
Optimisation par algorithme génétique

Influence du paramétrage

Swan Launay - Gabriel Vaubillon

Chapitre 1

Préambule

1.1 Remerciements

Ce rapport représente l’aboutissement du projet de 40 heures qui s’est déroulé entre octobre 2018 et avril 2019. Nous tenons tout d’abord à remercier Gilles Fraisse pour son soutien, sa pédagogie, mais aussi pour le temps qu’il a passé à nous accompagner.

Nous tenons aussi à remercier l’Université Savoie Mont Blanc (USMB) pour la mise à disposition des documents nécessaires à la réalisation de ce projet par le biais de la bibliothèque universitaire.

Nous remercions enfin Polytech Annecy - Chambéry pour nous avoir permis d’effectuer ce projet sur notre temps de travail universitaire et plus globalement pour nous avoir proposé un travail de ce type.

1.2 Résumé

L’optimisation par algorithme génétique permet d’obtenir de bonnes approximations de résolutions pour différents problèmes (avec un ou plusieurs objectifs). De façon générale, les algorithmes génétiques sont construits de telle façon à ce que l’on puisse faire varier certains paramètres. C’est ces mêmes paramètres qui vont déterminer la qualité et la fiabilité du résultat, mais aussi qui vont faire varier de façon plus ou moins significative le temps de résolution. Il s’agit alors de trouver un juste milieu entre le temps résolution et la fiabilité du résultat.

Chapitre 2

Introduction

Selon la *Théorie de l'évolution* [8] de Charles Darwin, l'évolution des espèces est issue de mutations aléatoires qui surviennent lors de la conception d'une nouvelle génération d'individus. Si cette mutation permet à l'individu d'être plus adapté à son milieu de vie, alors il aura plus de chance d'atteindre l'âge adulte et d'engendrer une nouvelle génération. Les algorithmes génétiques sont une application directe du darwinisme à l'informatique.

De nos jours, ces algorithmes permettent de résoudre des problèmes d'optimisation, c'est à dire de maximiser ou minimiser numériquement une ou plusieurs fonctions. La branche des mathématiques qu'est l'optimisation est un élément central dans le monde de l'ingénierie où on cherche dans de très nombreux cas à minimiser plusieurs problèmes à la fois à obtenir une solution de qualité dans un laps de temps court.

Ce rapport présente deux types d'algorithmes génétiques différents, le premier est adapté à une optimisation mono-objectif (un seul problème), alors que l'autre l'est pour des problèmes multiobjectifs. Pour chacun des deux algorithmes nous détaillerons à la fois le principe de résolution, mais aussi l'influence du paramétrage.

Table des matières

1	Préambule	I
1.1	Remerciements	I
1.2	Résumé	I
2	Introduction	II
3	Quelques définitions	IV
4	Optimisation à objectif unique	1
4.1	Algorithme	1
4.1.1	Principe	1
4.2	Optimisation unidimensionnelle / multidimensionnelle	2
4.3	Influence du paramétrage	2
4.3.1	Protocole	2
4.3.2	Taille de la population	2
4.3.3	Probabilité de croisement	4
4.3.4	Probabilité de mutation	4
5	Optimisation à objectifs multiples	7
5.1	Algorithme - NSGAII	7
5.1.1	Principe	7
5.2	Méthode de comparaison des résultats	7
5.3	Influence du paramétrage	8
5.4	NSGA III	8
6	Conclusion	9
A	Annexes	11
A.0.1	Programme	11
A.0.2	Figures	11

Chapitre 3

Quelques définitions

objectif : individus : solution optimale : pareto : pareto de référence : extremum local :

Chapitre 4

Optimisation à objectif unique

4.1 Algorithme

4.1.1 Principe

Hormis quelques variations dans leur conception, les algorithmes génétiques ont pratiquement tous le même principe de fonctionnement. Le logigramme ci-joint illustre le fonctionnement de base de l'algorithme d'un algorithme mono-objectif. On commence par générer de façon aléatoire une population de base qui servira d'initialisation de notre algorithme. Chaque individu de cette population est alors évalué, on lui attribue une valeur qui va déterminer sa position par rapport à la solution optimale recherchée (ici le minimum de la fonction). Une sélection est ensuite effectuée en gardant les meilleurs résultats, mais aussi en gardant un certain pourcentage de valeurs considérées comme de moyenne/faible qualité. Cette dernière opération permet entre autre à l'algorithme de ne pas tomber dans un extremum local. A partir de cette sélection, l'algorithme va procéder à deux étapes :

