse avec le bien 1 qu'avec le bien 2 lorsque $\alpha = 0.5$. Là encore, on observe que $W2(\alpha)$ vaut 3.5 fois cette courbe en V.

Les mécanismes 1HB-NR et 1HB-FR semblent légèrement moins efficace, avec 3 unités de bien obtenus en moyenne par chaque agent. Une investigation plus poussée est nécessaire pour déterminer si ce faible écart de performance entre les 3 mécanismes est fondé, ou simplement dû à une imprécision des résultats obtenus. Une piste serait par exemple de reproduire les résultats mais avec des épisodes plus longs, par exemple des épisodes de 50 manches [2].

Par ailleurs, lorsque les besoins sont totalement incompatibles ($\alpha = 0$), on constate que la richesse générée par chaque agent est indépendante de la règle d'inclusion. La raison est que lorsque $\alpha = 0$ l'agent 1 ne peut bénéficier que du bien 1, et l'agent 2 ne peut bénéficier que du bien 2. Par conséquent, autoriser les deux agents à consommer le bien fabriqué ne permet pas d'inciter les agents à coopérer, et ne permet pas non plus de générer davantage de richesse par rapport au cas où seul l'agent qui fait la meilleure offre est autorisé à consommer le bien fabriqué (puisque seul cet agent peut générer de la richesse avec le bien fabriqué). Au contraire lorsque les besoins sont totalement compatibles ($\alpha = 1$), on constate que la richesse générée par chaque agent vaut environ deux fois plus avec les règles MCT et NE comparé à la règle 1HB. Ce résultat révèle l'inadéquation de la règle d'inclusion 1HB pour le provisionnement d'un bien de club, dont la consommation est limitée à un agent alors que le bien, du fait de sa non-soustractibilité à la consommation, peut profiter à plusieurs agents sans coût additionnel.

2. La modification de la longueur des épisodes nécessite d'envisager l'ajustement de certains paramètres, comme par exemple le coefficient de dévaluation γ, de la longueur maximale de séquence des cellules LSTM, le nombre de manches qui conditionne l'arrêt d'un entraînement, la taille du tampon d'expérience, etc.

En théorie, les agents pourraient acquérir 8 unités de biens sur un épisode entier en coopérant parfaitement (par exemple avec l'acquisition du bien 1 aux manches 2,3,4,5,7,8,9,10). Cela ne peut toutefois être obtenu qu'avec une gestion parfaite du numéraire dépensé par les agents et des calculs d'arrondis dans la simulation, ce qui explique probablement pourquoi les agents ne parviennent à obtenir que 7 unités de bien (au lieu de 8) dans le meilleur des cas dans nos résultats.

Plus globalement, on remarque que la richesse moyenne générée $W1(\alpha)$, $W2(\alpha)$, et $W1(\alpha)+W2(\alpha)$ est plus faible avec la règle NR qu'avec les règles WR et FR. Ceci s'explique probablement par le fait que, pour WR et FR, aucun surplus de paiement n'est perdu, ce qui permet au budget total initialement prévu d'être presque entièrement dépensé sur le long terme ; contrairement à la règle NR, où lorsqu'un surplus de paiement est réalisé, la quantité de numéraire correspondante est perdue et le budget total initialement prévu ne peut pas être entièrement dépensé. On peut également constater une disparité plus importante des valeurs obtenues entre différentes sessions d'apprentissage, dont l'étendue des valeurs est représentée par une barre verticale, ce qui semble indiquer qu'il est plus difficile pour un agent de converger vers la stratégie optimale.

Nous proposons d'éliminer les mécanismes les plus inefficaces, c'est-à-dire tous les mécanismes basés sur la règle de redistribution NR (inefficacité qui provient de la perte des surplus de paiement) et les mécanismes basés sur la règle d'inclusion 1HB (inefficacité qui provient de l'exclusion infondée d'agents qui pourraient bénéficier du bien fabriqué sans coût supplémentaire). Il nous reste alors 3 mécanismes MCT-WR, MCT-FR, et NE-WR, dont nous illustrons à nouveau les résultats, de façon plus détaillée, sur la Figure 28 dans la page suivante.

L'indicateur $W1(\alpha)$ est révélateur de la coopération entre les deux agents. En effet, comme la richesse générée pour l'agent 1 est indépendante de α, et que les deux agents cherchent toujours à acquérir un maximum de biens, la richesse générée par l'agent 1 dépend de la coopération ou non de l'agent 2 à acquérir le bien 1. Lorsque l'agent 1 contribue seul à l'acquisition du bien 1, il acquiert entre 3 et 3.5 unités de bien 1 par épisode. Lorsque les deux agents coopèrent à l'acquisition du bien 1, ils parviennent à acquérir ensemble une quantité deux fois plus importante de bien, environ 6 à 7 unités de bien 1. Avec $W1(x)$, on voit aisément que la règle d'inclusion 1HB impose une rivalité entre les deux agents, qui n'acquiert jamais plus de 3.5 unités de bien 1, indépendamment de la règle de redistribution et de la valeur de α, pour la simple et bonne raison que la règle 1HB n'autorise qu'un seul agent parmi les deux à consommer le bien fabriqué. En revanche, avec la règle d'inclusion MCT et NE, la coopération entre les deux agents dépend de la valeur α. Avec $W1(\alpha)$, nous observons que :

— pour MCT, la coopération est nulle (resp. totale) lorsque $\alpha \leq 0.3$ (resp. $x \geq 0.5$). Ceci s'explique par le fait que l'agent 2 peut acquérir le bien 1 pour 50% du coût de fabrication lorsque l'agent 1 contribue au moins à 50% du coût de fabrication (le coût de fabrication étant alors divisé équitablement entre les deux agents), ou le bien 2 pour 100% du coût de fabrication. En fait, l'agent 2 peut avoir un intérêt à acquérir le bien 1 lorsque le rapport richesse générée / coût est meilleur que pour le bien 2, i.e. lorsque $\alpha/0.5 > (1-\alpha)/1$, c'est-à-dire lorsque $\alpha > 1/3$. Mais comme l'acquisition du bien 1 pour 50% du coût est conditionnée par la coopération des deux agents, un agent peut pénaliser l'autre en refusant la coopération. L'étendue des valeurs $W1(\alpha)$ obtenues pour $\alpha = 0.4$ indique que la coopération est toujours obtenue avec la règle WR, alors qu'elle n'est pas toujours obtenue avec les règles NR et FR.

— pour NE, la coopération est nulle (resp. totale) lorsque $\alpha \leq 0.4$ (resp. $\alpha \geq 0.6$). Ceci s'explique par le fait que l'agent 2 peut choisir librement de coopérer ou non avec l'agent 1, dans son propre intérêt et en toute liberté, sans que le mécanisme n'avantage ou ne pénalise l'agent en fonction de ce choix. Comme l'agent 2 génère une richesse plus importante avec le bien 1

qu'avec le bien 2 lorsque $\alpha < 0.5$, et réciproquement lorsque $\alpha > 0.5$, l'agent 2 choisi d'acquérir le bien 1 lorsque $\alpha < 0.5$ et choisi d'acquérir le bien 2 lorsque $\alpha > 0.5$. Lorsque $\alpha = 0.5$, l'agent peut choisir, a priori, indépendamment d'acquérir le bien 1 ou 2. Nos résultats indiquent que la coopération n'est pas toujours obtenue lorsque $\alpha = 0.5$.

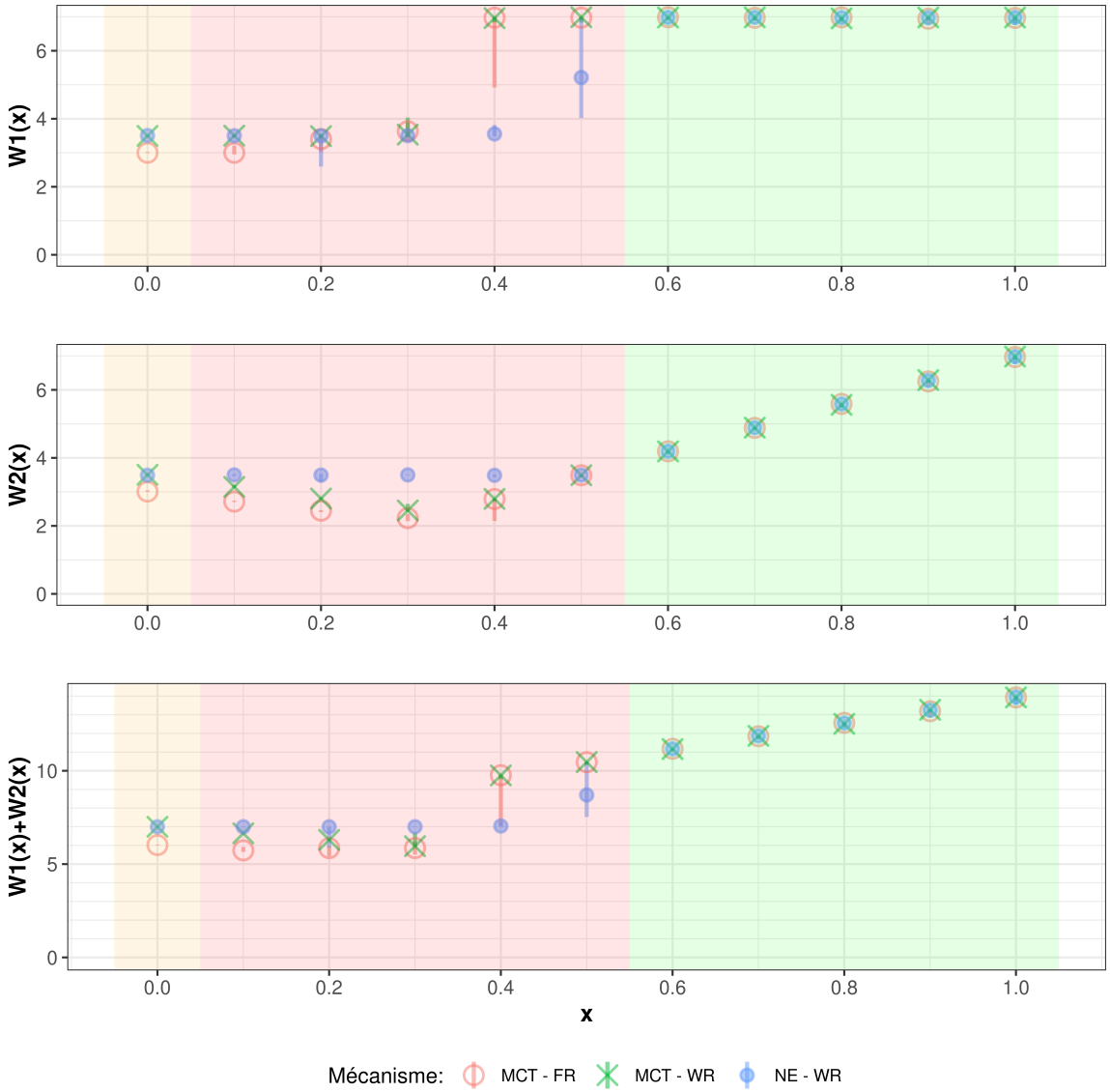


Figure 28 – *Richesse générée en moyenne au cours d'un épisode entier en fonction du niveau de compatibilité α des besoins des deux agents.*

$W1(\alpha)$ et $W2(\alpha)$ correspondent respectivement à la richesse générée en moyenne au cours d'un épisode entier par l'agent 1 et par l'agent 2. Les résultats sont générés pour différentes valeurs de $\alpha \in \{0.0, 0.1, 0.2, ..., 1.0\}$. Pour chaque valeur de α, l'entraînement des agents est effectué sur 1 million de manches, soit 100000 épisodes. La richesse moyenne générée pour un entraînement est calculée à partir des 100 derniers épisodes de l'entraînement. Le processus d'entraînement est répété 10 fois, avec de nouveaux agents à chaque fois, ce qui correspond donc à une série de 10 valeurs pour chaque valeur de α. Un marqueur indique la médiane d'une série, tandis que le bas et le haut de la barre verticale indiquent respectivement le minimum et le maximum d'une série.

