

N° d'ordre :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR & DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITE DJILLALI LIABES
FACULTE DES SCIENCES EXACTES
SIDI BEL ABBES

THESE DE DOCTORAT

Présentée par

Nadir Bachir MAHAMMED

Spécialité : Informatique

Option : Système d'information et de connaissances

Intitulée

*Contribution à l'amélioration
des processus métiers d'entreprise*

Soutenue le 13/05/2018

Devant le jury composé de :

Président : Pr. Mohamed Kamel Feraoun

Examineurs : Pr. Mimoun Malki

Dr. Zohra Slama

Dr. Amar Bensaber Djamel

Dr. Djamel Berrabah

Directeur de thèse : Pr. Sidi Mohamed Benslimane

Professeur

Professeur

MCA

MCA

MCA

Professeur

UDL-SBA

ESI-SBA

UDL-SBA

ESI-SBA

UDL-SBA

ESI-SBA

Année universitaire : 2017/2018

Dédicace

A ma Maman. Mon exemple dans la vie.

A mon jeune frère. Chevalier des temps nouveaux.

Remerciements

Tradition oblige, il faut toujours commencer par remercier, dans l'ordre, le président du jury ; pour avoir consenti à présider cette soutenance, les membres du jury ; pour avoir accepté d'évaluer ce travail. Suivi du directeur de thèse, pour avoir convenu à travailler ensemble, et pour tout l'effort, le suivi, la patience, et les conseils durant ces années de recherche.

Chose faite, j'aimerais, maintenant, prendre le temps de remercier les personnes qui comptent le plus pour moi, et avec qui (plus que jamais) ce travail n'aurait jamais pu se faire, ni même commencer.

Au risque de me répéter, je remercie mon directeur de thèse, Pr. Sidi Mohamed BENSLIMANE, qui contre toute attente, a accepté de me prendre sous son aile, de m'encadrer et de m'orienter au mieux que puisse le faire ... le MEILLEUR des encadrants !

Ma famille (bonjour le cliché !). Ma Maman qui n'a cessé de me pousser et me supporter, même si – je le devine – a eu assez de la phrase "Je dois étudier !", merci pour tout Maman. Mon petit frère, pour tout le sang-froid, le calme et l'indulgence dont il a fait preuve envers mon comportement, sachant que plus d'une fois, j'ai été un vrai "casse-pied", merci Gringo. Ma jeune sœur, merci pour l'aide, le soutien et les conseils avisés, tout au long de ce périple. T'inquiètes Marco, je ne t'oublie pas, merci pour le coup de main (You know !), et toutes les discussions intéressantes, depuis le tout début. Ma grande sœur, merci pour l'indéfectible présence, ici et en France. Enfin, ma Fillette, merci merci merci, pour avoir supporté toutes ces années, la phrase assassine "LA THESE !", et d'avantage le mode "tortue ninja", merci ma Puce.

Un spécial remerciement à M. Kostas VERGIDIS, pour la simplicité avec laquelle l'échange scientifique s'est fait. L'adage dit vrai "La science n'a pas de patrie".

J'aimerais conclure mais bon, "La bêtise consiste à vouloir conclure". Je remercie toute personne ayant pu être impliquée, de près ou de loin, à la réalisation de ce travail. Merci *les mecs*.

Résumé

En réponse à un environnement de plus en plus volatile et compétitif, les entreprises examinent comment leurs processus métiers peuvent être repensés, afin d'améliorer leurs performances et leur réactivité vis-à-vis du marché. Cependant, il y a un manque - terrible - d'approches globales pour l'identification, l'analyse et l'optimisation des conceptions de processus métiers. Le but de cette recherche est de proposer une approche permettant de développer un Framework d'optimisation évolutionnaire multi objectifs (ou multicritères) des conceptions (ou instances) de processus métiers. Ce Framework est capable de (i) représenter des conceptions de processus métiers d'une manière quantitative, (ii) composer d'une manière algorithmique des conceptions de processus métiers en fonction d'exigences spécifiques. (iii) Identifier les conceptions de processus métiers optimales en utilisant une proposition d'un nouveau algorithme évolutionnaire. Enfin, (iv) évaluer la qualité des nouvelles conceptions optimisées de processus métiers en usant d'une fonction de Fitness capable de traiter jusqu'à trois (03) critères d'optimisation.