1. Le croisement (Crossover) : il consiste à partager (croiser) les particularités de deux individus parents pour engendrer une nouvelle population d'individus conservant ces mêmes particularités. A noter qu'il existe plusieurs méthodes de croisement (un point / deux points ... etc) qui vont déterminer le nombre de points de rotation dans l'individu (Voir figure 4.2). *Remarque : On retrouve cette opération en biologie sous le nom de "brassage génétique" qui illustre le Principe d'hérédité de la théorie de Darwin [8].*

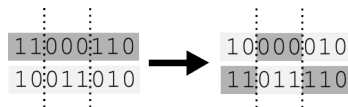


FIGURE 4.2 – Analogie binaire d'un croisement à 3 points de rotation

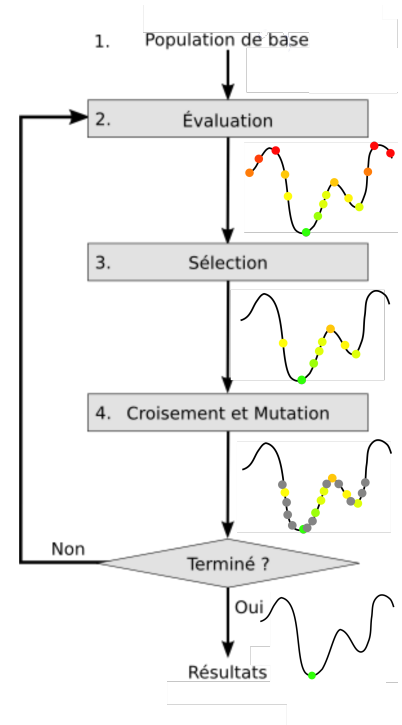


FIGURE 4.1 – Fonctionnement

2. La mutation : contrairement au croisement qui conserve les particularités des individus au cours des générations, la muta-

tion altère certains individus de manière aléatoire. Cette opération complète l'opération de sélection précédemment évoquée afin d'éviter de tomber dans un extremum local. *Remarque : cette opération illustre le Principe de variation de la théorie de Darwin en biologie [8].*

4.2 Optimisation unidimensionnelle / multidimensionnelle

Bien que l'optimisation mono-objectif signifie l'optimisation d'un seul problème, ce problème peut tout de même comporter plusieurs variables, on parle alors d'optimisation multidimensionnelle ($R^n \rightarrow R$). Il est possible de représenter graphiquement de tels problèmes jusqu'à une dimension égale à 2 (Voir figure 4.3).

4.3 Influence du paramétrage

Lors de son exécution, un algorithme génétique est soumis à un paramétrage préalablement défini par l'utilisateur. Ce paramétrage va agir sur le comportement et sur la capacité de l'algorithme à trouver des solutions de qualité. Dans le cadre de ce projet, c'est la l'influence de ce paramétrage qui nous intéresse, on cherche donc à faire varier les paramètres pour se rendre compte de l'influence qu'ils ont sur l'algorithme.

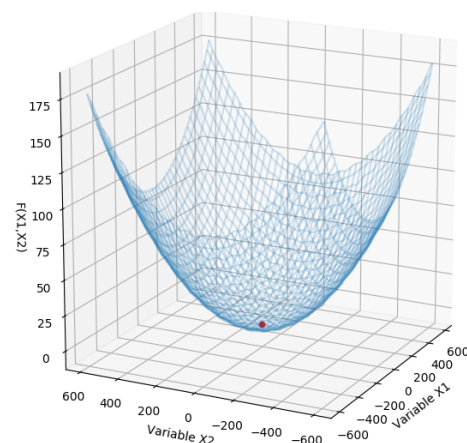


FIGURE 4.3 – Fonction Griewank [5] en bleu, solution de l'algorithme en rouge

4.3.1 Protocole

- Dans cette partie la fonction dont on cherche le minimum se nomme Ackley et est définie tel que :

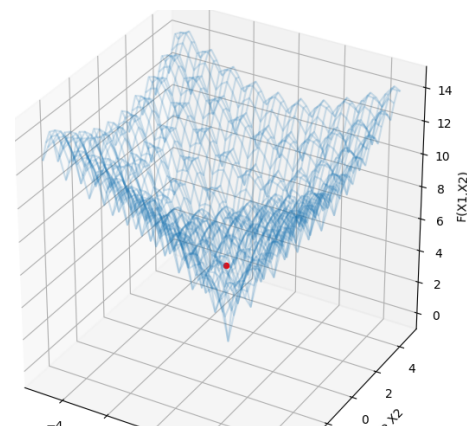
$$\begin{cases} \forall x, y \in [-5, 5], f(x, y) = -20 * e^{-0.2\sqrt{0.5(x^2+y^2)}} \\ \text{Solution : } f(0, 0) = 0 \end{cases}$$

- Un programme informatique a été réalisé pour faire varier ces paramètres automatiquement.

