

CONCEPTION D'UN SYSTÈME D'OPTIMISATION DES PLANS DE FEUX ROUTIERS

Projet Scientifique Collectif

6 janvier 2020

Clément BEAULIEU Antoine DOIZE Mathews LUCAS Charles LESCURE
Timothée PAQUATTE Skandere SAHLI

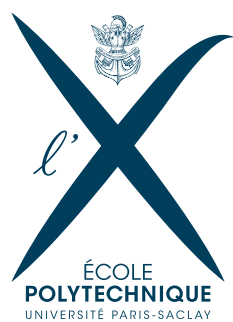


Table des matières

1	Introduction	4
1.1	Enjeux	4
1.2	Motivations et objectifs initiaux	4
1.3	Réévaluation des objectifs au cours de l'année	5
2	Algorithme génétique	6
2.1	Description générale et motivation	6
2.2	Structure classique d'un algorithme génétique	6
2.3	Description théorique des algorithmes génétiques	8
3	Recuit simulé	11
3.1	Présentation et sens physique	11
3.2	L'algorithme de Metropolis-Hastings	11
3.2.1	Principe	11
3.2.2	Avantages	12
3.2.3	Inconvénients	12
3.2.4	Un peu de théorie	13
3.3	Preuve de l'Algorithme de Recuit Simulé	13
4	Application de l'algorithme génétique et du recuit simulé	17
4.1	Présentation du problème	17
4.2	Comment fixer les paramètres : algorithme génétique	17
4.2.1	Probabilité de croisement	17
4.2.2	Probabilité de mutation	17
4.2.3	Taille de la population t	17
4.2.4	Nombre de générations G	18
4.3	Résultats finals : algorithme génétique	18
4.4	Ajustement des paramètres : méthode du recuit simulé	19
4.4.1	ajustement du paramètre $e^{-\Delta E/T(n)}$	20
4.4.2	ajustement du nombre d'itérations	21
4.4.3	Résultats : recuit simulé	22
4.5	Comparaison des algorithmes	23
5	La modélisation mathématique du trafic routier	25
5.1	Introduction	25
5.2	Les modèles microscopiques	25
5.2.1	Modèles à distance de sécurité	25
5.2.2	Limites du modèle	26
5.3	Les modèles macroscopiques	27
5.3.1	Le modèle LWR	27
5.3.2	Les modèles du second ordre	27
5.3.3	Le modèle cellulaire	28
5.3.4	Les modèles cinétiques	28
5.4	Conclusion	29
6	Conclusion	30
7	References Bibliographiques	31

1

Introduction

1.1 Enjeux

L'augmentation exacerbée de la densité routière lors de ces dernières décennies contraint les villes, et plus particulièrement les métropoles majeures, à intégrer cette problématique mondiale dans leur développement.

Les enjeux connexes sont multiples : économiques, financiers et écologiques. Outre le fait de permettre aux usagers de diminuer leur temps de trajet et d'économiser ainsi leur temps personnel et professionnel; l'optimisation des temps de trajet permet également de réduire les émissions atmosphériques des véhicules, aux conséquences environnementales et sanitaires urbaines bien connues. Les enjeux sont également économiques. Réduire les aléas des horaires d'arrivée d'un employé, dû aux perturbations routières, à son lieu de travail. La congestion d'un réseau routier engendre des dégradations conséquences des infrastructures, et de surcroît des surcoûts financiers qu'il convient de limiter.

Dans les villes, les feux tricolores sont les principaux moyens stratégiques de gestion du trafic puisqu'ils en permettent une gestion et une restriction aisée au niveau des intersections.

Durant les vingt dernières années, la recherche dans ce domaine a donné lieu à des avancées intéressantes, tant pour les séquences fixées à l'avance, que pour les systèmes de micro-régulation qui s'adaptent en temps réel au trafic grâce à des capteurs situés à proximité des intersections.

Ainsi, même si ces méthodes de contrôle peuvent bénéficier de l'expérience d'une large variété d'incidents routiers et de schémas de congestion de trafic, elles sont incapables de s'adapter à des situations inédites. De plus, elles ne permettent pas de coordonner les différentes intersections entre elles et de disposer d'une stratégie globale qui optimiserait les séquences des feux sur tout le réseau.

Le développement de théories informatiques et mathématiques novatrices (notamment les problèmes d'optimisation, intégrant le Machine Learning) a conduit à de nombreuses recherches sur ce sujet; le principal problème étant la dimension de l'espace des plans de feux de signalisation, gigantesque, rendant les calculs inextricables sur le plan computationnel.

Ainsi, le problème de la construction de modèles d'optimisation applicables à des réseaux routiers de taille conséquente reste actuellement largement théorique et partiellement résolue; les méthodes fonctionnant en temps réel aussi, à cause du manque d'information disponibles (acquises notamment grâce à des capteurs fixes ou des véhicules connectés).

1.2 Motivations et objectifs initiaux

Le thème de ce Projet Scientifique et Collectif a été proposé par une entreprise espagnole, nommée Aimsun (anciennement TSS), spécialisée dans le développement et la commercialisation d'un logiciel de simulation de trafic routier. Celui-ci ayant été adopté, depuis sa commercialisation, par plus de 4000 utilisateurs au sein d'agences gouvernementales et de centres de recherche, et notamment par la mairie de Paris.

Le thème proposé par Aimsun nous a semblé intéressant, parce qu'il revêt une dimension urbanistique majeure et mondialisée; et dont les outils informatiques et mathématiques qui permettent de l'appréhender sont éminemment d'actualité. Notamment, les méthodes d'apprentissage basées sur les réseaux de neurones ou les algorithmes génétiques ou encore le paradigme de métamodèles.

Développer nos connaissances dans ces domaines et renforcer notre compréhension des liens sciences appliquées/entreprise, cœur de notre cursus académique, nous semblait motivant. Comprendre ce lien via une entreprise leader de son secteur, qu'est Aimsun pour le secteur routier, nous semblait d'autant plus motivant. L'objectif initial de ce projet était de livrer à Aimsun un service d'optimisation de plans de feux routiers faisant intervenir son logiciel de simulation Aimsun Next.

1.3 Réévaluation des objectifs au cours de l'année

Une première difficulté était le choix de la licence qui nous permettrait de travailler sur le logiciel. Après échanges avec Ainmsun ainsi que la plupart des autorités de la direction des études, nous avons accepté de changer de type de licence : nous n'avions plus une licence normale comme les années précédentes mais une licence collective de type "classroom". Nous avons donc dû consacrer quelques semaines à la mise en place des infrastructures en collaboration avec le binet réseaux et la direction des systèmes d'information pour faire fonctionner cette licence. Cela a retardé la réalisation des objectifs initiaux.

Ensuite, après quelques manipulations, nous nous sommes rendu compte que l'interface héritée du groupe précédent ne fonctionnait pas avec cette licence. Nous avons donc dû remanipuler le code, avec l'aide de ce groupe, et après plusieurs semaines de travail il s'est avéré que la type de licence fourni n'avait pas les droits permettant de réaliser les manipulations nécessaires à la réalisation du projet. Nous avons donc averti Aimsun du problème et ils ont fait leur possible pour nous venir en aide. Cependant les autorisations ne dépendaient plus de leur pôle mais du service commercial, ce qui nous a empêché d'aller plus loin dans la réalisation de ces objectifs.

En parallèle de tous ces imprévus, nous avons tout de même consacré une équipe à la poursuite des recherches théoriques et à la conception d'algorithmes afin d'éviter de perdre du temps. Au vu de l'impasse dans laquelle nous nous sommes retrouvé, la plupart de ces recherches se sont malheureusement avérées inutiles.

Après concertation avec notre coordinateur PSC le 20 mars, nous avons suivi ses directives et tout en maintenant une fenêtre ouverte dans le cas où nous serions parvenu à faire fonctionner notre licence, nous avons décidé d'établir une étude plus théorique du problème.

Ainsi, nous avons formé trois pôles de recherche :

- Une étude théorique des algorithmes que nous devons appliquer au problème des feux (génétique et recuit simulé).
- Une étude algorithmique : l'implémentation des algorithmes génétique et du recuit simulé, pour la résolution du Vertex K-Center problem.
- Une présentation de l'état de l'art sur la modélisation du trafic routier.

2

Algorithme génétique

2.1 Description générale et motivation

Les algorithmes génétiques appartiennent plus largement à la classe des algorithmes évolutionnistes. Il s'agit d'algorithmes heuristiques dans le sens où ils se basent sur une heuristique pour approximer la solution d'un problème d'optimisation. Mais à la différence d'algorithmes d'approximation, il n'est pas nécessairement possible de borner l'approximation relativement à la solution optimale. Il peut exister des cas où l'approximation issue de l'heuristique est arbitrairement mauvaise.

Les algorithmes génétiques s'inspirent de la théorie de l'évolution de Darwin qui stipule que les espèces évoluent selon le principe de la sélection naturelle : les mieux adaptés au milieu survivent. Ce principe a inspiré les chercheurs, dont John Holland, pour développer une nouvelle classe d'algorithmes, basés sur l'utilisation d'opérateurs génétiques (sélection, croisement, mutation) afin d'optimiser les individus d'une population. Ces algorithmes ont été popularisés par David Goldberg dans son ouvrage *Algorithmes Génétiques* et destinés à résoudre des problèmes combinatoires ou d'optimisation que les algorithmes classiques peinaient à résoudre.

Les algorithmes génétiques font partie des algorithmes d'apprentissage non supervisé (*unsupervised learning*). Pour appliquer un algorithme génétique à un problème donné, il faut une manière de représenter chaque solution potentielle par un génome (tableau de bits ou de nombres) et une fonction d'évaluation de la qualité d'une solution (fonction de fitness). Des individus (solutions potentielles du problème) sont générés aléatoirement et forment une population. On fait ensuite évoluer la population de façon discrète, chaque étape étant appelée une génération. Pour passer à la génération suivante, les individus qui ont la meilleure fitness sont sélectionnés puis certains sont croisés deux par deux et certains de leur gènes sont mutés aléatoirement avec une certaine probabilité. En répétant ce processus de sélection, croisement et mutation, la fonction de fitness (adaptation des individus au problème) augmente, ce qui nous donne après un nombre de génération choisi à l'avance une bonne solution au problème. On peut par exemple les utiliser pour résoudre le problème du voyageur de commerce ou pour optimiser des emplois du temps.

2.2 Structure classique d'un algorithme génétique

Une des grandes forces des algorithmes génétiques est qu'ils ne supposent presque rien sur la fonction à optimiser. Contrairement aux méthodes d'optimisation classique, type descente de gradient, la régularité de la fonction n'est pas un paramètre à prendre en compte. Ainsi, ces algorithmes permettent de traiter des problèmes très complexe, la fonction à optimiser pouvant même être le résultat d'une simulation.

Nous prenons ici une fonction f de E^d dans \mathbb{R} .

Les Algorithmes Génétiques possèdent cinq étapes bien définies :

1. **Initialisation.** Une population initiale de N individus est tirée aléatoirement.
2. **Évaluation.** Chaque individu est évalué.
3. **Sélection.** Création d'une nouvelle population de N individus par l'utilisation d'une méthode de sélection appropriée.
4. **Reproduction.** Possibilité de croisement et mutation au sein de la nouvelle population.
5. **Retour** à la phase d'évaluation jusqu'à l'arrêt de l'algorithme.

Représentation des données. Il existe de nombreuses manières de représenter les données du problème, c'est-à-dire un individu de la population. La méthode choisie aura des conséquences sur la performance et la simplicité de l'algorithme. Historiquement, le codage binaire était majoritairement utilisé, car il permettait d'effectuer les opérations de mutations et de croisement simplement. Cependant, cette représentation n'est pas forcément adaptée à tous les problèmes, et présente certains défauts notamment en grande dimension. Actuellement, trois types de codage prédominent : le codage binaire, gray et réel.