Cette recherche propose une représentation technique pour les conceptions de processus métiers en utilisant, à la fois, une perspective visuelle et quantitative. Elle propose également l'algorithme de vérification inverse des processus métiers - ***Rear Process Verification Algorithm - (ReProVA)*** -, qui est un algorithme pour la vérification et la recomposition de nouvelles conceptions de processus métiers. L'évaluation de ces conceptions se fait par l'intermédiaire d'une fonction de Fitness adaptée à l'optimisation multi objectifs proposée. Le Framework d'optimisation proposé emploie la représentation technique citée, ***ReProVA***, la fonction de Fitness et un algorithme évolutionnaire amélioré, pour une optimisation combinatoire multi objectifs. Il est capable de générer une série de conceptions alternatives optimisées de processus métiers, en fonction des exigences données, le tout se basant sur un modèle de processus métier proposé.

Enfin, un scénario de processus métier inspiré de la vie réelle est présenté et utilisé. Afin d'évaluer, à la fois, la capacité du Framework dans l'optimisation évolutionnaire multicritères d'un scénario de processus métier tiré de la vie réelle, et la performance du dit algorithme évolutionnaire amélioré. Les résultats de l'expérimentation ont démontré la capacité du Framework proposé à automatiser le processus de composition, d'identification, de vérification et d'optimisation des conceptions de processus métiers avec des valeurs d'attributs optimisées.

Abstract

In response to an increasingly volatile and competitive environment, companies are examining how their business processes can be redesigned to improve their performance and responsiveness to the market. However, there is a terrible lack of holistic approaches to the identification, analysis and optimization of business process designs. The aim of this research is to develop a multi objective (or multi criteria) evolutionary optimization framework for business process designs (or instances) capable of (i) representing business process designs in a quantitative way, (ii) composing in an algorithmic way business process designs according to specific requirements. (iii) Identify optimal business process designs using a proposal for a new evolutionary algorithm. Finally, (iv) evaluate the quality of new, optimized business process designs using a Fitness function capable of processing up to three (03) optimization criteria.

This research provides a technical representation for business process designs using both a visual and a quantitative perspective. It also offers the Rear Process Verification Algorithm (*ReProVA*) algorithm, which is an algorithm for the verification and recomposition of new business process designs. The evaluation of these designs is done through a Fitness function adapted to the proposed multi objective optimization. The proposed optimization framework is at the heart of this research. It uses the cited technical representation, *ReProVA*, the Fitness function and an improved evolutionary algorithm, for multi objective optimization. The framework is able to generate a series of optimized alternative business process designs, based on given requirements and a proposed business process model.

Finally, a business process scenario inspired from real life is presented and used. In order to evaluate both the framework's capability in multi-criteria evolutionary optimization of a real-life business process scenario and the performance of the enhanced evolutionary algorithm. The results of the experiment demonstrated the ability of the proposed framework to automate the process of identifying, verifying and optimizing business process designs with optimized attribute values.

ملخص

للاستجابة إلى بيئة متقلبة وتنافسية متزايدة ، تدرس الشركات كيفية إعادة تصميم عملياتها التجارية لتحسين أدائها واستجابتها للسوق .ومع ذلك، هناك نقص رهيب في النهج الشاملة لتحديد وتحليل وتحسين التصميم عملية التجارية .والهدف من هذا البحث هو وضع إطار للتحسين التطوري متعدد الأهداف (أو متعدد المعايير) لتصاميم العمليات التجارية(أو الحالات) قادر على (1) تمثيل تصاميم العمليات التجارية بطريقة كمية، (2) إنشاء الأعمال بطريقة خوارزمية عملية التصميم وفقا لمتطلبات محددة. (3) تحديد التصميم المثلى لعملية التجارية باستخدام اقتراح لخوارزمية تطويرية جديدة .وأخيرا، (4) تقييم نوعية تصاميم العمليات التجارية الجديدة والأمثل باستخدام وظيفة اللياقة البدنية قادرة على معالجة ما يصل إلى ثلاثة (03) معايير الأمثل.

ويوفر هذا البحث تمثيلا فنيا لتصاميم العمليات التجارية باستخدام منظور مرئي وكمي .كما يقدم خوارزمية التحقق من العملية الخلفية(*ReProVA*) ، وهو خوارزمية للتحقق وإعادة تشكيل تصاميم عملية التجارية الجديدة .ويتم تقييم هذه التصميم من خلال وظيفة اللياقة البدنية تتكيف مع التحسين المتعدد الهدف المقترح .إطار التحسين الأمثل هو في صميم هذا البحث .ويستخدم التمثيل الفني استشهد، *ReProVA* ، وظيفة اللياقة البدنية وخوارزمية التطويرية المحسنة، لتحسين موضوع متعددة .الإطار قادر على توليد سلسلة من التصميم عملية بديلة الأمثل، استنادا إلى متطلبات معينة ونموذج عملية الأعمال المقترحة.