4.3.2 Taille de la population

L'exécution du programme pour la variation de la taille de la population retourne les graphiques des figures 4.5 et 4.6.

Le premier graphique (4.5) représente les données telles qu'elles sortent de notre algorithme, or bien que cette représentation soit précise et exacte, elle ne permet pas de tirer une conclusion quant à l'influence de ce paramètre. Afin d'éclaircir la figure on ramène un ensemble de points à une valeur moyenne de ces derniers, on obtient alors la figure



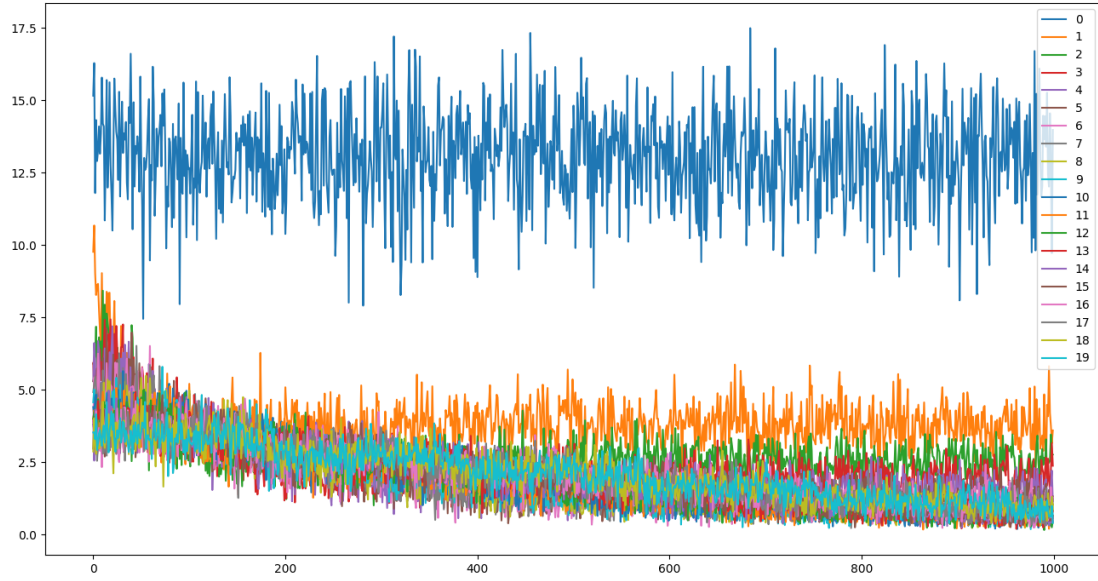


FIGURE 4.5 – Représentation de la solution au cours des générations - Variation de la taille de la population sur $[10, 2000]$ avec un pas de 10 - Autre paramètres : $P_{mutation} = 0.5, P_{crossover} = 0.5$

