

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان -

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Informatique

Spécialité : Système d'information et de connaissance(SIC)

Par : Heus Mohammed Housseem Eddine et Addoun Abdelhadi Hicham

Sujet

Optimisation de processus métier avec l'algorithme NSGA 2

Soutenu publiquement, le 28 / 09 /2020 , devant le jury composé de :

M.Abderaahim Alaedine

MCB

Encadreur

M.Belabed Amine

MCB

Président

M.Hadjila Fethallah

MCB

Examineur

Table des matières

I.1	Introduction.....	14
I.2	Optimisation monocritère	14
I.3	Optimisation multi objectifs.....	14
I.4	Processus métier.....	15
I.4.1	Qu'est-ce qu'un processus métier	15
I.4.2	Modélisation d'un processus.....	15
I.5	Heuristique	16
I.5.1	Définition	16
I.6	Métaheuristiques	17
I.6.1	Pourquoi passer au Métaheuristiques.....	17
I.6.2	Définition et principe	17
I.6.3	Classification des métaheuristiques	17
a.	Méta heuristique a parcours :	17
b.	Méta heuristique à population :	17
c.	Métaheuristiques à méthodes implicites :	17
d.	Métaheuristiques à méthodes explicites :	18
I.7	Les Algorithme évolutionnaires :	18
I.7.1	Définition	18
I.7.2	Description de mécanismes évolutionnaires de la littérature.....	19
I.8	Algorithme génétique.....	19
I.8.1	Principe et définition d'un algorithme génétique.....	19
I.8.2	Les opérateurs génétiques :	20
a.	La Sélection.....	20
b.	Le croisement.....	20
c.	La mutation	21
I.9	L'algorithme NSGA-II :	21
I.9.1	Principe de l'algorithme NSGA-II.....	21
I.9.2	Procédure de la recherche des solutions non dominées	22
I.9.3	Maintien de la diversité dans la population	23
I.10	Conclusion	23
II.1	Introduction.....	25
II.2	Sélections implémentable	25
II.2.1	Sélection proportionnelle :	25
a.	Sélection par roulette –RWS-	25
b.	Sélection universelle stochastique –SUS-.....	26
c.	Sélection d'échantillonnage à reste stochastique -RSS - :	26
d.	Sélection par rang :	26
II.2.2	Sélection par tournoi :	27
II.2.3	Sélection uniforme :	27

II.3	Les sélections naturelles non implémentées	28
II.3.1	La sélection massale.....	28
II.3.2	La sélection sur ascendance	28
II.3.3	La sélection sur descendance	29
II.3.4	La sélection sur collatéraux.....	29
II.4	La sélection par génération	30
II.5	Synthèse	30
II.6	Conclusion	34
III.1	Introduction.....	36
III.2	Partie I : Environnement de travail	36
III.2.1	Modèle mathématique pour la conception d'un processus métier	36
III.2.2	Matrice tâches ressources (task resources matrix - TRM).....	37
III.2.3	Librairie des taches	38
III.2.4	Fonction objective.....	38
III.2.5	Cas d'expérimentation	40
a.	Scénario de test	40
1)	Description du scénario : la prévision des ventes (Sales forecasting).....	40
2)	Librairie des tâches et ressources utilisées	41
III.3	Partie2 : Framework : BPO_Tool	43
III.3.1	Description du Framework : BPO_Tool	43
a.	Outil de développement	43
b.	Hiérarchie du Framework	43
c.	Librairies utilisées dans le Framework	44
d.	Machine d'exécution.....	45
e.	Fonctionnement du Framework	45
1)	Algorithme de la création de la population initiale	46
f.	Interfaces du BPO Tool	47
2)	Configurations	47
3)	Affichage des résultats (Plans).....	48
4)	Barre d'outils.....	49
5)	Les sorties de BPO Tool.....	49
III.3.2	Résultats et discussions	50
a.	Paramètres du Framework pour le cas d'expérimentation.....	50
b.	Résultats obtenus pour chaque sélection.....	51
1)	Sélection uniforme	51
2)	Sélection par tournoi	52
3)	Sélection par roulette.....	53
4)	Sélection par rang.....	54
5)	Sélection universelle stochastique.....	55
6)	Sélection d'échantillonnage & reste stochastique RSS :.....	56
7)	Sélection Massale	57
8)	Sélection sur ascendance	60

9)	Sélection sur descendance	61
10)	Sélection sur collatéraux	61
11)	Sélection par génération	62
12)	Récapitulation des résultats des sélections	62
c.	Comparaison entre les sélections (Discutions)	63
d.	Comparaison des résultats obtenus avec résultat de Vergidis.....	63
1)	Interprétation des résultats.....	66
III.4	Conclusion	67
IV.	Conclusion generale	Error! Bookmark not defined.
V.	Perspectives.....	69
VI.	Bibliographie.....	70

Les des tableaux

Tableau 1: Les principales méthodes de sélection naturelle	29
Tableau 2: synthèse finale	33
Tableau 3: Paramètres principaux utilisés pour la conception d'un processus.	37
Tableau 4: Exemple de TRM	37
Tableau 5 Exemple de librairie des taches	38
Tableau 6 Librairie des services web utilisés	42
Tableau 7 Ressources disponibles pour le scénario test.	42
Tableau 8 Description des packages	44
Tableau 9 Performance du machine	45
Tableau 10 description de (interface de configuration)	48
Tableau 11 description de l'interface du résultat	48
Tableau 12 Barre d'outils (Edit)	49
Tableau 13 Valeurs de paramètres pour le scénario du test	51
Tableau 14 Résultat de la sélection uniforme	51
Tableau 15 Résultat de la sélection par tournoi	52
Tableau 16 Résultat de la sélection par roulette	53
Tableau 17 Résultats de la sélection par rang	54
Tableau 18 Résultat de la sélection SUS	55
Tableau 19 Résultat de la sélection RSS	56
Tableau 20 Résultat de la sélection massale	57
Tableau 21 Résultat de la sélection sur assendance	60
Tableau 22 Résultats de la sélection sur descendance	61
Tableau 23 Résultats de la sélection sur collatéraux	61
Tableau 24 Résultats de la sélection par générations	62
Tableau 25 résultats de toutes les sélections	62
Tableau 26 premier resultat	63
Tableau 27 deuxième résultat	64

Liste des figures

Figure 1: Relation entre kilométrage et prix (Adel, 2009)	15
Figure 2 : Méta-modèle d'un processus métier (Hafedh Mili, 2004)	16
Figure 3 Classification des Métaheuristiques ¹	18
Figure 4 : Fonctionnement général d'un algorithme évolutionnaire (Mais HAJ-RACHID, 2010)	18
Figure 5 : schéma général des algorithmes génétiques (Kheireddine, 2012)	19
Figure 6 Représentation de l'opérateur de sélection	20
Figure 7: Croisement de deux parents (Dipama, 2010)	21
Figure 8: Mutation d'un gène. (Dipama, 2010)	21
Figure 9: Algorithme NSGA-II. (Deb, 2002)	21
Figure 10: Procédure de la recherche des solutions non dominées	22
Figure 11: Distance de surpeuplement (Deb, 2002)	23
Figure 12 : sélection par roulette (Goldberg, 1989)	25
Figure 13: Sélection universelle stochastique (Baker, 1987)	26

Figure 14 Sélection par rang	27
Figure 15 Sélection par tournoi	27
Figure 16: Types de sélections	28
Figure 17: Exemple de TRM basée sur la tâche.....	37
Figure 18: Représentation du rayon d'un cercle circonscrit du triangle CTS.....	38
Figure 19 Conception initiale des processus métier de la prévision des ventes.....	41
Figure 20 Packages du Framework	43
Figure 21 Librairies utilisées	44
Figure 22 Déroulement du BOP_Toot	46
Figure 23 Organigramme de l'algorithme de la création de la population initiale.....	47
Figure 24 interface de la configuration	47
Figure 25 Interface du résultat	48
Figure 26 Barre d'outils (Edit).....	49
Figure 27 Fichier XML d'un résultat.....	49
Figure 29 Conception du résultat	50
Figure 28 Ratio de solutions.....	50
Figure 30 Temps de calcul	50
Figure 31 Graphe de toutes les générations.....	51
Figure 32 Graphe des solutions pour la sélection uniforme.....	51
Figure 33 Conception de la meilleure solution pour la sélection uniforme	52
Figure 34 Graphe de toutes les générations.....	52
Figure 35 Graphe des solutions pour la sélection par tournoi.....	52
Figure 36 Conception de la meilleure solution (845-446) du tournoi.....	53
Figure 37 Conception de la meilleure solution (852-448) du tournoi.....	53
Figure 38 Graphe de solution de la sélection par roulette.....	54
Figure 39 Population de toutes les générations	54
Figure 40 Conception de la meilleure solution pour la sélection uniforme	54
Figure 41 Graphe de toutes les générations.....	55
Figure 42 Graphe des solutions pour la sélection par rang	55
Figure 43 Conception de la meilleures solution de la sélection par rang.....	55
Figure 44 Graphe des solutions pour la sélection SUS	56
Figure 45 Graphe de toutes les générations.....	56
Figure 46 Conception de la meilleure solution pour la sélection SUS.....	56
Figure 47 Graphe des solutions pour la sélection RSS	57
Figure 48 Graphe de toutes les générations.....	57
Figure 49 Conception de la meilleure solution pour la sélection RSS.....	57
Figure 50 Graphe des solutions pour la sélection massale.....	58
Figure 51 Graphe de toutes les générations.....	58
Figure 52 Conception de la meilleure solution (844-447) par massale.....	58
Figure 53 Conception de la meilleure solution (847-448) par massale.....	59
Figure 54 Conception de la meilleure solution (852-448) par massale.....	59
Figure 55 Conception de la meilleure solution (1053-554) par massale.....	60
Figure 56 Graphe des solutions pour la sélection sur ascendance	60
Figure 57 Graphe de toutes les générations.....	60
Figure 58 Graphe de toutes les générations.....	61
Figure 59 Graphe des solutions pour la sélection sur descendance	61
Figure 60 Graphe des solutions de la sélection sur collatéraux	61
Figure 61 Graphe de toutes les générations.....	61
Figure 62 Graphe de toutes les générations.....	62
Figure 63 Graphe des solutions pour la sélection par génération.....	62

Figure 75 Solution 1 Vergidis	64
Figure 76 Solution à 4 taches avec sélection Massale BPO Tool	65
Figure 77 Solution 2 Vergidis	65
Figure 78 Solution à 5 taches avec la sélection RSS avec BPO Tool	66

Remerciements

*Nous rendons grâce à **Dieu** tout puissant de nous avoir donné le courage et la volonté de mener à bien ce modeste travail.*

*Nous tenons à exprimer notre reconnaissance à notre encadreur **MONSIEUR Abderrahim Allaa Eddine** pour ses précieux conseils, sa disponibilité et ses encouragements tout au long de ce travail de mémoire.*

Nous adressons nos vifs remerciements à tous nos professeurs, ainsi qu'à tous les intervenants qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions sur ce projet.

Nous remercions nos chers parents qui ont toujours été là pour nous, ainsi que nos frères et sœurs pour leurs encouragements.

Notre gratitude va également aux membres du jury pour avoir bien voulu consacrer de leurs temps pour apprécier ce travail.

Et enfin, nous remercions nos amis et collègues pour leur support moral et intellectuel tout au long de notre cursus universitaire.

Merçi à toutes et à tous

Je dédie ce mémoire

À mes parents pour l'éducation qu'ils m'ont prodiguée, avec tous les moyens et au prix de tous les sacrifices qu'ils ont consentis à mon égard, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.

Je prie le Bon Dieu pour veiller sur eux en espérant qu'ils seront toujours fiers de moi.

À mes sœurs et frères.

Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

À la mémoire de mon frère Amine, tu seras toujours présent dans mon cœur.

À tous mes neveux, vous avez donné de la couleur à ma vie.

À tous mes professeurs, leurs générosités et leurs soutiens m'obligent à leur témoigner mon profond respect et ma loyale considération.

À mes amies SALAH EDDINE et ADDOUN HICHAM pour tous les moments inoubliables que nous avons passés ensemble.

À tous mes amis et collègues, à qui je souhaite bonne chance dans leur vie professionnelle.

À toute ma famille ; petits et grands, veuillez trouver dans ce modeste travail l'expression de mon affection.

Heus Mohammed Houssemeddne

Dédicaces

Rien n'est aussi beau à offrir que le fruit d'un labeur qu'on dédie du fond du cœur à ceux qu'on aime en exprimant toute notre gratitude et notre reconnaissance.

Je dédie ce modeste travail :

À mes chers parents, pour les sacrifices qu'ils ont consentis pour mon instruction et mon bien-être. Aucune dédicace ne saurait exprimer mon amour éternel et ma reconnaissance. Que ce travail soit l'accomplissement de vos vœux tant allégués et le fruit de votre soutien infailible. Merci MAMAN et PAPA pour tout. Que Dieu vous accorde santé, bonheur, une longue vie et faire en sorte que jamais je ne vous déçoive.

À mon frère et ma sœur en témoignage de mon affection fraternelle, je vous souhaite une vie pleine de bonheur. Je ne saurais guère exprimer l'amour que j'ai pour vous. Votre joie, votre réussite, votre gaieté me comblent de bonheur. Merci pour votre présence et votre soutien.

À ma chère fiancée ; je te dédie ce travail en témoignage de mon profond amour, merci pour tes conseils, ton aide et tes encouragements.

À mes amis Achir Youcef, Heus Housseem-Eddine, Benaissa Boumedianne, Bendiabdellah Mohammed, en témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de bonheur.

À toute la famille ADDOUN et SAOUDI ainsi qu'à tous mes amis de la promotion 2019/2020.

Addoun Abdelhadi Hicham

Introduction générale

Dans le monde de l'entreprise moderne, vue le marché changeant et la compétitivité croissante, un besoin fréquent est ressenti par l'entreprise à modifier la structure de leurs processus métiers pour faire des profits et même pour subsister. Ainsi, il ressort que la conception et la gestion des processus métiers soit un facteur clé pour booster l'efficacité et le côté concurrentiel de l'entreprise d'aujourd'hui.

L'amélioration des refontes des processus métiers est devenue un sujet de plus en plus attrayant, dans le large domaine de *business process intelligence*, ainsi que le *business process management*. Bien qu'il existe de nombreuses tentatives pour arriver à une optimisation réelle et efficace des processus métiers, rare sont les travaux ayant atteint des objectifs satisfaisants, et exploitables. Autre point digne d'intérêt est la nature informelle et semi automatisée des approches et autres méthodes appliquées sur la problématique de l'optimisation des processus métiers

Dans notre travail nous allons proposer :

Une approche pour l'optimisation multicritères automatisés des processus métiers. Le Framework proposé utilise un modèle mathématique de processus métier. Le modèle de processus métier est programmé et intégré dans notre Framework d'optimisation sur lequel est appliqué un algorithme évolutionnaire, dans notre cas nous allons utiliser l'algorithme NSGA II ce dernier aura comme mission de générer de nouvelles solutions

Enfin, pour valider la fiabilité de notre Framework (BOP_Tool), nous utilisons un cas d'étude réel. Nous reprenons celui proposé par (Kostas Vergidis, 2013) et comparons les résultats obtenus.

Problématique :

Allons nous obtenir des résultats plus fiables ?

La suite du mémoire se présente en trois chapitres :

Chapitre I : Cette partie servira à expliquer les différentes définitions et concepts utilisés dans notre travail de recherche.

Chapitre II : Cette partie se focalisera sur « la sélection », comme opérateur génétique dans l'algorithme évolutionnaire utilisé. Nos recherches nous ont conduites à recenser un nombre de sélection, que ce soit dans les algorithmes génétiques que dans le domaine de la recherche en biologie. Une synthèse sera introduite à la fin de cette partie.

Chapitre III : Le dernier chapitre consistera à présenter en détail notre framework et de décrire ses étapes fonctionnement. Nous présenterons par la suite le cas d'étude à tester, les résultats obtenus puis une discussion sur ces résultats.

Nous finirons ce travail avec une conclusion et les multiples perspectives s'ouvrant à nous, dans ce domaine de recherche.

Chapitre I

Optimisation multicritères des processus
métier avec les algorithmes génétiques.

I.1 Introduction

Résoudre un problème d'optimisation consiste à trouver la ou les meilleures solutions avec un ensemble de contraintes et/ou d'objectifs définis par l'utilisateur. Pour déterminer si une solution est meilleure qu'une autre, il est nécessaire que le problème introduise un critère de comparaison. Ainsi, la meilleure solution, appelée aussi solution optimale, est la solution ayant obtenu la meilleure évaluation au regard du critère défini.

Les problèmes d'optimisation sont des problèmes NP-Hard et donc ne possèdent pas, à ce jour, un algorithme général permettant de les résoudre en un temps polynomial. Ce problème de l'explosion combinatoire limite l'utilisation de méthodes exactes pour la résolution des problèmes de petite taille. Dans le cas de problèmes de grande taille, comme cela est souvent le cas dans les applications réelles, les méthodes approchées, qui sacrifient la complétude pour gagner l'efficacité, deviennent une alternative intéressante.

Les métaheuristiques d'inspiration biologique sont à l'origine de la plupart des métaheuristiques présentées ces dernières années, dans la littérature pour résoudre des problèmes combinatoires. Parmi ces métaheuristiques, nous pouvons citer les algorithmes génétiques, les réseaux de neurones, le recuit simulé et l'optimisation par colonie de fourmis.

Dans ce travail, nous nous intéressons à l'optimisation multi objectifs avec les algorithmes génétiques, pour résoudre le problème de l'optimisation des processus métier.

I.2 Optimisation monocritère

Résoudre un problème d'optimisation consiste à trouver une solution qui minimise ou maximise un critère particulier. Dans la plupart des cas, l'optimum découvert n'est pas unique. Ainsi il existe un ensemble de solutions minimisant ou maximisant le critère considéré. Nous pouvons décrire formellement un problème d'optimisation comme suit :

Considérons un problème de minimisation :

Soit $f: R^n \rightarrow R$

Et soit X un ensemble fermé de R^n

Alors l'ensemble des minimiseurs est : $\check{X} = \{\vec{x} \in X | \forall \vec{x}' \in X, f(\vec{x}') \geq f(\vec{x})\}$

Et le minimum de problème est : $f(\hat{x})$

L'ensemble X représente l'ensemble des solutions potentielles du problème. Cet ensemble est déterminé à l'aide de contraintes (souvent analytiques) données dans l'énoncé du problème. (Adel, 2009).

I.3 Optimisation multi objectifs

La plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs ou critères souvent contradictoires devant être optimisés simultanément. Alors que, pour les problèmes n'incluant qu'un seul objectif, l'optimum cherché est clairement défini, celui-ci reste à formaliser pour les problèmes d'optimisation multi objectifs. En effet, pour un problème à deux objectifs contradictoires, la solution optimale cherchée est un ensemble de points correspondant aux meilleurs compromis possibles pour résoudre notre problème.

Prenons le cas d'une personne souhaitant acheter une voiture d'occasion. La voiture idéale est celle qui est peu chère avec peu de kilomètres, Notre acheteur va donc devoir identifier les meilleurs compromis possibles correspondant a son budget (voir figure 1).

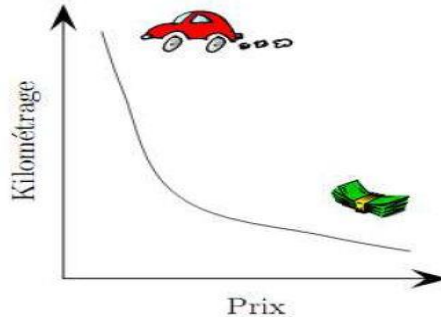


Figure 1: Relation entre kilométrage et prix (Adel, 2009).

Les problèmes d'optimisation multi-objectifs sont une généralisation à n fonctions objectif des problèmes d'optimisation classiques. Ils sont définis formellement comme suit :

Considérons un problème de minimisation :

Soit $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$

Et soit X un ensemble fermé de \mathbb{R}^n

Alors l'ensemble de minimiser est :

$$\hat{X} = \{\vec{x} \in X | \forall \vec{x}' \in X, (\exists i \in [1, \dots, m] f_i(\vec{x}) < f_i(\vec{x}')) \vee (\forall i [1, \dots, m] f_i(\vec{x}) = f_i(\vec{x}'))\}$$

Et le minimum de problème est : $f(\hat{x})$

D'après cette définition, il est clair que l'optimum n'est plus une simple valeur comme pour les problèmes a un objectif, mais un ensemble de points, appelé l'ensemble des meilleurs compromis ou le front Pareto (Adel, 2009).

I.4 Processus métier

I.4.1 Qu'est-ce qu'un processus métier

Un processus métier est un enchaînement d'activités à exécuter pour réaliser un objectif de l'entreprise. Cet enchaînement forme ce qu'il est convenu d'appeler le flux de contrôle du processus, c'est à dire sa logique d'exécution (Ferchichi, 2008).

I.4.2 Modélisation d'un processus

Il existe plusieurs méthodes pour modéliser un processus métier grâce (Bentellis, 2010). La représentation d'un processus métier est utile pour trois raisons :

Décrire un processus : La modélisation d'un processus permet de le décrire et de le documenter. Cette description peut avoir plusieurs cibles pour :

1. Des humains : dans ce cas la compréhension est importante.

2. Des machines : dans ce cas le formalisme de modélisation est plus important.

Analyser un processus : L'analyse d'un processus consiste en l'évaluation de ses propriétés. L'amélioration des processus se base sur l'analyse des processus existants pour identifier les étapes redondantes ou non optimales (Manolis Koubarakis, 2001).

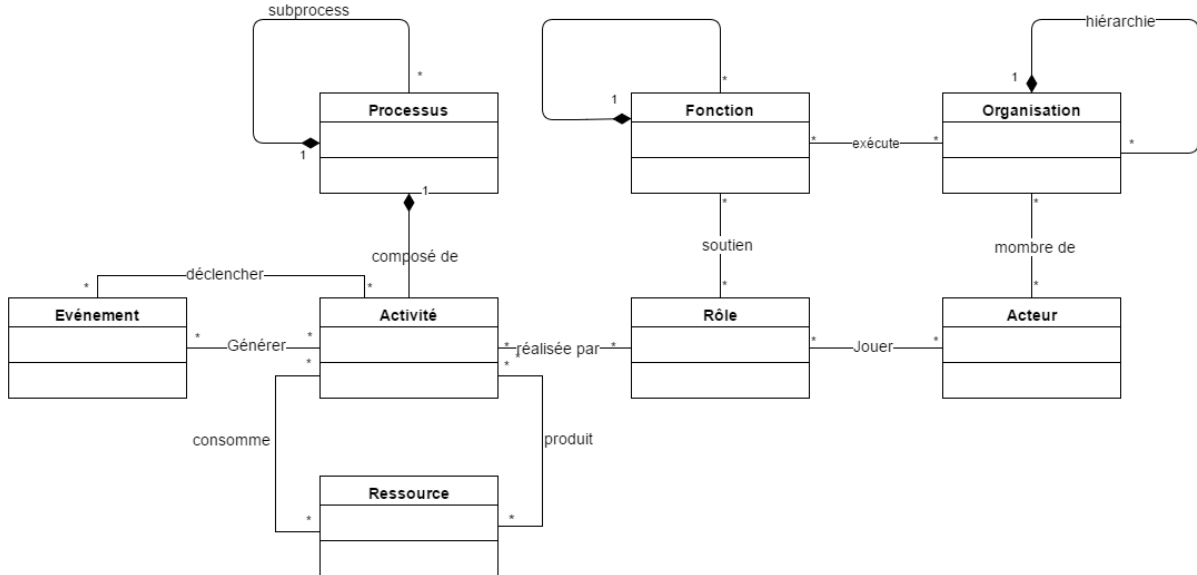


Figure 2 : Méta-modèle d'un processus métier (Hafedh Mili, 2004)

Les **activités** d'un processus sont exécutées par des **acteurs** jouant des rôles particuliers, consommant quelques ressources et produisant d'autres. Les activités peuvent être déclenchées par des **événements** et peuvent à leur tour produire des événements. Les activités d'un processus peuvent être liées par des dépendances de ressource (dépendances de producteur-consommateur) ou des dépendances de commande (une activité déclenchant une autre). Les acteurs opèrent à l'intérieur des frontières des organisations qui exécutent des fonctions métiers spécif. Les **rôles** peuvent soutenir des fonctions.