La Figure 29 montre $W1(\alpha)$ de façon plus détaillée pour $\alpha \in \{0.4, 0.5\}$ pour les 3 mécanismes les plus intéressants MCT-FR, MCT-WR, et NE-WR de façon plus détaillée pour $\alpha = 0.4$ et $\alpha = 0.5$, où la coopération peut être systématique ou aléatoire selon le mécanismes.

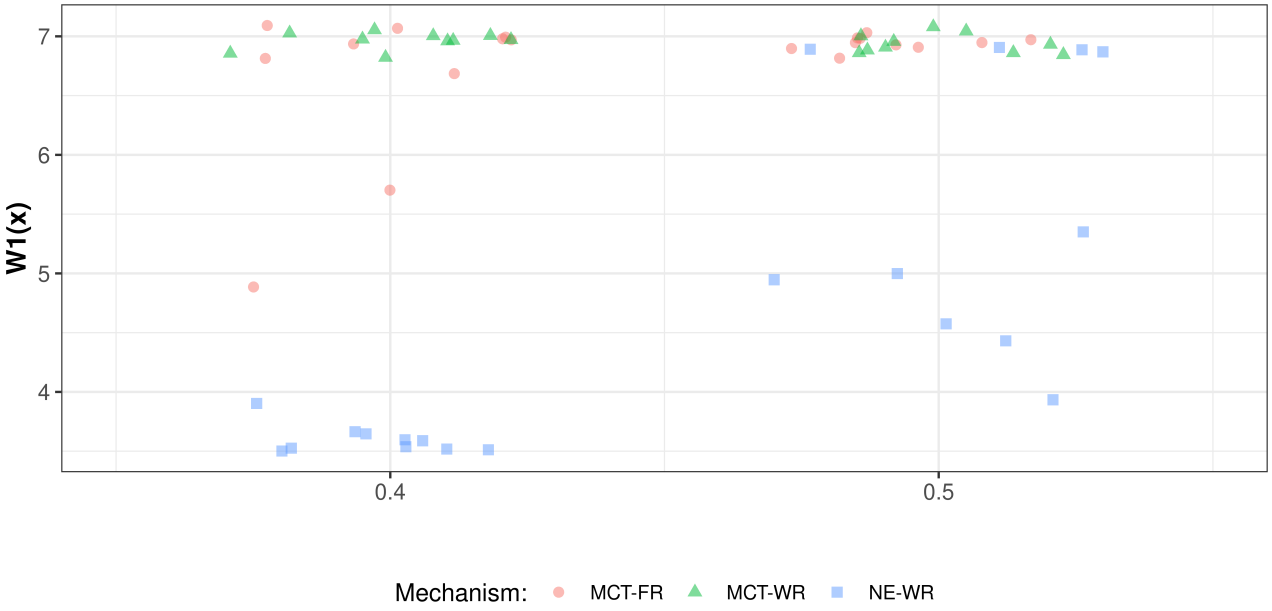


Figure 29 – *Richesse moyenne $W1(\alpha)$ générée par l'agent 1 au cours d'un épisode entier en fonction du niveau de compatibilité α des besoins des deux agents.*

Les résultats sont générés pour différentes valeurs de $\alpha \in \{0.4, 0.5\}$. Pour chaque valeur de α, l'entraînement des agents est effectué sur 1 million de manches, soit 100000 épisodes. La richesse moyenne générée pour un entraînement est calculée à partir des 100 derniers épisodes de l'entraînement. Le processus d'entraînement est répété 10 fois, avec de nouveaux agents à chaque fois, ce qui correspond donc à une série de 10 valeurs pour chaque valeur de α. Pour mieux distinguer les différents marqueurs, une valeur aléatoire d'amplitude 0.03 (resp. 0.1) est ajoutée à la position horizontale (resp. verticale) des marqueurs.

Deux mécanismes ressortent principalement du lot, ce sont les mécanismes MCT-WR et NE-WR. Ces deux mécanismes ont en commun la règle de redistribution WR. Ils se distinguent par la règle d'inclusion. D'une part, MCT domine (strictement) NE pour la seule valeur $\alpha \in \{0.4\}$ évaluée, mais c'est en réalité probablement le cas pour $\alpha \in]1/3, 1/2[$ (en raison du fait que l'agent 2 préfère acquérir le bien 2 et sera exclu à la consommation par la règle MCT lorsque le bien 1 sera fabriqué). D'autre part, NE domine (strictement) MCT pour les valeurs $\alpha \in \{0.1, 0.2, 0.3\}$ évaluées, mais c'est en réalité probablement le cas pour $\alpha \in]0, 1/3[$. Selon ces premiers résultats, il semble donc que ces deux mécanismes soient les meilleurs candidats du point de vu de la richesse moyenne générée au total par les deux agents. La prochaine étude de sensibilité va permettre d'apporter de nouveaux éléments de comparaison entre ces deux mécanismes.

4.6 Analyse de sensibilité au niveau d'équilibre des apports

Dans cette étude de sensibilité, nous cherchons à évaluer dans quelle mesure l'influence d'un agent varie en fonction de la proportion du pouvoir d'influence qui lui est alloué, relativement au pouvoir d'influence total. Concrètement, lorsque l'apport en numéraire d'un agent représente, par exemple, 30% du budget total, i.e. 30% de l'apport en numéraire agrégé par somme sur l'ensemble des agents, on souhaite idéalement que cela se traduise par le fait que cet agent acquiert en moyenne 30% du type bien qui l'intéresse le plus.

Ici, les deux agents $(1, 2)$ génèrent une unité de richesse pour la consommation respective d'une unité de bien $(1, 2)$. De plus, leurs besoins sont totalement incompatibles, c'est-à-dire que le bien 2 ne bénéficie pas à l'agent 1, et le bien 1 ne bénéficie pas à l'agent 2. Puisque l'objectif d'un agent est de maximiser la richesse qu'il génère individuellement, un agent i cherche (uniquement) à influencer le mécanisme pour acquérir le bien i. L'influence qu'un agent i a pu effectivement exercer sur le mécanisme peut donc se mesurer par la quantité de bien i qu'il obtient, ou de façon équivalente par la richesse que l'agent génère.

Notons (x_1, x_2) l'apport périodique de numéraire pour l'agent $(1, 2)$, et (y_1, y_2) la quantité de bien convoité et acquis par l'agent $(1, 2)$ au cours d'un épisode entier.

Définition 18 (Ratio d'influence allouée)**.** *Le* ratio d'influence allouée *à un agent i est noté $\xi_{in}(i)$ et défini comme $x_i/(x_1 + x_2)$.*

$\xi_{in}(i)$ correspond à l'influence allouée à l'agent i relativement à l'influence totale, allouée à l'ensemble des agents.

Définition 19 (Ratio d'influence effective)**.** *Le* ratio d'influence effective *d'un agent i est noté $\xi_{out}(i)$ et défini comme $y_i/(y_1 + y_2)$.*

$\xi_{out}(i)$ correspond à l'influence effective de l'agent i relativement à l'influence effective totale, de l'ensemble des agents.

Ces deux notions nous permettent de définir le *rendement d'influence*, qui reflète l'efficacité avec laquelle un agent arrive à influencer le mécanisme compte tenu de l'influence qui lui est allouée.

Définition 20 (Rendement d'influence)**.** *Le* rendement d'influence *d'un agent i est noté $\eta(i)$ et défini comme $\frac{\xi_{out}(i)}{\xi_{in}(i)} = \frac{y_i/(y_1+y_2)}{x_i/(x_1+x_2)} = \frac{y_i(x_1+x_2)}{x_i(y_1+y_2)}$*

Remarque 8. *Il est important de garder à l'esprit que le rendement d'influence peut ici être facilement défini parce que les besoins des agents sont totalement incompatibles. En effet, imaginons le cas délictueux d'un mécanisme basé sur la règle d'inclusion MCT, où l'agent* 1 *génère* $(1, 0)$ *unité de richesse pour la consommation du bien* $(1, 2)$*, tandis que l'agent* 2 *génère* $(0.6, 1)$ *unité de richesse pour la consommation du bien* $(1, 2)$*. Sans ambiguïté, on peut dire que l'agent* 1 *cherche à acquérir le bien* 1*. Mais comment considérer le cas de l'agent* 2 *? A priori, l'agent* 2 *bénéficie davantage du bien* 2 *que du bien* 1*. Mais la règle d'inclusion MCT peut forcer l'agent* 2 *à acquérir le bien* 1 *pour* 0.5 *unité de numéraire, en coopération avec l'agent* 1 *qui contribue à la même hauteur pour un bénéfice pourtant deux fois supérieur. Est-il juste de considérer que l'agent* 1 *a utilisé son numéraire volontairement pour influencer le mécanisme pour l'acquisition du bien* 2*, alors qu'il ne l'a fait que sous le résultat indirect de la contrainte de la règle d'inclusion MCT ? Et quand bien même cette possibilité serait envisagée, considérer dans ce cas que les deux agents ont influencé à la même hauteur le mécanisme pour la fabrication du bien est fortement discutable. Mais alors, dans quelle juste proportion l'influence de chaque*

agent pourrait-elle être imputée de la fabrication du bien 1 *? Trouver une définition du rendement d'influence qui fasse consensus dans une configuration où les besoins des agents sont partiellement compatibles paraît difficile, voire impossible.*

Nous souhaitons que, sur le long-terme, le rendement d'influence soit proche de 100% pour chaque agent, c'est-à-dire que chaque agent arrive à influencer le mécanisme dans les proportions prévues. Si le rendement d'influence d'un agent est sensiblement supérieur à 100% (resp. inférieur à 100%), alors cela signifie que l'agent arrive à influencer le mécanisme plus (resp. moins) que prévu, ce qui n'est pas souhaitable.