Initialisation. L'initialisation est importante dans la mesure où elle conditionne fortement la vitesse de convergence de l'algorithme. Lorsque l'on ne dispose d'aucune information sur l'optimum, il n'y a pas d'autre choix que de tirer aléatoirement une population, respectant les éventuelles contraintes. Cependant, il est possible que des résultats théoriques sur le problème permettent de réduire l'espace de recherche, ce qui permet souvent d'améliorer la convergence.

Sélection. L'étape de sélection s'inspire du phénomène de la sélection naturelle observée par les biologistes. L'idée sous-jacente est de conserver seulement les « meilleurs » (au sens de notre fonction) individus, et d'éliminer les plus mauvais. Il existe de nombreuses méthodes de sélection, les plus connues étant : fitness proportionate selection ou loterie biaisée. Selon cette méthode, chaque individu aura une probabilité d'être dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation. Autrement dit, si i est l'individu considéré, la probabilité qu'il appartienne à la nouvelle population est $p_i = \frac{f_i}{\sum_{j=1}^N f_j}$. Cette méthode est particulièrement adaptée lorsque la population considérée est grande.

Elle peut cependant engendrer une perte de diversité par la domination d'un super individu. Un autre inconvénient est sa faible performance vers la fin quand l'ensemble des individus se ressemblent. La Stochastic remainder without replacement selection évite ce genre de problème et donne de bons résultats en faible dimensions. Ce principe de sélection fonctionne comme suit :

- Pour chaque élément i , on calcule le rapport p_i
- Soit $e(r_i)$ la partie entière de r_i , chaque élément est reproduit exactement $e(r_i)$ fois.
- La loterie biaisée précédemment décrite est appliquée sur les individus affectés des fitness $r_i - e(r_i)$, avec $e(r_i)$ la partie entière de r_i .

Reproduction. Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants, ce qui permet d'évoluer à taille de population constante. Les individus sont regroupés par paires (il y a donc $N/2$ paires), et pour chaque couple d'individu, un entier est tiré aléatoirement entre 0 et $d - 1$. Cet entier définit la coordonnée où le slicing va être opéré. Enfin, le croisement ne s'effectue pas à coup sûr, mais avec une probabilité p_c . Le croisement le plus simple, et le plus courant, consiste à coupler les deux individus parents, comme sur le schéma : on peut étendre ce principe en découpant le chromosome non pas en 2 sous-chaînes mais en n sous-chaînes.

Ce type de croisement à découpage de chromosomes est très efficace pour les problèmes discrets. Pour les problèmes continus, un croisement « barycentrique » est souvent utilisé. Les deux enfants sont déterminés comme combinaison linéaire des parents :

$$C_1(i) = \alpha P_1(i) + (1 - \alpha) P_2(i)$$

$$C_2(i) = (1 - \alpha) P_1(i) + \alpha P_2(i)$$

où α est un coefficient de pondération aléatoire adapté au problème.

La reproduction et la sélection seule ne suffisent pas à assurer la correction de l'algorithme. En effet, cela ne nous permet pas de distinguer un extremum local d'un extremum global. Pour ce faire, on introduit l'opération de mutation.

Mutation. Pour éviter de rester «bloqué» dans un extremum local, l'étape de mutation entraîne la modification aléatoire de la population. Elle permet donc d'assurer que tous les états possibles sont atteignables par notre population (autrement dit l'ergodicité). Ainsi en toute rigueur, l'algorithme génétique peut converger sans croisement. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de la mutation sur le plan théorique. L'opérateur de mutation consiste généralement à tirer aléatoirement une coordonnée et à la remplacer par une valeur aléatoire.

2.3 Description théorique des algorithmes génétiques.

On modélise l'évolution de la population au cours de l'algorithme génétique par une chaîne de Markov. On cherche à trouver les maxima globaux de f une fonction cible.

On considère N la taille de la population. On note $X^{(k)}$ la population à la génération k représentée par un vecteur $(X_1^{(k)}, \dots, X_N^{(k)})$ où les $X_i^{(k)}$ sont à valeur dans E^d avec E fini. On a le diagramme suivant :

$$X^{(k)} \xrightarrow{\text{mutation}} Y^{(k)} \xrightarrow{\text{croisement}} Z^{(k)} \xrightarrow{\text{sélection}} X^{(k+1)}$$

Mutation. L'opérateur mutation ici considéré est le suivant : pour chaque composante du vecteur $X_i^{(k)}$ on tire une variable de Bernoulli de paramètre p_m (à fixer) qui détermine si la composante est mutée ou non. La nouvelle valeur obtenue est le résultat du tirage d'une variable aléatoire M sur E (que l'on pourra prendre uniforme pour fixer les idées).

Croisement. On fixe p_c la probabilité de croisement. On construit $Z^{(k)}$ à partir de $Y^{(k)}$ comme suit : on forme $N/2$ couples de $Y^{(k)}$. On tire une variable de Bernoulli de paramètre p_c pour chaque couple pour déterminer si croisement il y a ; si tel est le cas, on tire alors un entier de coupe i dans $1, \dots, d-1$ et on permute les séquences finales des séquences des couples.

Sélection. On sélectionne alors les meilleurs individus parmi $Z^{(k)}$. La probabilité de sélection de l'individu $Z_i^{(k)}$ est

$$p_i = \mathbb{P}(Z_i^{(k)}) = \frac{f(Z_i^{(k)})}{\sum_j f(Z_j^{(k)})}$$

La chaîne de Markov $(X^{(k)})_{k \in \mathbb{N}}$ est homogène, irréductible, apériodique et d'espace d'états fini ; un résultat classique sur les chaînes de Markov nous permet alors de conclure à l'existence d'une unique mesure de probabilité π invariante. Par le théorème ergodique, on sait même que pour tout x_0 dans $(E^d)^N$,

$$\pi(x) = \lim_{k \rightarrow \infty} \mathbb{P}(X^{(k)} = x | X^{(0)} = x_0)$$

On associe à cette mesure de probabilité le processus de fond noté $(X_\infty^{(k)})_{k \in \mathbb{N}}$, ie tel que pour tout $k \in \mathbb{N}$,

$$\mathbb{P}(X_\infty^{(k)} = x) = \pi(x)$$

On introduit les notations suivantes

$$\hat{f}(x) = \hat{f}(x_1, \dots, x_n) = \max\{f(x_i), i \in \{1, \dots, n\}\}$$

$$\hat{x} = \{x_i \in \operatorname{argmax} f(x)\}$$

On appelle processus de fond noté $(X_\infty^{(k)})_{k \in \mathbb{N}}$ le processus défini comme le **processus limite lorsque les perturbations génétiques ont disparu**. Il s'agit d'une chaîne de Markov sur E^N dont le mécanisme de transition correspond à la situation limite suivante :

- Les N composantes de $X_\infty^{(k+1)}$ sont choisies indépendamment et suivant la loi uniforme sur l'ensemble $\widehat{X_\infty^{(k+1)}}$
- Les individus dont l'adaptation n'est pas maximale en k , sont éliminés et n'apparaissent pas dans la génération $k + 1$.
- Les individus dont l'adaptation est maximale, ont des chances de survies égales.

Cette chaîne est tout d'abord piégée dans l'ensemble S des populations ayant la même adaptation (ou ensemble des population d'équi-adaptation),

$$S = \{x = (x_1, \dots, x_N) \in (E^d)^N, f(x_1) = f(x_2) = \dots = f(x_N)\}$$

Cette population représente les attracteurs de la chaîne, puis elle est absorbée par une population uniforme, de sorte que :

$$\forall x \in (E^d)^N, \mathbb{P} \left(\exists x_i \in \widehat{x}, \exists K \in \mathbb{N}, \forall k \geq K, X_\infty^{(k)} = x_i | X_\infty^{(0)} = x_{ini} \right)$$

Lorsque la population est devenue uniforme et en l'absence ici de perturbations, celle-ci n'est plus modifiée. Ceci peut également se traduire en définissant les populations uniformes comme les états absorbants de la chaîne $X_\infty^{(k)}$. Nous allons maintenant étudier la situation où ce processus est perturbé.

On étudie à présent la situation où le processus est perturbé. La modélisation part du processus de fond $(X_\infty^{(k)})$, décrit ci-dessus, qui est perturbé, aléatoirement, les perturbations sont indicées par le paramètre l . La chaîne de Markov $(X_\infty^{(k)})$ devient donc une suite de chaînes de Markov $(X_l^{(k)})$, dont le mécanisme de transition est donné par la succession des transformations générées par les opérateurs.

$$X_l^{(k)} \xrightarrow{\text{mutation}} U_l^{(k)} \xrightarrow{\text{croisement}} V_l^{(k)} \xrightarrow{\text{sélection}} X^{(k+1)}$$

Mutation $X_l^{(k)} \longrightarrow U_l^{(k)}$

Les mutations sont définies comme des perturbations aléatoires indépendantes des individus de la population $X_l^{(k)}$. On introduit la probabilité de transition de mutation entre les points i et j de E^d , notée $p_l(i, j)$. Il s'agit donc de la probabilité de mutation de i à j . Elle vérifie :

$$\sum_{j \in E^d} p_l(i, j) = 1$$

La probabilité de transition entre les points x et u de $(E^d)^N$ est :

$$\mathbb{P} \left(U_l^{(k)} = u | X_l^{(k)} = x \right) = p_l(x_1, u_1) \cdot p_l(x_2, u_2) \cdot \dots \cdot p_l(x_N, u_N)$$

On suppose que l'on dispose d'une distribution de probabilité de transition α irréductible telle que l'on ait le développement limité de p_l suivant, où a est un paramètre entier positif :

$$\forall i, j \in E^d, p_l(i, j) = \begin{cases} \alpha(i, j) \cdot l^{-a} + o(l^{-a}) & \text{si } i \neq j \\ 1 - \alpha(i, j) \cdot l^{-a} + o(l^{-a}) & \text{si } i = j \end{cases}$$

Cette définition permet de considérer p_l comme irréductible. Lorsque l'on fait tendre l vers $+\infty$, les perturbations disparaissent.

Croisement $U_l^{(k)} \longrightarrow V_l^{(k)}$

On gère les perturbations de croisement par une probabilité de transition q_l sur $E^d \times E^d$. On suppose encore que l'on dispose d'une probabilité de transition β irréductible nous permettant de faire le développement limité suivant, où b paramètre entier positif :

$$\forall (i_1, j_1) \in E^d \times E^d, \forall (i_2, j_2) \in E^d \times E^d,$$

$$q_l((i_1, j_1), (i_2, j_2)) = \begin{cases} \beta((i_1, j_1), (i_2, j_2)) \cdot l^{-b} + o(l^{-b}) & \text{si } (i_1, j_1) \neq (i_2, j_2) \\ 1 - \beta((i_1, j_1), (i_2, j_2)) \cdot l^{-b} + o(l^{-b}) & \text{si } (i_1, j_1) = (i_2, j_2) \end{cases}$$

Lorsque l tend vers $+\infty$, le croisement disparaît.