I.5 Heuristique

I.5.1 Définition

En optimisation combinatoire, une heuristique est un algorithme approché qui permet d'identifier en temps polynomial au moins une solution réalisable rapide, pas obligatoirement optimale. L'usage d'une heuristique est efficace pour calculer une solution approchée d'un problème et ainsi accélérer le processus de résolution exacte. Généralement une heuristique est conçue pour un problème particulier, En s'appuyant sur sa structure propre sans offrir aucune garantie quant à la qualité de la solution calculée. Les heuristiques peuvent être classées en deux catégories :

- Méthodes constructives qui génèrent des solutions à partir d'une solution initiale en essayant d'en ajouter petit à petit des éléments jusqu'à ce qu'une solution complète soit obtenue,
- Méthodes de fouilles locales qui démarrent avec une solution initialement complète (probablement moins intéressante), et de manière répétitive essaie d'améliorer cette solution en explorant son voisinage [Cours, Sidi Mohamed Douiri]

I.6 Métaheuristiques

I.6.1 Pourquoi passer au Métaheuristiques

Face aux difficultés rencontrées par les heuristiques pour avoir une solution réalisable de bonne qualité pour des problèmes d'optimisation difficiles, les métaheuristiques ont fait leur apparition. Ces algorithmes sont plus complets et complexes qu'une simple heuristique, et permettent généralement d'obtenir une solution de très bonne qualité pour des problèmes issus des domaines de la recherche opérationnelle ou de l'ingénierie dont on ne connaît pas de méthodes efficaces pour les traiter ou bien quand la résolution du problème nécessite un temps élevé ou une grande mémoire de stockage.

I.6.2 Définition et principe

Plusieurs définitions ont été proposées pour expliquer clairement ce qu'est un métaheuristique :

1. Un processus itératif qui subordonne et qui guide une heuristique, en combinant intelligemment plusieurs concepts pour explorer et exploiter tout l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions optimales, ou presque-optimales. (H. Osman, 1996).
2. Un métaheuristique est un processus itératif maître qui guide et modifie les opérations d'heuristiques subordonnés et produit efficacement des solutions de haute qualité. Il peut manipuler une solution unique complète (ou incomplète) ou une collecte des solutions à chaque itération (S. Voß, 1999).

I.6.3 Classification des métaheuristiques

Différentes classifications existent, selon le type, la recherche, etc. (Figure 3). Les métaheuristiques peuvent être classées selon leur fonctionnement :

a. Métaheuristique à parcours :

Les métaheuristiques les plus classiques sont ceux fondés sur la notion de parcours. Dans ce type d'algorithme ce dernier fait évoluer une seule fonction objective sur l'espace de recherche local à chaque itération puis la compare aux optimums. (Siarry, 2004).

b. Métaheuristique à population :

Dans cette famille les métaheuristiques utilisent la notion de population : Ils manipulent un ensemble de solutions en parallèle. Chaque élément de la population parcourt un certain nombre de solutions dans l'ensemble local. Parmi les algorithmes inclus dans cette classification : les algorithmes génétiques et les algorithmes de colonie de fourmis. (Siarry, 2004).

c. Métaheuristiques à méthodes implicites :

L'utilisation des méthodes implicites, avec les algorithmes génétiques, la distribution de probabilité n'est pas connue ou n'utilise pas le choix de l'échantillonnage entre deux itérations et ne suit pas une loi donnée. (Cheloud, 1999).

d. Métaheuristiques à méthodes explicites :

Ces méthodes utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution, comme leur nom l'indique, il estime chacune de leur itération, via une distribution de probabilité l'espace de recherche local optimal (Cheloud, 1999).

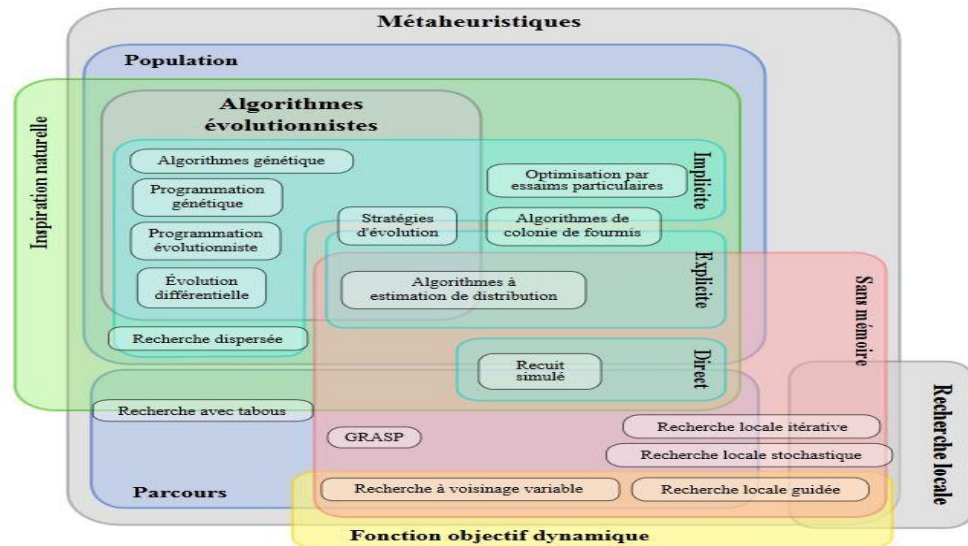


Figure 3 Classification des Métaheuristiques ¹

I.7 Les Algorithmes évolutionnaires :

I.7.1 Définition

Les Algorithmes évolutionnaires sont des méthodes d'optimisation stochastique basées sur une simulation brute de l'évolution naturelle des populations. Précisément, ce sont des techniques de programmation qui s'inspirent du principe de l'évolution des espèces décrit par Darwin. Le principe de base bien que simple, il n'en demeure pas moins puissant : il s'inspire de la théorie de Darwin sur l'évolution des espèces qui explique comment depuis l'apparition de la vie les espèces ont su évoluer de façon innovante et souple dans le sens d'une meilleure adaptation à l'environnement, tout en permettant aux seuls individus bien adaptés à leur environnement de se reproduire (Th. Back, 1997)

La figure (4) montre un schéma général de fonctionnement d'un AE.

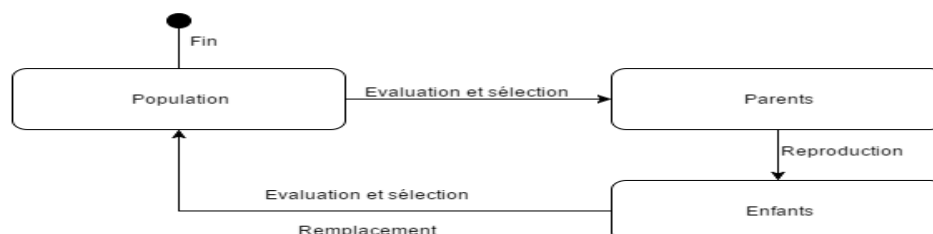


Figure 4 : Fonctionnement général d'un algorithme évolutionnaire (Mais HAJ-RACHID, 2010)

¹ https://commons.wikimedia.org/wiki/File:Metaheuristics_classification_fr.svg?uselang=fr

I.7.2 Description de mécanismes évolutifs de la littérature

Le but de cette section est de décrire plusieurs types d'éléments évolutifs rencontrés dans la littérature. (Mais HAJ-RACHID, 2010)

- Cinq types de codage.
- Deux types de création de population initiale.
- Trois types d'évaluation des individus.
- Six types de sélection des parents.
- Vingt-deux types de croisement.
- Six types de mutation.
- Quatre types de stratégie de remplacement pour la constitution de la génération suivante.
- Trois types de critères d'arrêt.

I.8 Algorithme génétique

I.8.1 Principe et définition d'un algorithme génétique

Un algorithme génétique (AG) est une métaheuristique qui manipule une population de solutions potentielles à la fois. Le mode de fonctionnement d'un AG est calqué sur les principes biologiques de la sélection naturelle et de la survie des individus les mieux adaptés à l'environnement (Darwin, 1876). La sélection naturelle est basée sur l'idée que les modifications des générations successives sont orientées par les pressions extérieures auxquelles sont soumises les espèces (e.g. la limitation des ressources, les modifications de l'environnement, les prédateurs et parasites, etc.). Il en résulte que les individus les mieux adaptés à l'environnement tendent à survivre plus longtemps et à se reproduire plus fréquemment. S'inspirant ainsi de ce mécanisme, (Holland, 1975) a posé les bases de la technique d'optimisation appelée "algorithmes génétiques". Mais c'est (Goldberg, 1989) qui par la suite s'est investi dans l'étude des AG et a développé la forme actuelle que nous connaissons.

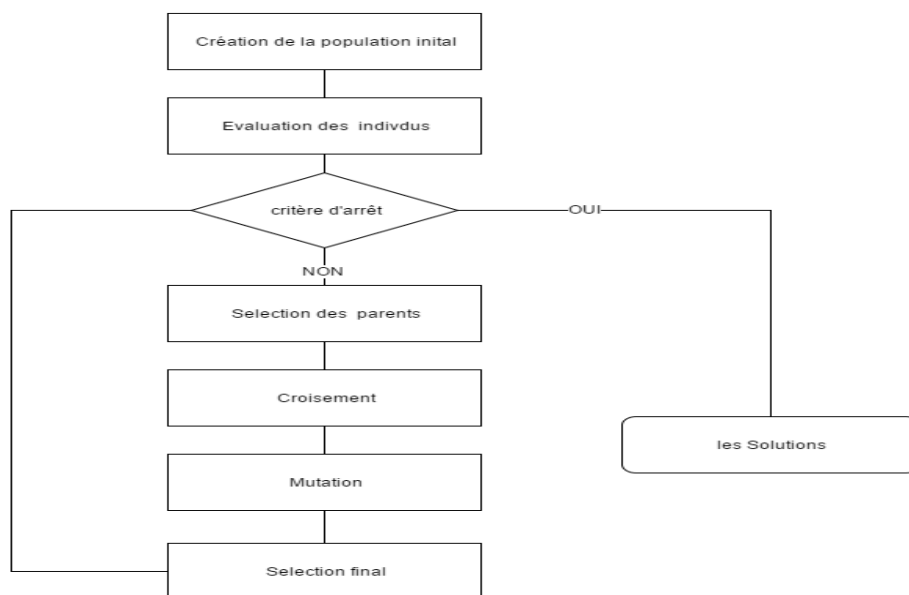


Figure 5 : schéma général des algorithmes génétiques (Kheireddine, 2012)

I.8.2 Les opérateurs génétiques :

Les AG utilisent principalement trois types d'opérateurs génétiques qui sont :

Le "*croisement*", la "*mutation*" et la "*sélection*". Ces opérateurs sont appliqués aux individus de la population avec l'espoir d'améliorer leurs caractéristiques génétiques et de les amener à s'adapter à leur environnement.

a. La Sélection

Cet opérateur est chargé de définir quels seront les individus de P qui vont être dupliqués dans la nouvelle population P' et vont servir de parents (application de l'opérateur de croisement).

Soit n le nombre d'individus de P, on doit en n/2 (l'opérateur de croisement nous permet de repasser à n individus).

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population (A.Spallanzani, 1999)

LA (figure 06) était créé à-partir du texte précédent pour montrer le principe de la sélection d'une façon facile



Figure 6 Représentation de l'opérateur de sélection

b. Le croisement

Une fois certains individus sélectionnés, on les fait se reproduire entre eux, pour cela on utilise l'opérateur croisement. C'est l'opérateur essentiel de recherche d'un algorithme génétique. Il combine les génotypes de deux individus pour en obtenir deux nouveaux. Avec cet opérateur, les génotypes sont vus comme une chaîne de nombres binaires. (Dejong & Spears, 1992) La figure 07 montre un exemple de croisement de deux parents codés en binaire. Un site de croisement est déterminé aléatoirement dans le chromosome des deux parents et deux nouveaux chromosomes enfants sont alors créés en échangeant tous les bits compris dans les portions avant le site de croisement.

L'application du croisement s'effectue suivant la probabilité (c p) définie par l'utilisateur, et qui indique si deux parents sélectionnés aléatoirement subiront le croisement.

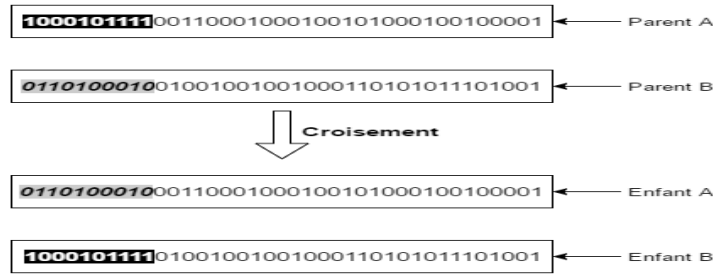


Figure 7: Croisement de deux parents (Dipama, 2010)

c. La mutation

Elle est généralement appliquée après le croisement des parents et aide à fournir de nouvelles caractéristiques génétiques, difficiles à obtenir par le seul opérateur de croisement. La figure 8 illustre le principe de la mutation d'un gène. Elle consiste à altérer un ou plusieurs gènes du chromosome d'un enfant donné, introduisant de ce fait une diversité dans la structure de la population. Cet opérateur permet ainsi l'exploration de l'espace des solutions. Il est caractérisé par la probabilité de mutation (mp) qui détermine si un enfant doit subir une mutation.

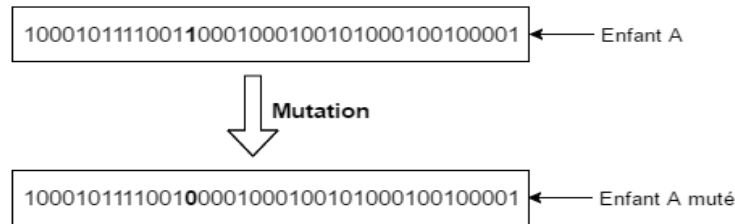


Figure 8: Mutation d'un gène. (Dipama, 2010)

I.9 L'algorithme NSGA-II :

L'algorithme NSGA-II ("*Non-dominated Sorting Genetic Algorithm*") présenté par (Deb, 2002) est un des algorithmes évolutifs les plus cités dans la littérature. Il est largement utilisé par plusieurs auteurs pour servir de base de comparaison avec leurs propres algorithmes. Nous donnerons dans les sections qui suivent une description détaillée du NSGAI qui servira de base pour le développement de l'algorithme proposé dans le Framework de notre travail.

I.9.1 Principe de l'algorithme NSGA-II

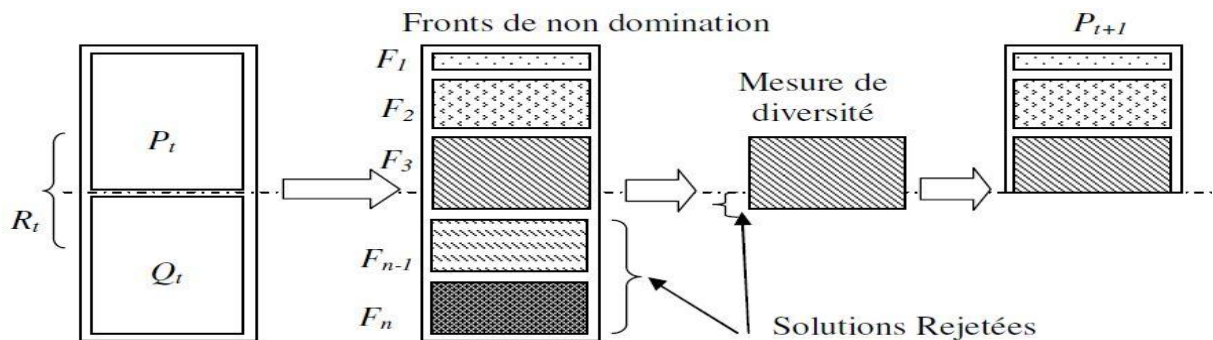


Figure 9: Algorithme NSGA-II. (Deb, 2002)

L'algorithme NSGA-II est un algorithme évolutif multi-objectif qui utilise un type de mesure du surpeuplement autour de chaque individu afin d'assurer la diversité de la population. Le principe de cet algorithme est illustré à la figure 09. Tout d'abord, une population $R_t = P_t \cup Q_t$ est formée, où P_t est la population des parents de taille N , et Q_t est la population des enfants (descendants) générés à partir des parents à travers les opérateurs de croisement et de mutation. La taille de la population totale R_t est $2N$. Cette population R_t composée des parents et des enfants est ensuite classée en différents fronts de non-dominance (F_1, F_2, \dots, F_{l_2}), en utilisant le concept de dominance de Pareto. (Deb, 2002)

Les individus faisant partie du premier front sont évidemment meilleurs que les individus du second front. Ces derniers sont à leur tour meilleurs que les individus du troisième front et ainsi de suite. L'accent est alors porté sur les meilleurs individus et donc dans ce cas, sur les individus du premier front, qui sont alors gardés dans une archive. Si le nombre d'individus de ce front est inférieur à la taille fixée N , alors tous les individus de ce front feront partie de la prochaine population $t+1 P$. Pour compléter la taille de la population $t+1 P$ à N , les nouveaux membres doivent être choisis dans les fronts restants, suivant leur rang. Ainsi les membres du second front F_2 sont choisis, puis les membres de F_3 et ainsi de suite jusqu'à obtenir une nouvelle population de taille N . Si tous les membres d'un front ne peuvent entrer dans la population sans un dépassement de la taille N , alors les membres de ce front sont classés suivant une mesure de diversité qui détermine la population autour de chaque individu pris séparément. Suivant ce classement, les individus les plus isolés sont admis dans la liste de $t+1 P$ jusqu'à compléter la taille de celle-ci. Les individus qui ne peuvent entrer dans la population sont simplement rejetés. (Deb, 2002)

I.9.2 Procédure de la recherche des solutions non dominées

Il existe différentes techniques ou algorithmes de recherche des solutions Pareto optimales, avec pour objectif de minimiser le temps de calcul (Deb, 2002). L'approche qui suit décrit la procédure, étape par étape pour trouver la population P' des solutions non dominées, à partir d'une population initiale totale P de taille N :

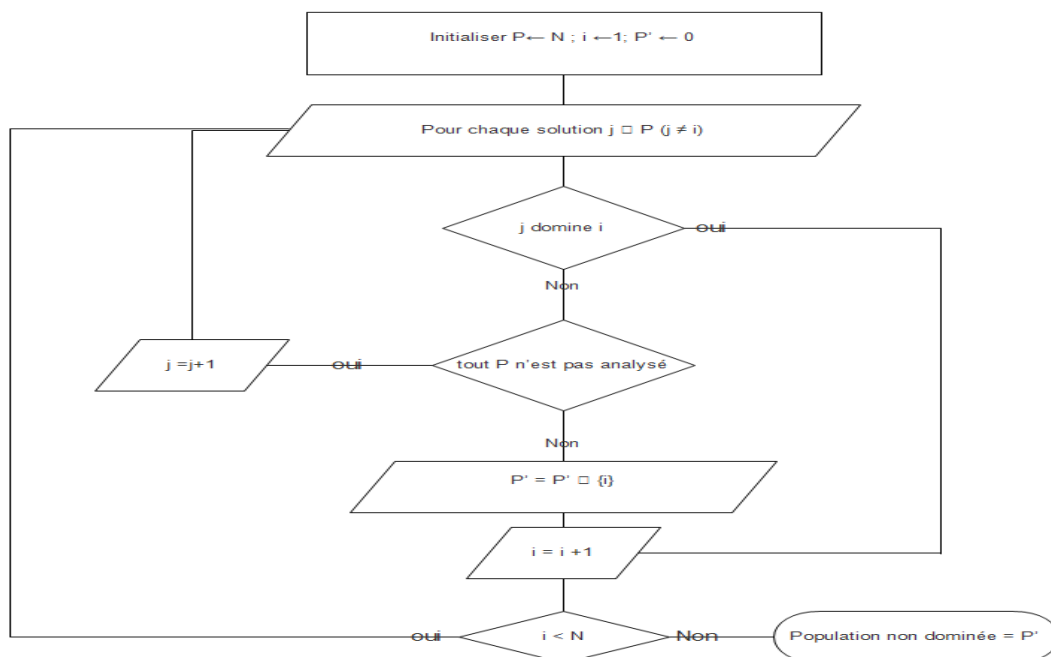


Figure 10: Procédure de la recherche des solutions non dominées

Nous avons désigné cette figure en basant sur les étapes de la procédure générale des algorithmes de recherche des solutions Pareto optimales.

Cette méthode compare donc chaque solution i avec le reste de la population P pour déterminer s'il existe un individu dans la population qui le domine. Si c'est le cas, on peut alors conclure qu'il existe au moins un individu dans la population qui est meilleur que i si l'on prend en compte toutes les fonctions objectives. La solution i ne peut donc appartenir à la liste des solutions non dominées P' . Cependant si aucun autre individu de la population ne domine i , celui-ci fait partie de la liste des solutions non dominées.

I.9.3 Maintien de la diversité dans la population

La distance de crowding d'une solution i (ou d'un individu) se calcule en fonction du périmètre formé par les solutions du même front les plus proches de i sur chaque objectif. La (figure 11) montre une présentation à deux dimensions associées à la solution i . Le calcul de la distance de crowding nécessite avant tout le tri des solutions selon chaque objectif, dans un ordre ascendant. Ensuite, pour chaque objectif, les individus possédant les valeurs limites (la plus petite et la plus grande valeur de fonction objective) se voient associés à une distance infinie (∞). Pour les autres solutions intermédiaires, on calcule une distance de crowding égale à la différence normalisée des valeurs de fonctions objectives de deux solutions adjacentes. Ce calcul est réalisé pour chaque fonction objective. La distance de crowding d'une solution est calculée en sommant les distances correspondantes à chaque objectif.

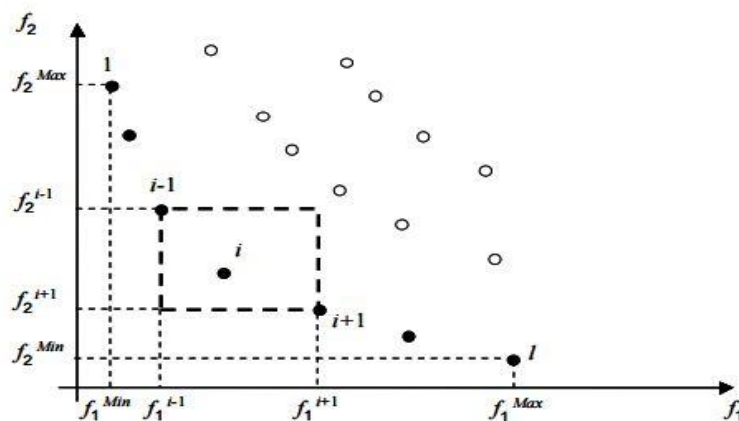


Figure 11: Distance de surpeuplement (Deb, 2002)

I.10 Conclusion

Nous avons présenté dans cette partie une vue généralisée des problèmes d'optimisation multicritère, tout d'abord par la définition de ce que sont les problèmes d'optimisation mono-objective et multicritère, puis nous avons détaillé les méta heuristique, vue qu'ils ont une grande capacité à résoudre ce type de problèmes, leur fonctionnement général et leur classification ainsi que les processus métier. Leur définition et la modélisation de ces processus dans le but de les optimiser.

La dernière section présente l'algorithme génétique ainsi que NSGA2, son principe, son fonctionnement.

Chapitre II

Les Sélections

II.1 Introduction

La sélection des parents à pour but de deviner les individus de la population courante qui seront autorisés à se reproduire (les “parents”). La sélection est fondée sur la qualité des individus, estimée à l’ aide de fonction d'adaptation. Cette opération est peut-être la plus importante puisqu’elle permet aux individus d’une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d’un individu sera directement reliée à son efficacité relative au sein de la population. Il existe plusieurs méthodes implémentées dans les algorithmes génétiques pour la reproduction. Ces dernières ont été inespérées depuis des sélections naturelles (E.Verrier & X.Rognon, 2009) biologiques mais il existe des sélections non implémentées.

