Optimisation des paramètres d'un algorithme génétique via le problème inverse dans le cadre d'un Vehicle Routing Problem

Lucas Anki
Supervisé par :
Alexandre Kornmann
Matéo Valdes

UFR de Mathématique et d'Informatique

26/08/2021

Introduction

SYNOVO GROUP
IT & SOFTWARE SOLUTIONS

Saphir

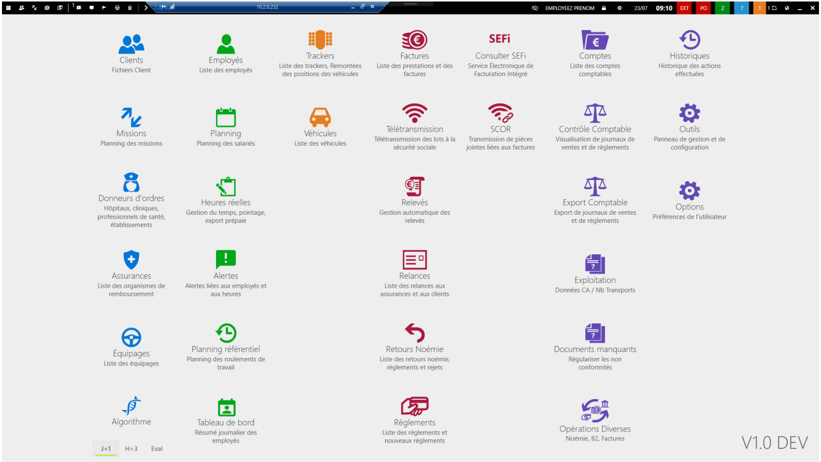


Figure: L'interface de Saphir

Vehicle Routing Problem (VRP)

Vehicle Routing Problem

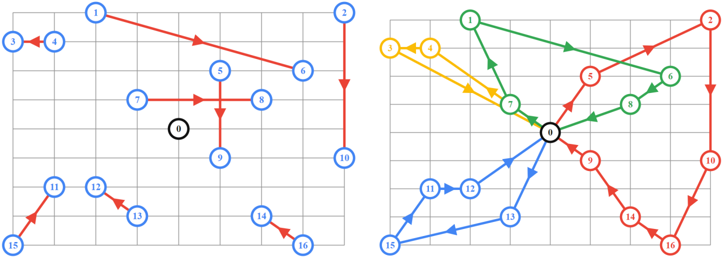
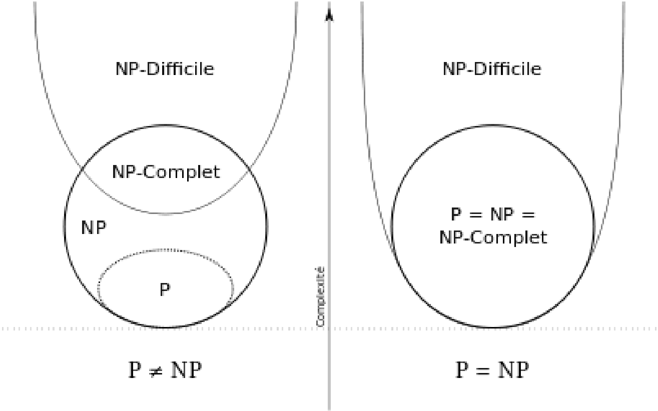


Figure: Un Vehicles Routing Problem with Pickups and Deliveries, et une solution possible

Vehicle Routing Problem (VRP)

Un problème NP-complet



Algorithme génétique

Algorithme génétique

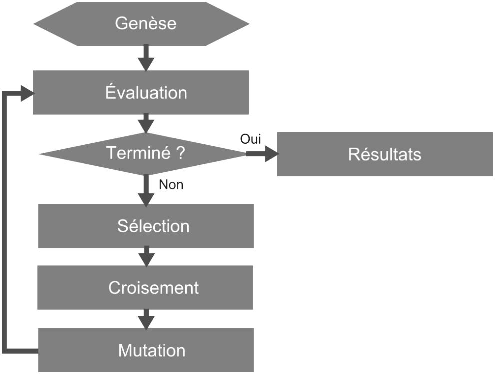


Figure: Principe d'un algorithme génétique de base

Algorithme génétique

Notre fonction d'évaluation

- Prend en entrée un planning, et lui assigne une note appelé fitness
- Plus le fitness est minimal, plus le planning est considéré comme bon
- La fonction d'évaluation sélectionne les meilleurs individus, favorisant la convergence vers une solution dans nos attente
- Elle est somme de plusieurs autres fonctions dépendant elles mêmes de plusieurs coefficients : des coefficients de pénalisation
- Pour deux jeux de coefficients de pénalisation, un même planning n'obtiendra pas forcément la même note

Formulation du problème

Problématique : Déterminer les coefficients de pénalisation optimaux permettant à l'algorithme génétique de proposer des plannings en accord avec les critères d'un client.