Sélection $V_l^{(k)} \rightarrow X_l^{(k+1)}$

L'opérateur de sélection est plus complexe mais permet la convergence vers les optima de f . On le modélise à l'aide d'une fonction de sélection notée F_l telle que :

$$\begin{aligned} F_l : \{1, \dots, N\} \times (\mathbb{R}_+)^N &\longrightarrow [0, 1] \\ (i, f_1, f_2, \dots, f_N) &\longmapsto F_l(i, f_1, f_2, \dots, f_N) \end{aligned}$$

F_l vérifie les propriétés suivantes :

- $F(\cdot, f_1, f_2, \dots, f_N)$ est une probabilité sur $1, \dots, N$
- $F(\cdot, f_1, f_2, \dots, f_N)$ est indépendante de l'indexation des f_i (on peut les permuter)
- La probabilité favorise les éléments i associés à des valeurs f_i élevées : Si $f_1 \geq f_2 \geq \dots \geq f_N$, alors

$$F_l(1, f_1, f_2, \dots, f_N) \geq F_l(2, f_1, f_2, \dots, f_N) \geq \dots \geq F_l(N, f_1, f_2, \dots, f_N)$$

On suppose que la probabilité de sélection s'écrit sous la forme :

$$\mathbb{P}(X_l^{(k+1)} = x | V_l^{(k)} = v) = \prod_{i=1}^N \Gamma_l(x_i, v_i)$$

La probabilité Γ_l de transition de sélection entre deux composantes x_i et v_i est donnée par :

$$\Gamma_l(x_i, v_i) = \sum_{k, x_k = v_k} F_l(k, f(v_1), f(v_2), \dots, f(v_N))$$

Conformément aux propriétés précédentes, on choisit usuellement la fonction de sélection F_l suivante :

$$F_l(i, f_1, f_2, \dots, f_N) = \frac{\exp(c \cdot f_i \cdot \ln(l))}{\sum_{j=1}^N \exp(c \cdot f_j \cdot \ln(l))}$$

Ce choix permet une sélection tendanciellement forte des éléments de la population réalisant une « grande » valeur de f .

Pour tout $x \in (E^d)^N$, lorsque l'on fait tendre l vers $+\infty$ on obtient à la limite le phénomène de sélection pour le processus de fond $(X_\infty^{(k)})$. Soit,

$$F_\infty(k, f(x_1), \dots, f(x_N)) = \frac{\mathbf{1}_{\hat{x}}(x_k)}{\text{card}(\hat{x})}$$

Il s'agit de la loi uniforme sur \hat{x} .

En mettant bout à bout les résultats précédentes, on obtient, $\forall x, y \in (E^d)^N$,

$$\mathbb{P}(X_l^{(k+1)} = y | X_l^{(k)} = x) \xrightarrow{l \rightarrow +\infty} \mathbb{P}(X_\infty^{(k+1)} = y | X_\infty^{(k)} = x)$$

Le sens de perturbation de la chaîne $(X_l^{(k)})$ par rapport au processus de fond $(X_\infty^{(k)})$ prend ici tout son sens.

3

Recuit simulé

3.1 Présentation et sens physique

Le recuit simulé est une méthode d'optimisation, différente des algorithmes génétiques, mais qui possède aussi l'avantage de s'appliquer à des fonctions très peu régulières.

Analogie. La méthode du recuit simulé a été inspiré non pas par la biologie, mais cette fois-ci par la physique des matériaux. Plus précisément, en métallurgie le recuit d'un métal est un procédé correspondant à un cycle de chauffage, suivi d'un maintien en température puis d'un refroidissement contrôlé permettant de modifier les caractéristiques de ce métal. Initialement, les atomes du métal se trouvent dans une certaine structure cristalline. Le cycle de chauffage permet de libérer ces atomes de cette structure. Une fois libéré, les atomes se regroupent lors du refroidissement pour reformer une structure cristalline. Si le processus est fait de manière assez « lente » (à interpréter en fonction des propriétés du métal considéré), la nouvelle configuration est alors souvent plus stable que la configuration initiale.

Physiquement, une configuration plus stable correspond à une énergie globale inférieure. Si le refroidissement est trop brusque, le métal cristallise dans un état instable (donc d'énergie plus élevée). Ce processus, conduit correctement, tend donc à amener le métal vers un état atteignant le minimum de l'énergie globale. Un résultat phare de physique statistique est que le mécanisme naturel de minimisation de l'énergie repose sur la distribution de probabilité de Boltzmann $p(E) \propto \exp(-E/kT)$.

Cette équation traduit le fait qu'un système à l'équilibre thermique à la température T , présente une distribution de Boltzmann de ces états d'énergie E . À basse température, il est possible, bien que la probabilité soit infime, de trouver le système dans un état de grande énergie. Cependant, il est possible que une fois dans cet état de plus haute énergie, le système puisse rejoindre un état de moindre énergie ; l'objectif final étant de parvenir d'un état métastable correspondant à un minimum local d'énergie, à l'état stable de plus basse énergie. Ce faisant, le système « emprunte » ponctuellement des états de plus haute énergie, d'autant plus défavorisés que la température est faible, mais qui, à terme, lui permettront de rejoindre l'état d'énergie minimum. C'est cette propriété qui permet d'assurer que le minimum atteint est bien un minimum global, et non un simple minimum local.

La probabilité qu'un système thermodynamique passe d'un état d'énergie E_1 à un état d'énergie E_2 , est donc égale à $p = \exp(-(E_2 - E_1)/kT)$, si $E_1 < E_2$. Si $E_1 > E_2$, on a $p = 1$, et la transition vers l'état de plus faible énergie se fait presque sûrement. Cependant, on remarque bien qu'une transition vers un état de plus haute énergie est possible, quoique très peu probable à basse température.

3.2 L'algorithme de Metropolis-Hastings

3.2.1 • PRINCIPE

Cet algorithme repose totalement sur l'analogie décrite plus haut. Plus formellement :

- La fonction à minimiser f joue le rôle de l'énergie E .
- Une solution possible S correspond à un état physique.

Le but est, en partant d'un état initial (donc d'une possible solution S), d'atteindre la solution optimale. On chauffe ensuite notre système en initialisant une température $T > 0$ afin d'autoriser le parcours des différents états. Une boucle est ensuite initiée où à chaque itération, l'état précédent est perturbé. La fonction est évaluée avec cette nouvelle solution possible S' , qui est systématiquement acceptée si $f(S') \leq f(S)$. Cette condition impose au système de tendre vers un état correspondant à un minimum local, éventuellement global, d'énergie. Afin de s'extraire de possibles minima locaux, la solution S' peut être acceptée avec une probabilité égale à $\exp(-(f(S') - f(S))/kT)$ même si $f(S') > f(S)$. Initialement, il est préférable de choisir T grand, de telle sorte que la probabilité de Boltzmann est voisine de 1 indépendamment du changement d'énergie : la grande majorité des nouveaux états sont par conséquent acceptés. Ainsi, lors des premières itérations une vaste région de l'espace des solutions possibles est parcourue. Lorsque la température décroît, le système dérive progressivement vers les régions de moindre énergie. Dans les derniers stades de l'algorithme, seules les configurations telles que $\delta E < 0$ sont acceptées, car nous considérons avoir trouvé la bonne zone de recherche. La solution finale est obtenue lorsque le système est quasiment gelé, à savoir lorsque le critère de terminaison est atteint.

Algorithm 1 Template for a Metropolis-Hastings algorithm, based on the choice of the neighbourhoods, the initialization, the constants k and T , and the termination criteria

Require: An optimization problem with a solution set \mathcal{S} , a solution function f to be minimized and a neighborhood structure $N(S)$ with $S \in \mathcal{S}$.

Ensure: A hopefully good solution.

Initialize $S \in \mathcal{S}$ suitably.

repeat $N \neq 0$

 Choose $S' \in N(S)$ uniformly at random.

if $f(S') \leq f(S)$ **then**

$S := S'$

else

 with probability $e^{(f(S)-f(S'))/(kT)}$ set $S := S'$

end if

until termination criteria fulfilled

return S

3.2.2 • AVANTAGES

Cet algorithme d'optimisation présente des avantages indéniables. Tout d'abord, il est facile à implémenter et peut s'appliquer à une multitude de problèmes. En effet, il ne nécessite aucun prérequis sur la fonction à minimiser, et ne repose pas sur des propriétés qui seraient intrinsèque à un certain type de problème. D'autre part, il a été constaté expérimentalement que l'algorithme est très performant pour des problèmes à grande dimension (ce qui est le cas du problème de trafic routier que nous voulions initialement résoudre). Enfin, nous disposons d'un résultat théorique affirmant que l'algorithme converge presque sûrement vers un minimum global sous certaines hypothèses. si la température ne décroît pas trop rapidement (plus précisément si la décroissance est plus faible que $C/\log(i)$, avec i correspondant au nombre d'itérations, et C une constante dépendant de la fonction à minimiser).

3.2.3 • INCONVÉNIENTS

L'un des plus gros inconvénients de cette méthode est la difficulté de choisir correctement les paramètres. En effet, il y a un compromis entre vitesse d'exécution (intimement lié à la loi de décroissance de la température choisie), et la qualité de la solution trouvée. La valeur de la température initiale a aussi une grande influence sur la vitesse de convergence. Enfin, il faut déterminer un critère de terminaison adapté au problème considéré, sous peine de voir les performances de l'algorithme diminuer fortement.

3.2.4 • UN PEU DE THÉORIE

Contrairement à beaucoup d'algorithmes heuristiques, il est prouvé que l'algorithme de recuit simulé converge presque sûrement vers la solution optimale, sous certaines hypothèses. Nous allons dans cette partie esquisser une preuve de ce résultat, et mettre en lumière les conditions suffisantes à la convergence. A la manière de ce que nous avons fait pour les algorithmes génétiques, la modélisation théorique retenue sera probabiliste et utilisera des chaînes de Markov.

Soit $U : E \rightarrow \mathbb{R}$ une fonction à minimiser. Nous traiterons ici le cas E fini par soucis de simplification. On note U_* la valeur minimale de U (qui existe car E est fini) et E_* l'ensemble de ses minima. Comme suggéré par l'algorithme, nous considérons la loi de Gibbs sur E : $\pi_T(x) = (1/Z_T) \cdot \exp(-U(x)/T)$.

En notant μ_* la loi uniforme sur E_* , on a $\|\mu_T - \mu_*\| \rightarrow 0$ quand $T \rightarrow 0$. En effet, pour x et y dans E tels que $U(x) > U(y)$, on a $\mu_T(x)/\mu_T(y) \rightarrow 0$ quand $T \rightarrow 0$. Pour a dans E_* , on a le résultat en écrivant $\mu_T(a) = \sum_{E_*} \mu_T(x)/\mu_T(a) + \sum_{(E \setminus E_*)} \mu_T(x)/\mu_T(a)$.

Ainsi, l'algorithme du recuit simulé minimise U en simulant une chaîne de Markov inhomogène convergent vers la loi μ_* lorsque $T \rightarrow 0$. Dans la suite, nous noterons $\beta = 1/T$.

3.3 Preuve de l'Algorithme de Recuit Simulé

Nous allons maintenant effectuer une étude théorique qui nous permettra de donner des conditions permettant d'assurer la convergence de l'algorithme.

Notons (X_n) une chaîne de Markov inhomogène, ν_n la loi de X_n , Q_n les matrices de transition associées.

Soit P une matrice de transition irréductible sur E vérifiant la condition de Doeblin. On a donc $\alpha > 0$, l un entier naturel et c une mesure de probabilité sur E tel que :

$$\forall x, y \in E, P^l(x, y) \geq \alpha c(y)$$

Soit $(\beta_n)_{n \in \mathbb{N}}$ une suite de réels positifs. On suppose que pour tout x, y dans E et $n \in \mathbb{N}$ on a :

$$Q_n(x, y) = \exp(-(H(y) - H(x))^+ \cdot \beta_n) P(x, y)$$

Enfin, on note μ_n la mesure invariante associée à la matrice de transition Q_n (qui existe car E est fini). On cherche ici à donner des conditions permettant d'avoir $\mathbb{P}(X_n > H_*) \xrightarrow[n \rightarrow +\infty]{} 0$. Nous avons montré que $\mu_n \xrightarrow[n \rightarrow +\infty]{} \mu_*$. On a donc $\mu_n(E \setminus E_*) \xrightarrow[n \rightarrow +\infty]{} 0$. Ainsi, il est intéressant d'observer la quantité $|\mathbb{P}(H(X_n) > H_*) - \mu_n(E \setminus E_*)|$.