Dans ce chapitre nous allons présenter les deux catégories (implémentée, non implémentée).

II.2 Sélections implémentable

Dans ce qui suit, nous allons citer les différentes méthodes de sélections implémentées dans les algorithmes génétiques :

II.2.1 Sélection proportionnelle :

Plusieurs stratégies de sélection des individus sont bastées sur le principe de sélection proportionnelle. Autrement dit, la chance qu'a un individu d'être sélectionné est proportionnelle à sa valeur de fitness. Ainsi,

a. *Sélection par roulette –RWS-*

Introduite par (Goldberg, 1989), la population est représentée comme une roue de roulette, ou chacun individu est représenté par une portion qui correspond proportionnellement à sa valeur de fitness.

La sélection d’un individu se fait en tournant la roue en face d’un pointeur fixe.

Cette procédure est répétée jusqu’à constitution complète de la population des parents.

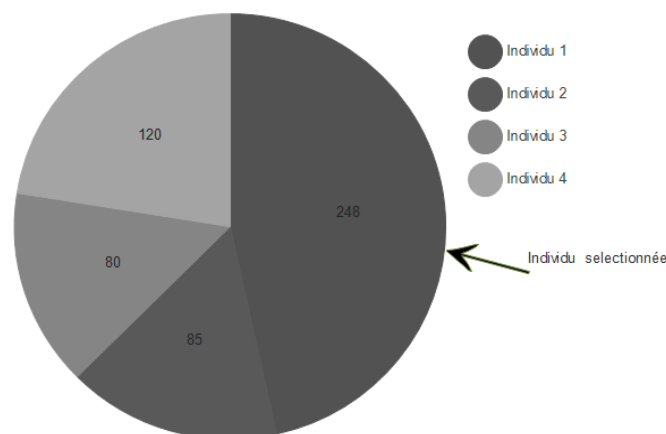


Figure 12 : sélection par roulette (Goldberg, 1989)

b. Sélection universelle stochastique –SUS-

Ce type est proposé par (Baker, 1987) utilise aussi une roulette partagée en autant de portions (individus), proportionnellement aux valeurs de fitness.

Mais cette fois, les individus sélectionnés sont désignés par un ensemble d'indicateurs équidistants ou une rotation de la roue de roulette sélectionne tous les parents simultanément.

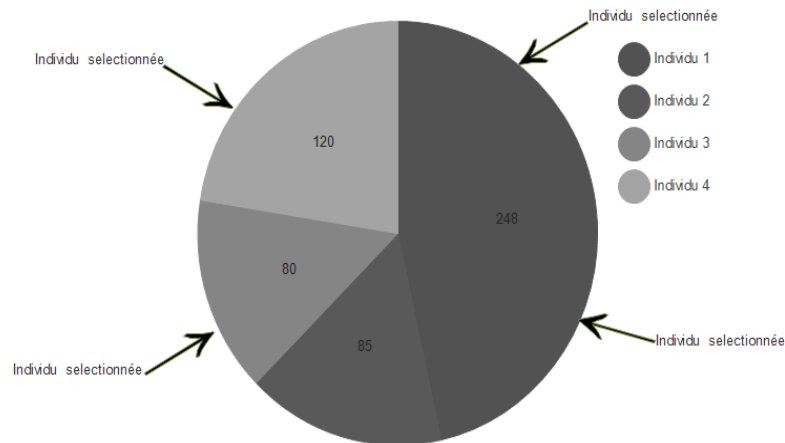


Figure 13: Sélection universelle stochastique (Baker, 1987)

c. Sélection d'échantillonnage à reste stochastique -RSS - :

Ce type a été illustré par (Whitley, 1994). Il utilise une procédure de traitement sur la valeur de fitness des individus comme la suit :

En supposant que la valeur de fitness d'une solution est un nombre décimal (2,45).

La partie entière de la valeur de fitness indique combien de copies sont directement placées dans la population de parents (2 copies dans cet exemple)

1. Puis, il faut déterminer si d'autres copies de cette solution doivent être encore ajoutées.
Dans un premier temps, l'auteur considère le reste de la valeur de fitness (0.45) comme une possibilité de placer une deuxième copie de cette solution.
2. Génère une valeur aléatoire (entre 0 et 1) (par exemple 0,6).
3. Fait une comparaison Si cette valeur est plus grande que (1- le reste), c'est-à-dire (1- 0.45 = 0,55) alors il ajoute une autre copie de cette solution à la population de parents.

d. Sélection par rang :

Se compose de deux étapes (Whitley D. , 1989). D'abord tous les individus de la population sont rangés selon leurs valeurs de fitness. Le rang est fait dans l'ordre décroissant (ou croissant), selon si on veut minimiser (ou maximiser) la fonction de fitness. Normalement, les individus de moins bonne qualité obtiennent un rang faible (à partir de 1). Et ainsi en itérant sur chaque individu on finit par attribuer le rang N au meilleur individu (où N est la taille de la population). Ensuite on effectue une sélection par roulette basée sur les rangs des individus où l'angle de chaque secteur de la roue sera proportionnel au rang de l'individu.

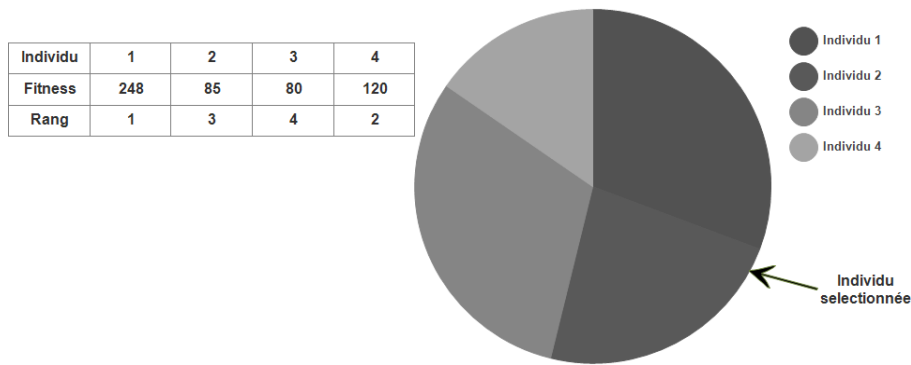


Figure 14 Sélection par rang

II.2.2 Sélection par tournoi :

Le principe consiste à choisir un sous-ensemble d'individus (S individus) aléatoirement dans la population, puis à sélectionner le meilleur individu dans ce groupe en fonction de sa fitness. Ce processus est répété jusqu'à l'obtention du nombre d'individus requis (Miller & Goldberg, 1995)

Le nombre de participants à un tournoi (S), appelée la taille de tournoi, est utilisé pour faire varier la pression de cette sélection (Freitas, 2002). Si ce nombre est grand, alors la pression sera forte et les faibles individus auront une petite chance d'être choisis. En général, un seul gagnant est choisi parmi les participants à un tournoi. Ce gagnant peut être choisi d'une façon déterministe ou probabiliste. Dans le cas déterministe, qui est pratiquement le plus utilisé, le gagnant est l'individu de meilleure qualité (meilleure fitness). Dans le cas probabiliste, chacun des participants peut être choisi en tant que gagnant avec une probabilité proportionnelle à sa fitness. Cela consiste à utiliser une procédure de sélection proportionnelle [comme la (roulette) citée ci-dessus] se référant seulement aux individus du tournoi. La méthode de tournoi la plus commune est le tournoi binaire (Bäck, 1992), où on choisit deux individus aléatoirement ($S=2$) puis on sélectionne le meilleur (qui a la valeur de fitness la plus élevée).

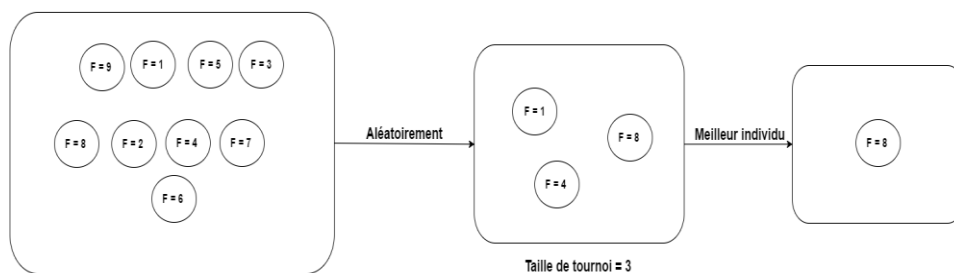


Figure 15 Sélection par tournoi

II.2.3 Sélection uniforme :

La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité $P= 1/N$ d'être sélectionné, où N est le nombre total d'individus dans la population.

Pour résumé, tous les types de sélection trouvés dans la littérature nous allons utiliser le schéma dans la figure 14 que nous l'avons inspiré à partir du texte précédent

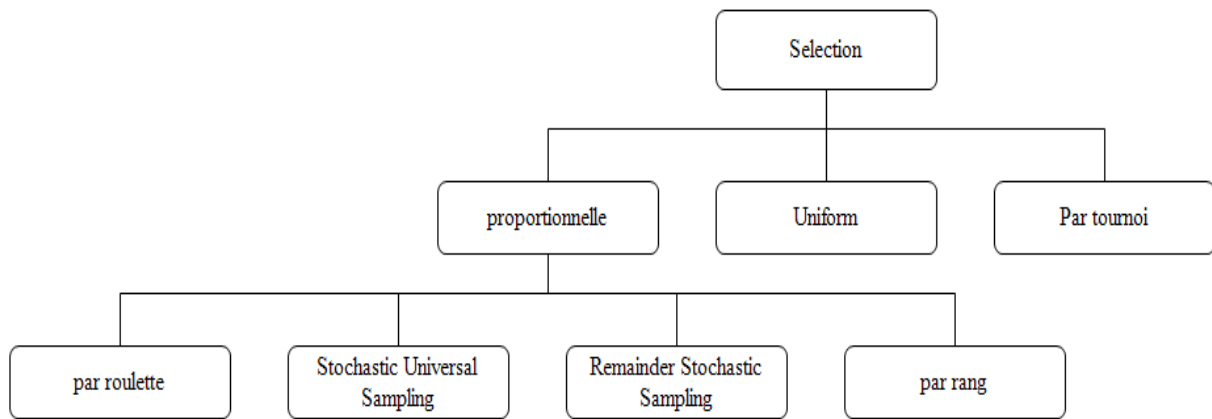


Figure 16: Types de sélections

II.3 Les sélections naturelles non implémentées

La sélection, dans le cadre de l'amélioration des espèces domestiques, est le processus par lequel certains individus sont choisis de préférence à d'autres pour engendrer la génération suivante. (E.Verrier & X.Rognon, 2009)

Il s'agit du mécanisme fondamental qui permet, à la nature comme aux humains, de modeler les attributs des animaux.

Dans ce qui suit nous allons citer les différents types de sélections naturelles dans la littérature

II.3.1 La sélection massale.

La sélection massale consiste à choisir les reproducteurs en fonction de leur(s) performance(s) individuelle(s). Pour obtenir une évaluation génétique des candidats, il suffit de contrôler les performances. Ainsi, la sélection massale n'implique pas la tenue d'un état civil, ce qui a des avantages dans les situations où aucun état civil n'est disponible. (E.Verrier & X.Rognon, 2009)

La sélection massale est la méthode de sélection la plus simple à mettre en oeuvre, ce qui en constitue l'avantage majeur. Elle autorise une forte intensité de sélection. Son principal inconvénient réside dans l'impossibilité qu'il y a de l'appliquer à certains caractères : ceux qui ne s'expriment que dans un sexe ou que l'on ne peut mesurer qu'après abattage. (E.Verrier & X.Rognon, 2009)

II.3.2 La sélection sur ascendance

La sélection sur ascendance consiste à choisir les reproducteurs en fonction des informations dont on dispose sur leurs parents ou sur des ancêtres plus éloignés. La mise en oeuvre de cette méthode (comme celle des autres méthodes généalogiques) repose sur la disponibilité de fichiers d'état civil fiables et tenus à jour. Ainsi, lorsque les enjeux génétiques ou commerciaux sont importants, il est nécessaire de procéder à une vérification de la filiation sur la base du polymorphisme biologique. (E.Verrier & X.Rognon, 2009)

La sélection sur ascendance est la sélection la plus précoce que l'on puisse faire, ce qui en constitue l'avantage majeur. On doit l'utiliser pour un premier tri des reproducteurs. Son principal inconvénient réside dans sa faible précision. Aussi, ce premier tri doit être affiné par la suite au moyen d'autres méthodes (E.Verrier & X.Rognon, 2009)

II.3.3 La sélection sur descendance

Dans un dispositif d'évaluation génétique en routine, dans des populations à générations chevauchantes, quand un animal a un descendant contrôlé (mesuré pour le caractère), l'information correspondante est incorporée à son index de valeur génétique. La mise en place d'une épreuve spécifique de sélection fondée sur la valeur moyenne de la descendance n'est, en revanche, organisée que dans certaines occasions : on parle d'épreuve de la descendance. Cette épreuve spécifique n'est commodément applicable que pour les mâles et, dans les faits, elle est essentiellement pratiquée pour les mâles des espèces peu ou pas prolifiques. La sélection sur descendance est la méthode qui permet d'obtenir le maximum de précision et elle est applicable pour tous les caractères. Ce sont là deux avantages majeurs. La sélection sur descendance représente cependant l'inconvénient d'allonger notablement l'intervalle de génération. Sur le plan pratique, la mise à l'épreuve de la descendance est coûteuse et nécessite des infrastructures importantes et une organisation rigoureuse : elle ne peut être le fait d'éleveurs individuels. (E.Verrier & X.Rognon, 2009)

II.3.4 La sélection sur collatéraux

La sélection sur collatéraux consiste à sélectionner les reproducteurs à partir de la moyenne des performances de leurs demi- ou pleins frères-sœurs. Elle est surtout développée dans les espèces où l'on peut disposer de familles nombreuses : volailles, poissons, lapins, porcs. Une des difficultés majeures de cette méthode est de limiter les effets d'environnement commun aux membres d'une même famille, ces derniers s'ils ne sont pas pris en compte pouvant en effet conduire à retenir prioritairement les candidats issus des familles ayant bénéficié d'un environnement particulièrement favorable. (E.Verrier & X.Rognon, 2009)

L'évaluation sur collatéraux est en général assez peu précise et sa fiabilité peut être mise en défaut en cas d'effet d'environnement commun, phénomène relativement courant pour des pleins frères-sœurs. Son intérêt réside surtout dans la possibilité d'évaluer des candidats pour des caractères qui ne s'expriment que dans un sexe ou dont la mesure nécessite l'abattage, et ce sans passer par l'évaluation sur descendance qui est longue et coûteuse. La sélection sur collatéraux est principalement employée chez les volailles, compte tenu des caractéristiques biologiques de ces espèces et de la maîtrise importante des conditions de milieu qui existe en sélection avicole. (E.Verrier & X.Rognon, 2009)

<i>Méthode</i>	<i>Information valorisée</i>
Sélection massale	Performance(s) individuelle(s)
Sélection sur ascendance	Performance(s) des parents Index des parents
Sélection sur descendance	Performance moyenne d'un petit échantillon des descendants
Sélection sur collatéraux	Performance moyenne des frères et sœurs (pleins f-s et demi-f-s)

Tableau 1: Les principales méthodes de sélection naturelle

II.4 La sélection par génération

Cette Méthode de sélection est basée sur la moyenne de fitness de la génération auxquelles l'individu appartient. C'est-à-dire que nous allons choisir juste les individus qui sont contenus dans la meilleure génération. Le choix de la meilleure génération ce diffère selon le type de problème (moyenne maximale pour un problème de maximisation ou moyenne minimale pour un problème de minimisation).

II.5 Synthèse

D'après les recherches, nous sommes arrivés à définir les sélections que nous avons trouvé et ce qu'est claire est que chaque méthode de sélection a ces propres caractéristiques et ça propre utilité.

Plusieurs chercheurs ont étudié la performance de GA en utilisant différentes stratégies de sélections ; mais presque aucun d'entre eux n'a testé sur le problème de l'optimisation des processus métiers. La performance de GA est généralement évaluée en termes de taux de convergence et le nombre de générations pour atteindre la solution optimale. (O. A. Jadaan, 2005) par exemple a comparé les résultats de GA entre la roue de la roulette proportionnelle et la méthode de sélection de roue de roulette à base de rang en utilisant plusieurs fonctions objectifs et a constaté que la sélection par rang est mieux que la sélection par roulette en nombre de générations à venir avec la solution optimale. Il a observé que la méthode de sélection de roue de roulette à base de rang est plus stable, plus rapide, certaine et plus robuste vers les solutions optimales que la roue de la roulette proportionnelle. D'autre part, (J. Zhong, 2005) a comparé roulette proportionnelle avec la sélection par tournoi, avec une taille du tournoi égale 6 à sept fonctions de test général et a conclu que l'algorithme avec la sélection du tournoi est plus efficace en matière de convergence que la sélection de la roue de la roulette proportionnelle. (Julstrom, 1999) a étudié l'efficacité de temps de calcul de deux types de probabilités de sélection fondées sur le rang; classement linéaire et probabilités de classement exponentielles et par rapport à la sélection du tournoi. Il a souligné que la sélection du tournoi est préférable à la sélection en fonction du rang parce que la sélection répétée du tournoi est plus rapide que le tri de la population d'attribuer des probabilités fondées sur le rang. En outre, (Mashohor, Evans, & Arslan, 2005) évaluer la performance du système d'inspection de circuits imprimés en utilisant la méthode de sélection trois GA ; déterministe (par rang), tournoi et la roue de la roulette et découvert que déterministe a la capacité d'atteindre le fitness maximal avec le plus bas nombre de générations pour toutes les images de test suivi par la roue de la roulette et la sélection du tournoi.

En dehors de cela, (Goldberg & Deb, 1991) a fait des études approfondies sur la sélection par roulette, par rang et par tournoi. Leurs études ont été réalisées pour comprendre le rapport de fitness et de la convergence du temps prévu. Ils ont constaté que la sélection par rang et la sélection du tournoi ont fait des résultats meilleurs que par roulette en termes de maintien d'une pression constante vers la convergence. Ils ont démontré en outre que la sélection par rang et la sélection par tournoi binaire ont des attentes identiques, mais recommandé la sélection du tournoi binaire en raison de sa complexité temporelle plus efficace.

Pour la complexité (E, Goldberg, & Deb, 1991) ont mentionné la différence entre ces méthodes de sélections selon leurs complexités et leurs résultats sont résumées dans le tableau (2).

De notre part nous avons essayé de collecter toutes les informations que nous pouvons dégager à partir des ouvrages et des articles que nous avons trouvés dans nos recherches et finalement nous avons sorti avec les résultats suivants :

Il y a des sélections qui nécessite une taille petite de population comme la sélection par roulette et la raison c'est qu'elle a une complexité de $O(n^2)$ ou n représente la taille de la population alors le plus grande sera la taille de la population le long sera le temps de calcul

Cependant il y des sélections qui ne dépend pas de la taille de la population comme la sélection par tournoi par exemple.

D'autre part nous avons constaté qu'il y a des méthodes qui servent à trouver la solution optimale dans un nombre bas d'itérations comme la sélection par roulette car elle assure que les individus le plus fit sera sélectionné pour la nouvelle génération donc si nous avons un problème de maximisation par exemple il y une forte chance d'arriver à la solution optimale (fitness le plus élevé) dans un nombre bas d'itération que nous l'avons approximé à 1 jusqu'à 100 itérations. Les autres sélections se différent ente un nombre moyen (de 100 à 1000) et élevé (1000 et plus).

Dans le tableau qui va suivre nous allons résumer toutes les sélections déjà mentionnées avec les avantages et les inconvénients de chaqu'une. Cette synthèse va nous aider à bien comprendre les caractéristiques et la déférence entre ces sélections-là par rapport à plusieurs critères (avantage, inconvénients, caractéristiques, taille de population... etc.).

Pour chaque méthode il y a deux critères contradictoires qui peut les évaluer (Diversité et convergence de solutions). Alors la meilleure méthode de sélection est celle qui peut garantir une moyenne entre ces deux critères là.

Le tableau (2) va démontrer les variances qui existe entre ces différentes méthodes de sélections.

Sélection	Auteur	Implémentable	Caractéristiques	Avantages	Inconvénients	Taille de population	Nombre d'itération	Complexité	Temps de calcul
Par roulette RWS	Caldberg 1989	Oui	<i>Principe de la roulette</i>	<i>Simple à implémenter.</i>	<i>Perte de diversité.</i>	<i>Petit</i>	<i>Bas</i>	$O(n^2)$	<i>Long</i>
Echantillon universal stock	Baker 1987	Oui	<i>Principe de la roulette</i>	Sélectionne tous les parents simultanément.	<i>Cette sélection ne prend pas en compte la diversité des solutions.</i>	<i>Peu importe</i>	<i>Elevé</i>	$O(n)$	<i>Court</i>
Remaindertochastic sampling RSS	Whitley 1994	Oui	<i>Utilise une procédure de traitement sur la valeur de fitness.</i>			<i>Peu importe</i>	<i>Elevé</i>	$O(n)$	<i>Court</i>
La sélection par rang	Whitkey 1989	Oui	<i>Classer les individus par rang</i>	<i>Tous les individus ont une chance d'être sélectionnés.</i>	<i>Peut dégrader la vitesse de convergence.</i>	<i>Petit</i>	<i>Bas</i>	$O(n \ln n)$ + <i>temps de sélection</i>	<i>Moyen</i>
La sélection par Tournoi	Brindle 1981	Oui	<i>Faire des tournois entre les individus</i>	<i>Tous les individus ont une chance d'être sélectionnés.</i>	<i>Peut dégrader la vitesse de convergence.</i>	<i>Peu importe</i>	<i>Bas</i>	$O(n)$	<i>Court</i>
				<i>Il préserve la diversité.</i>					
Uniforme		Oui	<i>Faire un choix au hasard</i>	<i>Simple à implémenter</i>	<i>Pas sûr de sélectionner les meilleurs parents</i>	<i>Peu importe</i>	<i>Moyen</i>	$O(n)$	<i>Court</i>

La sélection Massale		Pas encore	<i>Basé sur la performance de l'individu</i>	<i>Simple à mettre en œuvre</i>			.	.	.
La sélection sur ascendance		Pas encore	<i>Basée sur la Performance(s) des parents</i>	<i>Rapide</i>	<i>Peu précise</i>		.	.	.
La sélection sur descendance		Pas encore	<i>Performance moyenne d'un petit échantillon des descendants</i>	<i>Plus précise</i>	<i>Plus longue</i>		.	!	!
La sélection sur collatéraux		Pas encore	<i>Performance moyenne des frères et sœurs</i>
La sélection par génération		Pas encore	<i>Performance de l'ensemble de la génération de l'individu sélectionné</i>

Tableau 2: synthèse finale

II.6 Conclusion

Dans cette partie nous avons présenté un des opérateurs le plus important puisqu'elle permet aux individus d'une population de survivre : la sélection des parents.

Il existe plusieurs méthodes pour sélectionner un individu ces méthodes ont été inspirées depuis des sélections naturelles biologiques.

Nous avons classé ces méthodes de sélection en deux classes :

- 1- **Les sélections implémentées** : sont les sélections utilisées dans les algorithmes génétiques.
- 2- **Les sélections non implémentées** : sont les sélections jamais utilisées dans les algorithmes génétiques et utilisées dans la reproduction des animaux.

Pour bien comprendre chaque sélection nous avons expliqué le principe de chacune et l'année de la première utilisation et par qui.

Enfin de cette partie nous avons proposé un tableau (Synthèse) pour montrer la différence entre ces méthodes déjà mentionnées selon les axes (Auteur, année de création, implémenter, méthode mathématique).

Chapitre III

Expérimentation

III.1 Introduction

Le chapitre III qui est le dernier chapitre sera le fruit de notre travail qu'a duré plus de six mois, cette partie consiste à présenter les potentiels de notre Framework (BPO_tool v 2.0), ce dernier a comme rôle de générer des solutions possibles en passant par des étapes que nous allons les expliquer dans la suite de ce chapitre.

Nous allons utiliser des cas d'études réels (des scénarios) ces derniers nous allons les intégrer dans notre Framework en suite un ensemble de tests seront fait avec des paramètres déferents afin de trouver la solution optimale que nous cherchons.