Les règles de redistribution se différencient uniquement dans la façon dont les surplus de paiements sont redistribués. Lorsque le taux de charge à long-terme est strictement inférieur à 100%, les agents ne peuvent pas contribuer suffisamment, même en coopérant, pour fabriquer un bien pour chaque période. La rivalité entre les agents sera d'autant plus faible que le taux de charge sera faible. Par exemple, à partir d'un taux de charge de 100%, il est en théorie possible que les agents ne se "gênent" jamais, à condition qu'ils arrivent à coordonner leurs acquisitions. Au contraire, plus le taux de charge à long-terme λ est supérieur à 100%, plus les agents devront rivaliser entre eux pour acquérir leur bien. Par exemple, avec un taux de charge de 320% et une répartition des apports de $(50\%, 50\%)$, chaque agent reçoit 1.6 unité de numéraire par manche. Dans ces conditions, imaginons qu'un agent décide de révéler $PMA(1) \geq 1.6$ à chaque manche. Son adversaire doit alors nécessairement révéler $PMA(1) \geq 1.6$ pour avoir la possibilité d'acquérir un bien. Le surplus de paiement d'un agent "intelligent" sera donc vraisemblablement d'au moins 0.6 unité de numéraire par manche. Pour montrer de façon progressive les différences entre les règles de redistribution, qui ne peuvent apparaître qu'avec l'apparition de surplus de paiements, nous proposons donc d'évaluer les mécanismes pour des valeurs de taux de charge à long-terme λ choisies dans l'ensemble $\{100\%, 160\%, 320\%\}$.

Étant donné que le coût de fabrication maximum par manche vaut 1 (un seul bien peut être fabriqué à chaque manche pour un coût de 1), on fixe une matrice d'apports de numéraire $A = \frac{1}{1+\beta}[1, \beta]\lambda$, où β représente le niveau d'équilibre des apports entre les deux agents et λ représente le taux de charge à long-terme, i.e. l'apport total $x_1 + x_2$ divisé par le coût maximal qui peut être engendré par la fabrication des biens (ici égal à 1 puisque chaque bien fabriqué engendre un coût de 1 et au maximum un bien peut être fabriqué par manche).

Lorsque la valeur de β vaut 1, l'apport de numéraire est équilibré entre les deux agents. Plus β est éloigné de 1, plus l'équilibre d'apports de numéraire entre les deux agents est grand. Nous nous limitons ici à une valeur de $\beta \in [0.4, 1.0]$, ce qui correspond à un ratio d'influence compris dans $[0.4/1.4, 1/1.4] \approx [29\%, 71\%]$. La raison à cela est que lorsque $\beta < 0.4$ le déséquilibre est grand et l'acquisition d'un bien par l'agent le plus pauvre devient un évènement rare. Ceci rend la recherche d'une stratégie efficace plus difficile pour les agents, et le comportement obtenu sur des épisodes de 10 manches n'est plus représentatif du comportement sur le long-terme. Des travaux de recherche supplémentaires sont nécessaires pour évaluer la performance des mécanismes dans des conditions de déséquilibres plus extrêmes.

Dans l'analyse de sensibilité précédente, les mécanismes se distinguaient davantage par la règle d'inclusion que par la règle de redistribution. Ceci s'explique par le fait que le taux de charge λ était de 80%, et que par conséquent les surplus de paiements étaient faibles. On remarque dans ces nouveaux résultats que c'est encore le cas pour $\lambda = 100\%$. Néanmoins, dans cette deuxième analyse de sensibilité, les mécanismes se distinguent davantage par la règle de redistribution que par la règle d'inclusion lorsque $\lambda \in \{1.0, 1.6, 3.2\}$.

La Figure 30 représente le rendement d'influence d'un agent en fonction du ratio d'influence (= ratio du budget total) qui lui est alloué. Lorsque le revenu de l'agent 2 varie de 0 à 3.2, la proportion du budget total de l'agent 1 (resp. de l'agent 2) varie de 100% à 50% (resp. de 0% à 50%). Comme les 2 agents sont équivalents, le rendement d'influence est calculé à partir de l'agent 2 pour la plage de 0% à 50%, et à partir de l'agent 1 sur la plage de 50% à 100% (pour 50% exactement, le rendement d'influence est calculé à partir des 2 agents).

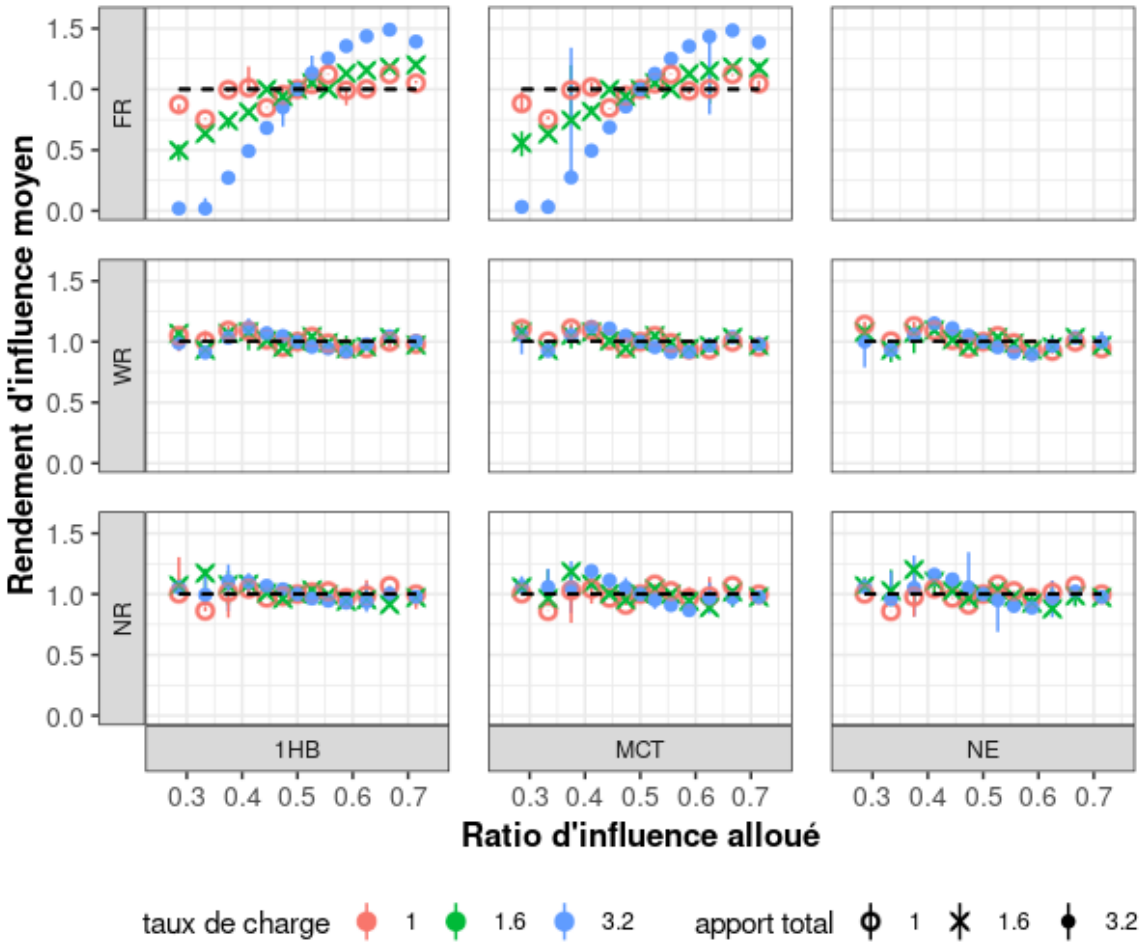


Figure 30 – *Rendement d'influence moyen d'un agent en fonction du ratio d'influence alloué à l'agent, défini comme l'apport de l'agent par rapport à l'apport total, et en fonction du taux de charge (de la fabrication) λ.*

Le rendement d'influence correspond à la richesse générée en moyenne au cours d'un épisode entier par un agent i. Le ratio d'influence alloué x de l'agent i vaut $x_i/(x_1+x_2)$. Les résultats sont générés pour différentes valeurs de ratio d'influence alloué $x \in \{0.4, 0.5, ..., 1.0\}$ et différentes valeurs d'apport total $x_1+x_2 \in \{1, 1.6, 3.2\}$, ici égal au taux de charge λ. Pour chaque paire de valeurs (x, λ), l'entraînement des agents est effectué sur 1 *million de manches, soit* 100000 *épisodes. Le rendement d'influence moyen d'un agent i généré pour un entraînement correspond à la richesse générée par l'agent i, et est calculé à partir des* 100 *derniers épisodes de l'entraînement. Le processus d'entraînement est répété* 10 *fois, avec de nouveaux agents à chaque fois, ce qui correspond donc à une série de* 10 *valeurs pour chaque paire de valeurs (x, λ).*

Contrairement à l'analyse précédente, on peut observer que la règle d'inclusion influe très peu sur le rendement d'influence, et que les mécanismes se distinguent donc essentiellement par leur règle de redistribution.

Les règles de redistribution NR et WR semblent être idéales du point de vue du critère du rendement d'influence. En effet, les écarts sont globalement contenus dans +/-25% et évoluent de façon symétrique autour de 100%, ce qui s'explique probablement par un effet de bord dû à la courte durée des épisodes (10 manches). Par exemple, pour un taux de charge de 3.2 et une ratio d'influence alloué de 0.27, la quantité de bien acquis par l'agent 1 quand la quantité de bien fabriquée par le mécanisme sur un épisode vaut 10 doit être de 2.7 pour que le rendement d'influence soit idéal et donc égal à 100%. Si les conditions font que l'agent acquiert en moyenne 2 biens (ou resp. 3) par période, le rendement d'influence obtenu vaudra environ 74% (resp. environ 111%).

La règle FR engendre un comportement clairement mauvais, avec un rendement d'influence qui s'éloigne sensiblement du rendement de référence dans certains cas. Par exemple, pour un taux de charge de 320% et un ratio d'influence inférieur à $0.5/1.5 = 1/3$, le rendement d'influence de l'agent vaut 0%, c'est-à-dire que l'agent n'arrive pas du tout à influencer le mécanisme pour acquérir un seul bien sur les 10 manches de l'épisode. La raison à cela est que la règle d'inclusion, 1HB ou MCT, priorise l'inclusion de l'agent qui révèle la meilleure offre, alors que dans le même temps la règle de redistribution FR garantit à un agent qui réalise un surplus de paiement de se le voir entièrement redistribué. Par exemple, si un agent révèle un PMA égal à 50 et qu'il remporte le bien, le mécanisme lui fait payer 50, puis lui rembourse l'entièreté de l'excès de contribution (égale à 49). Au final, l'agent obtient le bien pour 1 unité de numéraire. En fait, quel que soit le $x \geq 1$ révélé par l'agent, soit l'agent ne remporte pas le bien et il paie 0, soit l'agent remporte le bien avec la garanti de payer 1. Un qui reconnaît l'opportunité d'influencer le mécanisme sans en payer le coût, qui choisi de l'exploiter et qui dispose d'un apport en numéraire suffisamment important peut donc acquérir tous les biens en révélant un PMA le plus élevé possible (ou au moins plus élevé que son adversaire).

Ce résultat est doublement intéressant. Premièrement, il est très similaire au comportement de free-riders, à la différence près que l'agent ne cherche pas à révéler un PMA aussi faible que possible pour sous-payer un bien, mais il cherche à révéler un PMA aussi élevé que possible pour sur-prioriser son inclusion. Deuxièmement, la règle FR a été initialement prévue pour garantir a un agent de ne jamais sur-payer un bien, et ainsi éliminer tout intérêt de sous-évaluer le PMA révélé, rendant de ce fait un comportement rationnel compatible avec un comportement sincère. Au contraire ici, la garantie qu'a un agent de ne jamais sur-payer un bien lui procure un intérêt à sur-évaluer le PMA révélé, rendant de ce fait un comportement rationnel incompatible avec un comportement sincère.