- On ne peut pas simplement minimiser la sortie de la fonction d'évaluation !
- On va passer par des variables annexes

Formulation du problème

Un exemple de variable annexe : La distance à vide

- La distance à vide se calcule à partir d'un planning
- On dispose de plannings reflétant les attentes du client
- On dispose du plannings de sortie de l'algorithme génétique

On va faire en sorte de choisir les coefficients de pénalisation permettant d'avoir le planning de sortie de l'algorithme génétique le plus proche possible du planning client

Formulation du problème

- Y_i^{exp} les variables représentant la bonne qualité d'un planning client
- Y_i ces mêmes variables issues cette fois ci du planning de sortie de l'algorithme génétique
- $\theta_1, \ldots, \theta_n$ les coefficients de pénalisation dont dépend la fonction d'évaluation
- σ un coefficient

Le problème devient :

$$\min_{\theta_1,\ldots,\theta_n} \sum_{i=1}^{k} \frac{(Y_i^{exp} - Y_i)^2}{\sigma}$$

Méthodes d'optimisation

Les méthodes

- Dual/Simulated Annealing
- Optimisation bayésienne
- Differential Evolution
- Curve Fiting

Méthodes d'optimisation

Dual/Simulated Annealing

Le recuit simulé s'inspire inspire d'un processus de métallurgie consistant à alterner des cycles de refroidissement lent et de réchauffage, ce qui a pour effet de minimiser l'énergie d'un matériaux.

Par analogie avec le processus physique, on note E la fonction à minimiser, qui sera l'énergie du système. On note T un paramètre fictif représentant la température du système.

Méthodes d'optimisation

Optimisation bayésienne

- Très utile pour minimiser des fonctions "boite noire" dont on ne connait pas d'expression du gradient
- On l'utilise sur des fonctions couteuse à évaluer
- Utilise lorsque plusieurs évaluations d'une même fonction en un même point donnent des résultat différent

L'optimisation bayésienne consiste à utiliser un *prior*, c'est à dire une loi de probabilité qui se fonde sur les précédentes évaluations, afin d'obtenir des informations sur le comportement de la fonction. Ce prior permet ensuite de construire une fonction d'acquisition permettant de déterminer le prochain point à évaluer.

Un exemple sur une fonction simple

On va tester les différentes méthodes, selon la procédure suivante :

- Choisir une fonction mathématique à plusieurs paramètres, sous la forme $f(x, \beta)$
- Fixer les paramètres β
- Générer un certains nombres de couple $(x_i, y_i = f(x_i, \beta))$ (Remarque : On peut également générer des couples légèrement bruité $(x_i, f(x_i, \beta) + \alpha b$ avec α un coefficient et b suivant une loi uniforme $U(-1, 1))$
- Construire une fonction F prenant en entrée des paramètres α, et renvoyant $\chi^2 = \sum_i (\frac{y_i - f(x_i, \alpha)}{\sigma})^2$ avec σ un paramètre fixé
- Lancer les différents algorithmes d'optimisation afin de minimiser la fonction F

Un exemple sur une fonction simple

On considère la fonction suivantes à deux variables

$$\begin{aligned} f : \mathbb{R} \times \mathbb{R} &\to \mathbb{R} \\ (a, b) &\mapsto \frac{1}{1 + 2^{-(t-a)b}} - \frac{1}{1 + 2^{ab}} \end{aligned}$$

avec $t \in \mathbb{R}$ un paramètre fixé.

On génère 50 points t_i repartis linéairement dans l'intervalle $[-1, 1]$ et on fixe $a = 0.45$ et $b = 100$. On peut ensuite générer 50 observations $f(t_i, a, b)$ ainsi que 50 observations bruité $\tilde{f}(t_i, a, b)$.