On a $|\mathbb{P}(H(X_n) > H_*) - \mu_n(E \setminus E_*)| = \left| \sum_{x \in E \setminus E_*} (\nu_n(x) - \mu_n(x)) \right| \leq 2 \cdot \|\nu_n - \mu_n\|$. On se ramène donc à étudier la suite $(\|\nu_n - \mu_n\|)_{n \in \mathbb{N}}$.

On a par une récurrence immédiate que pour n et m dans \mathbb{N} , $\nu_{n+m} = \nu_n Q_{n+1} \dots Q_{n+m}$.

Ainsi, $\nu_{n+m} - \mu_{n+m} = (\nu_n - \mu_n) Q_{n+1} \dots Q_{n+m} + \sum_{k=1}^m (\mu_{n+k-1} - \mu_{n+k}) Q_{n+k} \dots Q_{n+m}$ (car $\mu_{n+m} Q_{n+m} = \mu_{n+m}$).

Nous allons chercher à majorer $(\|\nu_n - \mu_n\|)$.

On a pour $x \neq y$ et $n \in \mathbb{N}$: $Q_n(x, y) = \exp(-(H(y) - H(x))^+ \cdot \beta_n) \cdot P(x, y) \geq e^{-\kappa \beta_n} \cdot P(x, y)$ avec $\kappa = \max_{x, y \in E} (H(y) - H(x))$.

D'autre part, $Q_n(x, y) \leq P(x, y)$. D'où $Q_n(x, x) = 1 - \sum_{y \neq x} Q_n(x, y) \geq P(x, x) \geq e^{-\kappa \beta_n} \cdot P(x, x)$.

D'où pour $(x, y) \in E^2$,

$$Q_{n+1} \dots Q_{n+l}(x, y) \geq \exp(-\kappa \sum_{k=1}^l \beta_{n+k}) P^l(x, y) \geq \exp(-\kappa \sum_{k=1}^l \beta_{n+k}) \alpha c(y)$$

car P vérifie la condition de Doeblin. Posons donc $n = jl$ et $m = l$. Nous reviendrons ensuite au cas général.

En notant $P' = Q_{n+1} \dots Q_{n+l}$ et $\alpha_j = \exp(-\kappa \sum_{k=1}^l \beta_{jl+k}) \cdot \alpha$ on a $P'(x, y) \geq \alpha_j \cdot c(y)$.

Ainsi,

$$\begin{aligned} \|(\nu_n - \mu_n)P'\| &= \frac{1}{2} \sum_{y \in E} \left| \sum_{x \in E} (\nu_n(x) - \mu_n(x)) P'(x, y) \right| \\ &= \frac{1}{2} \sum_{y \in E} \left| \sum_{x \in E} (\nu_n(x) - \mu_n(x)) (P'(x, y) - \alpha_j c(y)) \right| \\ &\leq \frac{1}{2} \sum_{y \in E} \sum_{x \in E} |\nu_n(x) - \mu_n(x)| (P'(x, y) - \alpha_j c(y)) \\ &= \frac{1}{2} \sum_{x \in E} |\nu_n(x) - \mu_n(x)| (1 - \alpha_j) \\ &= (1 - \alpha_j) \|\nu_n - \mu_n\| \end{aligned}$$

D'autre part, on a pour M matrice de transition sur E et pour deux mesures μ et μ' sur E :

$$\begin{aligned} \|(\mu - \mu')M\| &= \frac{1}{2} \sum_{y \in E} \left| \sum_{x \in E} (\mu(x) - \mu'(x)) M(x, y) \right| \\ &\leq \frac{1}{2} \sum_{x \in E} \sum_{y \in E} |\mu(x) - \mu'(x)| M(x, y) \\ &= \frac{1}{2} \sum_{x \in E} |\mu(x) - \mu'(x)| = \|\mu - \mu'\| \end{aligned}$$

Nous obtenons donc $\|\nu_{(j+1)l} - \mu_{(j+1)l}\| \leq (1 - \alpha_j) \|\nu_{jl} - \mu_{jl}\| + \sum_{k=1}^l \|\mu_{jl+k-1} - \mu_{jl+k}\|$

En posant $z_j = \|\nu_{jl} - \mu_{jl}\|$ et $b_j = \sum_{k=1}^l \|\mu_{jl+k-1} - \mu_{jl+k}\|$ on a :

$$z_{j+1} \leq (1 - \alpha_j) \cdot z_j + b_j$$

Ici, si $\sum_{n \geq 0} \alpha_n = +\infty$ et $\frac{\beta_n}{\alpha_n} \xrightarrow{n \rightarrow +\infty} 0$, alors $z_n \xrightarrow{n \rightarrow +\infty} 0$ (nous ne démontrerons pas ce lemme, qui résulte de l'étude de suites réelles).

Revenons au cas général, et prenons p dans $\{1, \dots, l-1\}$. On a

$$\|\nu_{jl+p} - \mu_{jl+p}\| \leq \|(\nu_{jl} - \mu_{jl})Q_{jl+1} \dots Q_{jl+p}\| + \sum_{k=1}^l \|\mu_{jl+k-1} - \mu_{jl+k}\| \leq \|\nu_{jl} - \mu_{jl}\| + b_j$$

Ainsi, si $b_n \xrightarrow{n \rightarrow +\infty} 0$, on a $\|\nu_n - \mu_n\| \xrightarrow{n \rightarrow +\infty} 0$

Exprimée de cette manière, la condition sur (b_n) est difficile à vérifier. Nous allons donc donner une majoration de b_n pour tout $n \in \mathbb{N}$ dépendant de H et de (β_n) . On note, $\Delta(H) = \max_{x \in E} H(x) - H_*$ et pour $T \geq T' \geq 0$:

$$\|\mu_T - \mu_{T'}\| \leq \left| \frac{1}{T} - \frac{1}{T'} \right| \Delta(H)$$

En effet,

$$\begin{aligned} \|\mu_T - \mu_{T'}\| &= \sum_{x \in E} \left| \frac{1}{Z_T} e^{-\frac{H(x)}{T}} - \frac{1}{Z_{T'}} e^{-\frac{H(x)}{T'}} \right| \\ &\leq \sum_{x \in E} \frac{1}{Z_T} \left| e^{-\frac{H(x)}{T}} - e^{-\frac{H(x)}{T'}} \right| + \left| \frac{1}{Z_T} - \frac{1}{Z_{T'}} \right| \sum_{x \in E} e^{-\frac{H(x)}{T'}} \\ &\leq \sum_{x \in E} \left(\frac{1}{T'} - \frac{1}{T} \right) \Delta(H) \frac{1}{Z_T} e^{-\frac{H(x)}{T}} + \left(\frac{1}{T'} - \frac{1}{T} \right) \Delta(H) \frac{1}{Z_{T'}} \sum_{x \in E} e^{-\frac{H(x)}{T'}} \\ &= 2 \left(\frac{1}{T'} - \frac{1}{T} \right) \Delta(H) \end{aligned}$$

Où l'on a utilisé

$$\left| e^{-\frac{H(x)}{T}} - e^{-\frac{H(x)}{T'}} \right| = e^{-\frac{H(x)}{T}} \left| 1 - e^{-\left(\frac{1}{T'} - \frac{1}{T}\right)H(x)} \right| \leq \left(\frac{1}{T'} - \frac{1}{T} \right) \Delta(H) e^{-\frac{H(x)}{T}}$$

Et,

$$\left| \frac{1}{Z_T} - \frac{1}{Z_{T'}} \right| \leq \left(\frac{1}{T'} - \frac{1}{T} \right) \frac{\Delta(H)}{Z_{T'}}$$

En appliquant cette inégalité à b_j on obtient,

$$b_j \leq \sum_{k=1}^l |\beta_{jl+k-1} - \beta_{jl+k}| \Delta(H)$$

Pour résumer, si la suite $(\beta_n)_{n \in \mathbb{N}}$ vérifie les propositions suivantes :

- i) $\sum_{j \geq 0} \exp \left(-\kappa \sum_{k=1}^l \beta_{jl+k} \right) = +\infty$
- ii) $\frac{\sum_{k=1}^l |\beta_{jl+k-1} - \beta_{jl+k}|}{\exp \left(-\kappa \sum_{k=1}^l \beta_{jl+k} \right)} \xrightarrow{j \rightarrow +\infty} 0$
- iii) $\sum_{k=1}^l |\beta_{jl+k-1} - \beta_{jl+k}| \xrightarrow{j \rightarrow +\infty} 0$

Alors $\mathbb{P}(H(X_n) > H_*) \xrightarrow{n \rightarrow +\infty} 0$.

En prenant $(\beta_n)_{n \in \mathbb{N}}$ croissante nous pouvons simplifier les conditions précédentes :

- i) $\sum_{j \geq 0} \exp(-\kappa l \beta_{(j+1)l}) = +\infty$
- ii) $\beta_{(j+1)l} - \beta_{jl} \xrightarrow{j \rightarrow +\infty} 0$
- iii) $\frac{\beta_{(j+1)l} - \beta_{jl}}{\exp(-\kappa l \beta_{(j+1)l})} \xrightarrow{j \rightarrow +\infty} 0$

On remarque donc que le bon ordre de grandeur de décroissance de la température est logarithmique, ie $T_m = \frac{C}{\log(m)}$ avec $C > 0$. Trouvons une condition sur C pour que i) et iii) soient vérifiés.

$$\exp(-\kappa l \beta_{jl}) \underset{j \rightarrow +\infty}{\sim} (jl)^{-(\kappa l)/C}$$

et

$$\frac{\beta_{(j+1)l} - \beta_{jl}}{\exp(-\kappa l \beta_{(j+1)l})} \underset{j \rightarrow +\infty}{\sim} l^{-\kappa l/C} j^{-1-\kappa l/C}$$

Ainsi en prenant $C > \kappa l$, i) et iii) sont vérifiées pour $C > 0$.

Un théorème dû à Hajek amoindrit les hypothèses :

On a $\lim_{m \rightarrow +\infty} \mathbb{P}(H(X_n) > H_*) = 0$ si, et seulement si, $\lim_{n \rightarrow +\infty} T_n = 0$ et $\sum_{m \geq 1} e^{-C/T_m} = +\infty$ pour un certain $C > 0$ dépendant de H .

Revenons un peu à l'algorithme. Une décroissance logarithmique de T est très mauvaise la complexité. En pratique, on pourra préférer une convergence rapide à défaut d'un résultat précis. Ensuite, il faut que la matrice P puisse vérifier la condition de Doeblin qui impose une contrainte sur le choix du prochain point à construire.

4

Application de l'algorithme génétique et du recuit simulé

4.1 Présentation du problème

Nous avons implémenté ces algorithmes qui minimisent une fonction de R^n dans R^+ . Nous avons besoin d'un problème concret sur lequel les appliquer afin de les tester et de le comparer avec d'autres algorithmes. Après quelques recherches nous avons décidé de les appliquer au *Vertex k -center problem*. Il s'agit d'un problème d'optimisation NP-difficile que l'on peut présenter comme suit :

Etant donné n villes représentées par leurs coordonnées dans le plan euclidien, on veut implanter dans ces villes k entreprises de manière à minimiser la distance de chaque ville à l'entreprise la plus proche.

La fonction à minimiser est donc $f : x \rightarrow \max_{villes\ v} \{ \min_{entreprises\ e} \{ dist(e, v) \} \}$. Nous nous sommes fixé les valeurs suivantes afin de définir clairement le problème : $n = 100$ et $k = 10$.

4.2 Comment fixer les paramètres : algorithme génétique

Avant de tester notre algorithme génétique, nous avons besoin de fixer ses paramètres, qui étaient pour l'instant réglés arbitrairement à la main. Nous avons donc cherché les valeurs de ces paramètres qui le rendaient le plus performant possible. Ces paramètres sont le nombre de générations, la taille de la population, la probabilité de croisement et la probabilité de mutation.