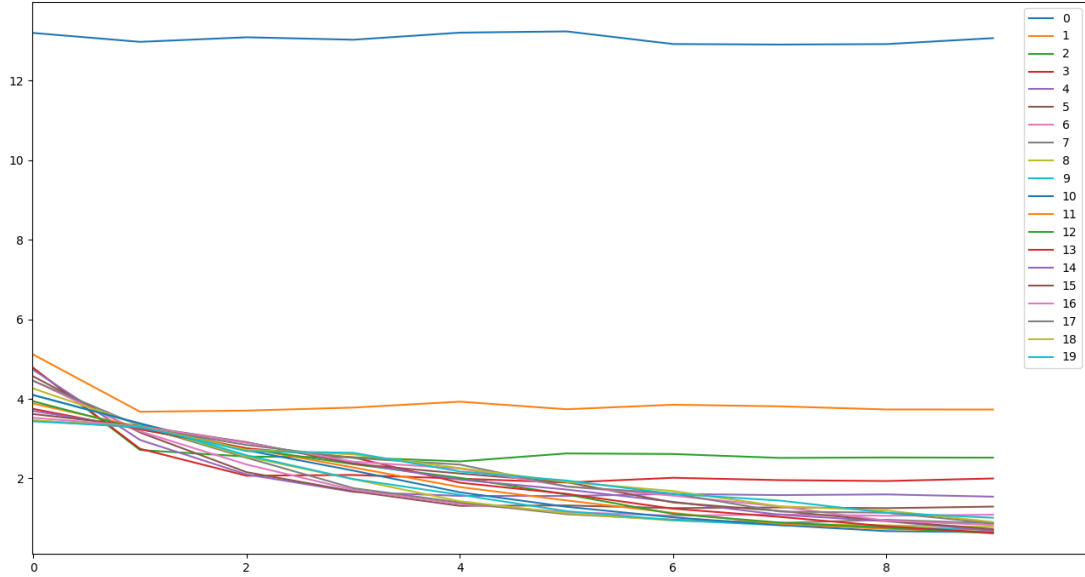


FIGURE 4.6 – Représentation "lissée" de la figure 4.5

4.6.

Dans un premier temps on remarque que la première courbe qui a été générée est largement au dessus des autres. Cette

courbe correspond à une taille de population de 10 alors que la suivante correspond à une population de 110. Dans les faits, quand la population est très faible (Moins de 100 individus) la convergence vers une solution de bonne qualité est très rapide, et donc n'est pas représentée de façon optimale sur ce graphique.

Si on met cette première courbe de côté, on remarque dans un second temps que lors des premières générations la convergence est semblable pour toutes les courbes, mais que plus on avance dans les générations, plus les courbes s'écartent.

4.3.3 Probabilité de croisement

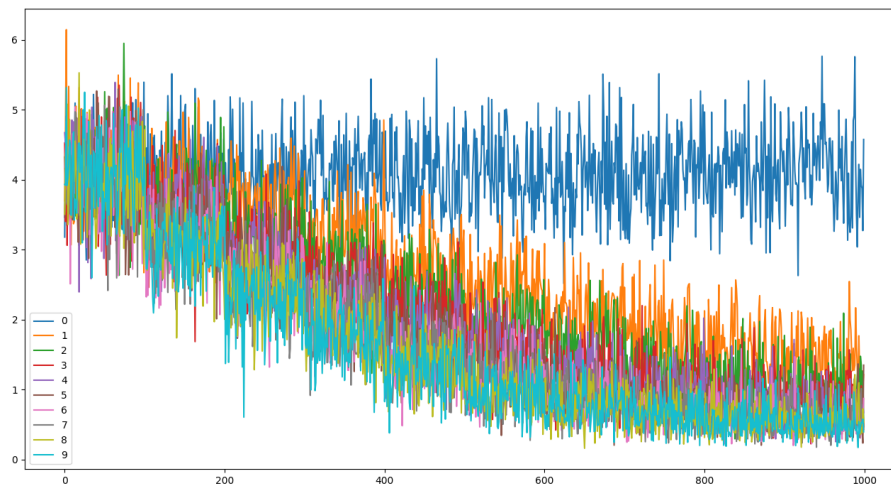


FIGURE 4.7 – Représentation de la solution au cours des générations - Variation de la taille de la population sur $[10, 2000]$ avec un pas de 10 - Autre paramètres : $P_{mutation} = 0.5, population = 1000$