Chapitre 5

Travaux associés

La littérature qui concerne la conception de mécanismes pour la provision de biens est gigantesque. Toutefois, les travaux font souvent intervenir de nombreuses hypothèses différentes, ce qui rend difficile la comparaison des résultats des différents travaux. Par exemple, le problème de provision peut concerner :

1. un unique bien ou de multiples biens,
2. une manche unique, plusieurs manches, une infinité de manches,
3. un bien peut être à faible ou forte soustractibilité, à faible ou forte excluabilité,
4. la classe de mécanisme étudiée peut être restreinte à certaines combinaisons de propriétés. Par exemple, la classe de mécanisme considérée peut être limitée aux mécanismes à budget non-déficitaire ou à budget équilibré, aux mécanismes strategyproof (compatible avec la révélation de préférences véridiques ou non) ou coalition strategyproof (compatible avec la révélation de préférences véridiques pour des agents pouvant se regrouper en coalition),
5. la valeur que les agents donnent à un bien peut être une information privée à chaque agent ou non,
6. les fonctions de préférences, de coût, etc. peuvent se limiter à des classes de fonction différentes (dérivables, monotones, convexes, etc.),
7. la sortie du mécanisme peut être déterministe ou non,
8. etc.

Et cette liste n'est pas exhaustive... Au final, être capable de prévoir si les résultats obtenus dans un travail donné sont pertinents lorsqu'une ou plusieurs des hypothèses diffèrent est un problème majeur. Un exemple révélateur est celui du mécanisme "Valeur de Shapley (pour bien non-rival excluable)", aussi dénommé Equal Cost Sharing with Maximal Participation (ECSMP). Ce mécanisme est étudié dans [Dobzinski et al., 2008], et l'un des résultats de ce papier est que ce mécanisme est le second-meilleur maximiseur de bien-être social dans la classe des mécanismes qui sont strategyproof, budget-équilibré, et qui satisfont un axiome d'égalité de traitement. Plus tard, Masso démontre dans [Massó et al., 2015] que ce mécanisme reste optimal dans le cas de deux agents, lorsque l'axiome d'égalité de traitement est abandonné et que la propriété de budget-équilibré est relâchée en la propriété de budget-non-déficitaire ; mais le mécanisme n'est pas optimal avec strictement plus de deux agents, sauf à ajouter la propriété de *faible monotonie de la demande* ou la propriété de *faible envie*. Un autre exemple révélateur peut être trouvé dans [Shichijo and Fukuda, 2016], où la classe de mécanismes étudiés se base sur 8 définitions distinctes (Voluntary Participation, No Positive Transfer, Symmetry, Strategy Proof, Group Strategy Proof, Non-bossy, Outside Independent, Weak Demand Monotonicity), qui s'ajoutent aux autres caractéristiques du problème étudié.

Choisir les travaux à évoquer ici en fonction de la proximité avec nos travaux est difficile en raison de la difficulté à en estimer, justement, la proximité. Nous choisissons donc d'évoquer les travaux qui 1) nous ont fortement inspirés dans notre travail, et 2) les travaux qui utilisent une méthodologie similaire à la notre.

5.1 Travaux qui nous ont fortement inspirés

Nos travaux ont fortement été inspirés par les papiers suivants :

— Dans [Cornelli, 1996] est étudié la vente d'un bien par un monopoliste à des agents dont les préférences pour le bien sont inconnues et où le coût de fabrication du bien est fixe. Dans certaines circonstances, on observe que des consommateurs sont prêts à payer différents prix pour la même quantité et la même qualité de bien. Francesca Cornelli explique que ce comportement n'est pas nécessairement le résultat de la générosité des consommateurs, et qu'il peut s'agir d'un comportement individuellement rationnel où le consommateur cherche à maximiser son utilité individuelle. Le cadre qui est considéré est celui où l'incertitude sur le fait que la contribution totale puisse couvrir le coût de fabrication du bien est grande (= coût de fabrication du bien élevé, nombre d'agents faible, incertitude élevée sur les préférences des agents). Dans ce cadre, il est pertinent de vendre le bien avant de le fabriquer, c'est-à-dire que le bien n'est pas fabriqué avant d'avoir la certitude que la contribution totale des agents couvre le coût de fabrication du bien. Deux raisons sont données pour expliquer que les consommateurs sont prêts à payer différents prix pour la même quantité et la même qualité de bien. La première raison est que, lorsque le bien est vendu avant d'être fabriqué, les consommateurs qui ont une valuation élevée du bien sont prêts à contribuer davantage par peur que le bien ne soit pas fabriqué du tout. La deuxième raison est qu'un agent qui contribue beaucoup ne veut pas exclure un autre agent qui contribue peu, dans la mesure ou l'exclusion de l'agent réduit la contribution totale, ce qui augmente le risque que le bien ne soit pas fabriqué du tout. Le papier étudie plusieurs mécanismes, où les agents paient que le bien soit fabriqué ou non, et où les agents peuvent être exclus à la consommation du bien. La conclusion suggère qu'il peut être optimal de vendre le bien avant de le fabriquer, et laisser libre les agents de contribuer autant qu'ils le souhaitent, i.e. aucun agent n'est exclus à la consommation du bien (règle d'inclusion NE dans notre papier). Il est enfin argumenté que ce mécanisme est d'autant plus approprié lorsque le coût de fabrication s'apparente à un flux, i.e. lorsque le coût de fabrication doit être payé répétitivement. Cette analyse semble justifier la pertinence et la bonne performance des mécanismes basés sur la règle d'inclusion NE (= aucun agent n'est exclus) dans notre contexte, malgré les différences suivantes entre les problèmes étudiés :

  1. dans nos travaux, l'objectif des agents est de maximiser l'utilité générée par le bien consommé, indépendamment du numéraire cumulé (vs maximiser la somme utilité générée + numéraire cumulé).
  2. dans nos travaux, la provision concerne de multiples biens (vs un unique bien) et la capacité de fabrication est limitée.
  3. dans nos travaux, la provision est étudiée sur plusieurs manches successives (vs une unique manche).

— Dans [Tabarrok, 1998], l'auteur explique que les problèmes de provision de biens publics peuvent être décomposés en deux sous-problème, 1) un problème de contribution, et 2) un problème de révélation. Le problème de contribution peut être résolu selon deux composantes : 1) la possibilité d'exclure un agent, qui permet d'éliminer l'intérêt que peut avoir un agent à ne pas contribuer lorsque les autres contribuent (attitude de free-riding), et 2) la garantie de remboursement de la contribution d'un agent, qui permet d'éliminer le risque de perdre sa contribution lorsque les autres ne contribuent pas. Le mécanisme proposé dans le papier, le contrat d'assurance dominant, est étudié sur une unique manche. Le comportement du mécanisme sur une provision étendue à plusieurs manches, comme dans notre contexte, est discuté mais nécessite

des travaux supplémentaires. De plus, nous cherchons à concevoir un mécanisme pour l'allocation de multiples biens, avec de potentielles contraintes sur les quantités de biens qui peuvent être fabriqués. Il nous paraît très complexe de prévoir comment le comportement du mécanisme se généraliserait dans ce nouveau contexte. Enfin, dans notre cas, l'objectif des agents diffère.

— Dans [Norman, 2004], les propriétés d'un mécanisme à seuil minimal fixe sont étudiées dans le contexte d'une grande économie. Comme dans [Cornelli, 1996], un bien n'est fabriqué que si la somme des contributions dépasse un seuil. La différence est qu'ici un agent peut consommer un bien fabriqué si et seulement si sa contribution est supérieure ou égale à un seuil minimal fixé. Norman met l'accent sur le fait que les résultats sont sensibles à la façon dont le coût de fabrication évolue avec la croissance de l'économie. Tandis que dans [Cornelli, 1996] l'étude se focalise sur un coût de fabrication élevé et un faible nombre d'agents, les travaux dans [Norman, 2004] se focalisent sur le cas où le nombre d'agents est très grand, mais où le coût requis pour fabriquer le bien évolue avec le nombre d'agent de façon à ce que ce coût reste significatif par agent. Dans ce cadre, le mécanisme à seuil minimal est strategyproof (révéler sa préférence véridique est une stratégie -faiblement- dominante) et approxime la solution optimale. Sur la base de ces résultats, nous avons cherché à imaginer un mécanisme dont le seuil minimal serait le plus petit et équitable pour tous les agents, ce qui nous a alors amené à concevoir la règle d'inclusion MCT et considérer un mécanisme MCT-NR. Par la suite, nous nous sommes aperçu que notre mécanisme MCT-FR s'apparente au mécanisme ECSMP, déjà existant de la littérature.

— Dans [Marks and Croson, 1998], différentes règles de redistributions sont étudiées dans le cadre de la provision d'un unique bien public à seuil, autrement dit un bien public non-rival et non-excluable à la consommation qui n'est fabriqué que lorsque la contribution totale dépasse un certain seuil. Les règles de redistributions spécifient comment les excès de contributions, au-dessus du seuil qui conditionne la fabrication du bien public, sont redistribués. Trois règles de redistributions sont considérées :

> *"a no rebate policy where excess contributions are discarded, a proportional rebate policy where excess contributions are rebated proportionally to an individual's contribution, and a utilization rebate policy where excess contributions provide some continuous public good."*
>
> - Extrait de [Marks and Croson, 1998], abstract.,

.

Ces travaux nous ont fait prendre conscience que le mécanisme ECSMP, qui différait du mécanisme que nous avions initialement imaginé par la seule règle de paiement, pouvait en fait être considéré comme un mécanisme identique au notre, mais avec une règle de redistribution différente. En particulier, ce sont ces travaux qui nous conduit à étendre le modèle de [Norman, 2004, Fang and Norman, 2010] avec l'ajout d'une règle de redistribution.

— Dans [Rothkopf, 2007] sont donnés 13 arguments pour lesquels le mécanisme Vickrey-Clarke-Groves, attrayant en théorie, n'est pas intéressant en pratique. Ce papier est intéressant dans le sens où il met en lumière les écarts constatés entre les modèles utilisés pour permettre des analyses théoriques, et la réalité qui est toujours plus complexe.

1. Un des arguments que nous trouvons particulièrement significatif est l'argument du faible équilibre de la stratégie dominante, et par la même la pertinence de la propriété strategyproof d'un mécanisme. En effet, une stratégie dominante n'est pas strictement dominante. La propriété strategyproof implique une "compatibilité avec une révélation véridique", mais elle n'impose pas une "incompatibilité avec une révélation non-véridique". L'exemple donné

dans [Rothkopf, 2007] pour concerne le mécanisme VCG et montre que, pour le cas considéré, l'agent perdant peut influencer sans conséquence pour lui-même l'utilité générée par l'autre agent négativement en ayant un comportement non-véridique.