وأخيرا، يتم عرض سيناريو عملية التجارية مستوحاة من واقع الحياة واستخدامها .من أجل تقييم كل من قدرة الإطار في متعددة المعايير الأمثل التطوري لسيناريو العملية التجارية واقع الحياة وأداء خوارزمية التطور المعزز .وأظهرت نتائج التجربة قدرة الإطار المقترح على أتمتة عملية تحديد التصميم العملية للأعمال والتحقق منها وتحسينها مع قيم السمات المحسنة.

Table des matières

Dédicace.....	2
Remerciements.....	3
Résumé	4
Abstract.....	6
ملخص	8
Table des matières	9
Liste des figures	13
Liste des tableaux	15
Introduction générale.....	1
1.1. Contexte	1
2.1. Problématique	2
3.1. Contribution.....	4
4.1. Feuille de route pour cette thèse	5
Chapitre 1 - Processus métier et optimisation multicritères	9
Introduction	10
1.1. Introduction aux processus métiers	10
1.1.1. Définition des processus métiers	11
1.1.2. Principaux éléments des processus métiers	14
1.1.3. Pourquoi automatiser les processus métiers ?	16
1.2. Optimisation évolutionnaire multicritères	17
1.2.1. Pourquoi l'optimisation multicritères ?.....	17
1.2.2. Introduction à l'optimisation évolutionnaire.....	19
1.2.3. Pourquoi utiliser l'optimisation évolutionnaire multicritères ?	20
Conclusion	21
Chapitre 2 - Algorithme évolutionnaire, NSGAI et opérateurs de sélection	24
Introduction	25

2.1.	Algorithmes évolutionnaires	25
2.1.1.	Les algorithmes génétiques	26
2.1.2.	Les stratégies évolutionnaires	26
2.1.3.	La programmation évolutive ou évolutionnaire	27
2.1.4.	La programmation génétique	27
2.1.5.	L'optimisation stochastique ou les algorithmes à estimation de densité	28
2.2.	Algorithmes génétiques et NSGAI.....	29
2.2.1.	Qu'est ce qu'un Algorithme génétique ?	29
2.2.2.	NSGAI plus qu'un simple algorithme génétique.....	33
2.2.3.	Fonctionnement général de NSGAI.....	34
2.3.	Sélection, plus qu'un opérateur	38
2.3.1.	Importance du choix de la sélection dans un algorithme génétique	38
2.3.2.	Techniques de sélection étudiées.....	39
2.3.3.	Synthèse.....	47
Conclusion		48
Chapitre 3 - Etat de l'art sur l'optimisation multicritères des processus métiers		51
Introduction		52
3.1.	Classification des méthodes de modélisation des processus métiers.....	52
3.1.1.	Les modèles schématiques.....	53
3.1.2.	Les modèles formels et/ou mathématiques	55
3.1.3.	Les langages des processus métiers	58
3.2.	Synthèse	60
Conclusion		64
Chapitre 4 - Représentation des processus métiers adaptée à l'optimisation multicritères.....		67
Introduction		68
4.1.	Spécification des processus métiers.....	68
4.1.1.	Définition adoptée pour une conception de processus métier.....	68
4.2.	Représentation proposée pour notre approche.....	70
4.2.1.	But et objectifs de l'approche proposée	70

4.2.2.	Représentation visuelle d'une conception de processus métier.....	71
4.2.3.	Représentation quantitative d'une conception de processus métier	72
4.2.4.	Matrice tâches/ressources	74
4.2.5.	Librairie des tâches.....	76
2.4.5.	Perspective visuelle contre perspective quantitative.....	77
4.3.	Algorithme de vérification inverse des conceptions de processus métiers	78
4.3.1.	Objectif de l'algorithme	79
4.3.2.	Exigences et résultats de ReProVA	81
4.3.3.	Principales étapes de ReProVA	82
4.3.4.	Algorithme pour "Elaborer un niveau fils"	86
4.3.5.	Algorithme pour "Mettre à jour le graphe et TRM"	87
Conclusion		89
Chapitre 5 - Framework pour l'optimisation multicritères des conceptions de processus métiers		91
Introduction		92
5.1.	Formulation du problème	92
5.2.	Challenges et défis de l'optimisation	94
5.3.	Framework d'optimisation proposé.....	96
5.3.1.	Fonctionnement du Framework.....	96
5.3.2.	Principales étapes du Framework	98
5.3.3.	Proposition de croisement.....	100
5.3.4.	Représentation d'une solution.....	108
Conclusion		109
Chapitre 6 - Expérimentation du Framework d'optimisation proposé.....		112
Introduction		113
6.1.	But du scénario réel de processus métier.....	113
6.1.1.	Contexte du scénario de la vie réelle.....	113
6.2.	Scénario test : La prévision des ventes	117
6.2.1.	Conception de processus métier initial.....	117
6.2.2.	Librairie des tâches.....	118