Un exemple sur une fonction simple

Grâce à ces observations, on construit nos fonctions F et $\tilde{F}$ à minimiser :

$$F : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$
$$(x, y) \mapsto \sum_{i=1}^{50} \left(\frac{f(t_i, a, b) - f(t_i, x, y)}{\sigma}\right)^2$$

$$\tilde{F} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$
$$(x, y) \mapsto \sum_{i=1}^{50} \left(\frac{\tilde{f}(t_i, a, b) - f(t_i, x, y)}{\sigma}\right)^2$$

avec $\sigma = 10^{-8}$

Un exemple sur une fonction simple

On fixe la recherche des paramètres dans l'ensemble $[0, 2] \times [10, 600]$.

Comme nous cherchons une solution dans un espace de dimension 2, il est possible d'avoir une représentation graphique de cet espace :

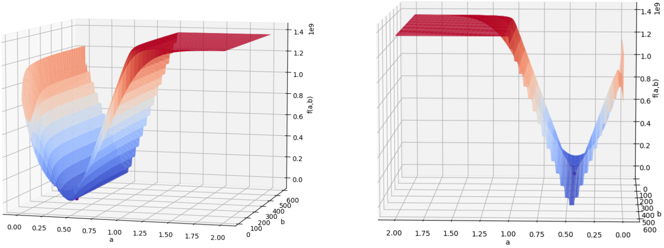


Figure: Visualisation de la fonction F à minimiser

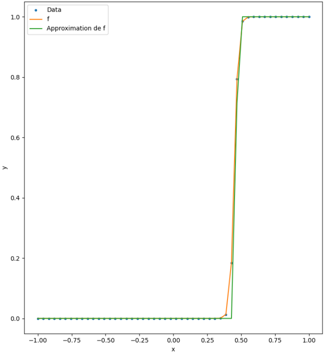
Un exemple sur une fonction simple

Méthode	Converge vers	Nombre d'appel de f
Dual annealing	$(0.45, 100.0)$	4121
Optimisation bayésienne	$(0.46, 600.0)$	300
Différential evolution	$(0.45, 100.0)$	3078
Gauss-Newton	$(0.45, 100.0)$	36
Levenberg-Marquardt	$(0.45, 100.0)$	36

Méthode	Converge vers	Nombre d'appel de f
Dual annealing	$(0.45, 99.97)$	4088
Optimisation bayésienne	$(0.44, 158.65)$	300
Différential evolution	$(0.45, 99.97)$	552
Gauss-Newton	$(0.45, 99.97)$	36
Levenberg-Marquardt	$(0.45, 99.44)$	45

Figure: Résultats pour F (en haut) et $\tilde{F}$ (en bas)

Un exemple sur une fonction simple



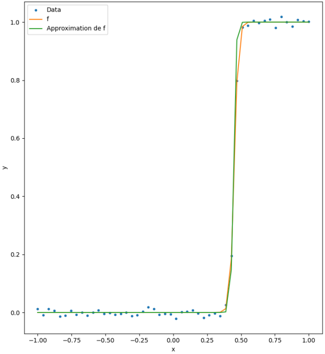


Figure: Résultat pour l'optimisation bayesienne les données bruité non bruité (à gauche) bruité (à droite)

Un exemple sur une fonction simple

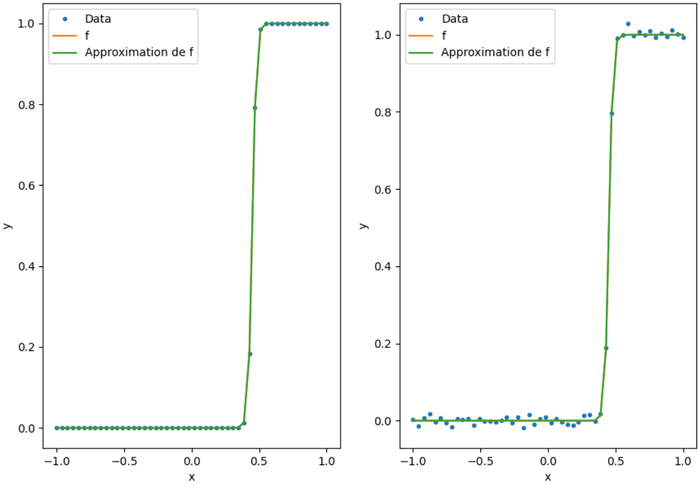


Figure: Résultat pour l'optimisation bayesienne les données bruité non

Un exemple sur une fonction simple

Analyse des résultats

On peut améliorer le nombre d'évaluation de la fonction.