2. Un autre argument est la tricherie par "faux-nom", qui consiste à soumettre des enchères via de fausses identités. A priori, lorsque les agents agissent dans l'intérêt d'un même groupe, il n'y a pas vraiment de raison à s'attendre ou à se prémunir contre ce type de comportement. Cependant, un mécanisme basé sur la règle d'inclusion MCT est sensible à ce type de comportement, puisque le seuil imposé par cette règle d'inclusion nécessite de considérer le nombre d'agents inclus, ce qui nécessite donc de connaître l'identité de chaque agent. dépend donc de "l'identité" de chaque agent. Au contraire, la règle d'inclusion NE est immunisé à ce type de comportement par nature, du fait qu'elle considère l'ensemble d'agents agrégé plutôt que chaque agent individuellement.

5.2 Travaux qui utilisent une méthodologie similaire à la nôtre

La méthode utilisée dans notre travail consiste à simuler les agents et intéractions d'agents dans un modèle décisionnel séquentiel localement constructif, c'est-à-dire dont chaque décision peut/doit tenir compte du passé et du futur, dans l'objectif de d'esquisser/théoriser le comportement émergent (au niveau individuel ou global). Cette méthode s'identifie à la partie des méthodes numériques appliquées à l'économie et dénommée Agents-based Computational Economics (ACE) [Tesfatsion, 2002], et plus généralement à la modélisation basée-agent.

— Dans [Bandyopadhyay et al., 2008], de multiples agents sont entraînés avec une technique d'apprentissage par renforcement. Le comportement d'agents dans un mécanisme d'enchères est modélisé par des agents artificiels joués dans un environnement de simulation. Les agents commencent avec une compréhension minimale de l'environnement, et analysent leurs gains et pertes dans le temps pour déterminer leurs futures offres. L'objectif de ces travaux est de répondre à la question ouverte suivante : est-ce que les offres réalisées par les agents convergent dans le temps vers l'équilibre théorique ? Les résultats obtenus dans ces travaux montrent que les offres réalisées convergent vers l'équilibre théorique dans un cas avec 2 agents, et les résultats obtenus dans un cas plus général avec n-agents permettent de renforcer la compréhension sur l'équilibre théorique.

> *"There results are promising enough to further consider the use of artificial learning mechanisms in reverse auctions and other electronic market transactions, especially as more sophisticated mechanisms are developed to tackle real-life complexities."*
>
> - Extrait de [Bandyopadhyay et al., 2008], abstract.,

— Dans [Tesauro and Kephart, 2002], l'applicabilité de la technique d'apprentissage par renforcement à agent unique Q-learning dans un contexte multi-agent, plus précisément avec deux agents. En effet, Q-learning garantit la convergence vers une stratégie optimale dans un environnement stationnaire, mais la convergence n'est pas garantit lorsque l'environnement est non-stationnaire, ce qui est le cas lorsque la stratégie de l'opposant n'est pas figée. Trois modèles économiques modérément réalistes sont considérés. Il est montré dans cette étude qu'une convergence simultanée vers des solutions optimales auto-cohérentes (self-consistent) est obtenu dans tous les modèles pour des faibles valeurs du coefficient de dévaluation (= faible considération des récompenses sur le "long-terme"). En revanche, ce n'est pas toujours le cas lorsque le coefficient de dévaluation est grand. Ces résultats indiquent qu'un algorithme d'apprentissage

par renforcement standard à agent unique tel que Q-learning peut, dans certains cas, échouer à converger dans un contexte multi-agent. Dans nos travaux, pour robustifier la convergence de l'apprentissage, nous avons utilisé l'algorithme IMPALA [Espeholt et al., 2018], spécifiquement conçu pour un contexte multi-agent.

Conclusion

Dans le cadre d'un problème de provision de biens de club sur plusieurs manches, nous avons analysé un ensemble générique de 8 mécanismes de provision qui sont basés sur des concepts rencontrés dans des mécanismes existants dans la littérature. La technique d'apprentissage par renforcement est utilisée pour apprendre à chaque service un comportement efficace qui maximise la richesse que le service génère, individuellement. Ce travail a été permis grâce à l'approche Agents-based Computational Economics, sans laquelle l'étude des mécanismes de provision aurait été laborieuse voire impossible, ce qui montre l'intérêt de cette approche.

Dans une première analyse, nous avons cherché à analyser la sensibilité des effets obtenus avec deux agents dotés d'une capacité d'influence identique mais dont les besoins pour deux biens de clubs sont plus ou moins compatibles/incompatibles. Dans une deuxième analyse, nous avons cherché à analyser la sensibilité des effets obtenus avec deux agents dont les besoins pour deux biens de clubs sont totalement incompatibles, l'apport en numéraire est plus ou moins équilibré/déséquilibre entre les deux agents, et le taux de charge (de la fabrication) varie de 100% à 320%.

Nos résultats montrent que lorsque l'objectif des agents est uniquement de maximiser la richesse générée individuellement, le problème de free-riders n'apparaît pas malgré l'absence de règle d'inclusion (ou d'exclusion). En effet, le problème de free-riders n'apparaît dans aucun des mécanismes que nous étudions. Ceci s'explique par le fait que les agents n'ont aucun intérêt à cumuler indéfiniment le numéraire. Le problème de free-riders est probablement révélateur d'un problème de modélisation plus profond, qui est qu'un agent ignore totalement le fait que la richesse sociale à un effet sur sa propre richesse. Cet argument est déjà avancé en 1976 par Becker, qui explique dans [Becker, 1976] comment la rationalité de groupe (notion d'altruisme) peut émerger d'une rationalité individuelle dans un modèle économique plus puissant, qui prend en compte de nouveaux effets.

> *"Economists, on the other hand, have relied solely on individual rationality and have not incorporated the effects of genetic selection."*
>
> - Extrait de [Becker, 1976], page 818.,

> *"I believe a more powerful analysis can be developed by joining the individual rationality of the economist to the group rationality of the sociobiologist. [...] I will show that models of group selection are unnecessary, since altruistic behavior can be selected as a consequence of individual rationality."*
>
> - Extrait de [Becker, 1976], page 818.,

Concernant les 4 critères d'évaluation définis dans la Section 4.4, nous faisons les observations qui suivent.

Critères de richesse sociale et richesse individuelle. Pour les deux critères, nos résultats montrent qu'un mécanisme basé sur la règle de redistribution NR est moins efficace qu'un mécanisme basé sur les règles WR et FR, ce qui s'explique par la perte des surplus de paiements. Un mécanisme basé sur la règle d'inclusion 1HB est moins efficace qu'un mécanisme basé sur les règles d'inclusions MCT et NE à la fois du point de vue de la richesse sociale et de la richesse individuelle générée. Ceci s'explique par l'exclusion infondée d'agents qui pourraient bénéficier des biens fabriqués, sans coût supplémentaire pour le mécansme. Les mécanismes basés sur la règle d'inclusion MCT ou NE ne sont

pas comparables du point de vue de la richesse sociale générée, parce que les mécanismes basés sur la règle MCT peuvent surperformer la règle NE pour certains cas, alors que l'inverse est aussi vrai pour d'autres cas. En effet, la règle MCT permet par le seuil de contribution minimal qu'elle impose de forcer la coopération entre les agents dans certains cas, ce qui permet d'extraire une richesse sociale générée plus importante. Mais cela se fait au détriment de la richesse individuelle générée par l'agent forcé de coopérer, un effet qui n'est pas souhaitable pour nous. Dans d'autres cas, la règle MCT ne permet pas de forcer la coopération entre les agents, ce qui engendre une perte de richesse sociale générée en raison de l'exclusion de l'agent qui refuse de coopérer malgré la contrainte. En conclusion, un mécanisme basé sur la règle NE semble préférable pour nous.

Critère d'élégance. Pour la règle d'inclusion, la règle NE est clairement la plus simple, et donc la plus élégante. Une classification pour la règle de redistribution n'est pas possible puisque la performance des mécanismes n'est pas égale selon les autres critères. Par exemple, la règle NR est plus simple que la règle WR, mais génère une richesse sociale et individuelle plus faible.

Critère du rendement d'influence. La performance des mécanismes étudiés se résume à la règle de redistribution, et nous pouvons classer les règles comme NR = WR > FR.

Synthèse. Compte tenu de l'ensemble des critères que nous considérons, le mécanisme NE-WR sort du lot. Ce mécanisme est simple, génère une richesse sociale élevée, et favorise la génération de richesse individuelle comme nous le désirons. De plus, ce mécanisme montre un rendement d'influence presque idéal sur des épisodes de 10 manches, ce qui laisse présager d'un rendement d'influence idéal sur des épisodes plus longs. Le mécanisme NE-WR semble donc être un bon candidat pour résoudre notre problème. Nous ne voyons a priori aucune raison à ce que tous les effets désirables, obtenus ici dans une configuration limitée à 2 agents et 2 biens binaires, ne se transposent pas dans une configuration plus réaliste.

Quatrième partie

Conclusion générale et perspectives

Chapitre 1

Conclusion générale

Dans ces travaux de thèse, nous nous sommes efforcés à proposer une solution au problème d'équilibrage de charge efficace et adaptatif du flux de données des véhicules vers le Cloud, en tenant compte des propriétés temporelles des informations qui sont liées aux données transmises. Plutôt que de chercher à résoudre ce problème de bout-en-bout, nous avons opté pour une décomposition multi-échelle temporelle sur 4 niveaux hiérarchiques, une décomposition classiquement utilisée en théorie de recherche opérationnelle.

Au niveau des véhicules, nous proposons l'utilisation d'un algorithme d'ordonnancement à temps-réel souple, embarqué dans chaque véhicule. Dans cette solution, une fonction d'utilité est associée à chaque évènement généré par le véhicule. Cette fonction spécifie la valeur générée pour le système global en fonction de la date de réception de l'évènement dans le Cloud. Le modèle utilisé est peu coûteux en mémoire et en calculs, et est donc adapté pour un système embarqué. De nombreux algorithmes existent déjà dans la littérature, et nous avons donc choisi un sous-ensemble pertinent d'algorithmes à évaluer. Il s'agit plus précisément d'algorithmes gloutons et myopes, basés sur différentes heuristiques relativement simples. Par nature, les décisions prises par ces algorithmes concernent un horizon futur très court, ce qui permet une très bonne réactivité aux perturbations extérieures, comme par exemple une variation de la bande passante, l'apparition imprévue d'évènements, etc. Une évaluation comparative rigoureuse de la performance des algorithmes a été réalisée, dans laquelle nous avons pris de nombreuses précautions pour limiter l'introduction de biais. Pour sélectionner le meilleur algorithme, il faut s'assurer que les scénarios utilisés dans l'évaluation sont représentatifs de la réalité. Cependant, des scénarios représentatifs d'une réalité future, même à court terme, ne sont pas disponibles aujourd'hui. Pour palier à ce problème, nous avons utilisé une méthode de génération de scénarios aléatoire utilisée dans [Aldarmi and Burns, 1999], méthode que nous avons proposé d'étendre pour générer une plus grande diversité de scénarios. Bien sûr, l'augmentation de la diversité des cas étudiés ne permet pas d'éliminer entièrement le risque du biais de sélection, où nous aurions pu faire des choix arbitraires pour obtenir des résultats favorables a une opinion forgée a priori. Cela ne permet pas non plus de garantir la représentativité des scénarios étudiés. Néanmoins, cela permet d'améliorer la transparence de nos résultats, en facilitant la recherche de biais par une évaluation plus fine des résultats. Nos travaux indiquent qu'un des algorithmes est un très bon candidat et permettrait de maximiser la valeur cumulée de façon très satisfaisante, et que la recherche d'un algorithme plus performant n'a que peu d'intérêt.