6.2.3.	Scénario de processus métier complet	119
6.3.	Fonction de Fitness proposée	120
6.3.1.	Pourquoi une - nouvelle - fonction de Fitness ?	120
6.2.3.	Fonction de Fitness utilisée par le Framework proposé.....	121
6.3.	Expérimentation du Framework proposé.....	122
6.3.3.	Paramètres du Framework proposé avec le scénario test.....	123
6.3.4	Evaluation et interprétation des résultats.....	123
Conclusion		133
Conclusion générale		136
7.1.	Principales observations	136
7.1.1.	Etude bibliographique.....	136
7.1.2.	Représentation des processus métiers	137
7.1.3.	Framework pour l'optimisation multicritères des processus métiers	138
7.1.4.	Algorithme évolutionnaire proposé	139
7.1.5.	Validation avec un scénario de processus métier réel	139
7.2.	Limites de la recherche	140
7.2.1.	Limites de la méthodologie de la recherche	140
7.2.2.	Limites de la représentation proposée.....	141
7.2.3.	Limites du Framework d'optimisation proposé.....	142
7.3.	Perspectives et recherches futures	143
Bibliographie		146

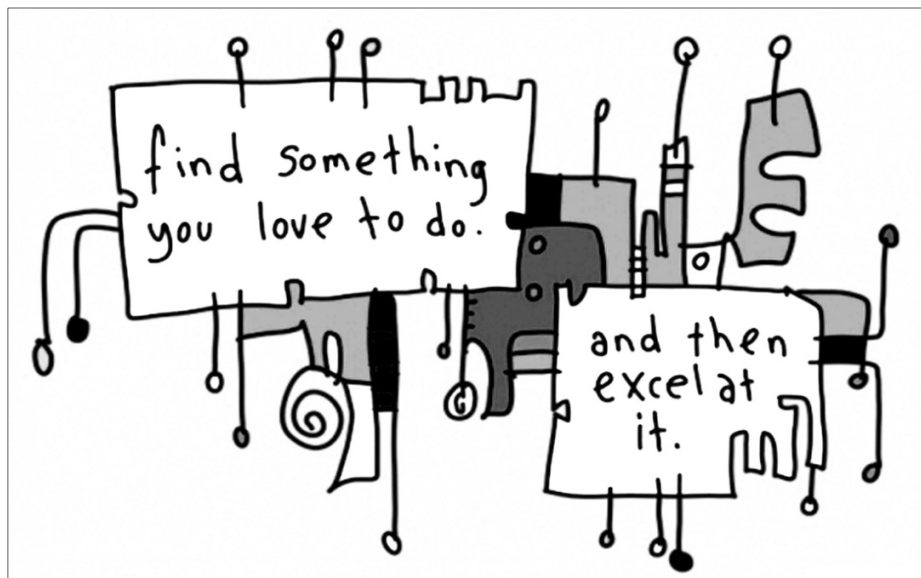
Liste des figures

Figure 1.1 Structure de la thèse.....	6
Figure 1.2 Meta modèle d'un processus métier	15
Figure 2.1 Exemple de croisement : Croisement à un point	32
Figure 2.2 Exemple de mutation : Mutation à un point	33
Figure 2.3 Fonctionnement général de NSGAI.....	35
Figure 2.4 Ranking dans NSGAI.....	36
Figure 2.5 Pseudo-code de NSGAI	36
Figure 2.6 Distance de peuplement.....	37
Figure 2.7 Procédure de la sélection par tournoi	40
Figure 2.8 Exemple de sélection par roulette.....	41
Figure 2.9 Procédure de sélection exponentielle par rang	44
Figure 2.10 Procédure de la sélection par troncation.....	45
Figure 3.1 Classification des techniques de modélisation des processus métiers	53
Figure 3.2 Représentation d'un processus métier en cube.....	56
Figure 3.3 Framework proposé par (Farsani et al., 2013).....	58
Figure 3.4 Correspondance entre le cycle de vie en BPM et un algorithme évolutionnaire	59
Figure 3.5 Processus métier de la gestion des exigences de crédit	59
Figure 3.6 Classification des travaux selon les techniques de modélisation pour processus métier.....	60
Figure 4.1 Exemple de représentation visuelle d'une conception de processus métier	72
Figure 4.2 Représentation proposée pour une tâche	73
Figure 4.3 Représentation détaillée proposée pour une conception de processus métier	75
Figure 4.4 Exemple de TRM pour représenter la Tâche 01	75

Figure 4.5 Exemple d'élaboration du graphe d'une conception de processus métier.....	80
Figure 4.6 Principales étapes de ReProVA.....	83
Figure 4.7 Principales étapes pour la composition et la vérification d'une conception de processus métier.....	85
Figure 4.8 Algorithme pour "Elaborer un niveau fils".....	86
Figure 4.9 Etapes pour "Mettre à jour le graphe et TRM".....	88
Figure 5.1 Composants du Framework d'optimisation proposé.....	96
Figure 5.2 Entrées et sorties du Framework proposé.....	97