III.2 Partie I : Environnement de travail

III.2.1 Modèle mathématique pour la conception d'un processus métier

L'ensemble des tâches n_d appartenant à une conception particulière d'un processus métier est $N_d = \{t_1, t_2, t_3, \dots, t_{nd}\}$. L'ensemble des r_d ressources dans la conception $R_d = \{r_1, r_2, r_3, \dots, r_{rd}\}$ inclut les sous-ensembles R_{in} et R_{out} qui stockent les ressources d'entrée et les ressources de sortie du processus métier, respectivement. La conception d'un processus métier utilise toutes les ressources de R_{in} et produit toutes les ressources de R_{out} . En outre, chaque tâche i dans la conception d'un processus métier possède t_{in} ressources d'entrée stockées dans $I_i \subseteq R_d$ et t_{out} ressources de sortie stockées dans $O_i \subseteq R_d$. Enfin, chaque tâche i a des valeurs d'attribut p stockées dans l'ensemble TA_i et les attributs du processus p correspondants sont stockés dans l'ensemble PA .

Paramètre	Description
n_d	Nombre de tâches dans la conception
r_d	Nombre de ressources dans la conception
t_{in}	Nombre des ressources d'entrée d'une tâche
t_{out}	Nombre des ressources de sortie d'une tâche
r_{in}	Nombre des ressources d'entrée d'un processus
r_{out}	Nombre des ressources d'entrée d'un processus
P	Nombre des attributs d'une tâche dans un processus
N_d	Ensemble des n_d tâches
R_d	Ensemble des r_d ressources
I_i	Ensemble des t_{in} ressources d'une tâche i
O_i	Ensemble des tout ressources d'une tâche i
R_{in}	Ensemble des r_{in} ressources

R_{out}	Ensemble des r_{out} ressources
TA_i	Ensemble des valeurs des attributs d'une tâche i
PA	Ensemble des valeurs des attributs du processus p

Tableau 3: Paramètres principaux utilisés pour la conception d'un processus.

III.2.2 Matrice tâches ressources (task resources matrix - TRM)

La matrice qui vise à capturer le séquençage des tâches et les motifs formulés dans la conception du processus métier. A cet effet, la matrice cartographie les entrées et les sorties des tâches relatives à la conception du processus. Cette matrice est appelée matrice tâche/ressource (*TRM*). Cette matrice à deux dimensions avec la taille $n_d \times r_d$. Les lignes de TRM présentent les n_d tâches de la conception d'un processus métier, tandis que les colonnes présentent les r_d (nombre de ressources utilisées dans le processus métier) ressources de la conception d'un processus métier. Chaque case de TRM montre la relation entre la tâche et la ressource.

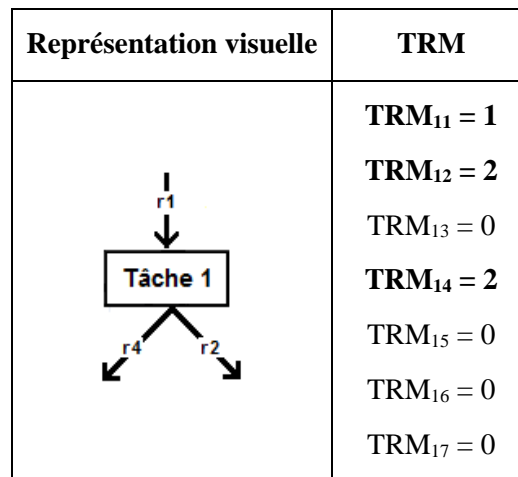


Figure 17: Exemple de TRM basée sur la tâche

	r1	r2	r3	r4	r5	r6	r7	r8	r9
Tâche 1	1	2	0	2	0	0	0	0	0
Tâche 2	0	0	0	1	0	2	0	0	1
Tâche 3	0	0	2	0	0	0	1	0	0
Tâche 4	0	1	0	0	0	0	0	2	0
Tâche 5	0	0	0	0	0	1	0	0	2
Tâche 6	0	0	1	0	2	0	0	1	1

Tableau 4: Exemple de TRM

Pour une tâche $i \in N_d$ et une ressource $j \in R_d$, nous avons :

Si $r_i \in I_i$ alors $TRM_{ij} = 1$ (Si la ressource appartient à l'ensemble des ressources d'entrée de la tâche alors sa relation est marquée par 1)

Si $r_i \in O_i$ alors $TRM_{ij} = 2$ (Si la ressource appartient à l'ensemble des ressources de sortie de la tâche alors sa relation est marquée par 2).

Si $r_j \notin I_i$ et $r_j \notin O_i$ alors $TRM_{ij} = 0$ (Si la ressource n'appartient ni aux entrées, ni aux sorties de la tâche i , alors sa relation est marquée par 0).

III.2.3 Librairie des taches

La librairie des taches est l'ensemble des taches qui peut être inclus dans un processus donné

Chaque tâche dans cette librairie est caractérisée par un nombre fini d'attribues. (Numéro de la tâche, nom de la tâche, les ressources d'entré, les ressources de sortie, SDP, SFT, SDD).

Le tableau suivant présente un exemple sur la forme de librairie.

<i>Numéro de la tache</i>	<i>Nom de la tache</i>	<i>Ressources d'entré</i>	<i>Ressources de sortie</i>	<i>SDP</i>	<i>SFT</i>	<i>SDD</i>
<i>0</i>	<i>Fax.com</i>	<i>1.2</i>	<i>6</i>	<i>220</i>	<i>110</i>	<i>350</i>
<i>1</i>	<i>DB Business Vérification</i>	<i>2.4</i>	<i>3.5</i>	<i>250</i>	<i>200</i>	<i>300</i>
<i>2</i>	<i>StrikeIron Lookup Business</i>	<i>5</i>	<i>1</i>	<i>200</i>	<i>180</i>	<i>290</i>
<i>3</i>	<i>GraphMagic's Graph Chart Web Service API</i>	<i>5.8</i>	<i>2</i>	<i>203</i>	<i>107</i>	<i>368</i>

Tableau 5 Exemple de librairie des taches

III.2.4 Fonction objective

$F(x)$ est la fonction objective pour l'optimisation multicritères de nos processus métier. Le nombre de critères (objectifs) utilisé par $F(x)$ est de trois (03), la minimisation du prix de livraison du service (C), la maximisation du taux de réussite du service (S) et la minimisation de la durée de livraison du service (T).

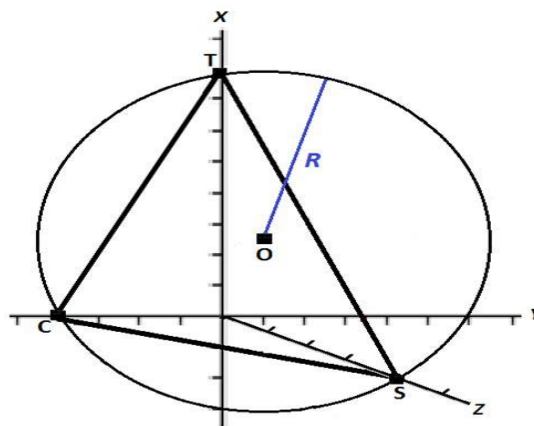


Figure 18: Représentation du rayon d'un cercle circonscrit du triangle CTS.

Pour représenter cette fonction, nous nous sommes aidés de la loi des sinus. La figure 15 représente la schématisation de notre fonction objective. Le rayon R du cercle circonscrit du triangle CTS (de type quelconque) symbolise $F(x)$. Ce choix s'est imposé à nous par le fait de l'absolu nécessité de ne favoriser aucun objectif en particulier, et vu que le rayon (R en l'occurrence) est équidistant des trois sommets du triangle CTS (*i.e.* $OT = TS = OC$).

Il serait bon de préciser que chaque processus métier obtenu par notre Framework comporte trois critères C , T et S avec :

$$C = \sum_1^{n_d} c_i, T = \sum_1^{n_d} t_i, S = \sum_1^{n_d} s_i$$

Où

n_d : le nombre maximal de tâches par processus métier.

c_i : le prix de livraison d'une tâche i composant un processus métier.

s_i : taux de réussite d'une tâche i faisant partie d'un processus métier.

t_i : la durée de livraison d'une tâche i d'un processus métier.

Soit C , T , S trois points dans un plan à trois dimensions (X , Y , Z), avec les coordonnées suivantes :

$$C(x_c, y_c, z_c)$$

$$S(x_s, y_s, z_s)$$

$$T(x_t, y_t, z_t)$$

Le rayon d'un cercle circonscrit du triangle CST se calcule ainsi :

$$2R = \frac{ST}{\sin \widehat{SCT}} = \frac{CS}{\sin \widehat{CTS}} = \frac{TC}{\sin \widehat{TSC}}$$

Suivant le calcul de la norme d'un vecteur :

$$\|SC\|^2 = \|TS\|^2 + \|CT\|^2 + 2(TS)(CT) \cos \widehat{SCT}$$

Ce qui implique :

$$\cos \widehat{SCT} = \frac{\|TS\|^2 + \|CT\|^2 - \|SC\|^2}{2(TS)(CT)}$$

Sachant que pour tout angle x :

$$\cos^2 X + \sin^2 X = 1$$

Nous pouvons déduire que :

$$\sin \widehat{SCT} = \sqrt{1 - \left(\frac{\|TS\|^2 + \|CT\|^2 - \|SC\|^2}{2(TS)(CT)} \right)^2}$$

Après avoir ajouté la loi des sinus, nous obtenons :

$$2R = \frac{SC}{\sqrt{1 - \left(\frac{\|TS\|^2 + \|CT\|^2 - \|SC\|^2}{2(TS)(CT)} \right)^2}}$$

Dernière étape est l'utilisation des coordonnées des différents points du triangle CTS . (*e.g.* $\|SC\| = \sqrt{(x_c - x_s)^2 + (y_c - y_s)^2 + (z_c - z_s)^2}$).

III.2.5 Cas d'expérimentation

Le Framework de l'optimisation proposée est orientée vers l'optimisation tri-objective. Il met également l'accent sur les problèmes min-max-min, à savoir la minimisation du premier objectif, la maximisation du second et la minimisation du troisième.

Basé sur la spécification des attributs de service (Lakin, 1996):

Le prix de livraison du service (SDP ou *Service Delivery Price*) est sélectionné comme premier objectif. SDP spécifie le montant d'argent que le client du service devra payer pour la consommation d'un service distinct, à savoir le coût pour utiliser le service.

Le deuxième objectif est la maximisation du taux de réussite du service (SFT ou *Service Fulfilment Target*). SFT spécifie la promesse du fournisseur de services d'une livraison efficace et transparente des prestations déterminées à tout consommateur de service autorisé ; qui demande le service dans les délais de service définis. Il est exprimé comme le nombre maximal promis de livraisons de service individuelles réussies en ce qui concerne les comptes totaux de livraisons de service individuelles. SFT peut être mesuré et calculé par le service consommateur ou par groupe de consommateurs et peut être soumis à différents moments.

Le troisième objectif à minimiser est la durée de livraison du service (SDD ou *Service Delivering Duration ou Service Providing Duration*). Avec cet attribut, le laps de temps maximal admissible est spécifié pour rendre tous les avantages propres au service déclenché par le consommateur de service (ou client). SDD représente l'intervalle de temps respectif, à partir du déclenchement du service défini par un consommateur de service autorisé jusqu'au rendu complet de tous les avantages exprimés au lancement du service par le client.

Ces trois attributs de services seront utilisés comme les objectifs d'optimisation pour les scénarios de la vie réelle.

a. Scénario de test

1) Description du scénario : la prévision des ventes (Sales forecasting)

Ce scénario décrit le processus métier de la prévision des ventes (Grigori, 2004). Ce processus est considéré comme semi-automatique car elle implique l'interaction de certaines applications, mais nécessite l'intervention humaine dans le processus de génération et de visualisation des prévisions demandées. Nous nous proposons d'automatiser entièrement le processus en sélectionnant et mettant en œuvre des services web pertinents, afin de proposer un ensemble de conceptions de processus métier optimisés qui répondent aux exigences du scénario présenté, ayant des valeurs d'attribut optimales.

Figure 19 montre la conception des processus métier génériques pour le scénario sur la prévision des ventes. Il comporte deux ressources d'entrée : (a) le nom de l'entreprise et (b) la demande de mise à jour du marché. La première ressource est nécessaire pour le service Web afin d'extraire les données pertinentes pour l'entreprise spécifiée. La deuxième ressource est une demande de mise à jour du marché qui doit être prise en considération pour la prévision des ventes. La conception initiale du processus consiste en cinq étapes : la première étape consiste à récupérer les informations financières de l'entreprise pour les prévisions pertinentes. Parallèlement à cela, les dernières nouvelles du marché doivent être mises à jour, car elles seront prises en compte pour aboutir à des prévisions pertinentes (étape 2). Le résultat de ces deux étapes conduit une simulation qui génère les prévisions de ventes (étape 3). Cette prévision est alors tracée dans un graphique (étape 4), puis communiquée à la

personne la demandant (étape 5). Le résultat de ce processus est un rapport contenant les résultats prévisionnels.

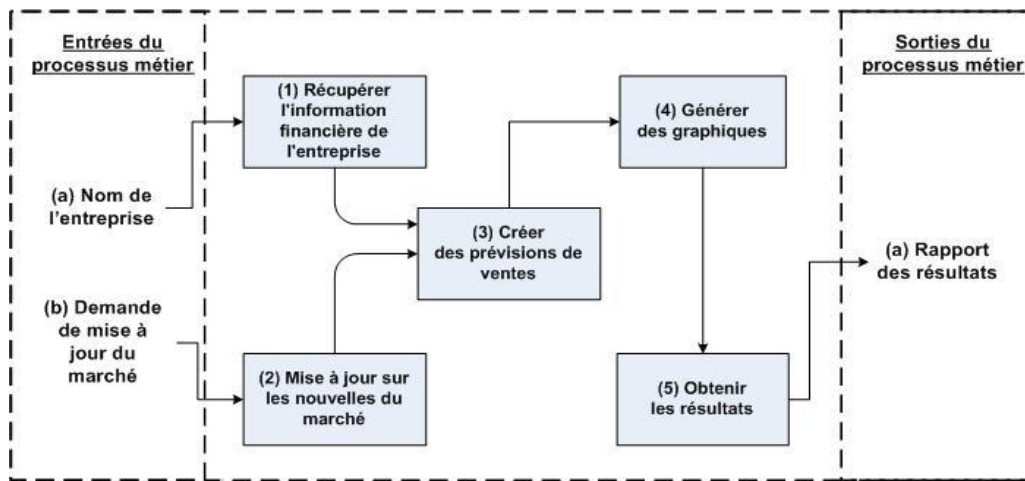


Figure 19 Conception initiale des processus métier de la prévision des ventes

2) Librairie des tâches et ressources utilisées

Nous allons définir l'ensemble des services web à utiliser par le logiciel d'optimisation, pour obtenir de nouvelles conceptions de processus métier respectant le scénario présenté. Ces services web sont regroupés dans une librairie (Tableau 6). Des recherches pertinentes sur les librairies en ligne pour sélectionner de services web ont donné lieu à une sélection de 20 services Web de différents fournisseurs qui peuvent potentiellement mettre en œuvre notre scénario (Vergidis, 2008). Les services web de la librairie sont disposés dans l'ordre alphabétique, au nombre de 20, numérotés de 0 et 19.

N°	Nom de tâche	Entrée	Sortie	SDP	SFT	SDD
0	D&B Business Verification	3,1	0,5	208	113	339
1	Fax.com	8,2	4	219	109	307
2	Gale Group Business Information	0,3	5	229	115	352
3	Gale Group Business Intelligence	5,8	0,5	202	109	309
4	GraphMagic's Graph & Chart Web Service API	8,2	2	203	108	355
5	interfax.net	5,7	4	200	103	336
6	Lokad Business time-series forecasting and analysis	3,6	8	218	112	351
7	Midnight Trader Financial News	3	7	206	103	316
8	StrikeIron Company Search	1,3	0,3	211	109	350
9	StrikeIron Get Business Prospect	3,1	0,5	224	103	359

10	StrikeIron Lookup Business	3	3, 0	220	113	343
11	Wall Street Horizon Real-Time Company Earnings	3, 1	5	218	107	332
12	Xignite Get Balance Sheet	3, 1	5	206	105	314
13	Xignite Get Chart Url	8	2	203	110	374
14	Xignite Get Chart Url Preset	8	2	226	105	340
15	Xignite Get Growth Probability	5, 7	8	222	114	322
16	Xignite Get Market News Headlines	6	7	228	100	289
17	Xignite Get Market Summary	6	7	215	102	374
18	Xignite Get Topic Chart	3, 5, 7	8	229	108	337
19	Xignite Get Topic Data	3, 5, 7	2, 8	208	105	373

Tableau 6 Librairie des services web utilisés

Pour chaque service web, les ressources d'entrée et de sortie se trouvent dans les colonnes adjacentes. Le logiciel d'optimisation proposé est testé pour trois objectifs SDP, SFT et SDD. Comme les fournisseurs de services ne fournissent pas d'informations détaillées sur les performances de chaque service Web, les valeurs attribuées sont basées sur la distribution uniforme d'une petite plage de valeurs sur chaque service Web. Cela rend les services Web disponibles compétitif (avec une certaine différence entre eux). Le tableau 2 présente les différentes ressources qui seront utilisées par les services web dans la librairie.

N°	Nom de la ressource
0	Business details
1	Business query
2	Chart / graph
3	Company name
4	Fax (on-line)
5	Financial data
6	Market update request
7	Recent market trends
8	Time-series forecast

Tableau 7 Ressources disponibles pour le scénario test.

III.3 Partie2 : Framework : BPO_Tool

III.3.1 Description du Framework : BPO_Tool

Le Framework utilisé est une application de bureau réalisée avec le langage de développement Java. Cet outil propose une solution pour faire une optimisation multi objective des processus métiers.

BPO_Tool utilise l'algorithme génétique NSGA II à l'aide de plusieurs méthodes de sélection (les sélections mentionnées dans le chapitre II) et donne comme résultat des solutions faisable et des fois optimales pour des exemples (scénario à testés). L'avantage major de notre Framework c'est qui il donne la structure de chaque solution trouvée et propose la possibilité de sauvegarder la meilleure solution en XML.

a. Outil de développement

Le Framework était développé en utilisant les outils suivants :

Version de JDK

Nous avons opté pour la version de Java, JDK 1.8.01¹

Java Netbeans-8.0.1

NetBeans est un environnement de développement intégré (EDI) gratuit Utilisé principalement pour le développement d'applications avec Java et Android²

b. Hiérarchie du Framework

Afin de bien structuré notre Framework nous étions obligé de choisir une certaine organisation pour le code source alors nous avons séparé les classes par des packages selon la fonctionnalité de chaque classe. La figure suivante va montrer comment cette organisation est faite

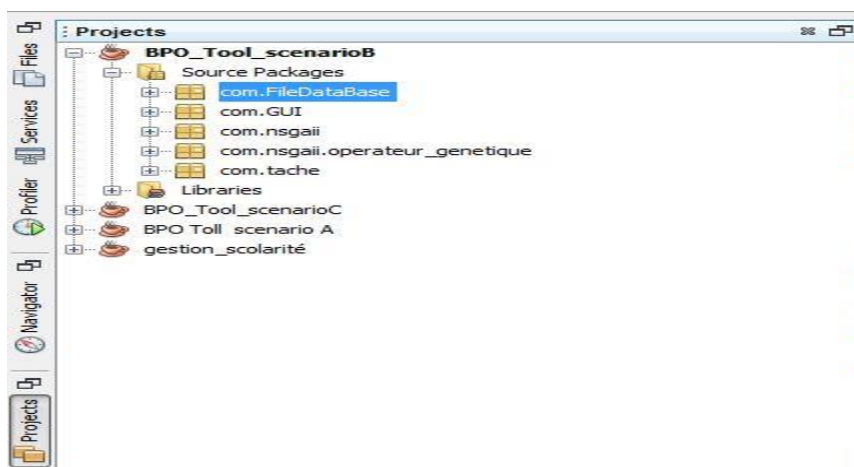


Figure 20 Packages du Framework

1- <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

2- <https://netbeans.org/downloads/8.0.1/>

<i>Nom du package</i>	<i>Description du package</i>
Com.GUI	Contient les classes qui concernent l'interface du Framework.
Com.nsgaii	Contient les classes qui concernent le fonctionnement de l'algorithme NSGA II.
Com.nsgaii. Operateur_génétique	Contient les classes qui concernent tout les opérateurs génétiques (Les méthodes de sélections, Croisement, Mutation).
Com. Tache	Contient les classes qui concernent le coté processus métier et l'algorithme de la création de la population initiale.

Tableau 8 Description des packages

c. *Librairies utilisées dans le Framework*

BPO_Tool propose plusieurs utilitaires pendant son exécution notant la possibilité de représenter les solutions dans un plan et la possibilité d'exporter les valeurs de chaque solutions (SDP, SFT, SDD) dans un fichier Excel et plusieurs d'autres fonctionnalités, alors pour cela nous étions obligés d'importer des bibliothèques externes. Ces dernières sont représentées dans la figure suivante

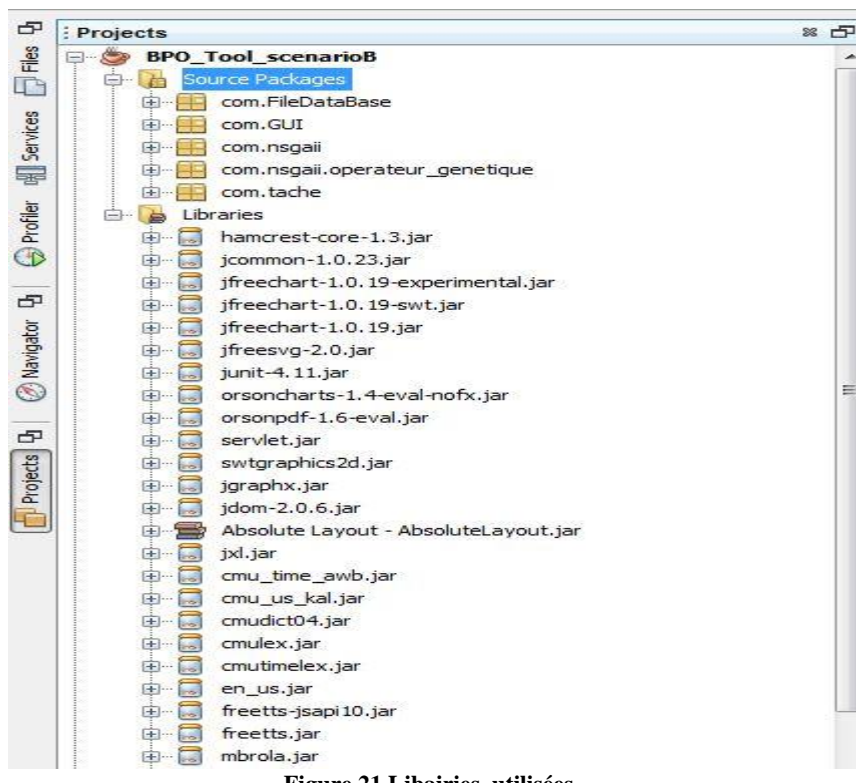


Figure 21 Librairies utilisées

JfreeChart 1.0.19.jar :

C'est une API Java permettant de créer des graphiques et des diagrammes de très bonne qualité. Cette API est open source et sous licence LGPL .

Nous avons utilisé cette dernière pour représenter les individus (processus métier) dans des plans à deux démentions.

JGraphX.jar :

C'est une API qui permet de dessiner des graphes dans une application Java, nous avons l'utilisé pour représenter graphiquement la structure du processus.

JDom.jar :

C'est une API open source Java dont le but est de représenter et manipuler un document XML de manière intuitive pour un développeur Java sans requérir une connaissance pointue de XML. Dans ce Framework le rôle de Jdom est d'importer la librairie des tâches et les inputs et les outputs de processus initial ou pour exporter la meilleure solution sous la forme d'un fichier XML.