Au niveau du Cloud, nous proposons une solution de contrôle basé-marché, où un Mécanisme d'Allocation Centralisé situé dans le Cloud a pour objectif de déterminer périodiquement la quantité optimale de chaque type de données que la flotte de véhicules doit remonter vers le Cloud, en fonction d'un budget alloué à un niveau supérieur et sur une période plus grande. Le contrôle basé-marché est en effet adapté à ce problème de contrôle, puisque ce paradigme est adapté aux systèmes complexes dont le contrôle serait impossible ou très difficile à réaliser autrement. De plus, cela permet de garantir facilement que les coûts opérationnels ne sont pas supérieurs au budget prévu, et aussi de trouver un

compromis entre optimisation globale de la richesse sociale générée au moindre coût par l'ensemble des services eHorizon et la satisfaction des besoins individuels à chaque service, qui est priorisé pour éviter une situation de famine. Chaque service dispose d'un pouvoir d'influence, matérialisé par du numéraire fourni périodiquement, qui peut être utilisé ou cumulé librement par les services. Cela permet à un service d'adapter dynamiquement son pouvoir d'influence, en cumulant par exemple du numéraire lorsque les besoins sont faibles, pour disposer d'un pouvoir d'influence ponctuellement supérieur lorsque les besoins sont plus importants. Nos travaux montrent essentiellement que le problème de free-riders n'apparaît pour aucun des mécanismes de provision considérés, probablement en raison du fait que dans notre cas, l'objectif des services consiste seulement à maximiser la richesse générée individuellement. De ce fait, les services n'ont aucun intérêt à adopter un comportement de free-rider et à cumuler du numéraire indéfiniment au détriment de la richesse qu'ils génèrent. Lorsqu'un service n'a pas d'intérêt à cumuler le numéraire indéfiniment, l'exclusion à la consommation de données n'est plus un problème et devient donc plus difficile à justifier.

Chapitre 2

Perspectives

Nos travaux réalisés dans la Partie II, qui concernent l'évaluation de la performance d'algorithmes en-ligne à temps-réel souple, sont aisément reproductibles et il est donc facile d'y apporter de nouvelles contributions. Une piste pourrait être par exemple de rechercher un modèle explicatif plus complexe de la performance des algorithmes en fonction des caractéristiques des scénarios. Une autre piste pourrait être de considérer une configuration avec un coût de préemption fixe, coût qu'il est très facile d'intégrer dans l'heuristique des différents algorithmes, et de vérifier que les bonnes performances des algorithmes se conservent dans cette nouvelle configuration.

Pour les travaux réalisés dans la Partie III, qui concernent l'étude générique de mécanismes de provision de multiples biens de clubs non-binaires, de nombreuses perspectives sont possibles :

- dans un objectif de validation/invalidation/consolidation des résultats obtenus, il est possible de rechercher des hyperparamètres plus pertinents que ceux que nous avons utilisés, d'utiliser d'autres d'algorithmes d'apprentissages, de substituer les agents simulés par des humains pour comparer les effets obtenus, etc.
- dans un objectif d'extension des travaux réalisés, il est possible
  - de reproduire les résultats obtenus pour réaliser une analyse plus fine de la stratégie utilisée par les agents,
  - d'envisager des études dans des configurations ou les agents peuvent se regrouper en coalition,
  - d'analyser le comportement d'agents dans un processus de participation volontaire constitué de 2 étapes successives, similairement à [Saijo and Yamato, 1999], dans lequel les agents choisiraient dans une première étape le mécanisme avec lequel ils souhaitent participer, et révèleraient simultanément au mécanisme leur PMA dans une seconde étape. On pourrait imaginer que le mécanisme serait déterminé aléatoirement en cas de désaccord, ou d'une manière plus réfléchie.
  - de confronter les agents fictifs entraînés avec des humains afin de comparer la qualité de la stratégie obtenue par les agents simulés avec celle d'un humain,
  - d'évaluer de nouveaux mécanismes, en introduisant de nouvelles règles par exemple,
  - etc.

Une perspective plus intéressante pourrait certainement concerner une analyse de sensibilité du pouvoir d'influence d'un agent à un déséquilibre de la capacité "cognitive" d'un agent. Par exemple, on pourrait faire varier le nombre de couches ou de neurones d'un agent simulé, pour voir dans quelle mesure un agent "moins intelligent" serait pénalisé. On pourrait alors imaginer un nouveau critère selon lequel, dans un mécanisme désirable, le pouvoir d'influence d'un agent peu intelligent face à son adversaire serait relativement bien conservé, pour répondre par exemple à un besoin éthique. On pourrait aussi imaginer le contraire, où un agent peu intelligent face à son adversaire serait fortement pénalisé, pour répondre par exemple à un besoin évolutionnaire dans lequel la pression sélective doit être forte.

Notre travail peut également susciter de nouvelles discussions sur l'intérêt d'exclure certains biens non-soustractibles à la consommation fabriqués dans notre société, comme par exemple les propriétés intellectuelles. En effet, Michele Boldrin et David K. Levine questionnent déjà dans [Boldrin et al., 2008] l'intérêt des brevets et copyrights :

> *"'"Intellectual monopoly" hinders the competitive free market and the authors agree that patents and copyrights as they exist today should be eliminated."*
>
> - Extrait de [Boldrin et al., 2008], abstract.,

Cela s'aligne avec nos résultats, qui montrent que l'exclusion n'est pas nécessaire ou justifiée pour maximiser la richesse sociale. Cependant, nous avons considéré un modèle très simplifié qui n'est pas forcément suffisamment représentatif de notre réalité économique. Des travaux supplémentaires sont donc nécessaires pour vérifier si la réduction des contrôles de la propriété intellectuelle serait une bonne chose pour notre société.

Enfin, dans notre travail nous avons laissé de côté le problème de contrôle au niveau L1, qui doit faire le lien entre le niveau L1 et le niveau L3. Il faut à ce niveau trouver une solution qui permette d'agir sur les véhicules pour que la quantité de chaque type de données qu'il est prévu de remonter au cours d'une période soit effectivement remonté. Cela suppose, bien évidemment, que la planification réalisée au niveau L3 est faisable. Il faudra donc un système qui permette de prédire au mieux la quantité de chaque type de données que peut remonter la flotte de véhicules dans chaque période, ce qui se traduira par des contraintes arbitraires que prendra alors naturellement en compte le mécanisme de provision que nous avons défini.

Bibliographie

Adwan Alanazi and Khaled Elleithy. Real-time qos routing protocols in wireless multimedia sensor networks : Study and analysis. *Sensors*, 15(9) :22209–22233, 2015. (Cité en page 19.)

Saud Ahmed Aldarmi and Alan Burns. Dynamic value-density for scheduling real-time systems. In *Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on*, pages 270–277. IEEE, 1999. (Cité en pages ix, 25, 26, 27, 29, 30, 33, 34, 37 et 98.)

Ramiro Alvarez and Mehrdad Nojoumian. Comprehensive survey on privacy-preserving protocols for sealed-bid auctions. *Computers & Security*, 88 :101502, 2020. (Cité en page 55.)

MA Ameen, Ahsanun Nessa, and Kyung Sup Kwak. Qos issues with focus on wireless body area networks. In *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on*, volume 1, pages 801–807. IEEE, 2008. (Cité en page 19.)

Hal R Arkes and Catherine Blumer. The psychology of sunk cost. *Organizational behavior and human decision processes*, 35(1) :124–140, 1985. (Cité en pages 34 et 43.)

Hagit Attiya, Leah Epstein, Hadas Shachnai, and Tami Tamir. Transactional contention management as a non-clairvoyant scheduling problem. *Algorithmica*, 57(1) :44–61, 2010. (Cité en page 17.)

Subhajyoti Bandyopadhyay, Jackie Rees, and John M Barron. Reverse auctions with multiple reinforcement learning agents. *Decision Sciences*, 39(1) :33–63, 2008. (Cité en page 93.)

S. Baruah, G. Koren, B. Mishra, A. Raghunathan, L. Rosier, and D. Shasha. On-line scheduling in the presence of overload. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 100–110, 1991. doi : 10.1109/SFCS.1991.185354. (Cité en page 23.)

Sanjoy Baruah, Gilad Koren, Decao Mao, Bhubaneswar Mishra, Arvind Raghunathan, Louis Rosier, Dennis Shasha, and Fuxing Wang. On the competitiveness of on-line real-time task scheduling. *Real-Time Systems*, 4(2) :125–144, 1992. (Cité en pages 17, 22 et 36.)

Sanjoy Baruah, Vincenzo Bonifaci, Gianlorenzo DAngelo, Haohan Li, Alberto Marchetti-Spaccamela, Suzanne Van Der Ster, and Leen Stougie. The preemptive uniprocessor scheduling of mixed-criticality implicit-deadline sporadic task systems. In *2012 24th Euromicro Conference on Real-Time Systems*, pages 145–154. IEEE, 2012. (Cité en page 37.)

Gary S Becker. Altruism, egoism, and genetic fitness : Economics and sociobiology. *Journal of economic Literature*, 14(3) :817–826, 1976. (Cité en page 95.)

Michele Boldrin, David K Levine, et al. Against intellectual monopoly. 2008. (Cité en page 101.)

Allan Borodin, Nathan Linial, and Michael E Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM (JACM)*, 39(4) :745–763, 1992. (Cité en page 36.)

Alan Burns, Divya Prasad, Andrea Bondavalli, Felicita Di Giandomenico, Krithi Ramamritham, John Stankovic, and Lorenzo Strigini. The meaning and role of value in scheduling flexible real-time systems1. *Journal of systems architecture*, 46(4) :305–325, 2000. (Cité en page 23.)

Giorgio Buttazzo, Marco Spuri, and Fabrizio Sensini. Value vs. deadline scheduling in overload conditions. In *Real-Time Systems Symposium, 1995. Proceedings., 16th IEEE*, pages 90–99. IEEE, 1995. (Cité en pages 22, 23, 30 et 37.)

Giorgio C Buttazzo. *Hard real-time computing systems : predictable scheduling algorithms and applications*, volume 24. Springer Science & Business Media, 2011a. (Cité en pages 21 et 22.)

Giorgio C Buttazzo. Hard real-time computing systems : Predictable scheduling algorithms and applications (real-time systems series). 2011b. (Cité en pages 20 et 36.)

Ruggiero Cavallo. Efficiency and redistribution in dynamic mechanism design. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 220–229, 2008. (Cité en page 62.)

Dazhi Chen and Pramod K Varshney. Qos support in wireless sensor networks : A survey. In *International conference on wireless networks*, volume 233, pages 1–7, 2004. (Cité en page 19.)

Jian-Jia Chen. Scheduling aperiodic jobs on uniprocessor systems. [Powerpoint slides]. Retrieved from https://ls12-www.cs.tu-dortmund.de/daes/media/documents/teaching/courses/rts/aperiodic.pdf, April 2016. (Cité en page 24.)

Edward H Clarke. Multipart pricing of public goods. *Public choice*, pages 17–33, 1971. (Cité en pages 46 et 51.)