Par exemple pour Dual annealing, on peut fixer le nombre d'appel maximum de la fonction à 500 par exemple. On peut également passer sur du simulated annealing en enlevant la minimisation locale.

	Converge vers	Nombre d'appel de f
Données non bruité	$(0.45, 100.0)$	500
Données bruité	$(0.45, 102.01)$	500

Application au cas de la fonction d'évaluation de l'algorithme génétique

Rappel : On cherche à déterminer les meilleurs coefficients de pénalisation, dont dépend la fonction d'évaluation et par conséquent l'algorithme génétique, permettant d'avoir en sortie de l'algorithme génétique un planning conforme avec le planning type du client

Problème : Ce planning de sortie est celui ayant eu la note, attribuée par la fonction d'évaluation, la plus basse au cours des générations de l'algorithme génétique. Mais pour justement attribuer cette note au planning, on a besoin de coefficients de pénalisation, alors que c'est ce que l'on cherche à déterminer.

Application au cas de la fonction d'évaluation de l'algorithme génétique

$\rightarrow$ Pour contrer ce problème, on ne minimisera pas directement la note, mais plutôt des variables annexes intrinsèque à chaque planning représentant la bonne qualité d'un planning.

$\rightarrow$ On pourra donc comparer ces variables pour un planning issu de l'algorithme génétique avec ces mêmes variables issu d'un planning type client.

Application au cas de la fonction d'évaluation de l'algorithme génétique

Des exemples de variables

- La distance à vide : Elle représente le nombre total de mètres où des véhicules ont roulés sans patient à l'intérieur.
- La prise de service anticipée : Elle indique le nombres total d'heures où les employées ont pris leur service en avance. Par exemple, si un employé commence normalement le matin à 6h, mais qu'on lui a assigné une mission à 5h30, et qu'il doit faire 20 minutes de trajet depuis la base pour se rendre à cette mission, alors cet employé aura $20 + 30 = 50$ minutes de prises de services anticipée
- Le retard : Il représente le retard cumulé de toutes les missions

Application au cas de la fonction d'évaluation de l'algorithme génétique

Des exemples de variables

Voici un exemple de ces variables extraites de quatre planning type d'un client générique :

	Distance à vide	Prise service anticipée	Retard
Planning 1	168153	18900	6619
Planning 2	133677	8100	1171
Planning 3	117518	9000	3248
Planning 4	208888	12600	4954

Regardons maintenant la différence entre la plus grande et la plus petite valeur de chaque variable :

	Distance à vide	Prise service anticipée	Retard
Max-Min	91370	10800	6448

Application au cas de la fonction d'évaluation de l'algorithme génétique

Des exemples de variables

Il y a un écart dans les ordres de grandeurs de ces variables.

On normalise l'ancienne formulation :

$$\chi^2 = \sum_{i=1}^{k} \left(\frac{Y_i^{exp} - Y_i}{\sigma}\right)^2$$

par :

$$\chi^2 = \sum_{i=1}^{k} \frac{1}{\sigma^2} \left(1 - \frac{1 + Y_i}{1 + Y_i^{exp}}\right)^2$$

Application au cas de la fonction d'évaluation de l'algorithme génétique

Les coefficients de pénalisation

Ils servent de paramètres à des fonctions de pénalisation, fonctions qui servent à construire la fonction d'évaluation de l'algorithme génétique.
Ils pénalisent certains aspects d'un planning, ce qui permet à l'algorithme génétique de converger vers une solution acceptable pour nous.

Voici quelques exemples de ces coefficients de pénalisation :

- Distance à vide : Coefficient servant à contrôler la distance à vide .
- TapFinancial : Coefficient servant à contrôler la rentabilité d'une mission
- Retard : Coefficient servant à contrôler le retard

Application au cas de la fonction d'évaluation de l'algorithme génétique

Les coefficients de pénalisation

- On constate donc que certains coefficients pénalisent des aspects que l'on a déjà évoqué dans les variables
- On en déduit que si l'on souhaite par exemple minimiser le coefficient de la distance à vide, il sera intéressant de construire notre χ^2 avec la variable représentant justement la distance à vide
- Pour chaque coefficients de pénalisation que l'on souhaitera optimiser, on utilisera donc pour la construction du χ^2 des variables reflétant l'impact de ce paramètre sur le planning