Jxl.jar :

C'est une API java utilisé pour créer et manipuler des fichiers Excel. Nous avons utilisé jxl.jar pour sauvegarder les individus de la population et les solutions dans un document Excel.

d. Machine d'exécution

Pour le développement et l'exécution du Framework nous avons utilisé une machine avec les performances suivantes :

<i>Processeur</i>	<i>I3 1.7 GHz</i>
<i>RAM</i>	<i>4 Gb</i>

Tableau 9 Performance du machine

e. Fonctionnement du Framework

BPO_Tool pendant son exécution utilise deux algorithmes essentiels, une pour créer la population initiale qui contient les individus à traiter (processus métier) et bien sur NSGA II qui sert à faire l'optimisation multi objective des individus.

Le premier algorithme était créé par nous, afin de pouvoir créer des individus valides.

Le mot valide ici signifie que chaque individu doit respecter un nombre de conditions pour qu'il soit utile dans notre traitement. Ces conditions sont :

- 1- Les entrés de chaque processus doivent être incluses dans l'ensemble des entrés du scénario utilisé
- 2- Les sorties de chaque processus doivent être incluses dans l'ensemble des sorties du scénario utilisé
- 3- Les taches qui forment le processus doivent être bien reliées c'est-à-dire que chaque sortie de la tache A doit être l'entrée de la tache B sachant que A et B sont deux tâches liées entre elles

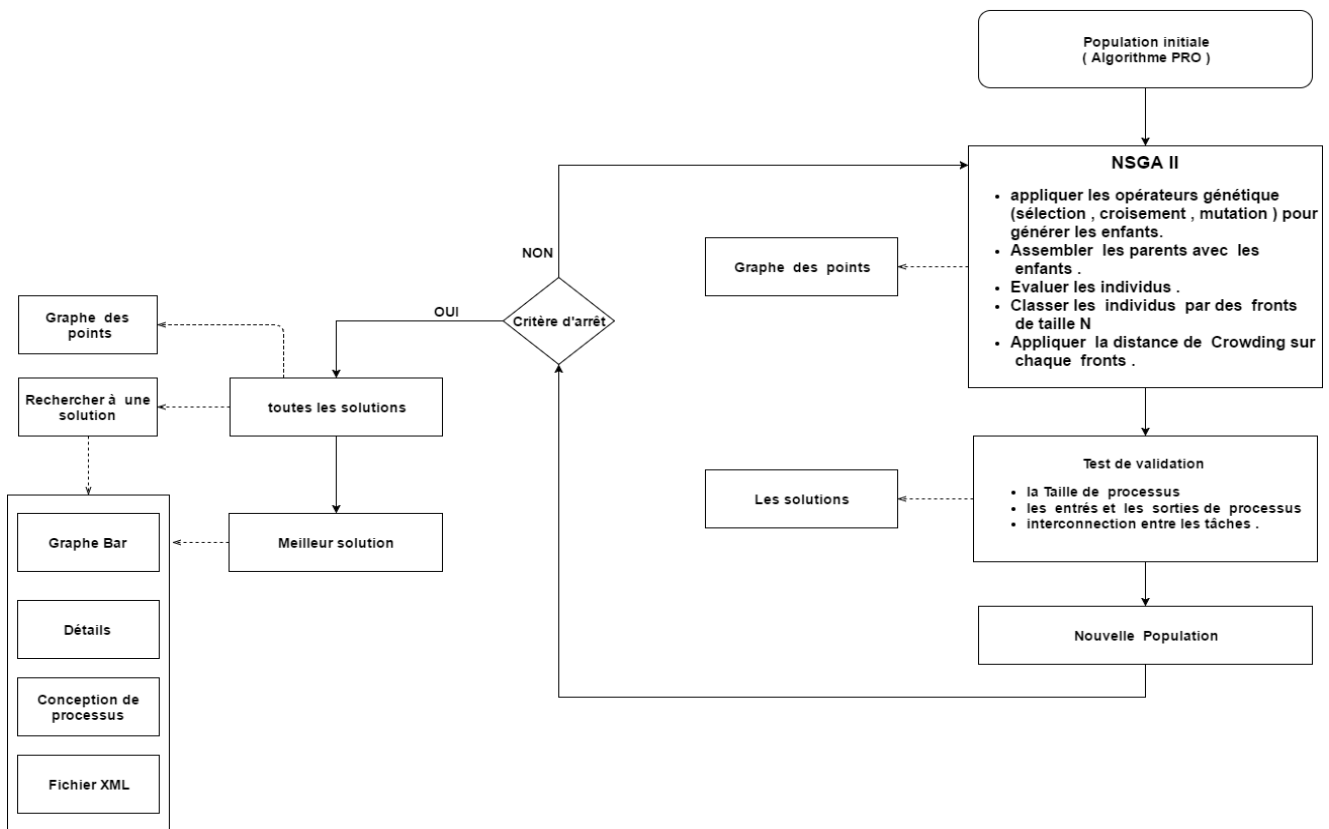


Figure 22 Déroulement du BOP_Toot

1) Algorithme de la création de la population initiale

Cet algorithme est le responsable de la création de la population des individus valides qui seront traité par notre Framework afin de pouvoir les optimiser.

Le principe de cet algorithme est :

D’abord il faut choisir deux taches aléatoirement et ajouter leurs outputs à **la liste des inputs possibles**, ensuite nous devons ajouter les deux taches sélectionnées à **la liste des tâches sélectionnées** et les Supprimer de la librairie pour ne les choisir par une autre fois pendant la procédure de la création du processus.

Après ça, il faut sélectionner une autre tâche, si les inputs de cette tâche n’appartiennent pas à **la liste des inputs possibles**, nous devons alors refaire ça jusqu’à arriver à trouver une tâche qui convient. Lorsque nous trouvons une tâche valide nous devons ajouter cette dernière à **la liste des tâches sélectionnées** et ajouter ces outputs à **la liste des inputs possibles**. Et nous supprimons cette tâche de la librairie. A chaque fois nous sélectionnons une tâche nous allons créer les liens d’interconnexion entre les tâches sélectionnées.

Nous devons vérifier la taille de processus (taille = nombre des tâches sélectionnées) à l’ajout de chaque tâche.

Si la taille de processus inférieur de la taille demandée alors nous recommençons la procédure.

Sinon, l’algorithme a produit un processus métier avec des tâches interconnectées avec une taille prédéfinie dans la partie de la configuration.

La figure 23 résumera le fonctionnement de notre algorithme.

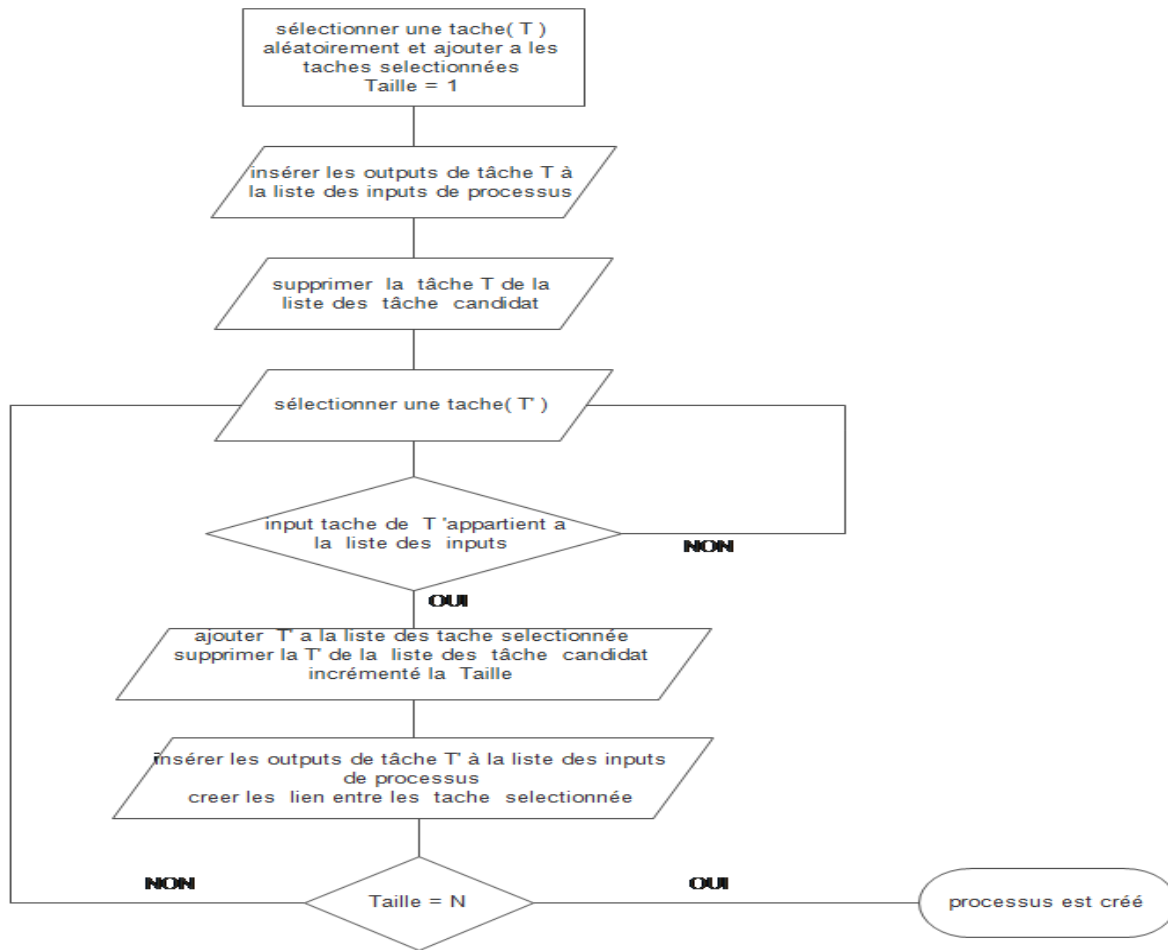


Figure 23 Organigramme de l'algorithme de la création de la population initiale

f. Interfaces du BPO Tool

2) Configurations

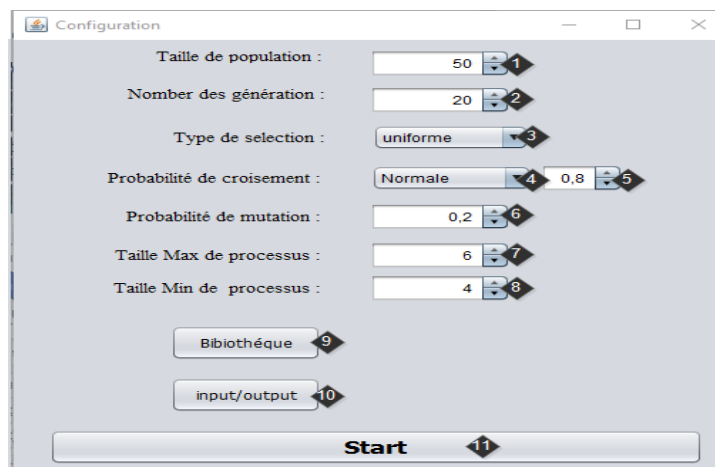


Figure 24 interface de la configuration

<i>Numéro</i>	<i>Description</i>
<i>1</i>	<i>Configurer le nombre des individus dans la population</i>
<i>2</i>	<i>Configurer le nombre des itérations</i>
<i>3</i>	<i>Sélectionner le type de la sélection voulue</i>
<i>4</i>	<i>Sélectionner le type de croisement voulu</i>
<i>5</i>	<i>Probabilité de croisement</i>
<i>6</i>	<i>Probabilité de mutation</i>
<i>7</i>	<i>Configurer la taille minimale du processus</i>
<i>8</i>	<i>Configurer la taille maximale du processus</i>
<i>9</i>	<i>Importer la librairie des taches</i>
<i>10</i>	<i>Importer le scénario du test</i>
<i>11</i>	<i>Lancer l'exécution du BPO Tool</i>

Tableau 10 description de (interface de configuration)

3) Affichage des résultats (Plans)

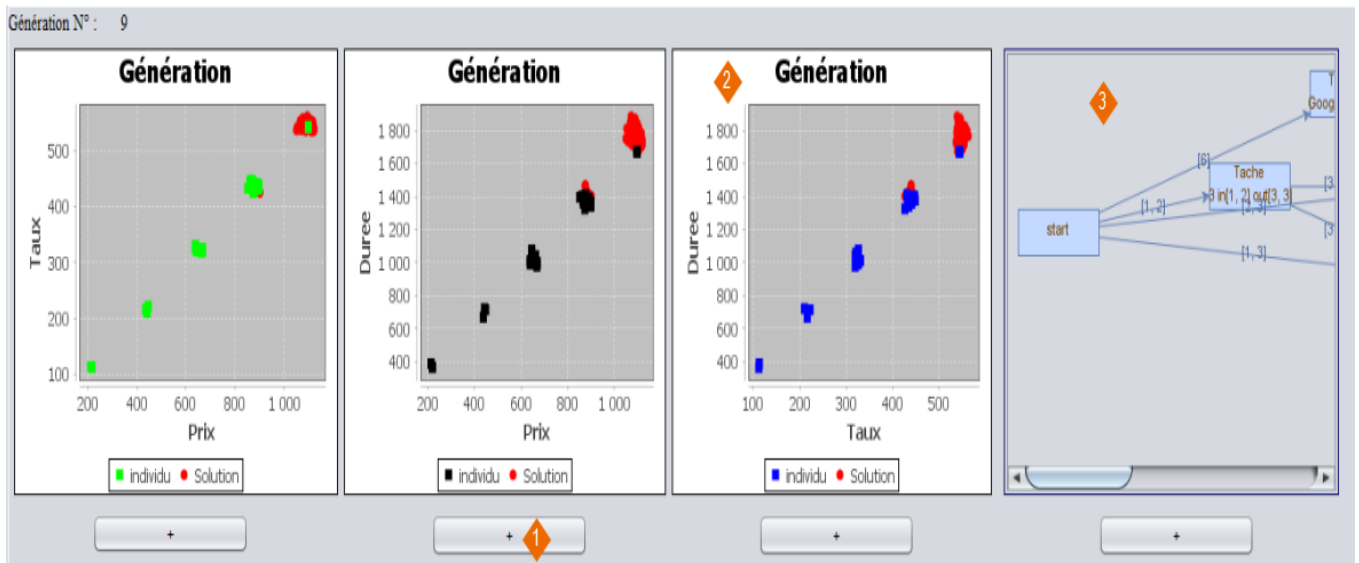


Figure 25 Interface du résultat

Numéro	Description
1	Botton sert à grandir le plan qui affiche les points
2	Un plan qui affiche les points (individus) selon 2 critères
3	La représentation graphique de la conception de la meilleure solution dans cette génération

Tableau 11 description de l'interface du résultat

4) Barre d'outils

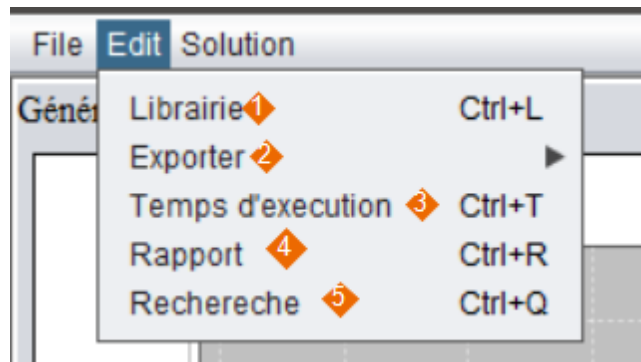


Figure 26 Barre d'outils (Edit)

Numéro	Description
1	Affiche la librairie des taches qui contient toutes les taches du scénario utilisé
2	Exporter les résultats sous Excel ou sous format d'une image
3	Temps d'exécution du BPO Tool
4	Le rapport entre le nombre des individus el le nombre des solutions
5	Recherche une solution selon leur durée/ prix ou satisfaction

Tableau 12 Barre d'outils (Edit)

5) Les sorties de BPO Tool

Notre Framework a la possibilité d'exporter les résultats sous format XML et de montrer la conception des meilleurs résultats et en plus de ça BPO Tool donne aussi plusieurs informations utiles sur les solutions comme le temps de calcul et la ration des solutions faisables, tous ces derniers nous allons les montrer dans les figures ().