4.3.4 Probabilité de mutation

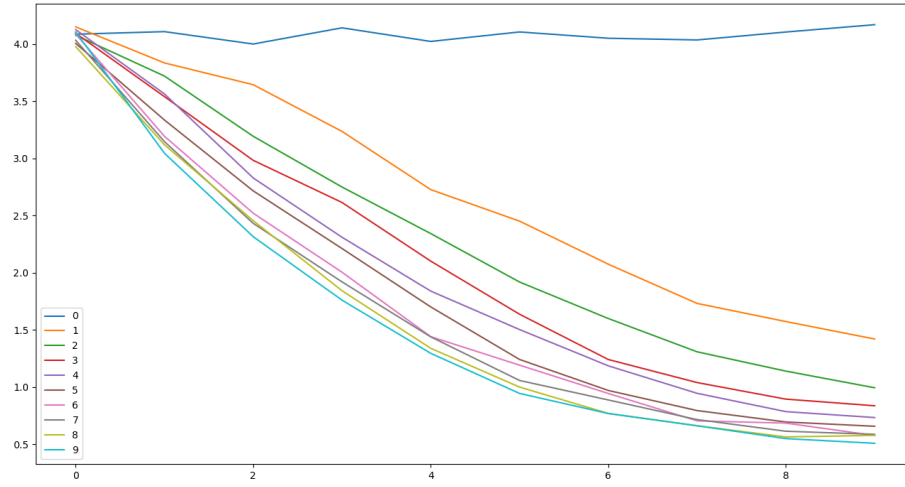


FIGURE 4.8 – Représentation "lissée" de la figure 4.5

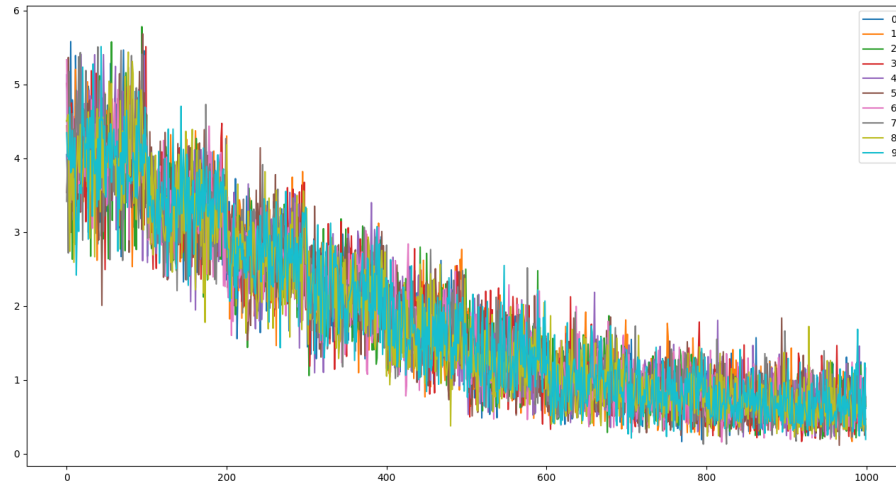


FIGURE 4.9 – Représentation de la solution au cours des générations - Variation de la taille de la population sur $[10, 2000]$ avec un pas de 10 - Autre paramètres : $population = 1000, P_{crossover} = 0.5$

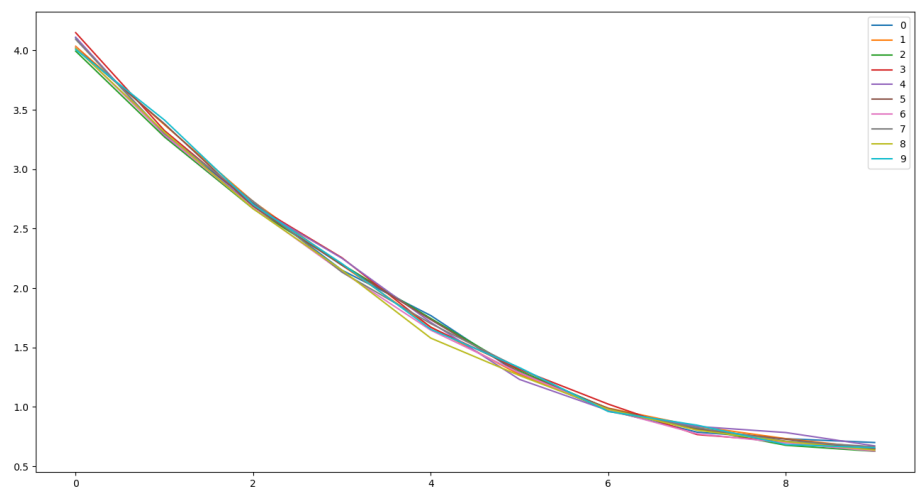


FIGURE 4.10 – Représentation "lissée" de la figure 4.5

Chapitre 5

Optimisation à objectifs multiples

5.1 Algorithme - NSGAII

L'algorithme d'optimisation NSGAII (Non dominated sorting genetic algorithm) est un algorithme connu et utilisé à l'échelle mondiale. + Multi obj