Scott H Clearwater and James J Yeh. *Market-based control : A paradigm for distributed resource allocation*. World Scientific, 1996. (Cité en pages 41 et 43.)

Francesca Cornelli. Optimal selling procedures with fixed costs. *Journal of Economic Theory*, 71(1) : 1–30, 1996. (Cité en pages 91 et 92.)

Avinash Dixit and Mancur Olson. Does voluntary participation undermine the coase theorem ? *Journal of Public Economics*, 76(3) :309–335, 2000. (Cité en page 51.)

Stan Dmitriev. Autonomous cars will generate more than 300 tb of data per year. https://www.tuxera.com/blog/autonomous-cars-300-tb-of-data-per-year/, 2017. [Online ; accessed 02-August-2021]. (Cité en page 4.)

Shahar Dobzinski and Mukund Sundararajan. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 38–47, 2008. (Cité en page 56.)

Shahar Dobzinski, Aranyak Mehta, Tim Roughgarden, and Mukund Sundararajan. Is shapley cost sharing optimal ? In *International Symposium on Algorithmic Game Theory*, pages 327–336. Springer, 2008. (Cité en page 90.)

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala : Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018. (Cité en pages 72 et 94.)

Hanming Fang and Peter Norman. Optimal provision of multiple excludable public goods. *American Economic Journal : Microeconomics*, 2(4) :1–37, 2010. (Cité en pages 53, 56 et 92.)

Amos Fiat, Richard M Karp, Michael Luby, Lyle A McGeoch, Daniel D Sleator, and Neal E Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4) :685–699, 1991. (Cité en page 36.)

Janet Fleetwood. Public health, ethics, and autonomous vehicles. *American journal of public health*, 107(4) :532–537, 2017. (Cité en page 2.)

Qingqing Gan and Torsten Suel. Improved techniques for result caching in web search engines. In *Proceedings of the 18th international conference on World wide web*, pages 431–440. ACM, 2009. (Cité en page 17.)

Michael R Garey and David S Johnson. Computers and intractability : A guide to the theory of npcompleteness (series of books in the mathematical sciences), ed. *Computers and Intractability*, 340, 1979. (Cité en page 23.)

Louis-André Gérard-Varet. incitatifs au développement de procédures expérimentales de révélation des préférences. 1998. (Cité en page 49.)

Sujit P Gujar and Yadati Narahari. Redistribution mechanisms for assignment of heterogeneous objects. *Journal of Artificial Intelligence Research*, 41 :131–154, 2011. (Cité en page 62.)

Mingyu Guo and Vincent Conitzer. Strategy-proof allocation of multiple items between two agents without payments or priors. In *AAMAS*, pages 881–888. Citeseer, 2010. (Cité en page 51.)

Carlos Herrera. *Cadre générique de planification logistique dans un contexte de décisions centralisées et distribuées*. PhD thesis, Université Henri Poincaré-Nancy 1, 2011. (Cité en page 9.)

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) : 1735–1780, 1997. (Cité en page 73.)

Jean Ibarz, Michaël Lauer, Matthieu Roy, Jean-Charles Fabre, and Olivier Flébus. Optimizing vehicle-to-cloud data transfers using soft real-time scheduling concepts. In *Proceedings of the 28th International Conference on Real-Time Networks and Systems*, RTNS 2020, page 161–171, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375931. doi : 10.1145/3394810.3394818. URL https://doi.org/10.1145/3394810.3394818. (Cité en page 11.)

Muhammad Awais Javed, Elyes Ben Hamida, and Wassim Znaidi. Security in intelligent transport systems for smart cities : From theory to practice. *Sensors*, 16(6), 2016. ISSN 1424-8220. doi : 10.3390/s16060879. URL https://www.mdpi.com/1424-8220/16/6/879. (Cité en pages ix et 6.)

E Douglas Jensen, C Douglas Locke, and Hideyuki Tokuda. A time-driven scheduling model for real-time operating systems. In *RTSS*, volume 85, pages 112–122, 1985. (Cité en pages 21 et 36.)

Bala Kalyanasundaram and Kirk Pruhs. Speed is as powerful as clairvoyance. *Journal of the ACM (JACM)*, 47(4) :617–643, 2000. (Cité en page 36.)

Richard M Karp. On-line algorithms versus off-line algorithms : How much is it worth to know the future ? In *IFIP Congress (1)*, volume 12, pages 416–429, 1992. (Cité en page 36.)

Terence Kelly et al. Utility-directed allocation. In *First Workshop on Algorithms and Architectures for Self-Managing Systems*, volume 20, 2003. (Cité en page 79.)

Gilad Koren and Dennis Shasha. D^over : An optimal on-line scheduling algorithm for overloaded uniprocessor real-time systems. *SIAM Journal on Computing*, 24(2) :318–339, 1995. (Cité en page 36.)

U Kramer and G Reichart. Automation and safety systems engineering aspects of selected prometheus functions. In *Second Prometheus Workshop, Stockholm*, 1989. (Cité en page 2.)

Tak-Wah Lam, Tsuen-Wan Johnny Ngan, and Kar-Keung To. Performance guarantee for edf under overload. *Journal of Algorithms*, 52(2) :193–206, 2004. (Cité en page 36.)

Zhi Li, Christopher M Anderson, and Stephen K Swallow. Uniform price mechanisms for threshold public goods provision with complete information : An experimental investigation. *Journal of Public Economics*, 144 :14–26, 2016. (Cité en page 65.)

C Douglass Locke. Best-effort decision making for real-time scheduling. *(Ph. D Thesis), Computer Science Department, CMU*, 1986. (Cité en page 22.)

Melanie Marks and Rachel Croson. Alternative rebate rules in the provision of a threshold public good : An experimental investigation. *Journal of public Economics*, 67(2) :195–220, 1998. (Cité en pages 53 et 92.)

Jordi Massó, Antonio Nicolo, Arunava Sen, Tridib Sharma, and Levent Ülkü. On cost sharing in the provision of a binary and excludable public good. *Journal of economic theory*, 155 :30–49, 2015. (Cité en pages 55, 56 et 90.)

R Preston McAfee and John McMillan. Auctions with entry. *Economics Letters*, 23(4) :343–347, 1987. (Cité en pages 64 et 75.)

Paul Milgrom and John Roberts. Économie, organisation et management, August 2016. (Cité en page 49.)

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016. (Cité en page 72.)

Amir Mosavi, Pedram Ghamisi, Yaser Faghan, and Puhong Duan. Comprehensive review of deep reinforcement learning methods and applications in economics. *arXiv preprint arXiv :2004.01509*, 2020. (Cité en page 66.)

Hervé Moulin. Serial cost-sharing of excludable public goods. *The Review of Economic Studies*, 61 (2) :305–325, 1994. (Cité en pages 46, 49 et 56.)

Herve Moulin and Scott Shenker. Strategyproof sharing of submodular access costs : Budget balance versus efficiency. *Available at SSRN 42940*, 1996. (Cité en page 56.)

Peter Norman. Efficient mechanisms for public goods with use exclusions. *The Review of Economic Studies*, 71(4) :1163–1188, 2004. (Cité en pages 42, 53, 56 et 92.)

Vincent Ostrom, Elinor Ostrom, and ES Savas. Public goods and public choices. *1977*, pages 7–49, 1977. (Cité en pages ix, 47 et 48.)

ML Patkin, GN Rogachev, and NG Rogachev. Neural net based multi-agent mobile robots control system : Practical implementation on different platform. In *IOP Conference Series : Materials Science and Engineering*, 2019. (Cité en pages ix et 67.)

Christine Plumejeaud. *Modèles et méthodes pour l'information spatio-temporelle évolutive*. PhD thesis, Université de Grenoble, 2011. (Cité en page 8.)

Prabhakar Raghavan and Marc Snir. Memory versus randomization in on-line algorithms. In *International Colloquium on Automata, Languages, and Programming*, pages 687–703. Springer, 1989. (Cité en page 36.)

Ievgen Redko and Charlotte Laclau. On fair cost sharing games in machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4790–4797, 2019. (Cité en page 66.)

G Reichart. Driving future vehicles. chapter 37 : Human and technical reliability. *Publication of : TAYLOR AND FRANCIS LTD*, 1993. (Cité en pages 2 et 3.)

Libby Rittenberg. *Principles of microeconomics*. Flat World Knowledge, 2008. (Cité en page 49.)

Michael H Rothkopf. Thirteen reasons why the vickrey-clarke-groves process is not practical. *Operations Research*, 55(2) :191–197, 2007. (Cité en pages 92 et 93.)

Tatsuyoshi Saijo and Takehiko Yamato. A voluntary participation game with a non-excludable public good. *Journal of Economic Theory*, 84(2) :227–242, 1999. (Cité en page 100.)

Ahtia Salo and Martin Weber. Ambiguity aversion in first-price sealed-bid auctions. *Journal of Risk and Uncertainty*, 11(2) :123–137, 1995. (Cité en page 66.)

Paul A Samuelson. Aspects of public expenditure theories. *the Review of Economics and Statistics*, pages 332–338, 1958. (Cité en page 48.)

Bruce Schneier. Invited talk : The coming ai hackers. In *International Symposium on Cyber Security Cryptography and Machine Learning*, pages 336–360. Springer, 2021. (Cité en page 79.)

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv :1707.06347*, 2017. (Cité en page 72.)

Margot Sève. *La régulation financière face à la crise*. Primento, 2013. (Cité en page 49.)

Tatsuhiro Shichijo and Emiko Fukuda. Equal cost sharing for the good with network externalities. *Discussion paper new series*, 2016(5) :1–7, 2016. (Cité en pages 65, 75 et 90.)

Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015. (Cité en page 2.)

Daniel D Sleator and Robert E Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2) :202–208, 1985. (Cité en page 22.)

SPGlobal. S&p500 information technology | s&p down jones indices. https ://www.spglobal.com/spdji/en/indices/equity/sp-500-information-technology-sector/#data, September 2021. Accessed : 2021-09-01. (Cité en page 48.)

John A Stankovic, Marco Spuri, Marco Di Natale, and Giorgio C Buttazzo. Implications of classical scheduling results for real-time systems. *Computer*, 28(6) :16–25, 1995. (Cité en page 22.)

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer, 1999. (Cité en page 72.)

Alexander Tabarrok. The private provision of public goods via dominant assurance contracts. *Public Choice*, 96(3-4) :345–362, 1998. (Cité en page 91.)

Gerald Tesauro and Jeffrey O Kephart. Pricing in agent economies using multi-agent q-learning. *Autonomous agents and multi-agent systems*, 5(3) :289–304, 2002. (Cité en page 93.)

Leigh Tesfatsion. Agent-based computational economics : Growing economies from the bottom up. *Artificial life*, 8(1) :55–82, 2002. (Cité en page 93.)

ValuePenguin. Information technology sector : Overview and funds. https ://www.valuepenguin.com/sectors/information-technology#software, September 2021. Accessed : 2021-09-01. (Cité en page 48.)

PJ Th Venhovens, JH Bernasch, JP Löwenau, HG Rieker, and M Schraut. The application of advanced vehicle navigation in bmw driver assistance systems. *SAE transactions*, pages 936–945, 1999. (Cité en page 3.)