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Solution SDP=" 895" SFT=" 450" SDD=" 1322">
3    <task>
4      <num_tache>3</num_tache>
5      <name_tache>Gale Group Business Intelligence</name_tache>
6    </task>
7    <task>
8      <num_tache>15</num_tache>
9      <name_tache>Xignite Get Growth Probability</name_tache>
10   </task>
11   <task>
12     <num_tache>8</num_tache>
13     <name_tache>StrikeIron Company Search</name_tache>
14   </task>
15   <task>
16     <num_tache>16</num_tache>
17     <name_tache>Xignite Get Market News Headlines</name_tache>
18   </task>
19 </Solution>

```

Figure 27 Fichier XML d'un résultat

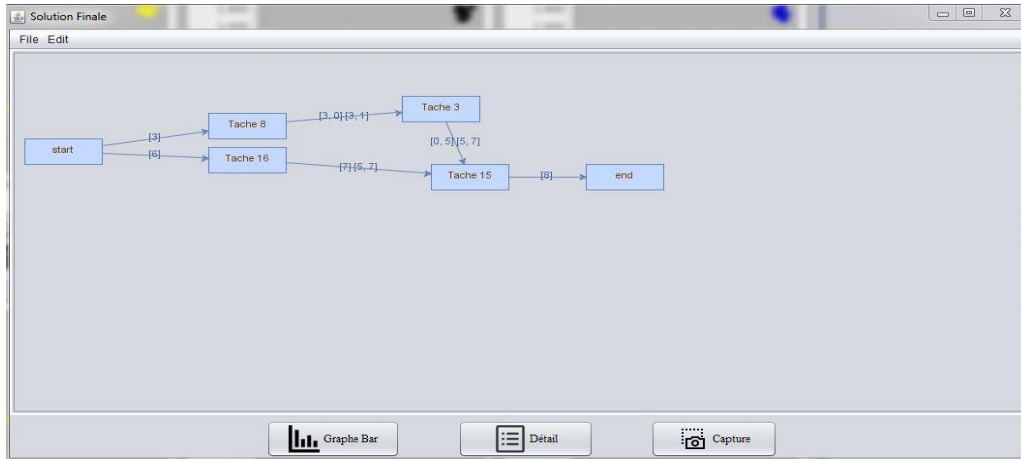


Figure 29 Conception du résultat

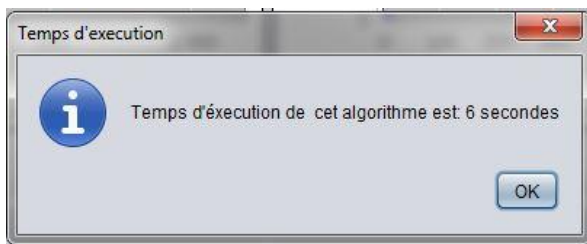


Figure 30 Temps de calcul

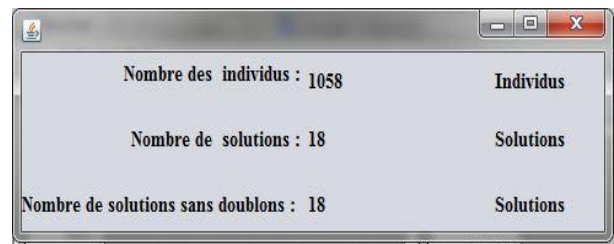


Figure 28 Ratio de solutions

III.3.2 Résultats et discussions

a. Paramètres du Framework pour le cas d'expérimentation

Après avoir recueilli toutes les informations nécessaires (bibliothèque de services web et les ressources pour chaque service web), les paramètres du problème peuvent être définis. Le tableau 3 résume les valeurs de paramètres utilisés par notre logiciel d'optimisation, pour notre cas d'expérimentation.

Paramètre	Désignation	Valeur
Taille librairie des tâches	n	20
Taille du processus métier initial	n_d	5
Taille minimum d'un processus métier	n_{\min}	4
Taille maximum d'un processus métier	n_{\max}	6
Ressources disponibles	r	9
Ressources d'entrée (Input) du processus métier initial	t_{in} / t_{out}	0, 1
Ressources de sortie (Output) du processus métier initial	r_{in} / r_{out}	8

Attributs par tâche / processus	p	3
prix de livraison du service	SDP	200 - 230
taux de réussite du service	SFT	100 - 115
durée de livraison du service	SDD	285- 375

Tableau 13 Valeurs de paramètres pour le scénario du test

Pendant plusieurs essais nous avons remarqué que les résultats se stagnent après 20 itérations et que nous n'avons pas besoins d'utiliser un nombre grand de populations, BOP_Tool a pu trouver des solutions faisables et bonnes juste avec 500 individus. Alors nous allons résumer les paramètres que nous avons utilisé pendant les tests dans le tableau se dessous

b. Résultats obtenus pour chaque sélection

Dans cette partie nous allons présenter les différents résultats trouvé pour chaque sélection après nous allons récapituler les résultats donnée dans tableau afin de pouvoir les comparais et pour essayer de trouver qu'elle est la meilleurs sélection.

1) Sélection uniforme

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
500	20	142 sur 11426	49 Secondes	(838-446)

Tableau 14 Résultat de la sélection uniforme

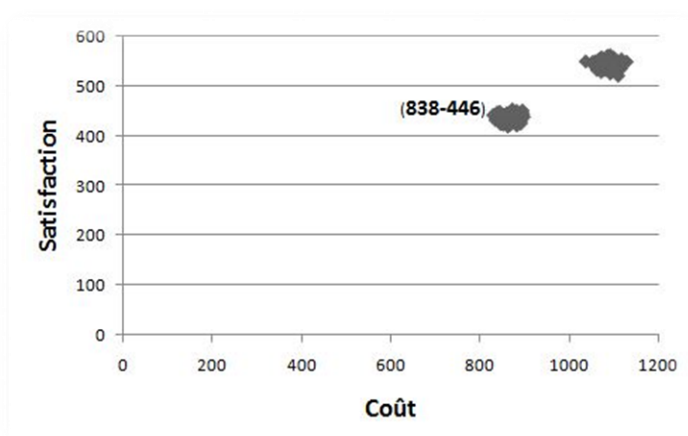


Figure 32 Graphe des solutions pour la sélection uniforme

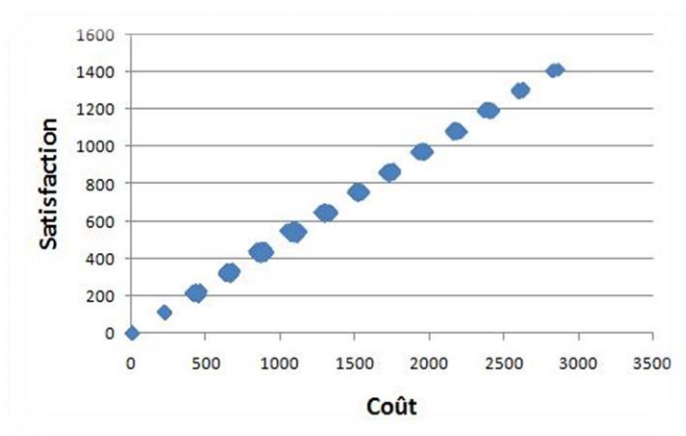


Figure 31 Graphe de toutes les générations

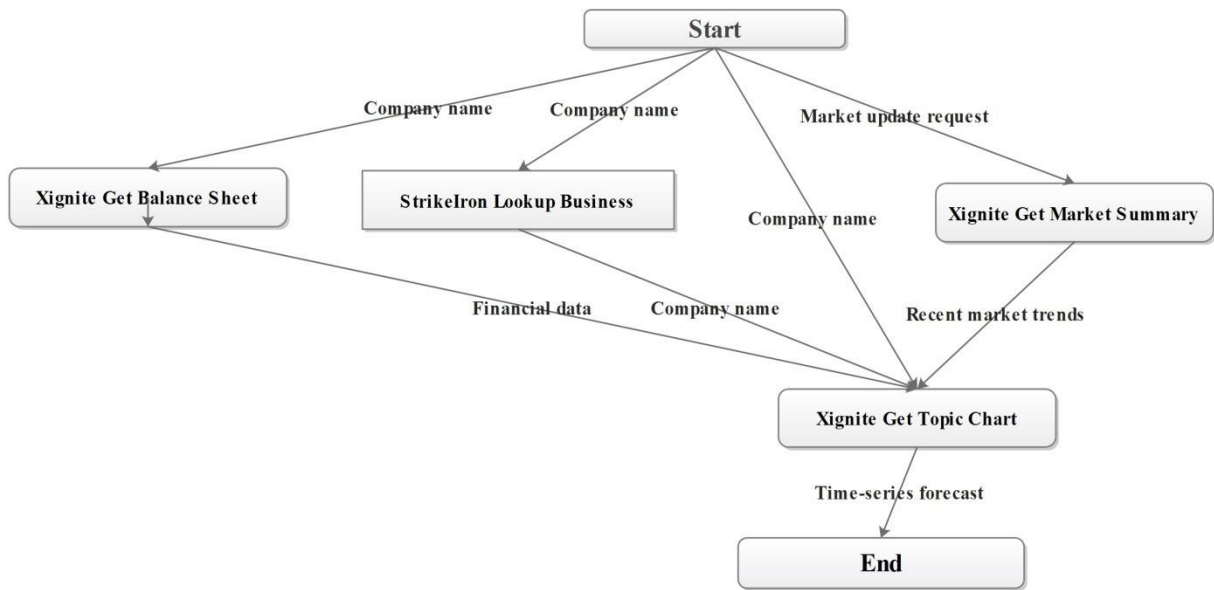


Figure 33 Conception de la meilleure solution pour la sélection uniforme

2) Sélection par tournoi

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleurs solutions
500	20	157 sur 10752	52 Secondes	(845-446) (852-448)

Tableau 15 Résultat de la sélection par tournoi

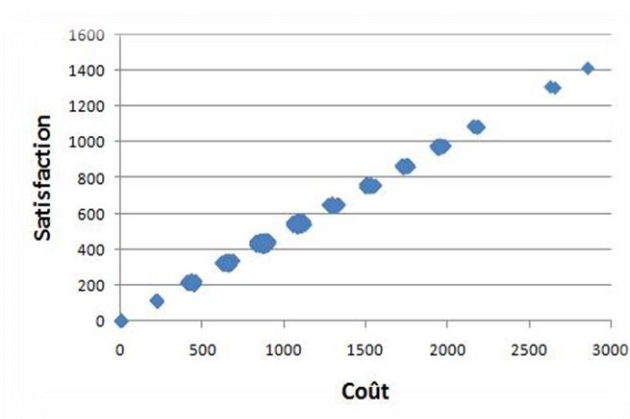


Figure 34 Graphe de toutes les générations

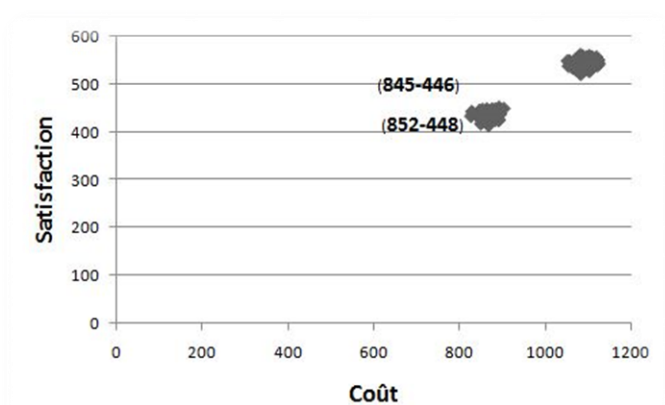


Figure 35 Graphe des solutions pour la sélection par tournoi

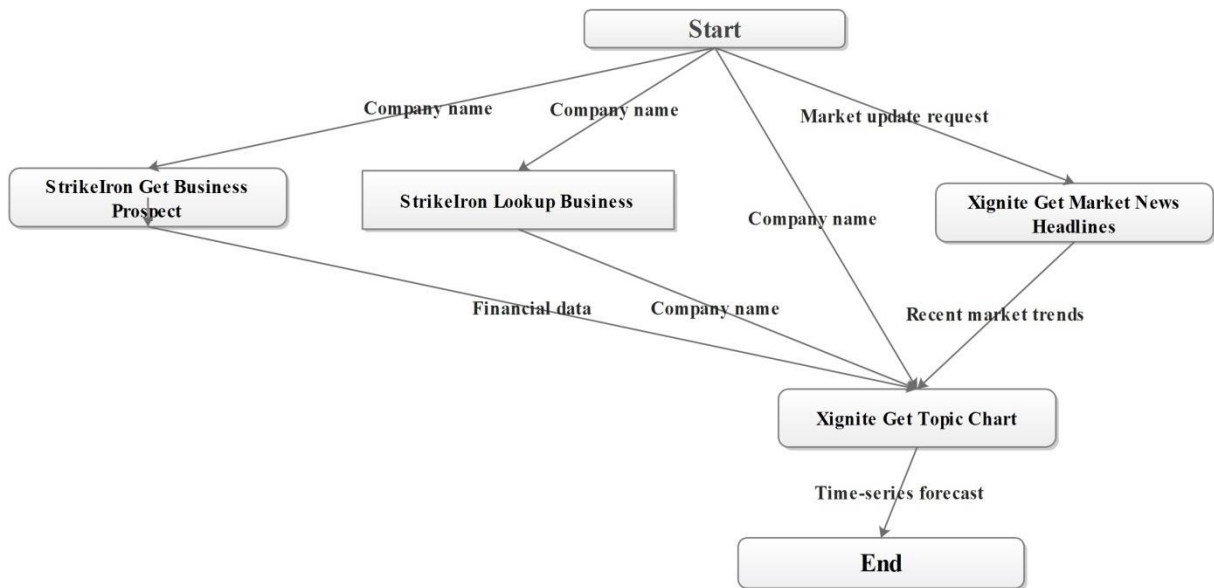


Figure 36 Conception de la meilleure solution (845-446) du tournoi

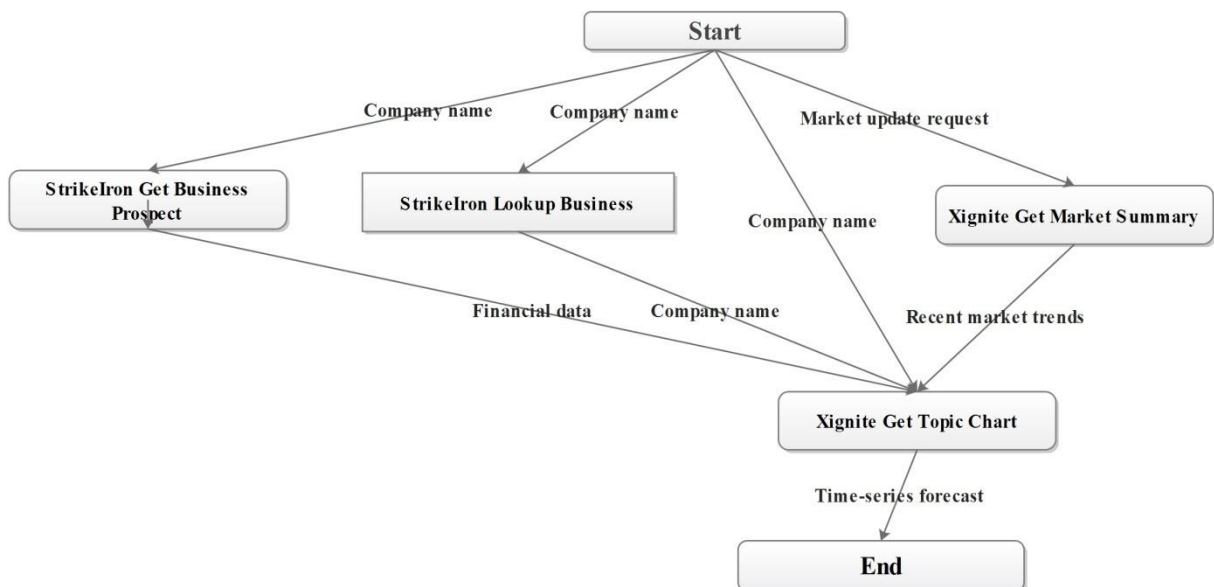


Figure 37 Conception de la meilleure solution (852-448) du tournoi

3) Sélection par roulette

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleurs solutions
200	20	154 sur 10554	360 Secondes	(851-447)

Tableau 16 Résultat de la sélection par roulette

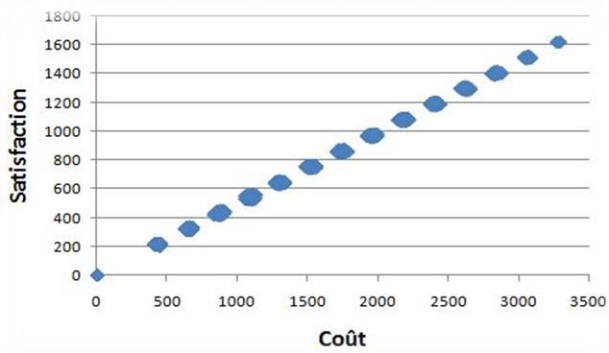


Figure 39 Population de toutes les générations

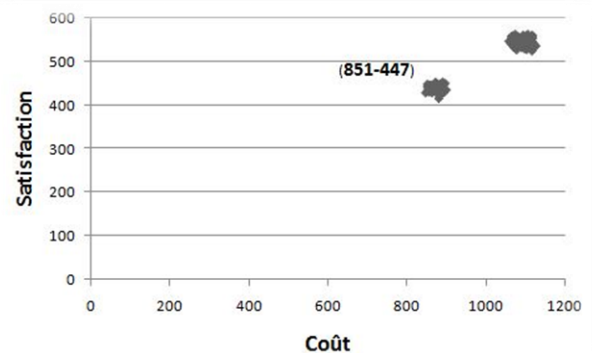


Figure 38 Graphe de solution de la sélection par roulette

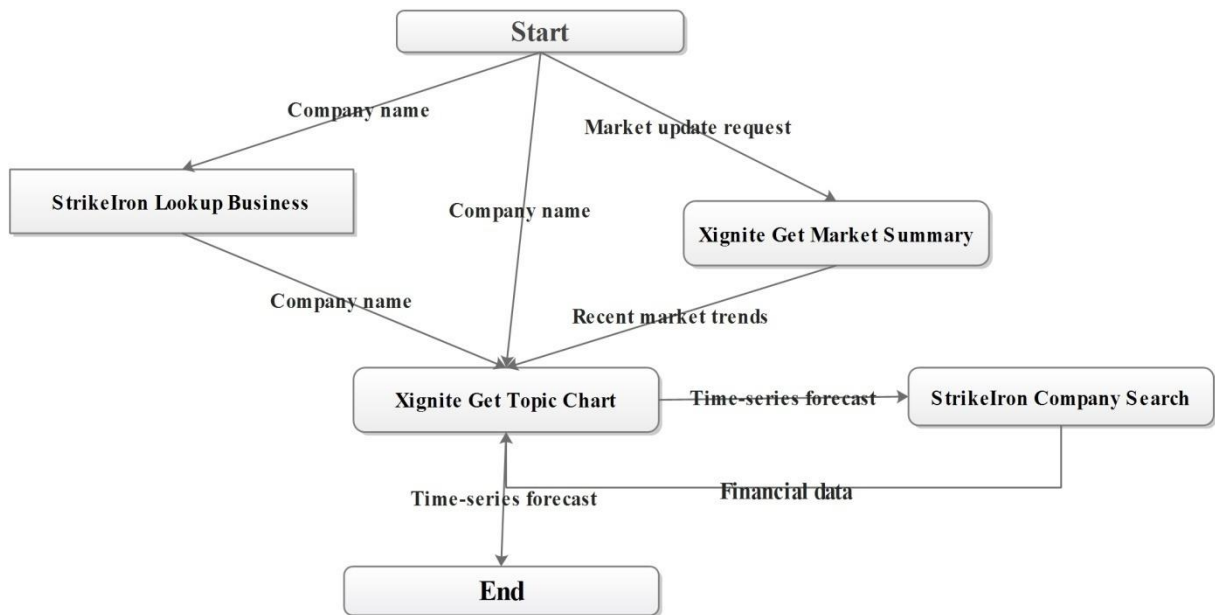


Figure 40 Conception de la meilleure solution pour la sélection uniforme

4) Sélection par rang

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleurs solutions
200	20	157 sur 4200	209 Secondes	(852-448)

Tableau 17 Résultats de la sélection par rang

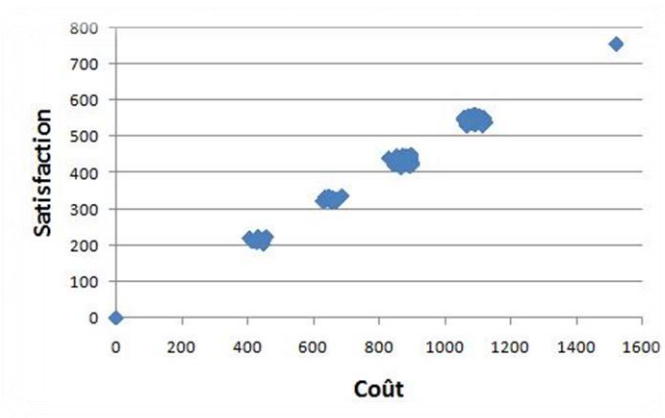


Figure 41 Graphe de toutes les générations

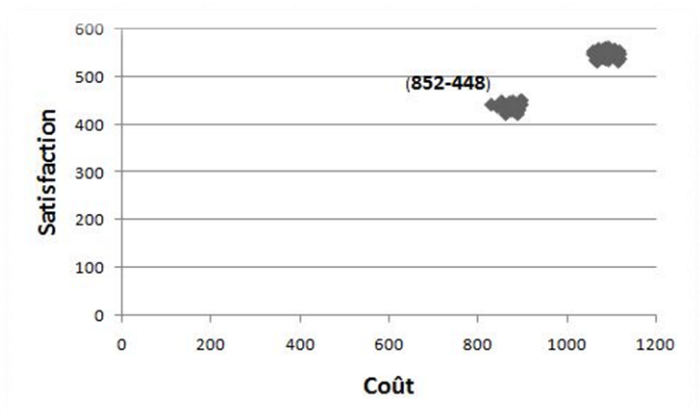


Figure 42 Graphe des solutions pour la sélection par rang

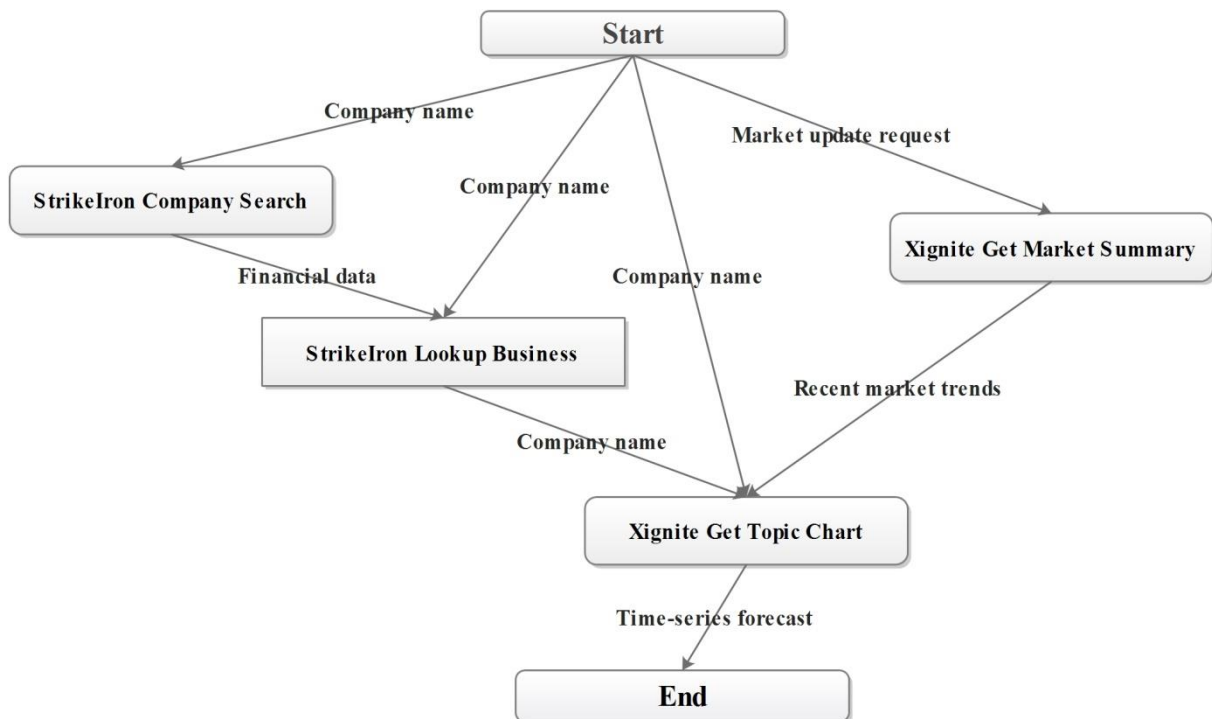


Figure 43 Conception de la meilleure solution de la sélection par rang

5) Sélection universelle stochastique

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
200	20	70 sur 7352	212 Secondes	(845-446)

Tableau 18 Résultat de la sélection SUS

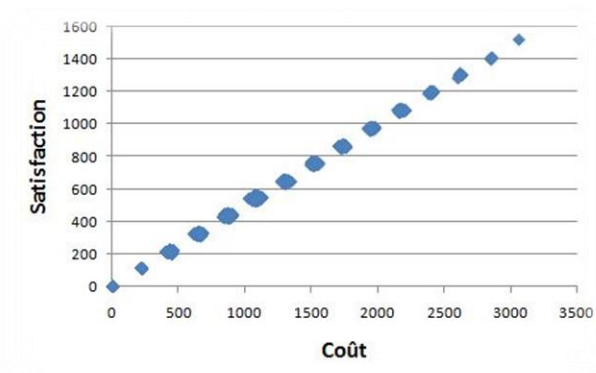


Figure 45 Graphe de toutes les générations

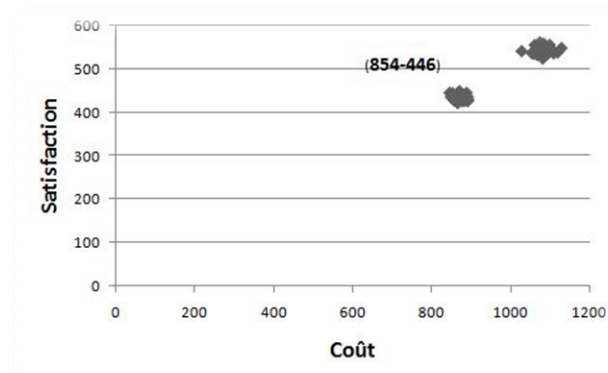


Figure 44 Graphe des solutions pour la sélection SUS

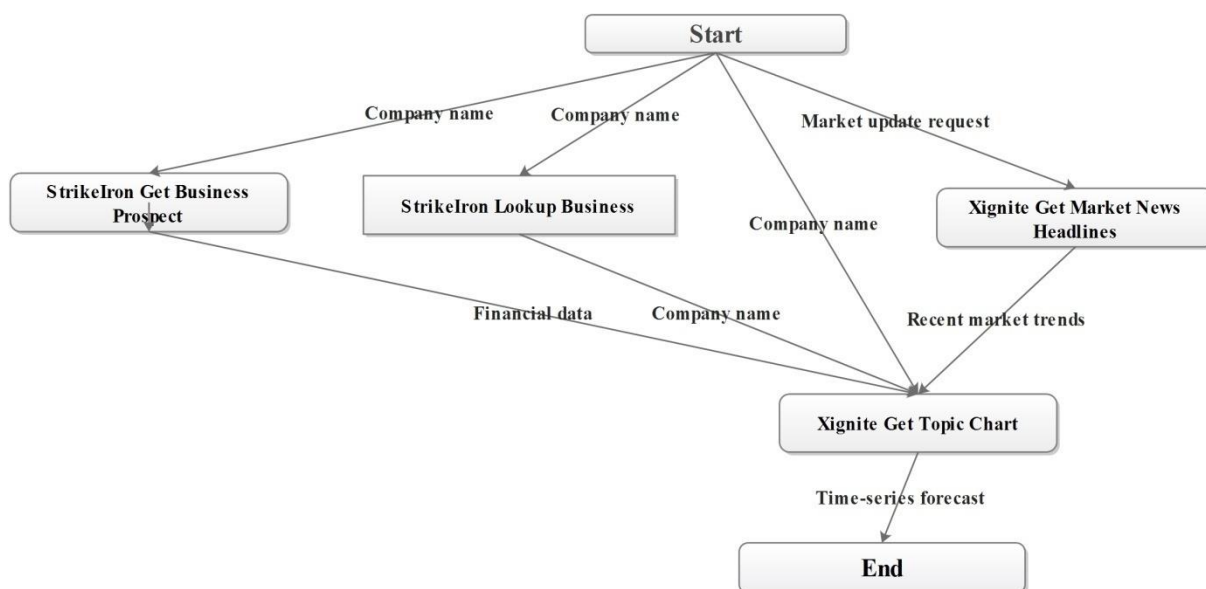


Figure 46 Conception de la meilleure solution pour la sélection SUS

6) Sélection d'échantillonnage & reste stochastique RSS :

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
500	20	172 sur 11376	64 Secondes	(845-446)

Tableau 19 Résultat de la sélection RSS

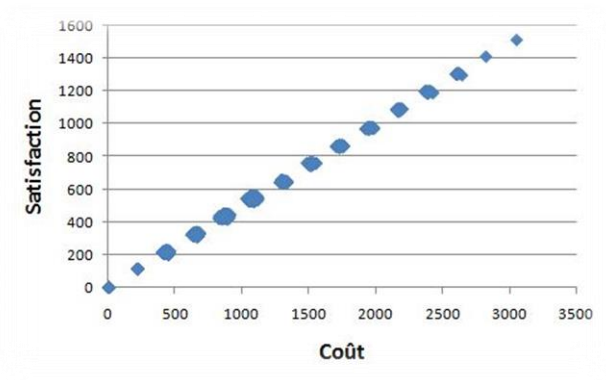


Figure 48 Graphe de toutes les générations

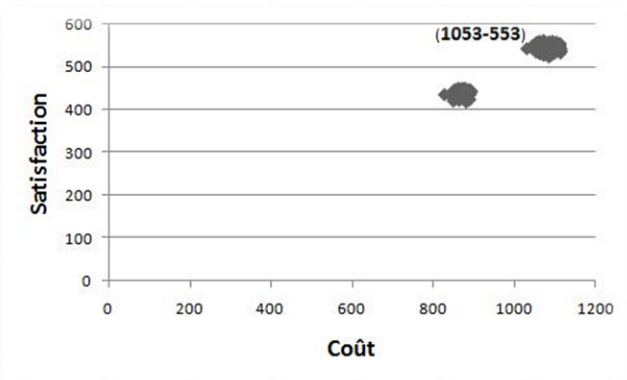


Figure 47 Graphe des solutions pour la sélection RSS

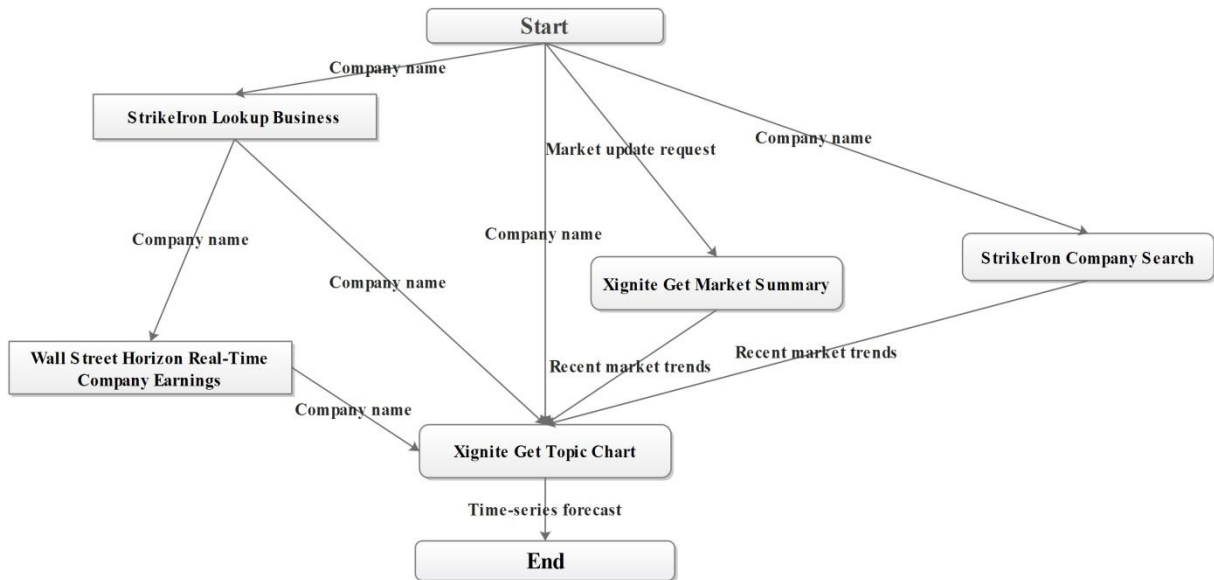


Figure 49 Conception de la meilleure solution pour la sélection RSS

7) Sélection Massale

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
200	20	72 sur 4400	142 Secondes	(844-447) (847-448) (852-448) (1052-554)

Tableau 20 Résultat de la sélection massale

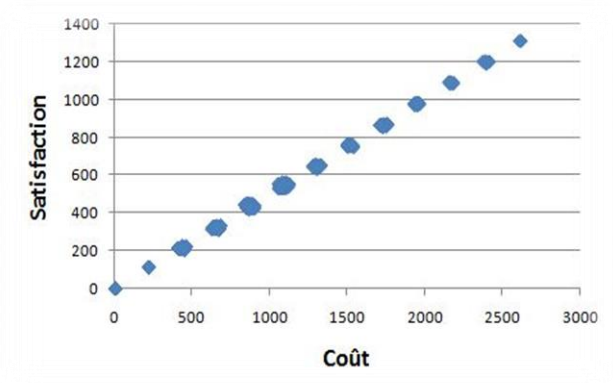


Figure 51 Graphe de toutes les générations

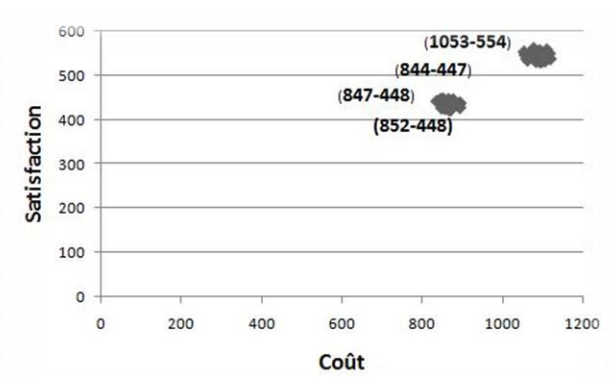


Figure 50 Graphe des solutions pour la sélection massale

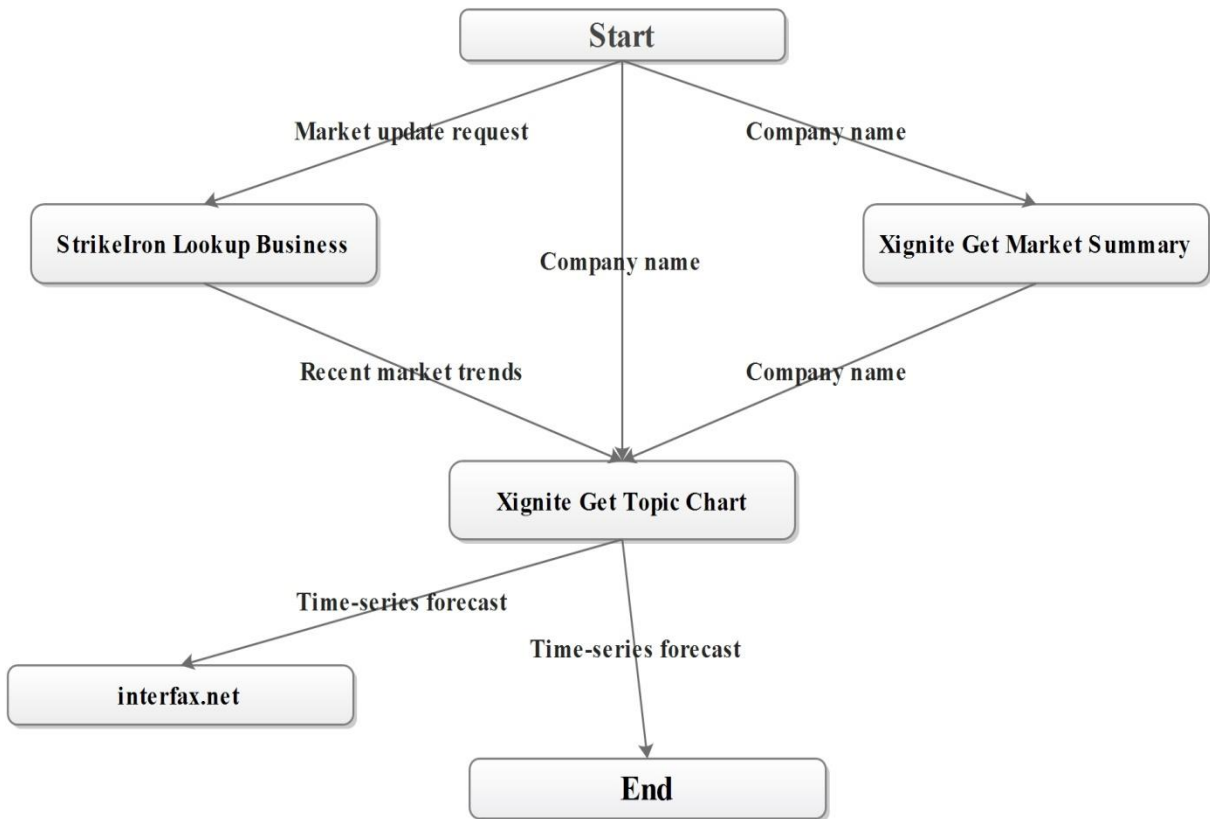


Figure 52 Conception de la meilleure solution (844-447) par massale

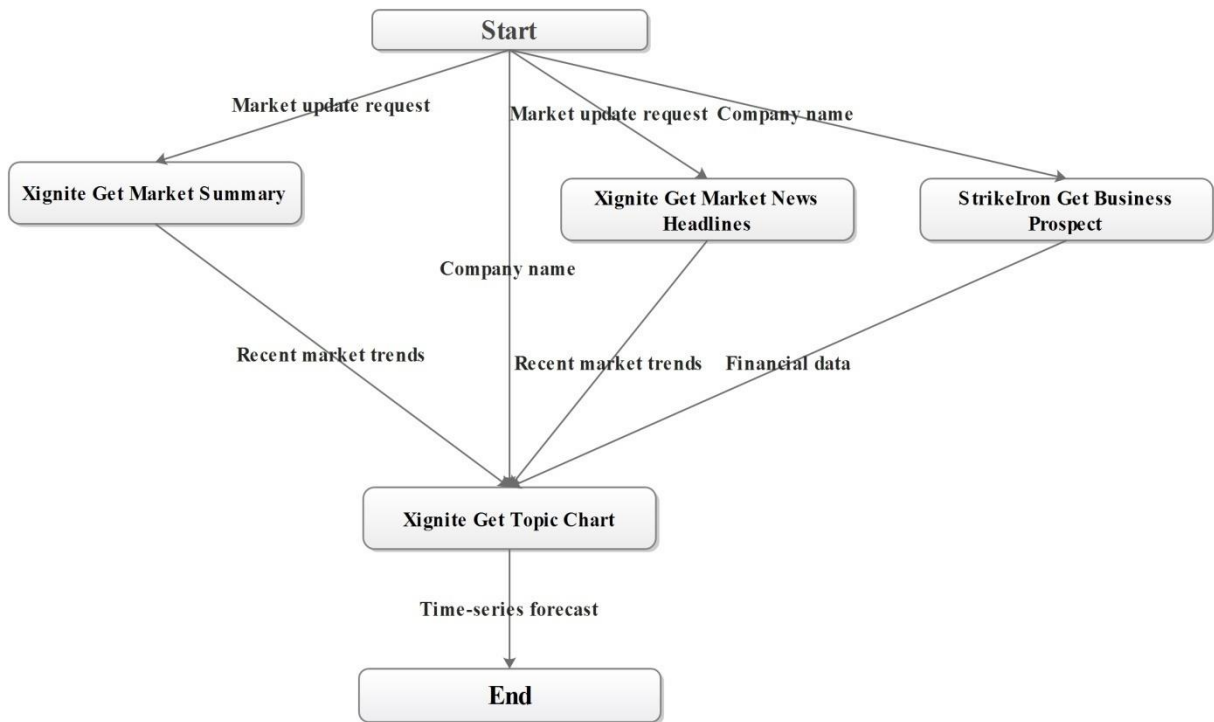


Figure 53 Conception de la meilleure solution (847-448) par massale

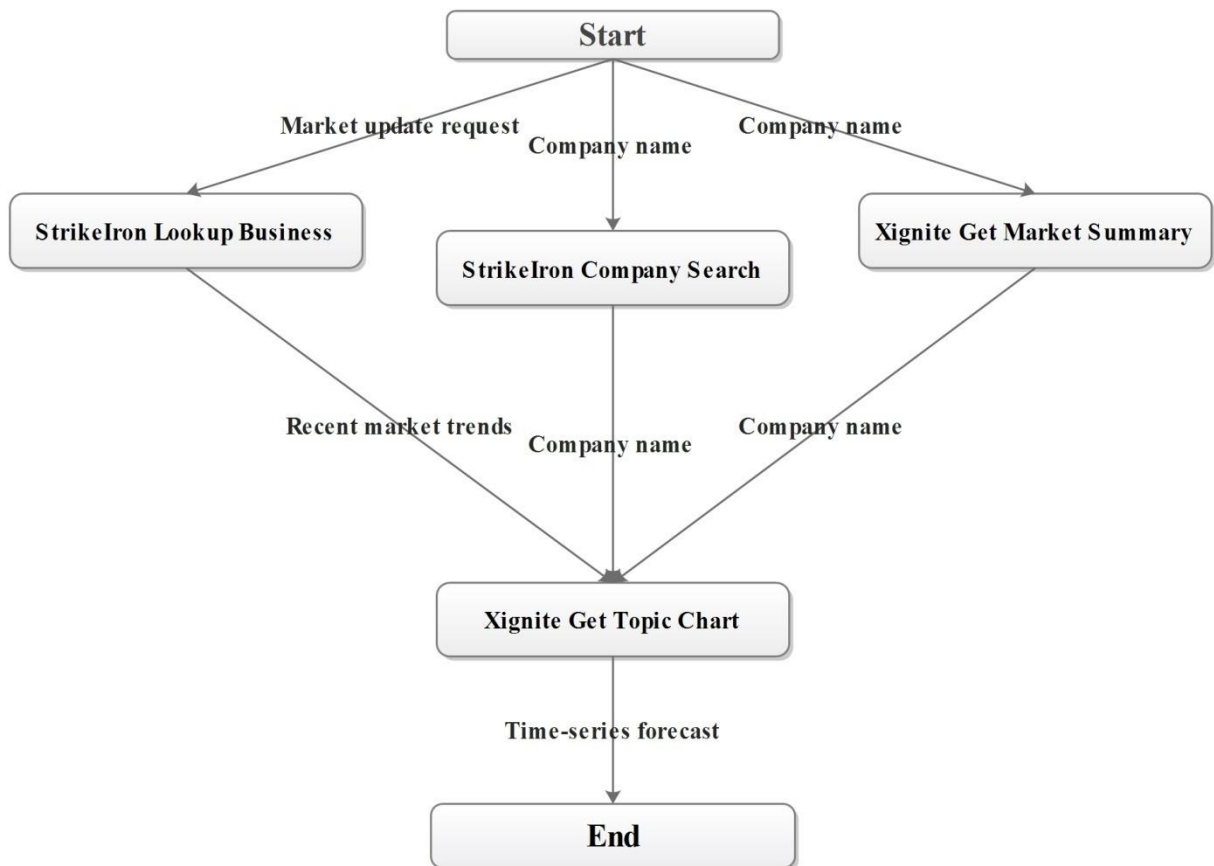


Figure 54 Conception de la meilleure solution (852-448) par massale

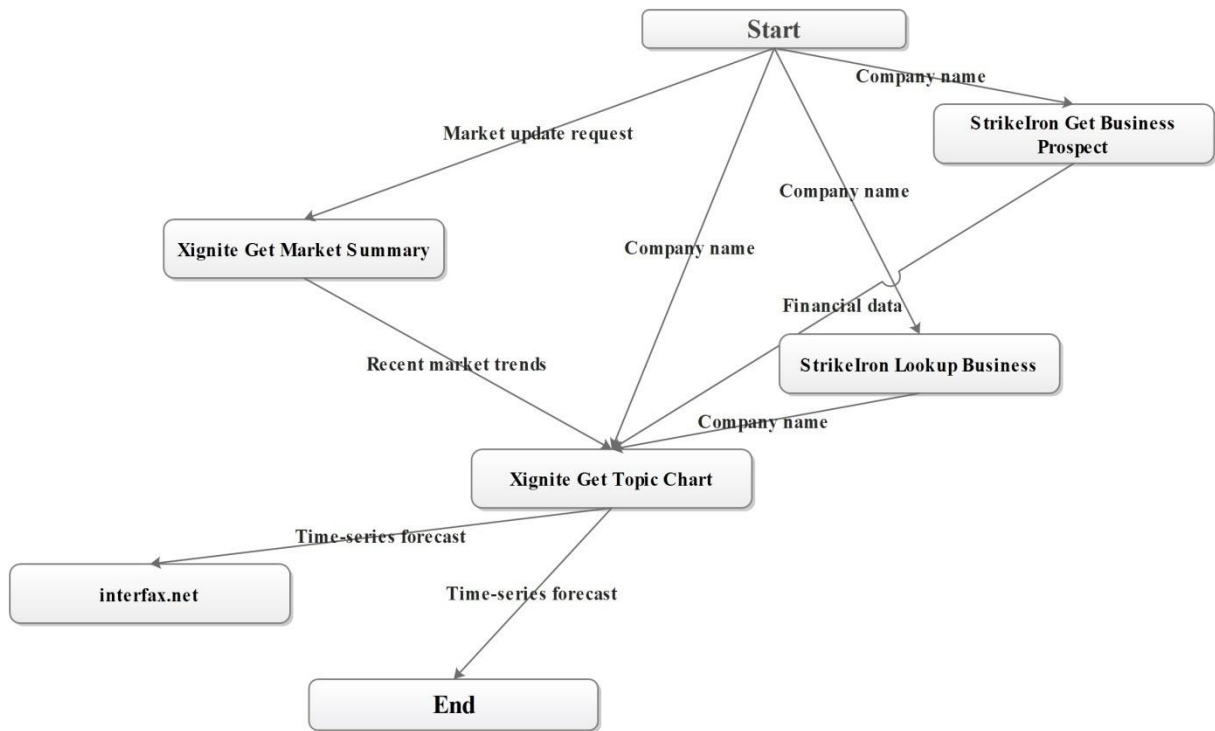


Figure 55 Conception de la meilleure solution (1053-554) par massale

8) Sélection sur ascendance

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
50	10	20 sur 559	169 Secondes	-

Tableau 21 Résultat de la sélection sur ascendance

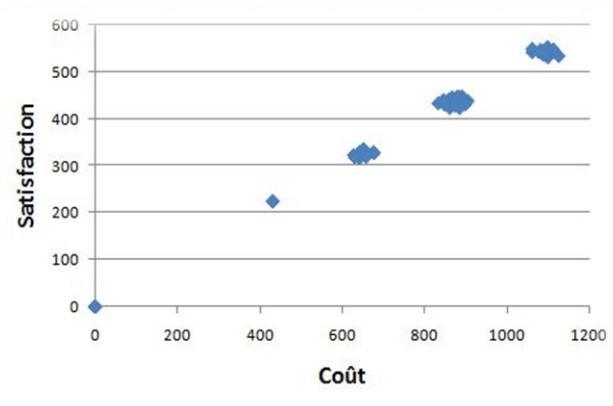


Figure 57 Graphe de toutes les générations

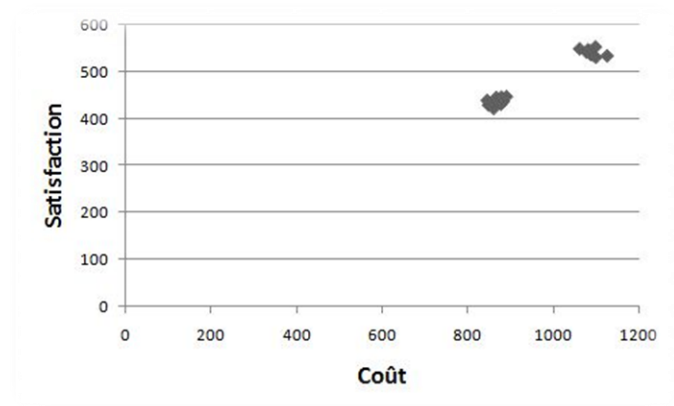


Figure 56 Graphe des solutions pour la sélection sur ascendance

9) Sélection sur descendance

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
50	10	14 sur 562	453 Secondes	-

Tableau 22 Résultats de la sélection sur descendance

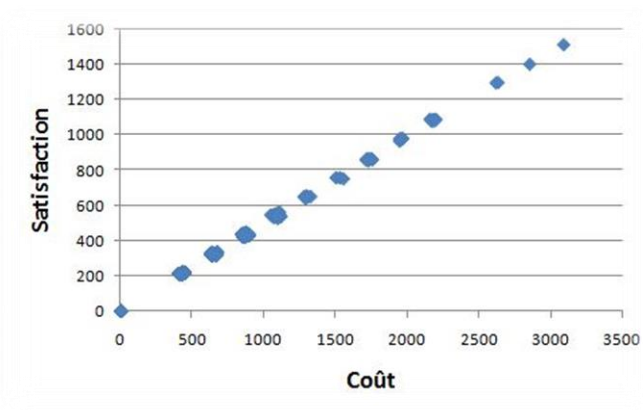


Figure 58 Graphe de toutes les générations

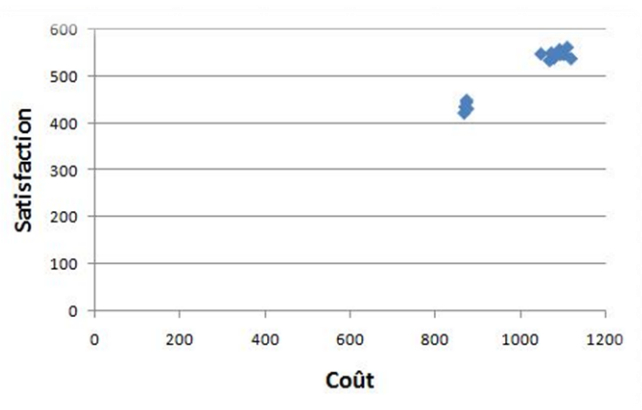


Figure 59 Graphe des solutions pour la sélection sur descendance

10) Sélection sur collatéraux

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
50	10	19 sur 552	73 Secondes	-

Tableau 23 Résultats de la sélection sur collatéraux

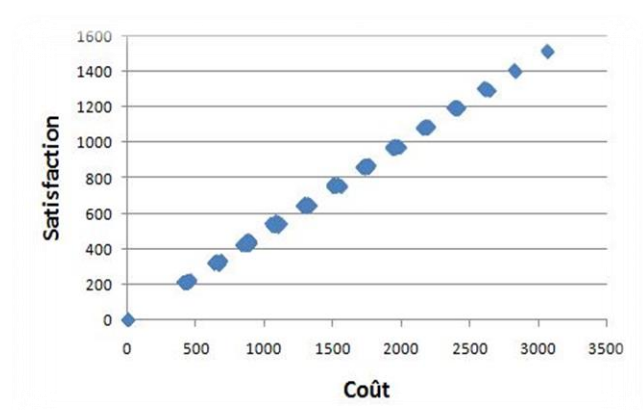


Figure 61 Graphe de toutes les générations

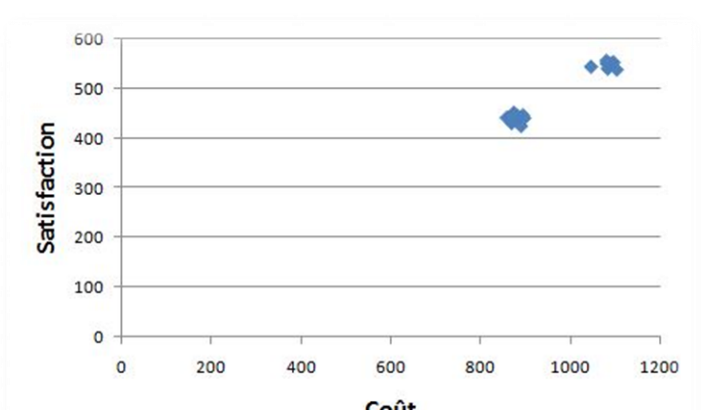


Figure 60 Graphe des solutions de la sélection sur collatéraux

11) Sélection par génération

Taille de population	Nombre de générations	Ratio (solution/individus)	Temps	Meilleures solutions
50	10	18 sur 556	36 Secondes	-

Tableau 24 Résultats de la sélection par générations

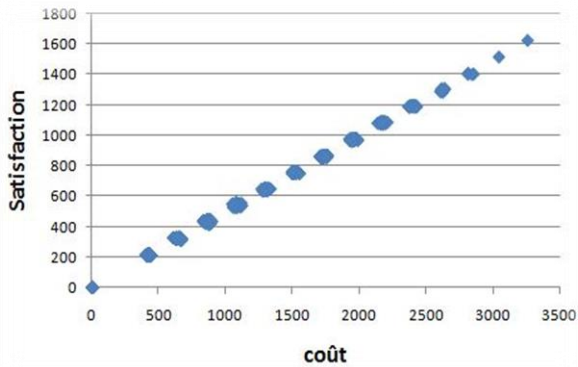


Figure 62 Graphe de toutes les générations

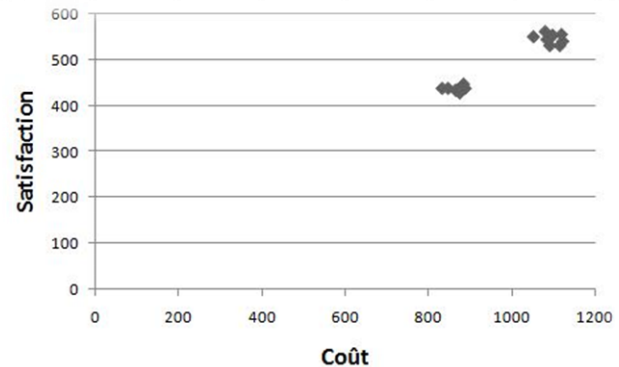


Figure 63 Graphe des solutions pour la sélection par génération