5.1.1 Principe

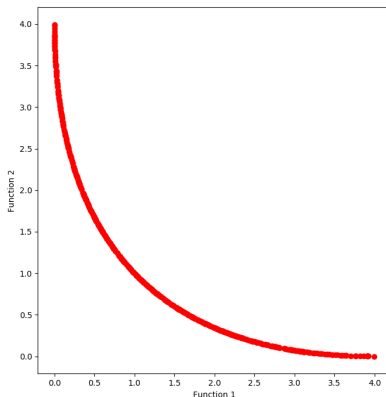


FIGURE 5.1 – Légende

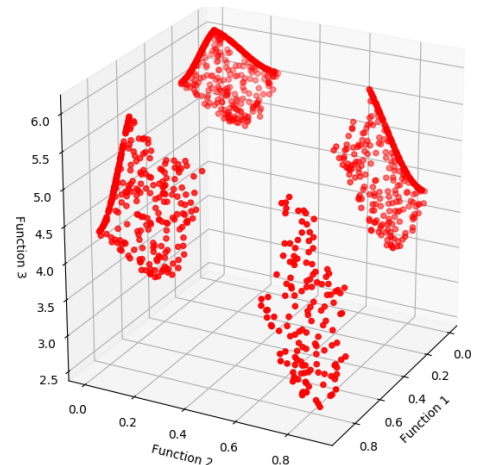
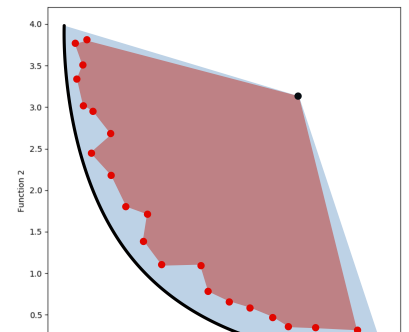


FIGURE 5.2 – Légende

+ Diff avec algo standard : celui la est élitiste

5.2 Méthode de comparaison des résultats

Le type solution recherché dans la cadre d'une optimisation multi-objectif n'est pas le même que dans un problème mono-objectif. Quant bien on cherchait une et une seule valeur (ou un n-uplet) avec le premier algorithme, l'objectif ici est de minimiser plusieurs fonction à la fois. Notre but final étant de d'étudier l'influence du paramétrage il est essentiel de pouvoir caractériser l'évolution de chaque algorithme. Dans le cas présent, pour un problème multi-objectif il est possible d'étudier l'évolution du



front de pareto géométriquement en calculant l'aire, le volume ou l'hyper-volume de ce dernier. Prenons un exemple simple dans un problème où l'on cherche à minimiser deux fonctions (Figure 5.3). La courbe noire représente le front de pareto de référence et les points rouges un front trouvé à l'aide de l'algorithme NSGAI (Peu d'individus et peu de générations). On pose un point sur le plan et on calcule les aires des front noir et rouge en tenant compte du point A. On obtient alors deux valeurs réelles comparables qui permettent d'étudier la convergence du front de pareto au cours des générations (Plus les valeurs sont semblables, plus la qualité du pareto est bonne).

5.3 Influence du paramétrage

5.4 NSGA III

Chapitre 6

Conclusion

+ Il existe d'autre algo utilisés : [http ://www.laas.fr/files/MOGISA/sem_Alain - Berro.pdf](http://www.laas.fr/files/MOGISA/sem_Alain - Berro.pdf) p.31 +
Il faut relier avec le metiering

Bibliographie

- [1] Algorithme génétique. fr.wikipedia.org/wiki/Algorithme_génétique.
- [2] Mathematical optimization. en.wikipedia.org/wiki/Mathematical_optimization.
- [3] Multi-objective optimization. en.wikipedia.org/wiki/Multi-objective_optimization.
- [4] Optimisation (mathématiques). [fr.wikipedia.org/wiki/Optimisation_\(mathématique\)](http://fr.wikipedia.org/wiki/Optimisation_(mathématique)).
- [5] Test functions for optimization. en.wikipedia.org/wiki/Test_functions_for_optimization.
- [6] Alain Berro. Algorithme évolutionnaire pour l'optimisation multiobjectif. www.laas.fr/files/MOGISA/sem_Alain-Berro.pdf.
- [7] Jean-Christophe Culioli. *Introduction à l'optimisation*. 2012.
- [8] Charles Darwin. *L'Origine des espèces*. 1862.
- [9] Kamal Mohammedi Idir Belaidi, Brahim Merdjadoui. *Un modèle sur les algorithmes génétiques et le front de pareto pour l'optimisation multi-objectif*. PhD thesis, Université M'Hamed Bougara de Boumerdes, 2007.
- [10] Michel Minoux. *Programmation mathématique*. 2007.

Annexe A

Annexes

A.0.1 Programme

L'ensemble des programmes qui ont été utilisé pour réaliser ce projet sont mis à disposition sur notre page GitHub :

<https://github.com/Akashita/TPE-NSGAI>

Ce travail est soumis à la license *GNU General Public License v3.0*
Pour plus de détails veuillez vous référer au site suivant :

<https://www.gnu.org/licenses/gpl-3.0.en.html>

A.0.2 Figures