4.2.1 • PROBABILITÉ DE CROISEMENT

Nous avons écrit un programme de test qui testait 30 fois l'algorithme pour chaque valeur d'un paramètre sur une plage définie. En effet, effectuer 30 tests et ne garder que la moyenne des résultats permet d'atténuer le facteur aléatoire du problème (placement des villes) et de l'algorithme afin d'avoir des résultats plus significatifs. Dans le cas de la probabilité de croisement nous l'avons naturellement faite varier entre 0 et 1 en testant chaque dixième (0, 0.1, 0.2, ..., 1) (Figure 1).

Ainsi la faible variation (entre 235 et 238) montre que les croisements ne sont pas nécessaires à l'algorithme génétique pour trouver la meilleure solution possible, ils aident cependant l'algorithme à explorer plus vite l'espace des solutions.

4.2.2 • PROBABILITÉ DE MUTATION

Comme pour la probabilité de croisement, nous avons testé 11 valeurs entre 0 et 1 afin d'étudier l'impact de la mutation sur l'algorithme (Figure 2a). Nous nous attendions à des résultats différents de ceux sur le croisement : en effet la mutation est censée être déterminante pour le processus de convergence contrairement aux croisements. D'après ce graphique il est très clair que la valeur de la probabilité de mutation influe beaucoup la convergence et la valeur idéale semble proche de 0.05. Nous avons alors raffiné notre recherche en faisant les mêmes calculs pour une probabilité de mutation qui varie entre 0 et 0.2 (Figure 2b).

La valeur idéale de la probabilité de mutation que nous cherchions est donc environ 0.06, ce qui est à peu près ce qui est conseillé par la littérature.

4.2.3 • TAILLE DE LA POPULATION T

La taille de la population est un paramètre critique au même titre que le nombre de générations car la complexité de l'algorithme est en $O(Gtkn)$ et le temps de calcul est un paramètre important lors de la conception

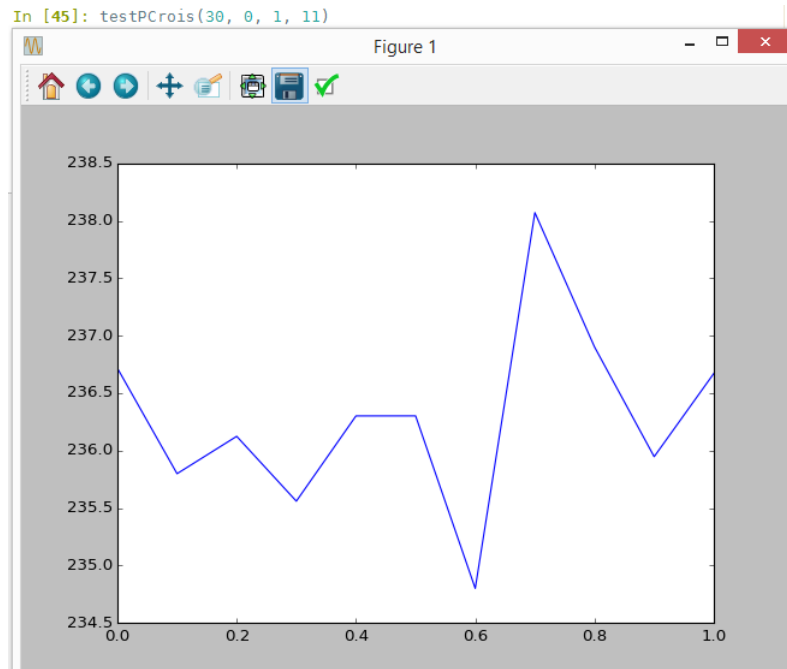


FIGURE 1 – Moyenne de la fitness des 30 essais en fonction de la probabilité de croisement

d'un programme. En fixant le nombre de générations à 100 l'algorithme a déjà un temps d'exécution de 8 secondes quand $t = 50$, ce qui signifie qu'à 500 individus une exécution prend 80 secondes. Ainsi il devient long de faire des tests par centaines pour étudier sa valeur idéale. Nous avons testé des valeurs de t jusqu'à 2000 (Figure 3) et nous en avons déduis qu'une taille de 500 individus était suffisante, étant donné le temps de calcul déjà conséquent qu'elle représente.

4.2.4 • NOMBRE DE GÉNÉRATIONS G

Le nombre de générations est simple à fixer : il suffit d'afficher la fitness du meilleur élément à chaque génération (Figure 4) et de regarder à quel moment il n'y a plus d'amélioration. Nous avons effectué ce test 5 fois en allant jusqu'à la génération 400 (ce qui nécessite déjà 5mn par exécution). Ainsi avons nous décidé de conserver cette valeur de 400 générations qui paraît adaptée, afin de trouver une solution satisfaisante sans faire exploser le temps de calcul.

4.3 Résultats finals : algorithme génétique

Ayant ainsi fixé les paramètres de l'algorithme génétique nous étions donc en mesure de le tester dans les meilleures conditions et de le comparer à d'autres algorithmes. Pour cela nous avons codé deux algorithmes concurrents : l'algorithme Sh, créé par David Shmoys en 1995 spécialement pour ce problème, qui est polynomial en n en temps et très rapide (quelques millisecondes), et un algorithme naïf qui teste $G \times t$ placements aléatoires d'entreprises et garde la meilleure (où G est le nombre de générations de l'algorithme génétique et t la taille de la population de l'algorithme génétique) qui a une complexité $O(Gtn)$ et un temps d'exécution très proche de celui de l'algorithme génétique. Les résultats de cette comparaison (Figures 5 et 6) sont clairs : l'algorithme génétique est meilleur à chaque fois. L'écart avec l'algorithme naïf est significatif et il ne faut pas oublier que le temps de calcul est le même, donc l'algorithme génétique apporte une nette amélioration. De plus il se base sur des concepts simples et compréhensibles, ce qui en fait un algorithme d'optimisation de choix.

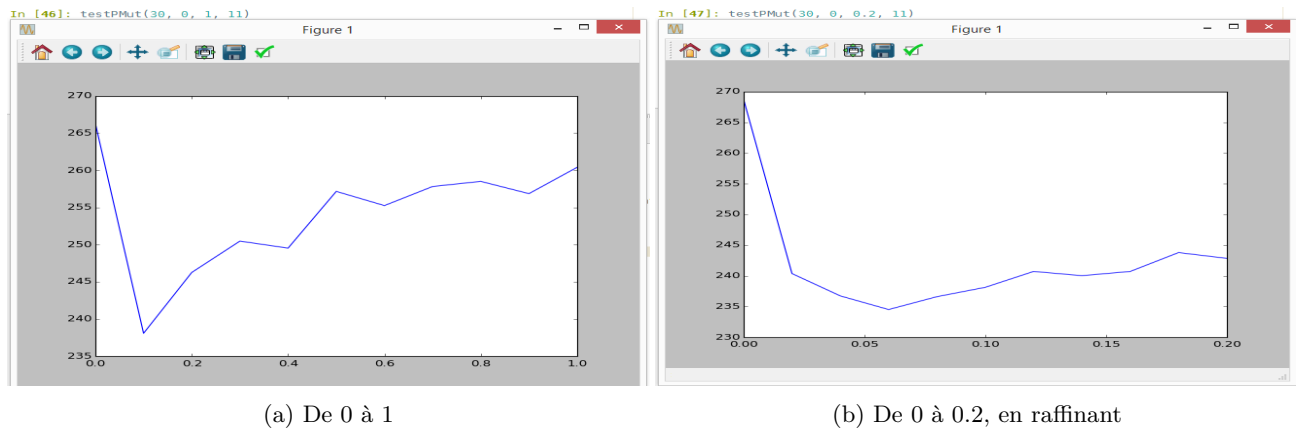


FIGURE 2 – Moyenne de la fitness des 30 essais en fonction de la probabilité de croisement

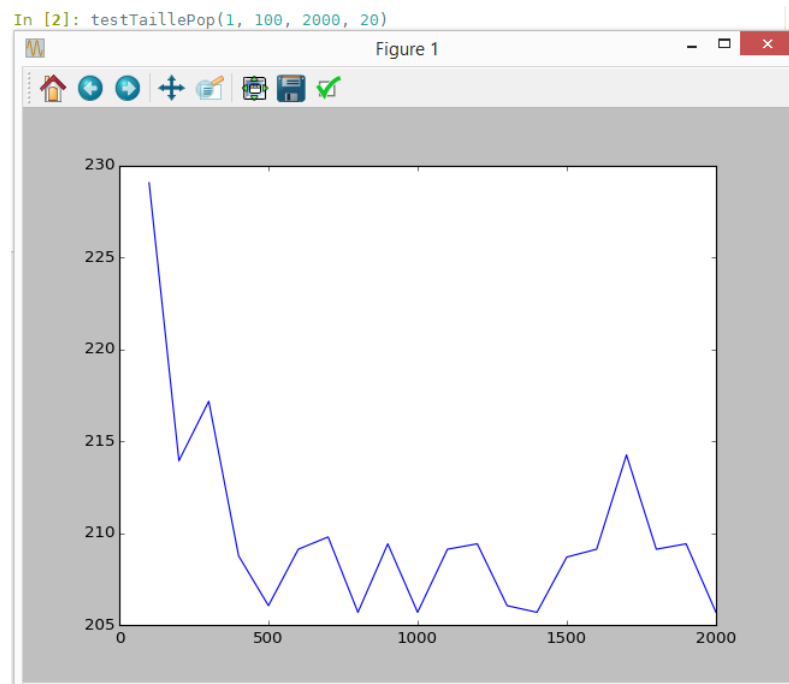


FIGURE 3 – Résultat de l'algorithme en fonction de la taille de la population

4.4 Ajustement des paramètres : méthode du recuit simulé

Le recuit simulé est une autre méthode permettant de répondre au problème proposé. Après avoir ajusté ses paramètres, nous pourrions comparer les performances du recuit simulé et de l'algorithme génétique.

Le recuit simulé permet théoriquement d'obtenir un minimum global au bout d'un temps infini. Il s'agit donc de trouver un bon compromis entre temps de calcul et qualité de la solution : pour cela nous utiliserons des méthodes du recuit simulé exponentiel et arithmétique.