12) Récapitulation des résultats des sélections

Sélection	Taille de population	Nombre de génération	Ratio	Temps	Nombre de meilleures solutions
Uniforme	500	20	142/ 11426	49 s	1
Tournoi	500	20	157/ 10752	52 s	2
Par roulette	500	20	154/ 10554	360 s	1
RSS	500	20	172/ 11376	64 s	1
Par rang	200	20	77/ 4200	209 s	1
SUS	200	20	70/ 7352	212 s	1
Massale	200	20	72/ 4400	142 s	4
Sur ascendance	50	10	20/ 559	169 s	0
Sur descendance	50	10	14/ 562	453 s	0
Sur collatéraux	50	10	19/ 552	73 s	0
Par génération	50	10	18/ 556	36 s	0

Tableau 25 résultats de toutes les sélections

c. Comparaison entre les sélections (Discussions)

Il est clair que chaque sélection a ses propres caractéristiques et c'est exactement ce que nous avons affirmé dans la partie théorique (synthèse). Dans la suite du texte, nous nous concentrons sur les résultats du tableau 26.

Les résultats obtenus pour les sélections uniforme, par tournoi, RSS ont pu être trouvés avec près de 142 solutions, dans un temps court [49 64] secondes. Ceci avec une population initiale égale à 500 individus. Il faut garder en tête que la complexité de ces sélections est faible ($O(n)$).

Par contre, nous remarquons qu'avec les mêmes paramètres, la sélection par roulette a trouvé le même nombre de solutions, mais dans un temps plus important. Cette sélection est un peu lourde à agir, par rapport aux sélections précédemment citées, avec une complexité de ($O(n^2)$).

Après plusieurs essais, nous avons remarqué que la sélection par rang et la sélection universelle stochastique ont pu aboutir à des solutions faisables avec un nombre moyen d'individus. Ce qui explique l'utilisation de 200 individus, pour la taille de la population initiale. Nous sommes arrivés à conclure que la sélection par rang a pris le triple du temps consommé par la sélection uniforme, celle par tournoi et enfin ; RSS. Idem, avec une taille de population initiale inférieure à celle utilisée par les autres sélections. La consommation du temps pour ce type de sélection peut s'expliquer par la nécessité de trier la population initiale en premier lieu (ce qui augmente le temps consommé), puis choisir des individus selon leur rang.

Les sélections biologiques (excepté la sélection massale) n'ont pas pu trouver beaucoup de solutions faisables, ceci est causé en partie par le nombre réduit d'individus composant la population initiale (pour des raisons de complexité de traitement). Pour bien fonctionner, ces méthodes nécessitent un accès au fichier contenant le détail de chaque individu de la population, pour appliquer une recherche séquentielle pour les parents et les enfants de tous les individus.

Finalement, nous pouvons affirmer que du point de vue du temps d'exécution les sélections, uniforme, par tournoi, RSS sont les meilleures. Par contre, si nous nous intéressons au nombre de solutions trouvées, la sélection Massale et la sélection par tournoi ont poussé BPO Tool à arriver à trouver le plus de solutions

d. Comparaison des résultats obtenus avec résultat de Vergidis

Dans cette phase nous allons faire une comparaison avec les résultats obtenus par (Vergidis, 2008) selon les trois critères suivants : (nombre de tache, prix et satisfaction)

<i>Critères</i>	<i>Résultat de vergidis</i>	<i>Résultat de BPO Tool</i>
<i>Nombre de taches</i>	<i>4</i>	<i>4</i>
<i>Prix</i>	<i>853</i>	<i>844</i>
<i>Satisfaction</i>	<i>446</i>	<i>447</i>
<i>Taille de population</i>	<i>250</i>	<i>200</i>
<i>Nombre d'itération</i>	<i>25000</i>	<i>20</i>

Tableau 26 premier resultat

<i>Critères</i>	<i>Résultat de vergidis</i>	<i>Résultat de BPO Tool</i>
<i>Nombre de taches</i>	<i>6</i>	<i>5</i>
<i>Prix</i>	<i>1056</i>	<i>1053</i>
<i>Satisfaction</i>	<i>553</i>	<i>553</i>
<i>Taille de population</i>	<i>250</i>	<i>500</i>
<i>Nombre d'itération</i>	<i>25000</i>	<i>20</i>

Tableau 27 deuxième résultat

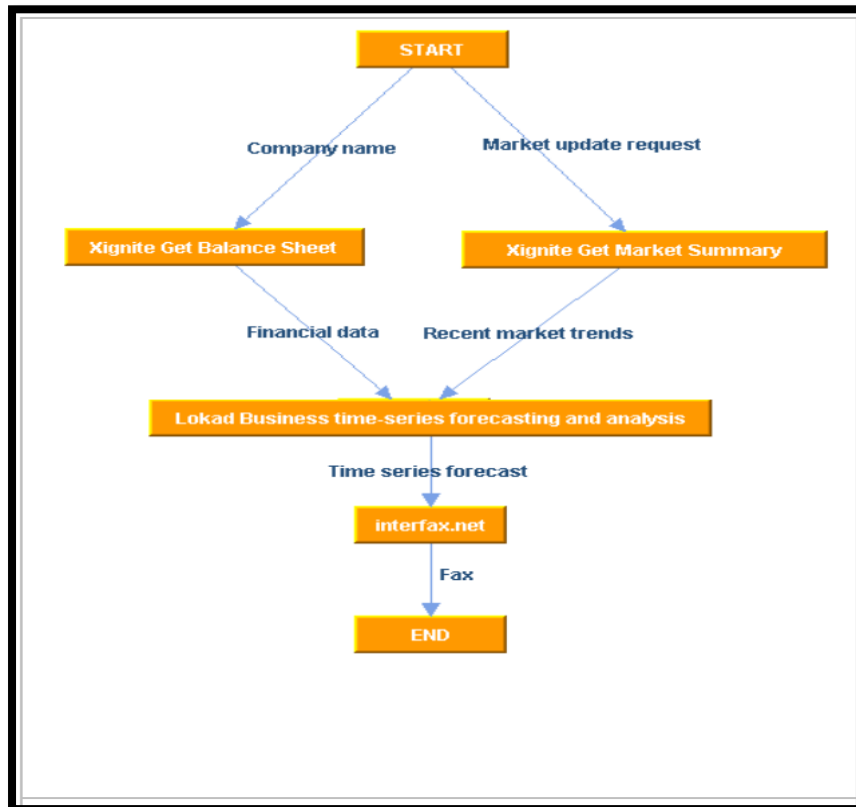


Figure 64 Solution 1 Vergidis

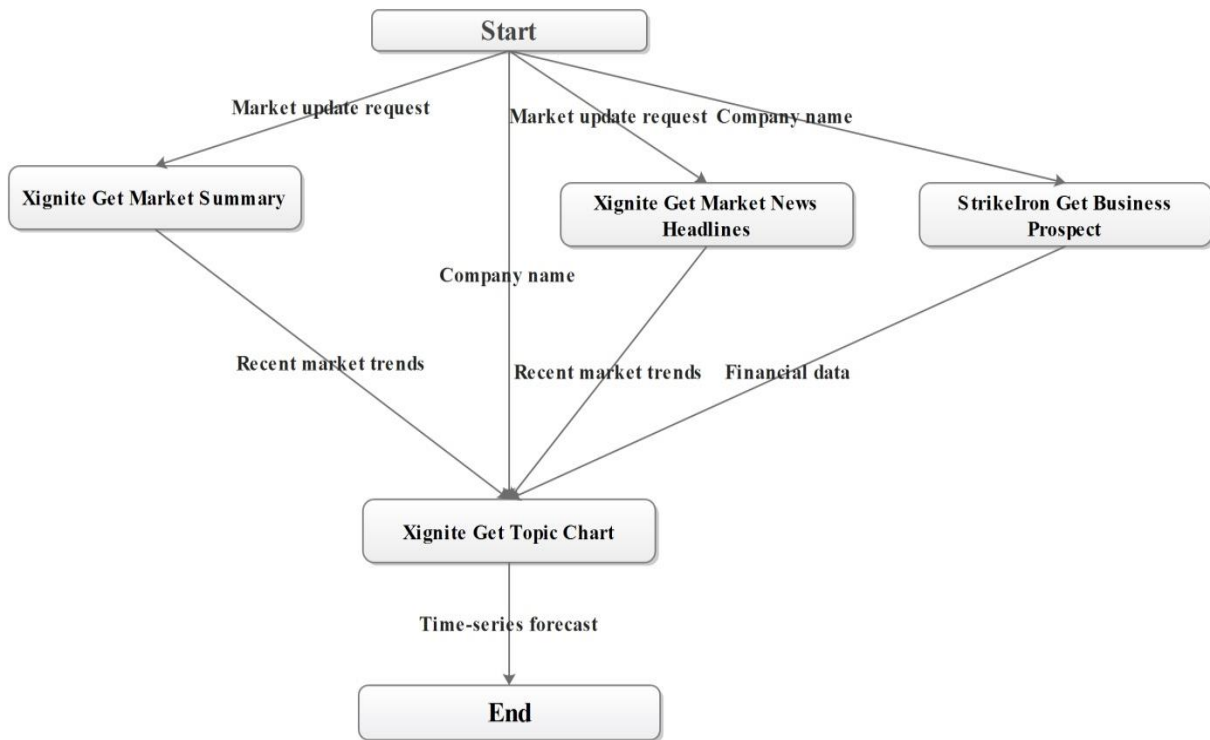


Figure 65 Solution à 4 taches avec sélection Massale BPO Tool

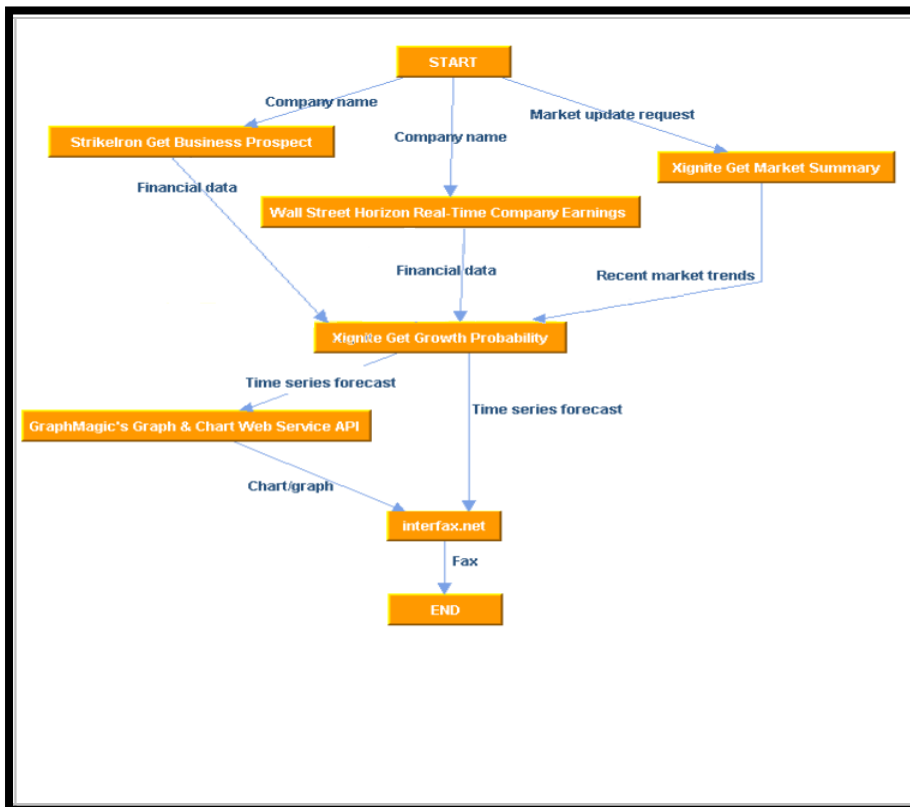


Figure 66 Solution 2 Vergidis

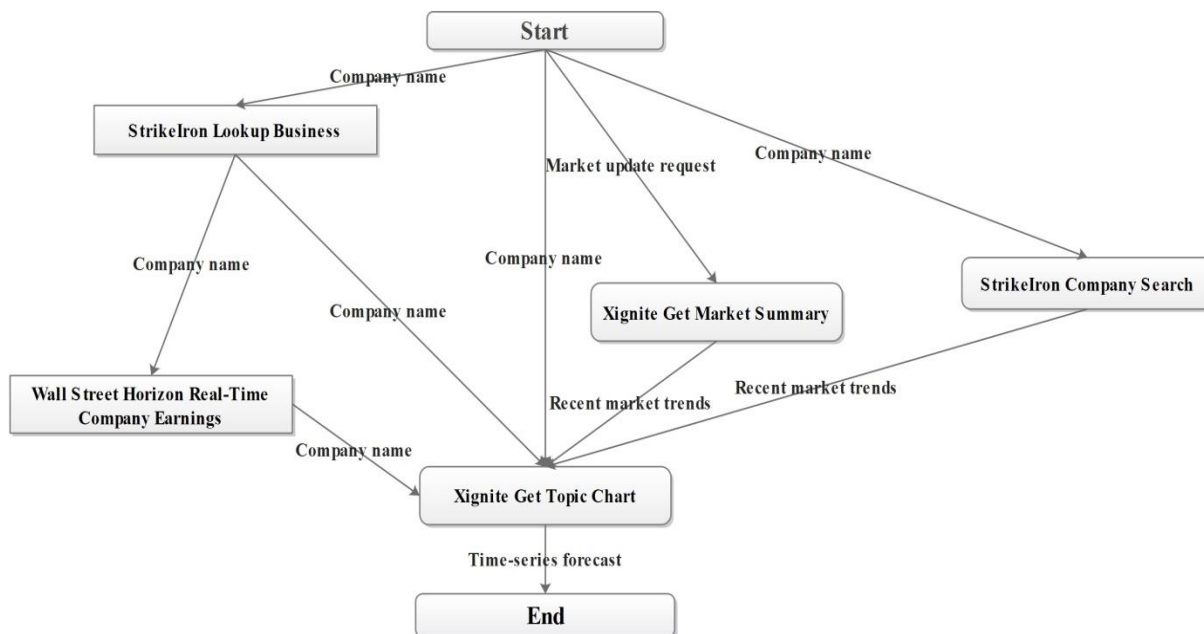


Figure 67 Solution à 5 tâches avec la sélection RSS avec BPO Tool

1) Interprétation des résultats

Nous avons pu trouver des meilleurs résultats que ceux proposés par (Vergidis K. T., 2015) en utilisant sept types de sélection qui sont la sélection uniforme, par roulette, par tournoi, universelle stochastique, échantillonnage à reste stochastique (RSS) et finalement la sélection Massale. Les paramètres utilisés pour chaque sélection pendant l'exécution du BP Tool sont regroupés dans le tableau 26.

BPO Tool a donné des résultats dans un temps raisonnable en utilisant les sélections (tournoi, uniforme, RSS) par contre la sélection par roulette a pris le triple du temps d'exécution. La sélection par tournoi de taille 5 a provoqué une convergence remarquable (figure 35).

Le fait de choisir un petit nombre d'itérations a été décidé après les différents essais que nous avons pu faire. Ce choix s'est imposé par la constatation de la création d'individus invalides (ne respectant pas la structure d'un processus métier) après un certain nombre d'itérations. Notre analyse nous conduit à déduire que l'opérateur de croisement en est la cause. Par sa nature, il provoque le changement de la structure du processus métier, ce qui provoque une élimination de ce dernier par les tests de vérification et de validation.

La sélection Massale a donné les meilleurs résultats par rapport aux autres sélections. Elle a pu nous aider à trouver plus de 72 solutions faisables avec 4 solutions meilleures que celle de (Vergidis K. T., 2015). La raison derrière ce constat est l'efficacité dans sa logique de fonctionnement. Cette sélection est basée sur les apparences que nous souhaitons obtenir dans la nouvelle génération. Pour cela, nous avons adapté cette méthode pour qu'elle choisisse un parent avec le coût le plus bas, la satisfaction la plus élevée et la durée la plus courte. Ce traitement nous a aidés à trouver quatre bonnes solutions par rapport à la solution trouvée par (Vergidis K. T., 2015).

III.4 Conclusion

Dans ce chapitre nous avons testé le potentiels de notre Framework en lui appliquent dans un cas réel (un scénario de test (Sales forecasting)). Nous avons aussi décrit notre Framework, leur fonctionnement, leur contenu.

Le chapitre III a montré les résultats trouvés par notre Framework. Ces résultats Là ont été comparés avec celui de (Vergidis K. T., 2015). Dans cette phase du mémoire nous avons prouvé que BPO_Tool a pu dépasser celle de Vergidis car elle a pu trouver une solution optimale qui est bonne que la solution trouvé par lui, en utilisent le même scénario et les mêmes paramètres.

Conclusion générale :

Notre thème vise à étudier l'application de l'optimisation multi-objective évolutive vers le domaine des processus métiers. L'optimisation multicritères des processus métiers est considérée comme le problème de conception des processus métier réalisables avec des valeurs d'attributs optimales telles que la durée et le coût. L'approche proposée consiste en l'application d'algorithmes d'optimisation multi-objectifs évolutifs (NSGA2), dans une tentative de générer une série diversifiée de processus métier optimisés pour les besoins d'un processus métier donné.

Dans ce rapport nous avons fait une étude bibliographique sur les concepts de bases qui contiennent notre travail (thème).

Cette étude est divisée en deux grandes parties essentielles : concepts de bases et les sélections.

La partie de concepts de bases a présenté le problème de l'optimisation multi-objectif en démontrant l'optimisation monocritère après l'optimisation multicritère, ensuite nous avons défini les processus métier et le rapport entre ces processus métier et l'optimisation multicritère.