Carl A Waldspurger, Tad Hogg, Bernardo A Huberman, Jeffrey O Kephart, and W Scott Stornetta. Spawn : A distributed computational economy. *IEEE Transactions on Software Engineering*, 18 (2) :103–117, 1992. (Cité en page 42.)

Qing Wang, Jiechao Xiong, Lei Han, Peng Sun, Han Liu, and Tong Zhang. Exponentially weighted imitation learning for batched historical data. In *NeurIPS*, pages 6291–6300, 2018. (Cité en page 72.)

Michael P Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of artificial intelligence research*, 1 :1–23, 1993. (Cité en page 42.)

Michael P Wellman. Market-oriented programming : Some early lessons. *Market-based control : a paradigm for distributed resource allocation*, pages 74–95, 1996. (Cité en page 42.)

Wuthichai Wongthatsanekorn, Matthew J Realff, and Jane C Ammons. Multi-time scale markov decision process approach to strategic network growth of reverse supply chains. *Omega*, 38(1-2) : 20–32, 2010. (Cité en page 9.)

Mohamed Younis, Kemal Akkaya, and Moustafa Youssef. Handling qos traffic in wireless sensor networks. In *Encyclopedia on Ad Hoc and Ubiquitous Computing : Theory and Design of Wireless Ad Hoc, Sensor, and Mesh Networks*, pages 257–279. World Scientific, 2010. (Cité en page 19.)

Robertas Zubrickas. The provision point mechanism with refund bonuses. *Journal of Public Economics*, 120 :231–234, 2014. (Cité en page 75.)

Annexe : validation de l'écart de performance entre DVD1 et DVD2

Nous voulons vérifier si la différence de performance observée entre les algorithmes DVD1 et DVD2 est statistiquement significative. Comme montré dans la Figure 31, clairement aucune des valeurs HVR obtenues par tous les algorithmes ne suit la distribution d'une loi normale. Par conséquent, les méthodes ANOVA ou t-student ne sont pas directement applicables.

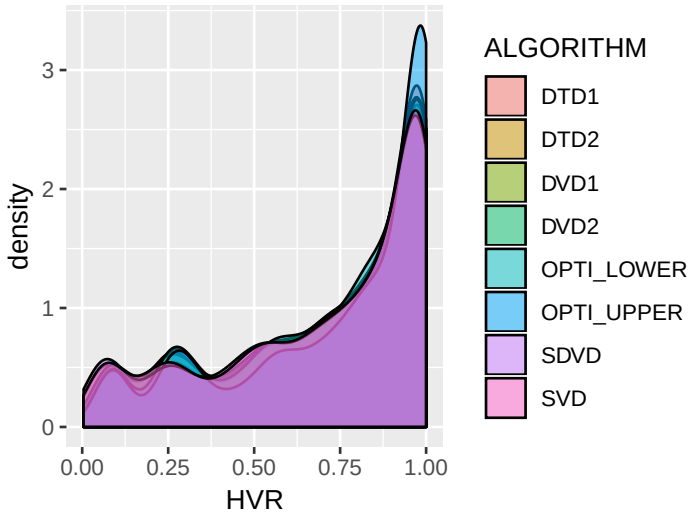


Figure 31 – *Distribution des valeurs HVR de tous les algorithmes pour tous les 3000 scénarios simulés.*

Une alternative à ce problème est de calculer la différence de performance entre les algorithmes 2-à-2, et faire une analyse de ces différences. Malheureusement, la différence de HVR entre les algorithmes DVD1 et DVD2, affichée Figure 32, ne suit pas la distribution d'une loi normale non plus.

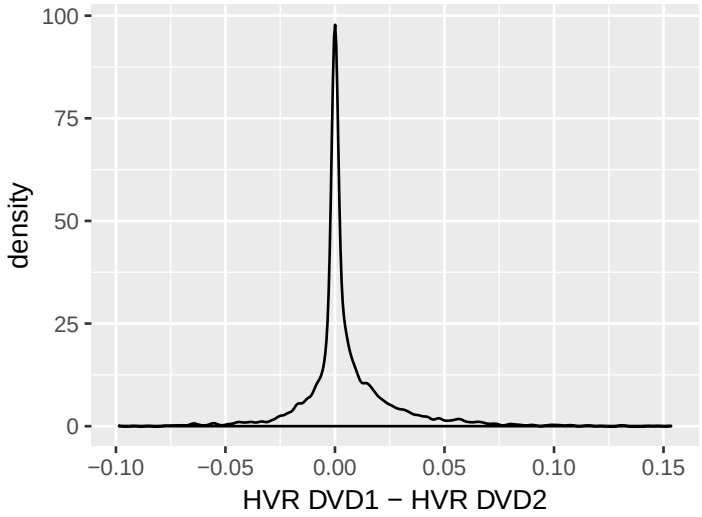


Figure 32 – *Distribution de la différence HVR DVD1 - HVR DVD2 pour tous les 3000 scénarios simulés.*

Finalement, nous choisissons d'appliquer la méthode bootstrap. Dénotons X, Y deux vecteurs colonne tel que X_i, Y_i est le HVR obtenu respectivement par l'algorithme DVD1 et l'algorithme DVD2 pour le même scénario i. Puisque l'algorithme DVD1 sur-performe l'algorithme DVD2 invariablement

du taux de charge, nous considérons les résultats obtenus sur tous les 3000 scénarios générés. Nous calculons la différence de performance entre l'algorithme DVD1 et l'algorithme DVD2 $D = X - Y$.

Maintenant faisons l'hypothèse *H0* qui suit : les algorithmes DVD1 et DVD2 n'ont pas une différence de performance statistiquement significative. Si *H0* est valide, nous ne devrions pas être capable de distinguer les résultats obtenus d'un algorithme ou de l'autre.

Nous calculons la différence moyenne obtenue en échangeant aléatoirement les valeurs HVR des algorithmes $\sigma_i = D \circ R$, où R est un vecteur aléatoire dont les coefficients sont échantillonnés aléatoirement dans $\{-1, 1\}$ et $(\circ)$ est le produit par élément. Le processus est répété pour $i \in 1, 2, .., N, N = 100000$.

La distribution résultante des valeurs obtenues $\{\sigma_1, \sigma_2, ..., \sigma_N\}$, montré Figure 33, suit une loi de distribution normale. C'est une très bonne approximation de la distribution que l'on pourrait espérer obtenir en reproduisant les résultats si l'hypothèse *H0* est valide : l'intervalle de confiance à 95% est dans $I = [-2.57 \times 10^{-6}; 1.60 \times 10^{-6}]$. Dans nos résultats, la différence de performance moyenne vaut $mean(X1 - X2) = 5.0 \times 10^{-3}$ et cette valeur n'appartient pas à I, donc l'hypothèse *H0* n'est pas valide et nous pouvons conclure que nos résultats sont statistiquement significatifs.

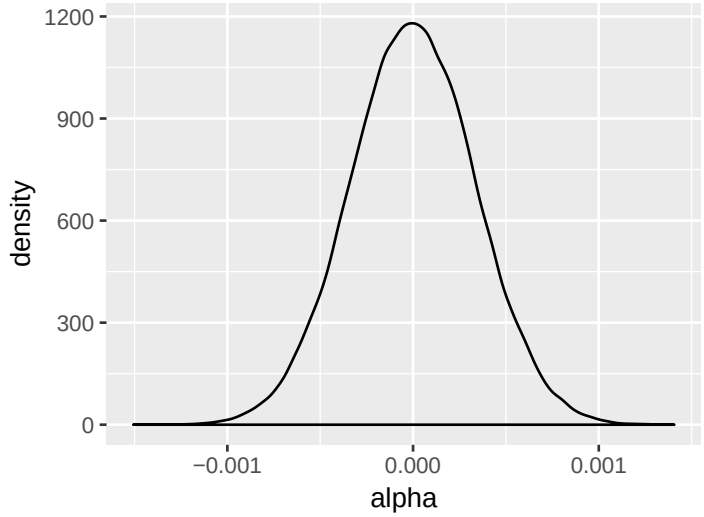


Figure 33 – *Distribution des valeurs échantillonnées* $\{\alpha_1, \alpha_2, ..., \alpha_N\}$ *avec* $N = 100000$.

RÉSUMÉ : Pour améliorer l'expérience de la mobilité, une piste consiste à éliminer les manques ou déficiences d'informations et de capacités et possibilités de communication, qui seraient à l'origine de 90% de tous les accidents de trafic. À cette fin, il est envisagé que des millions de véhicules connectés agissent en tant que mineurs d'informations dans un système distribué massif. Dans ce système, chaque véhicule est équipé de plusieurs capteurs pour capter les informations locales de l'environnement. Les informations capturées par l'ensemble des véhicules sont transférées vers le Cloud, où elles sont exploitées par des services pour générer une connaissance globale et fréquemment mise à jour de l'environnement. La connaissance globale de l'environnement ainsi générée rend possible une meilleure anticipation des situations futures, dans un horizon électronique qui s'étend au-delà de la perception des capteurs embarqués.

L'importante quantité de données générées par la flotte de véhicules ainsi que le caractère fortement dynamique de l'environnement impose une optimisation efficace et adaptative du flux de données transférées des véhicules vers le Cloud. Dans cette thèse, nous proposons des éléments de solution à ce problème.

Nous proposons d'abord un modèle multi-échelle-temporelle du problème avec 4 niveaux hiérarchiques, un type de modèle usuellement rencontré en recherche opérationnelle. Les différentes échelles de granularité temporelle considérées sur les différents niveaux permettent une réactivité adaptée pour répondre avantageusement aux variations imprévues dans un horizon futur plus ou moins grand. Au niveau des véhicules, à l'échelle temporelle la plus fine, nous formulons le problème comme un problème d'ordonnancement basé-valeur en nous appuyant sur des concepts de temps-réel souple. Nous réalisons une évaluation expérimentale comparative d'un ensemble d'algorithmes gloutons en-ligne, judicieusement choisis pour leur forte capacité adaptative et leur faible complexité. Nous proposons d'étendre une méthode de génération aléatoire de scénarios, ce qui permet d'une part de générer un ensemble de scénarios plus variés pour une meilleure couverture de l'évaluation, et d'autre part d'obtenir des résultats plus riches et donc potentiellement plus transparents. Nos résultats indiquent qu'un algorithme simple permet de résoudre efficacement le problème.

Au niveau du Cloud, à une échelle temporelle moins fine, il faut concilier les besoins des différents services, sachant que leurs besoins peuvent être difficiles à estimer, qu'il ne sera pas possible de tous les satisfaire, et que certains services sont plus importants que d'autres. Nous cherchons un moyen d'abstraire l'expression concrète du besoin, et de contrôler l'influence de chaque service sur le flux de données des véhicules vers le Cloud. Nous envisageons une solution de contrôle basé-marché pour résoudre ce problème. Le pouvoir d'influence est matérialisé par du numéraire, et distribué périodiquement aux services. Les services ont la liberté et la responsabilité d'utiliser au mieux le numéraire pour influencer, au travers d'intéractions avec un mécanisme de provision, le flux de données. Nous construisons une multitude de mécanismes basés sur des concepts, que nous évaluons grâce à l'utilisation de techniques d'apprentissage par renforcement. Nos résultats indiquent qu'un mécanisme simple et peu coercitif est un bon candidat pour résoudre efficacement notre problème.