Les méthodes de recuit simulées étudiées seront :

- le cas particulier de l'algorithme de Metropolis-Hasting (température constante)
- le recuit simulé exponentiel (variation géométrique de la température)

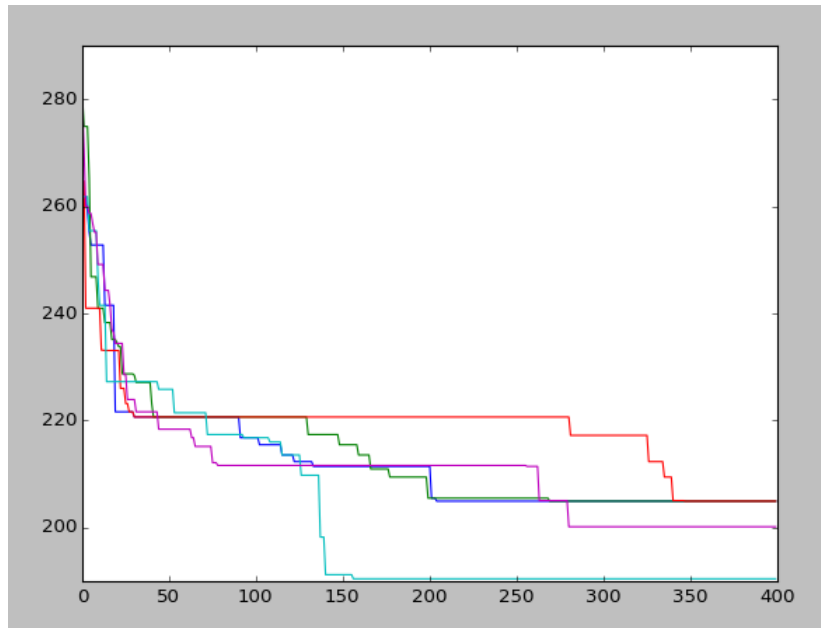


FIGURE 4 – Fitness du meilleur individu en fonction des générations

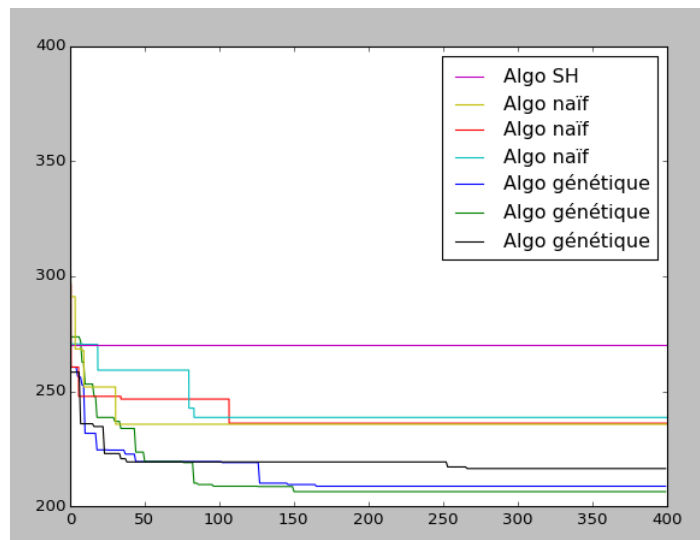


FIGURE 5 – Comparaison de l'évolution des trois algorithmes présentés au cours des générations

-le recuit simulé arithmétique (variation affine de la température)

Notre repère sera un algorithme naïf qui génère des solutions aléatoires.

4.4.1 • AJUSTEMENT DU PARAMÈTRE $e^{-\Delta E/T(n)}$

A chaque itération, ce paramètre donne la probabilité d'accepter une remontée de potentiel dans la recherche de solution.

Si on applique ce paramètre sans ajuster la température, il se trouve que la valeur de ΔT est trop élevée ce qui engendre un paramètre $e^{-\Delta E/T(n)}$ trop proche de 0 : il n'y a jamais de remontée de potentiel comme le montre la figure.

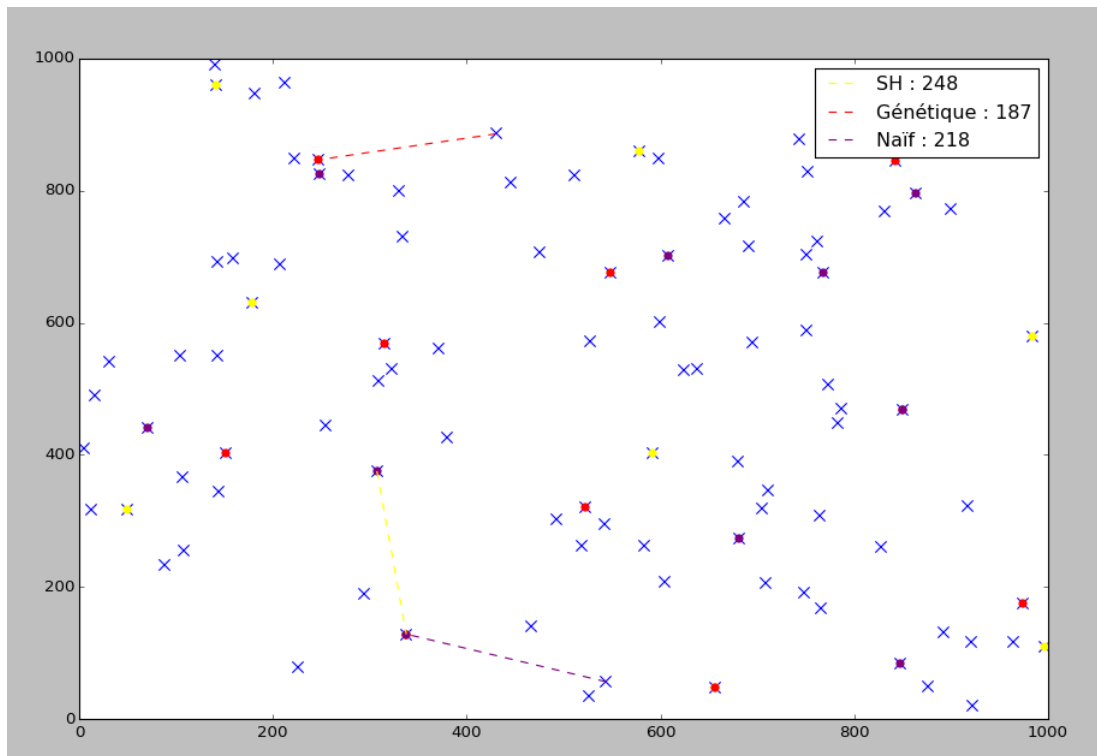


FIGURE 6 – Représentation graphique du problème et de ses solutions par les algorithmes

Il est donc nécessaire d'ajuster ce paramètre afin qu'il soit proche de 1 en début d'exécution de l'algorithme. Pour cela, il suffit d'ajuster T pour qu'il commence proche de ΔE moyen. Nous commençons donc par calculer une énergie moyenne calculée à partir de solutions aléatoires du problème.

Cela nous permet d'obtenir un algorithme de Metropolis efficace : qui accepte les remontées de potentiel (du moins en début d'algorithme).

Ensuite, il faut travailler sur la variation de $T(n)$ au cours du temps.

De façon générale, la nature de la méthode définit la façon dont varie T au cours du temps : d'abord il est fixé à T_0 calculé plus haut, mais ensuite il peut varier de façon affine au cours du temps (décroissance arithmétique : à chaque itération $T(n+1) = T(n) - r$) ou bien de façon exponentielle (décroissance exponentielle : à chaque itération $T(n+1) = r * T(n)$).

Afin de fixer le paramètre r , nous avons choisi de nous concentrer sur la valeur de T_{fin} à atteindre, et d'adapter r .

Pour cela, nous avons réalisé plusieurs essais du recuit simulé en faisant varier la décroissance, et il s'est avéré que l'état d'"équilibre" était atteint lorsque la température était de l'ordre de $0.01T_{init}$.

4.4.2 • AJUSTEMENT DU NOMBRE D'ITÉRATIONS

La méthode étudiée permet théoriquement d'atteindre un minimum global au bout d'un temps de calcul infini. Une approche pragmatique consiste à étudier l'évolution de la qualité des solutions trouvées en fonction du nombre d'itérations au cours d'une méthode du recuit simulé. Une fois que la qualité de la solution commence à stagner on considère qu'on a une bonne solution.

Étant donnés les paramètres du problèmes, on peut alors établir plusieurs observations :

-La qualité des solutions cesse d'évoluer significativement à partir de 1500 itérations, quelle que soit la méthode utilisée. Pour réaliser un grand nombre de simulations on peut se permettre de s'arrêter à 1500 itérations par simulation et garder des résultats assez précis.

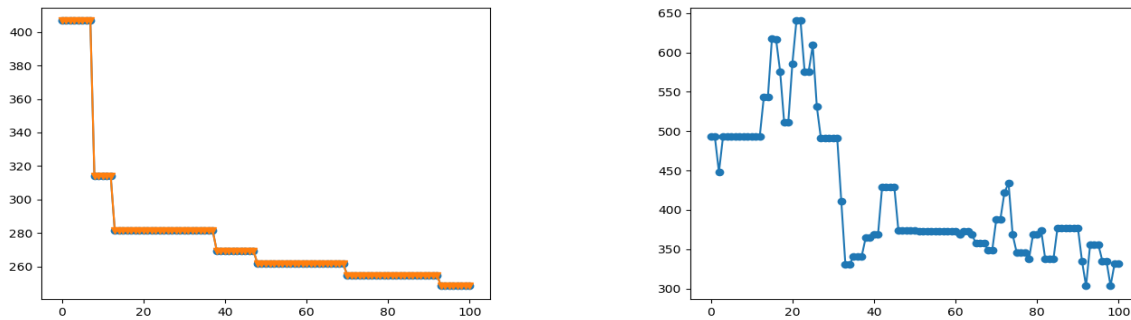


FIGURE 7 – Ajustement de la température initiale

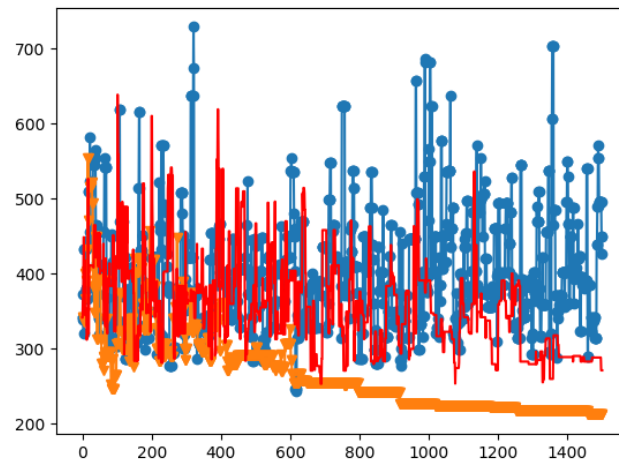


FIGURE 8 – en bleu : $T_f = 0.9 * T_{init}$ en rouge $T_f = 0.1 * T_{init}$ en orange $T_f = 0.01 * T_{init}$

-Globalement, le recuit simulé (qu'il soit algorithmique ou exponentiel) permet d'obtenir des solutions de qualité comparable, alors qu'une algorithm naïf reste bloqué à une certaine valeur du potentiel.

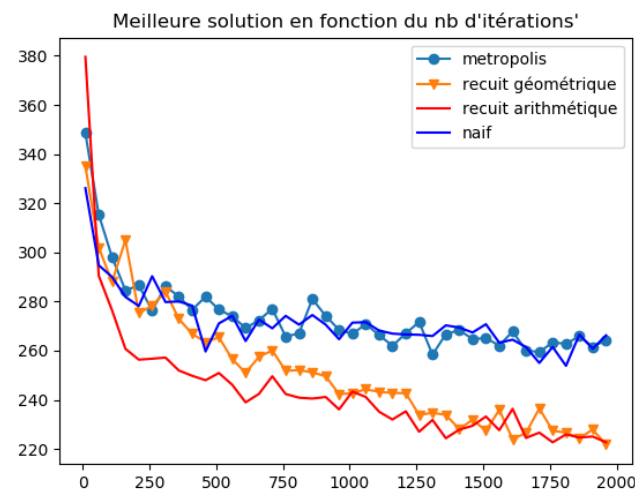
-Il semble que l'algorithme arithmétique soit légèrement plus efficace lorsqu'on réalise entre 200 et 300 itérations, mais cette différence s'efface dès que l'on dépasse les 1000 itérations.

4.4.3 • RÉSULTATS : RECUIT SIMULÉ

Ainsi, la méthode du recuit simulé permet d'obtenir un meilleur résultat qu'un algorithme naïf sans temps de calcul excessif. Au final, on obtient les moyennes suivantes pour des simulations à 1500 itérations :

- L'algorithme naïf obtient des solutions de qualité moyenne de 257 en moins de 0.1s
- Le recuit simulé arithmétique obtient des solutions de qualité moyenne de 233.1 en 14.7s
- Le recuit simulé géométrique obtient des solutions de qualité moyenne de 232.8 en 14.7s

On peut observer ci-dessous les solutions renvoyées par les méthodes naïve et une méthode du recuit simulé exponentiel sur une même instance du problème (sur la première figure les entreprises sont repérées par des carrés bleus, et sur la seconde elles sont repérées par des carrés rouges). La solution naïve présente une distance max de 373 contre 217 pour la solution trouvée par le recuit, pour un même nombre d'itérations.