Pour arriver à expliquer l'algorithme NSGA II qui est notre méthode pour optimiser les processus métier, nous avons utilisé une hiérarchie dans la définition des concepts en passant par c'est quoi les heuristiques, Métaheuristiques et ces classes, Les algorithmes évolutionnaires ensuite les algorithmes génétiques et finalement nous avons arrivé à présenter le principe de notre fameuse algorithme NSGA II.

Dans la deuxième partie nous avons présenté une des opérateurs les plus intéressants dans les algorithmes génétiques **La sélection**. Dans cette partie nous avons présenté les différents types de sélection existants dans la littérature. Et nous avons classé les méthodes de sélection en deux classes. (Implémentées et non implémentées).

Et en fin de cette partie nous avons réalisé une synthèse finale pour résumer tous les types trouvés.

Dans la partie des tests nous avons conclu que l'algorithme de création de la population initiale joue un rôle très important dans notre Framework car elle donne plus de possibilité d'installer un individu qui peut être une solution pour notre problème après.

Finalement nous pouvons dire aussi que les résultats obtenus avec les différents types de sélection ont soutenu les recherches que nous avons fait pour les sélections et qui ont été résumés dans la fin du chapitre II avec une synthèse.

IV. Perspectives

Vue la nature du problème traité et la multiplicité des composantes utilisées, plusieurs perspectives se présentent à nous. Nous citerons :

- Ajouter plus de trois critères.
- Développer la fonction Objective.
- Implémenter NSGA III.
- Développé l'efficacité des sélections biologiques
- Implémenter un croisement qui évite l'élimination des processus

V. Bibliographie :

- A.Spalanzani. (1999). *Algorithmes évolutionnaires pour l'étude de la robustesse des Systèmes de reconnaissance automatique de la parole*. thèse de doctorat,, Université Joseph Fourier-Grenoble, Grenoble.
- Adel, S. (2009). *Résolution des Problèmes Multi Objectifs à Base de Colonies de Fourmi*. Batna.
- Bäck, T. (1992). Self adaptation in genetic algorithms. In *Toward a Practice of Autonomous Systems*. MIT Press , 263-271.
- Baker, J. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms* , 14-21.
- Bentellis, A. (2010). *Intégration des Applications d'Entreprises Une approche basée objectif pour la gestion des processus métier flexibles*. thèse Doctorat informatique, université Mentouri , Constantine.
- Cheloud, R. (1999). *Généralités sur les méthodes d'optimisation*. université De Cergy, France.
- Darwin, C. (1876). *L'origine des espèces au moyen de la sélection naturelle ou la lutte pour l'existence dans la nature*. Paris: Éditions Reinwald.
- Deb, K. (2002). A Fast and Elitist Multiobjective Genetic Algorithm NSGA II. *IEEE Transactions on evolutionary computation* , 6 (2), 182-197.
- Dejong, K., & Spears, W. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence Journal* , 1-26.
- Dipama, J. (2010). *Optimisation Multi-Objectif des systèmes énergétiques*. Thèse Doctorat, Ecole polytechnique de Montréal , Montréal.
- E, D., Goldberg, & Deb, K. (1991). *A Comparative Analysis of Selection Schemes*. Urbana-Champaign: Morgan kaufmann publishers .
- E.Verrier, & X.Rognon. (2009). *Amélioration génétique des animaux*. Paris: Polycopié AgroParisTech.
- Ferchichi, A. (2008). *Contribution à l'intégration des processus métier :Application à la mise en place d'un référentiel qualité multi-vues*. école centrale de Lille et recherche Opérationnelle Innovation, Lille.
- Freitas, A. (2002). *Data mining and knowledge discovery with evolutionary algorithms*. Springer,.
- Goldberg. (1989). *Genetic algorithms in search, optimization, and machine learning*. Massachusetts: Addison Wesley Publishing Company, Inc.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*.
- Goldberg, D., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *G.J.E. Rawlins* , 69-93.
- Grigori, D. C.-C. (2004). Business Process Intelligence. , *Computers in Industry* , 53, 321-343.
- H. Osman, G. L. (1996). *Meta-Heuristics: Theory & Applications*. Kluwer Academic Publishers , 609.
- Hafedh Mili, G. T. (2004). *Business Process Modeling Languages: Sorting Through the Alphabet Soup*. rapport de recherche, Montréal.

- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introduction analysis with applications to biology*. University of Michigan Press.
- J. Zhong, X. H. (2005). Comparison of Performance between Different Selection Strategies on Simple Genetic algorithms. *Web Technologies and Internet Commerce* .
- Julstrom, B. A. (1999). *It"s All the Same to Me: Revisiting Rank-Based Probabilities and Tournaments*. Department of Computer Science, Cloud State University, Cloud State.
- Kheireddine, L. (2012). *Optimisation multi-objectifs par les algorithmes génétiques et application à la commande des systèmes*. DOCTORAT ES SCIENCES, Université Mentouri, Constantine .
- Kostas Vergidis, C. T. (2013). An automated optimisation framework for the development of re-configurable business processes: a web services approach. *International Journal of Computer Integrated Manufacturing* , 37-41.
- Lakin, R. C. (1996). 'BPR Enabling Software for the Financial Services Industry,. *Management Services* , 18-20.
- Mais HAJ-RACHID, C. B.-C. (2010). *Différentes opérateurs évolutionnaires de permutation: sélections, Croisement et mutation*.
- Manolis Koubarakis, D. P. (2001). Information Systems; A Formal Framework for Business Process Modelling and Design. *Elsevier Science Ltd* , 299-319.
- Mashohor, S., Evans, J. R., & Arslan, T. (2005). Elitist Selection Schemes for Genetic Algorithm based Printed Circuit Board Inspection System. 974-978.
- Miller, B., & Goldberg, D. E. (1995). Genetic Algorithms, Tournament Selection and the Effect of noise. *Complex System* , 193-212.
- O. A. Jadaan, L. R. (2005). Improved Selection Operator for GA . *Journal of Theoretical and Applied Information Technology* .
- S. Voß, S. M. (1999). *Meta-Heuristics - Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers .
- Siarry, C. (2004). *Un nouveau méta heuristique pour l'optimisation difficile : la Méthode des essais particuliers*,. université de paris, Paris.
- Th. Back, D. F. (1997). *Handbook of Evolutionary Computation* . England.
- Vergidis, K. T. (2015). An Automated Optimization Framework for the Development of Re-Configurable Business Processes: A Web Services Approach. *In-ternational Journal of Computer Integrated Manufacturing* , 28, 41-58.
- Vergidis, K. T. (2008). Business Process Optimisation using an Evolutionary Multi-Objective Framework. *Congress on Evolutionary Computing* , 668-675 .
- Whitley, D. (1994). A Genetic Algorithm Tutorial. *Statistics and Computing*. 65-85.
- Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank based allocation of reproductive trials is best. *Morgan Kaufmann publishers* , 116-121.

Résumé :

Dans notre travail nous avons proposé une approche pour l'optimisation multicritères automatisés des processus métiers.

le framework proposé utilise un modèle mathématique ; ce modèle est intégré dans le framework d'optimisation ainsi que les différentes méthodes d'optimisation.

nous avons fait une étude comparative en reprenant celui proposé par vergidis en utilisant l'algorithme NGSA.

Abstract :

In our work we have proposed an approach for the automated multicriteria optimization of business processes.

the proposed framework uses a mathematical model; this model is integrated .into the optimization framework as well as the various optimization methods we did a comparative study using the one proposed by vergidis using the NGSA .algorithm

خلاصة

لقد اقترحنا في عملنا نهجًا للتحسين الآلي متعدد المعايير لعمليات الأعمال. يستخدم الإطار المقترح نموذجًا رياضيًا ؛ تم دمج هذا النموذج في إطار التحسين بالإضافة إلى طرق التحسين المختلفة. أجرينا دراسة مقارنة باستخدام تلك التي اقترحها vergidis باستخدام خوارزمية NGSA.