Application au cas de la fonction d'évaluation de l'algorithme génétique

Procédure

Voici la procédure que l'on va donc suivre dans un premier temps:

- On fixe le nombre de coefficients de pénalisation à optimiser et on choisit les variables qui serviront à construire le χ^2
- On génère un certains nombre de données clients avec un jeu de coefficients de pénalisation fixé
- De ces données on y extrait les variables du *chi*^{2} , qui seront donc nos variables d'observations

Application au cas de la fonction d'évaluation de l'algorithme génétique

Procédure

- On construit une fonction f prenant en entrée des coefficients de pénalisation, effectuant l'algorithme génétique pour chaque jeu de donnée client et pendant un certain nombre de génération, en utilisant les coefficients d'entrée, et renvoyant les variables des planning de sortie des run de l'algorithme génétique
- On construit notre fonction à optimiser, prenant en entrée les coefficients de pénalisation, appelant la fonction précédente f en récupérant les variables d'exploitation, et renvoyant le chi^2 construit à l'aide des variables d'observation et d'exploitation
- On utilise les différents algorithmes d'optimisation sur cette fonction

Application au cas de la fonction d'évaluation de l'algorithme génétique

Les différents coéfficients de pénalisation ont généralement des valeurs allant de 10^3 à 10^{14} pour certains, par conséquent dus à l'énorme différence d'ordre de grandeur entre ces valeurs, on ne cherchera pas à les optimiser directement.

A la place, on écrira les pénalités sous la forme 10^x et on cherchera à optimiser x.

Application au cas de la fonction d'évaluation de l'algorithme génétique

Recherche d'un coefficient de pénalisation

On applique la procédure précédente.

- On génère 6 jeux de données client en fixant les coefficients de pénalisation, dont le coefficient de la distance à vide, que l'ont fixe à 10^7
- On utilise 3 variables pour le chi^2 : la distance à vide (dav), le retard (ret), et la prise de service anticipée(psa)

On cherche à retrouver la puissance du coefficient de pénalisation sous la forme 10^x avec $x = 7$.
On effectue la recherche dans l'intervalle $[3, 11]$.
On arrêtera les algorithmes génétiques à 2500 itérations.

Application au cas de la fonction d'évaluation de l'algorithme génétique

Recherche d'un coefficient de pénalisation

Coéfficient générateur	7	
Méthode	Converge vers	Nombre d'appel de f
Dual annealing	7.179	170
Optimisation bayésienne	7.007	100
Différential evolution	6.993	100
Gauss-Newton	8.191	7
Levenberg-Marquardt	9.140	5

Application au cas de la fonction d'évaluation de l'algorithme génétique

Recherche d'un coefficient de pénalisation

- Converge rapidement proche de la solution pour Dual Annealing, optimisation bayésienne et differential evolution

Chaque algorithme a été arrêté manuellement lorsque suffisamment proche de la solution : Une itération d'optimisation, c'est à dire 6 fois 2500 générations, prends entre 30 minutes et une heure pour ce jeux de données.

- Ne converge pas pour les méthodes de curve fitting, et s'arrête trop vite

Cela est dû au coté heuristique de l'algorithme génétique.

Application au cas de la fonction d'évaluation de l'algorithme génétique

Recherche d'un coefficient de pénalisation

Méthode	dav1	psa1	ret1
Valeurs observations	906481	16500	8926
Dual annealing	899535	17700	9425
Optimisation bayésienne	927113	17700	8680
Différential evolution	915529	16500	8958

dav2	psa2	ret2	dav3	psa3	ret3
331618	8400	1744	194852	9000	6914
337768	8400	3190	178235	9000	7900
337768	8400	3190	194852	9000	6914
331618	8400	1744	200716	9000	5922

Conclusion

- L'étude préliminaire à permis de retenir plusieurs méthodes d'optimisation
- L'application a la fonction d'évaluation pour un coefficient de pénalisation à déterminer à permis de retenir 3 méthodes
- Les tests sur quatre paramètres, non évoqués ici, ont permis de retenir uniquement Dual annealing, en gardant l'option de l'optimisation bayésienne

Le but de la fin du stage consiste à continuer les tests avec plus de coefficients, sur des données issues de clients dont on ne connait pas à l'avance la configuration de coefficients de pénalisation optimale.