Ainsi, pour le cas particulier de ce problème, les recuits simulés géométrique et arithmétique sont d'efficacité à peu près égales, et permettent d'obtenir des résultats sensiblement meilleurs.

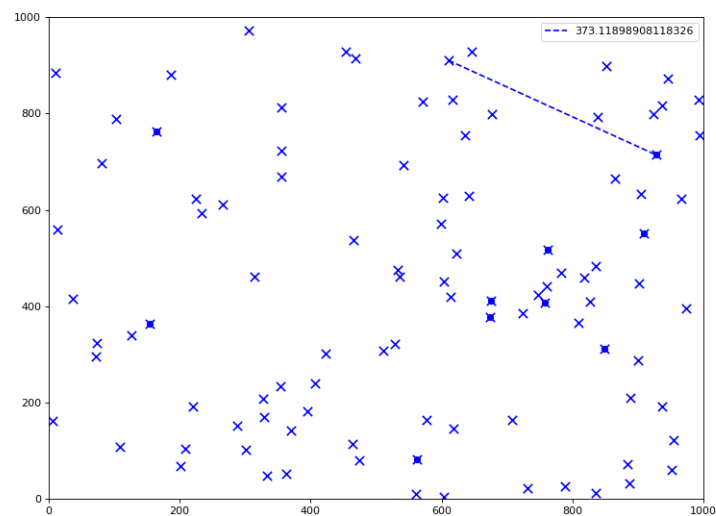


FIGURE 9 – Représentation d'une solution naïve

4.5 Comparaison des algorithmes

Maintenant que nous avons fixé les paramètres de chacun de nos 2 algorithmes (recuit simulé et algorithme génétique) nous cherchions à les comparer. Pour cela nous les avons testé plusieurs fois chacun en modifiant aléatoirement le placement des villes à chaque fois, afin de les comparer en atténuant le côté aléatoire du problème. Voici donc les résultats :

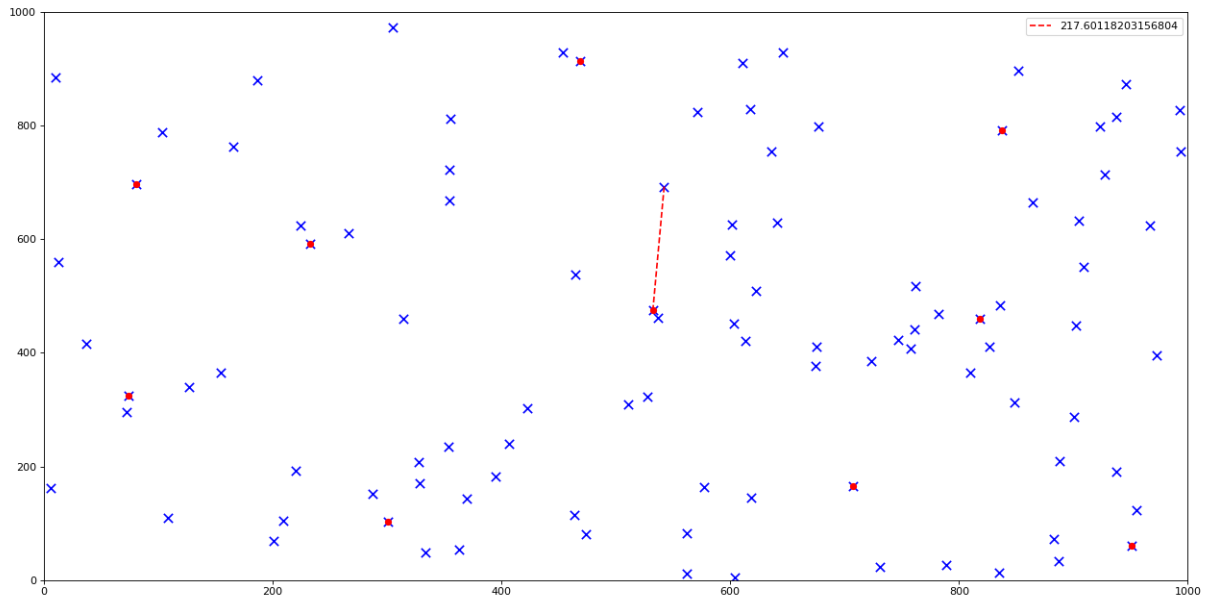


FIGURE 10 – Représentation d'une solution par le recuit

	Résultat moyen	Temps de calcul (s)
Algorithme SH	274.0	0.003
Algorithme naïf (ité = 62 500)	240.6	95.3
Algorithme génétique (t=180, G=80)	220.0	22.3
Algorithme génétique (t=250, G=250)	206.8	108.3
Algorithme génétique (t=500, G=400)	204.9	333
Recuit simulé géométrique (ité = 2000)	218.6	21.3
Recuit simulé arithmétique (ité = 2000)	217.8	21.3
Recuit simulé géométrique (ité = 8000)	211.4	105.8
Recuit simulé arithmétique (ité = 8000)	206.1	105.8

On peut en déduire que les deux algorithmes sont aussi efficaces l'un que l'autre dans la résolution de ce problème donné. La qualité de la solution dépend du temps de calcul et on voit ici qu'au moins 100s sont nécessaires pour obtenir une solution satisfaisante, bien que les données du problème ne soient pas de grande dimension ($n = 100$ et $k = 10$). Cependant le temps de calcul, qui est évidemment un problème pour une quelconque utilisation pratique, peut probablement être réduit en utilisant un autre langage plus efficace que Python, comme le C++, ou en apportant quelques optimisations au code. Cependant notre priorité ici n'était pas l'optimisation du temps de calcul mais la recherche du meilleur compromis possible.

5

La modélisation mathématique du trafic routier

5.1 Introduction

Dans cette partie, nous livrons un état de l'art de la modélisation mathématique du trafic routier, qui est l'objet de recherches depuis les années 1950 environ. On peut définir mathématiquement le trafic routier comme un ensemble de parcours, un parcours étant un chemin reliant un point d'origine à un point de destination.

L'étude du trafic routier se scinde généralement en deux approches distinctes. La première étudie la manière dont les conducteurs vont faire le choix de leur trajet, de la voie qu'ils vont emprunter, etc. La seconde, qui est celle que nous avons choisie ici, est l'étude de l'écoulement du trafic, notamment autour des phénomènes de poursuite et de congestion.

Les modèles que nous allons présenter sont séparés en deux catégories : les microscopiques, qui étudient le comportement de chaque véhicule individuellement, et les macroscopiques, qui renseignent sur des caractéristiques plus globales du trafic. Il existe également des modèles mésoscopiques, qui étudient le comportement des véhicules par paquets, mais ceux-ci sont moins courants d'utilisation.

5.2 Les modèles microscopiques

Des modèles microscopiques ont comme objectif la description du comportement individuel des conducteurs. C'est considéré que les usagers suivent un comportement en poursuite, correspondant au comportement d'un conducteur en réponse aux actions du véhicule qui le précède. Dans ce cas, il est courant de distinguer deux cas types de conduite :

- La conduite libre, le véhicule ne perçoit pas le véhicule qui le précède du fait d'un écart intervéhiculaire suffisamment important et dans ce cas, le conducteur circule à sa vitesse maximale désirée.
- La conduite en poursuite, le conducteur souhaite circuler à sa vitesse libre, mais son véhicule leader ne le lui permet pas, le contraignant ainsi à adopter une vitesse inférieure à sa vitesse désirée.

5.2.1 • MODÈLES À DISTANCE DE SÉCURITÉ

Ces modèles cherchent à décrire simplement la dynamique du seul véhicule en relation avec son prédécesseur, de sorte à respecter une certaine distance de sécurité. C'est grosso modo la base mathématique du logiciel AIMSUM.

Le premier modèle de ce type là a été le modèle de Pipes[Pipes, 1953], selon lequel « une bonne règle pour suivre un autre véhicule à une distance sûre est de laisser au moins l'équivalent d'une longueur de son propre véhicule tous les dix miles par heure [soit 16,1 km/h] ». On a donc :

$$S_i(t) = L_i \left(1 + \frac{v_i(t)}{16.1} \right) \quad (1)$$

Où S_i est la distance de sécurité entre i et son prédécesseur $i+1$, L_i est la longueur de la voiture i et v_i est la vitesse du véhicule i .

Dans les modèles de distance de sécurité il est possible de distinguer les modèles de collision avoidance. Ce type modèle vise plutôt à retranscrire la trajectoire d'un véhicule en fonction d'une distance de sécurité minimale. Le développement de ce type de modèles s'appuie sur une utilisation des simples équations newtoniennes du mouvement. Le modèle de Kometani et Sasaki spécifie que l'interdistance minimale permettant d'éviter toute collision est donnée par :

$$S_i(t) = \alpha v_{i-1}^2(t) + \beta v_i^2(t + T) + \gamma v_i(t + T) + \delta \quad (2)$$

Où α et β représentent l'inverse de la capacité maximale de décélération respectivement du véhicule leader et du véhicule suiveur et γ et δ sont homogènes respectivement à l'inverse et à une distance. La valeur T représente un temps de réaction.

Ce modèle peut être amélioré en ajoutant la taux de freinage maximal b_i^{max} et la vitesse désirée $V_{i,désirée}$. Ainsi, la vitesse du véhicule en $t+T$ devient la valeur minimal entre la vitesse de sécurité V_{s_i} et la vitesse maximale (V_{max}) qu'il peut atteindre avec sa puissance mécanique. Alors la vitesse est exprimé comme :

$$V_i(t+T) = \min(V_{max_i}, V_{s_i}) \quad (3)$$

Où :

$$V_{s_i}(t+T) = b_i^{max}T + \sqrt{(b_i^{max}T)^2 - b_i^{max} \left[2S_i(t) - Sc_i - v_i(t)T - \frac{v_{i-1}(t)^2}{b_{i-1}^{max}} \right]} \quad (4)$$

$$V_{max_i}(t+T) = v_i(t) + 2.5a_i^{max}T \left(1 - \frac{v_i(t)}{V_{i,désirée}} \right) \sqrt{0.025 + \frac{v_i(t)}{V_{i,désirée}}} \quad (5)$$

La quantité Sc_i est l'interdistance minimale, c'est à dire la valeur minimal qui assure que le véhicule suiveur ne pénètre jamais dans le véhicule analysé. La décélération maximale du véhicule leader (b_i^{max}) est estimée visuellement par le conducteur du véhicule suivant.

Finalement ce modèle a été réprisé par Krauss dans [Krauss, 1997]. Ses travaux proposent un modèle discret en temps dans lequel la vitesse est recalculée selon le processus suivant :

1. On considère l'inéquation assurant que le véhicule i n'entre pas en collision lors d'un freinage avec son véhicule leader. Celle-ci est donnée par une relation sur les distances de freinage du véhicule leader et du véhicule suiveur, fonctions des vitesses de chacun d'eux ; Étant donné la distance entre deux véhicules la vitesse permettant d'éviter la collision est égale à la solution cette inéquation. Krauss assure alors que :

$$V_{i,sécurité}(t) = V_{i-1}(t) + \frac{S_i(t) - V_{i-1}(t)T}{\frac{V_i(t) + V_{i-1}(t)}{b_i(t) + b_{i-1}(t)} + T}$$

Le produit $V_{i-1}T$ est considéré comme étant l'interdistance désirée et $b_i(t)$ est la décélération du véhicule, dans ce modèle supposée identique pour l'ensemble des véhicules ;

2. La vitesse désirée est alors prise égale à :

$$V_{i,désirée} = \min(V_i(t) + a_i(t)\Delta t, V_{t,sécurité}, V_{max})$$

où $a_i(t)$ est l'accélération du véhicule et V_{max} la vitesse maximale de la route ;

3. La vitesse est alors mise à jour, en considérant que $V_i(t + \Delta t) = \max(0, V_{désirée} - \epsilon)$ où ϵ représente une perturbation aléatoire introduite pour représenter les cas déviants d'une conduite optimale. Enfin, la position du véhicule est donnée par :

$$x(t + \Delta t) = x(t) + V_i(t + \Delta t)\Delta t$$

5.2.2 • LIMITES DU MODÈLE

À cause des hypothèses simplificatrices le modèle utilisé a quelques limitations. Selon Leclercq [Leclercq, 2002] :

- Ce type de modèle considère, normalement, conducteurs uniformes. Notamment à cause de cette uniformité il aurait des incohérences physiques et psychologiques, vu que chaque personne à son propre temps de réaction et que ne sont pas tous les conducteurs que veulent rouler à la vitesse maximale, par exemple.
- Ce modèle n'est utile qu'en situations de trafic dense, puisque si la distance entre deux véhicules est trop grande il n'y a pas de risque de collision et donc, chaque véhicule roule à sa vitesse désirée.

5.3 Les modèles macroscopiques

Ces modèles sont dédiés à l'étude du trafic de manière globale et non plus à l'échelle des interactions entre véhicules proches. Ils permettent l'étude du trafic dans de grands réseaux routiers, notamment dans l'optique de la création d'infrastructures pouvant influencer sur ces derniers.

Ces modèles font intervenir différentes variables, mais trois d'entre elles leur sont communes :

- Le débit $Q(x, t)$: il s'agit du nombre de véhicules passant par le point d'abscisse x par unité de temps.
- La concentration $K(x, t)$, ou densité spatiale instantanée : il s'agit du nombre de véhicules présents sur une section de longueur dx autour du point d'abscisse x
- La vitesse de flot $V(x, t) = \frac{Q(x, t)}{K(x, t)}$, dont on peut montrer qu'elle est égale à la vitesse moyenne spatiale autour du point d'abscisse x

5.3.1 • LE MODÈLE LWR

Ce modèle n'est pas le plus élaboré des modèles macroscopiques, mais son efficacité suffit à faire de lui le modèle le plus utilisé dans le cadre de la modélisation macroscopique du trafic. Développé par Lighthill, Whitham et Richards en 1955, il consiste en un système de trois équations à trois inconnues, analogue à un système que l'on retrouve en dynamique des fluides. Une de ses principales caractéristiques, qui est d'ailleurs une limitation, est qu'il ne peut décrire qu'un trafic à l'équilibre. Il induit donc que le trafic passe instantanément d'un état d'équilibre à un autre, sans régime transitoire. Nous verrons plus tard des modèles qui résolvent cette limitation. Le système d'équations de LWR est le suivant :

$$\begin{cases} Q(x, t) = K(x, t) \times V(x, t) \\ \frac{\partial Q(x, t)}{\partial x} + \frac{\partial K(x, t)}{\partial t} = 0 \\ V(x, t) = V_e(K(x, t)) \end{cases}$$

- La première équation définit simplement la vitesse comme étant égale au quotient du débit par la concentration.
- La deuxième est l'équation de conservation. Elle lie la variation temporelle de la concentration en un point à la variation spatiale du débit. Physiquement, elle consiste à dire que la variation de la concentration correspond au nombre de véhicules qui rentrent dans une section donnée pendant un laps de temps, moins le nombre de véhicules qui en sortent.
- La troisième équation est celle du diagramme fondamental. Elle décrit la situation d'équilibre évoquée précédemment, en imposant à la vitesse de flot d'être à tout instant et en tout point égale à une vitesse à l'équilibre, fonction uniquement de la concentration. Le diagramme fondamental, décrivant la fonction V_e est le seul paramètre du modèle.

Malgré sa définition par un système de 3 équations, le modèle LWR est en réalité un système du premier ordre, c'est-à-dire qu'il peut être condensé en une unique équation :

$$\frac{\partial K(x, t)}{\partial t} + Q'_e(K(x, t)) \times \frac{\partial K(x, t)}{\partial x} = 0$$

avec $Q_e(x, t) = K(x, t) \times V_e(K(x, t))$.

Il existe plus d'une vingtaine d'allures fréquemment utilisées pour les diagrammes fondamentaux. Généralement, elles distinguent une situation de trafic fluide pour les faibles concentrations d'une situation de trafic congestionné à forte concentration.

Une des principaux avantages du modèle LWR est l'existence de solutions analytiques facilement calculables dans les cas simples. C'est ce qui justifie le fait qu'il soit souvent utilisé malgré le défaut évoqué précédemment.

5.3.2 • LES MODÈLES DU SECOND ORDRE

Les modèles du second ordre, datant des années 70, ont l'avantage de permettre l'existence d'état de non-équilibre et de régimes transitoires en remplaçant l'équation d'équilibre de LWR par une équation exprimant

l'accélération du flux.

Ils peuvent être regroupés sous la forme du système à deux équations suivant, qui est cette fois irréductible.

$$\begin{cases} \frac{\partial K}{\partial t} + \frac{\partial KV}{\partial x} = 0 \\ \frac{\partial V}{\partial t} + V \frac{\partial V}{\partial x} = \frac{1}{\tau}(V_e(K) - V) - \frac{1}{K}C^2(K) \frac{\partial K}{\partial x} \end{cases}$$

Le paramètre C est appelé célérité caractéristique du trafic, il est lié à la vitesse à laquelle se propage l'information dans le système. Plusieurs expressions de ce paramètre sont régulièrement utilisées. τ est quand à lui le temps de réaction des conducteurs, que l'on considère généralement uniforme.

Malgré leur plus grande précision, les modèles du second ordre sont moins utilisés que LWR car les solutions en sont plus difficiles à calculer.

5.3.3 • LE MODÈLE CELLULAIRE

Ce modèle datant de 1993 propose une approche radicalement différente de celle décrite précédemment.

Il commence par discrétiser le paramètre t , ainsi que l'espace.

Le réseau est divisé en cellules de longueur égales à la distance parcourue en une unité de temps par un véhicule circulant à une vitesse V_{max} , que l'on suppose majorer la vitesse du flot.

On note $n_i(t)$ le nombre de véhicules présents dans la cellule i à l'instant t , $N_i(t)$ le nombre maximal de véhicules que peut contenir la cellule i à l'instant t , et $y_i(t)$ le nombre de véhicules entrant dans la cellule i à l'instant t .

Cela donne la loi de transmission suivante :

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t)$$

où l'on pose $y_i(t) = \min\left(n_{i-1}(t), Q_i(t), \frac{w}{V(x,t)}(N_i(t) - n_i(t))\right)$.

avec $Q_i(t)$ donné par le diagramme fondamental suivant :

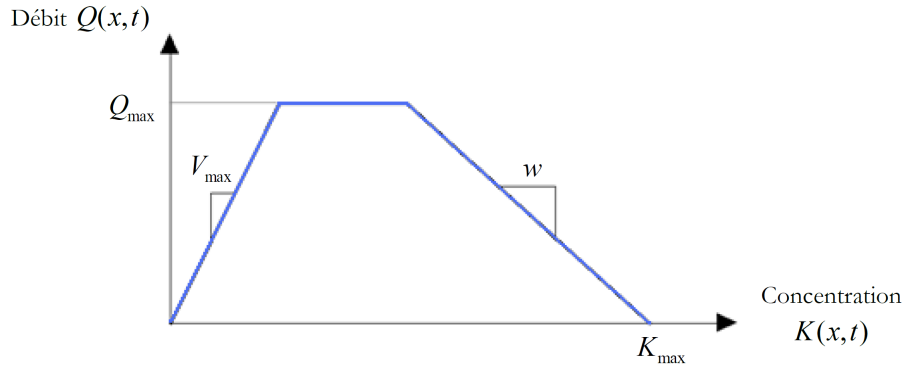


FIGURE 11 – Diagramme fondamental du modèle cellulaire

w est un paramètre du modèle.

5.3.4 • LES MODÈLES CINÉTIQUES

Ces modèles des années 70, inspirés de la théorie cinétique des gaz de Boltzmann, permettent de prendre en compte l'aspect aléatoire du trafic.

La grandeur K y est observée comme une densité particulière, liée à une distribution des vitesses $f(x, v, t)$ par la formule suivante :

$$K(x, t) = \int_0^{+\infty} f(x, u, t) du$$

On définit à nouveau la vitesse comme le rapport du débit sur la concentration, soit

$$v = \frac{\int_0^{+\infty} f(x, u, t) u du}{\int_0^{+\infty} f(x, u, t) du}$$

Les équations qui définissent le modèle sont ensuite relativement classique, à ceci près qu'elles font intervenir un paramètre stochastique : la probabilité qu'un véhicule a de dépasser un véhicule qu'il rattrape. Ce paramètre est généralement défini de manière inversement proportionnelle à K .

Ces modèles ne sont cependant que très peu utilisés, car il sont très coûteux en temps de calcul et leurs solutions analytiques sont très difficiles à calculer. Ils présentent cependant un cadre d'étude intéressant, par exemple pour le passage du microscopique au macroscopique, en raison de leur analogie avec la théorie de Boltzmann.

5.4 Conclusion

Chaque modèle a ses propres limitations, comme aussi un domaine d'utilisation optimale. En effet, l'utilisation d'un modèle hors de son domaine donne souvent des résultats décevants.

Alors, avant de pouvoir réaliser une quelconque étude portant sur le trafic routier, il est nécessaire de préciser les échelles spatiale et temporelle de cette étude [Bourrel, 2003]. Cela traduit en outre la volonté de l'expérimentateur d'observer plus ou moins finement le réseau. Par ailleurs, certains auteurs ont développé le principe de l'hybridation des modèles dans lesquels il est possible d'utiliser les modèles adéquats selon la recherche de finesse souhaitée et il y a quelques logiciels de simulation permettent un couplage entre l'échelle microscopique et l'échelle macroscopique.

6

Conclusion

Après une étude théorique de certaines classes d'algorithmes d'optimisation (algorithmes génétiques et recuit simulé) nous les avons testés sur un problème NP-difficile, le *Vertex k -center problem*, afin de vérifier que la pratique était en accord avec la théorie.

Il s'est avéré qu'après avoir trouvé les paramètres adaptés à chacun, l'application de ces algorithmes était efficace sur le problème en question, relativement aux autres algorithmes "témoins". Ces algorithmes, relativement simples à comprendre et à mettre en place, sont donc de très bonnes solutions à mettre en place dans le cas d'un problème d'optimisation à faible régularité, typiquement dans le cas de simulations où la fonction à optimiser n'est pas dérivable.

Afin de revenir à notre problème initial de trafic routier, malgré l'impossibilité de lui appliquer nos algorithmes faute de logiciel de simulation, nous avons tout de même réalisé une étude théorique et mathématique de la modélisation du trafic routier.

7

References Bibliographiques

L.A. PIPES, An operational analysis of traffic dynamics, Journal of Applied Physics, volume n24, mars 1953, pp. 274-281.

S. KRAUSS, Microscopic modeling of traffic flow : investigation of collision free vehicle dynamics, University of Cologne, Germany, 1997, 129 pages

L. LECLERCQ, Modélisation dynamique du trafic et applications à l'estimation du bruit routier, Mémoire de thèse, LICIT, Octobre 2002, 310 pages.

E. BOURREL, Modélisation dynamique de l'écoulement du trafic routier : du macroscopique au microscopique, Mémoire de thèse, INSA Lyon, Décembre 2003, 261 pages.

G. COSTESEQUE, Modélisation du trafic routier : passage du microscopique au macroscopique, Mémoire de master, ENPC, 2011, 103 pages.