Figure 5.3 Principales étapes du Framework proposé.....	100
Figure 5.4 Processus de la représentation en clés aléatoires.....	102
Figure 5.5 Croisement sélectif et mise à jour de la dominance.....	105
Figure 5.6 Recombinaison avec le croisement sélectif.....	106
Figure 5.7 Calcul du vecteur de dominance.....	107
Figure 5.8 Différentes formes d'une solution durant les étapes du Framework.....	109
Figure 6.1 Conception initiale du processus métier "La prévision des ventes".....	117
Figure 6.2 Représentation du rayon du cercle circonscrit du triangle CTS.....	122
Figure 6.3 Comparaison des solutions avec une taille = 4 tâches / scénario test.....	127
Figure 6.4 Comparaison des solutions avec une taille = 5 tâches / scénario test.....	128
Figure 6.5 Comparaison des solutions avec une taille = 6 tâches / scénario test.....	128
Figure 6.6 Résultats obtenus avec la sélection binaire par tournoi et le croisement proposé.....	129
Figure 6.7 Comparaison des techniques de sélection employées avec xNSGAIII selon la valeur de Fitness.....	130
Figure 6.8 Evolution de la Fitness des solutions en fonction des techniques de sélection utilisées.....	131

Liste des tableaux

Tableau 1.1 Définitions des processus métiers dans la littérature.....	14
Tableau 2.1 Terminologie utilisée avec un algorithme génétique.....	30
Tableau 2.2. Synthèse sur les types de sélection dans la littérature	46
Tableau 3.1 Synthèse des travaux sur l'optimisation multicritères des processus métiers ..	64
Tableau 4.1 Principaux paramètres mathématiques d'une conception de processus métier	73
Tableau 4.2 Exemple de matrice tâches/ressources	76
Tableau 4.3 Exemple d'une librairie des tâches avec trois attributs.....	77
Tableau 5.1 Paramètres du problème d'optimisation des conceptions de processus métiers	93
Tableau 5.2 Représentation d'une conception de processus métier en clés aléatoires	103
Tableau 5.3 Représentation d'un individu avec son vecteur de dominance	104
Tableau 6.1 Ressources disponibles pour le scénario test.....	118
Tableau 6.2 Librairie des tâches pour le scénario test	119
Tableau 6.3 Paramètres utilisés avec le scénario test	120
Tableau 6.4 Paramètres du Framework pour l'expérimentation du scénario test.....	124
Tableau 6.5 Comparaison des résultats avec ceux de (Vergidis et al., 2008, 2015)	125
Tableau 6.6 Comparaison des résultats pour le scénario test	126
Tableau 6.7 Améliorations apportées par la fonction de Fitness proposée.....	131
Tableau 6.8 Taux d'amélioration des résultats du Framework proposé.....	132



Introduction générale

1.1. Contexte

La bonne gestion d'une entreprise exige la connaissance, la compréhension et le meilleur alignement possible des processus métiers avec les objectifs de l'entreprise. La gestion de ces processus est connue sous le nom de "Gestion des processus métiers", appelée communément BPM, pour *Business Process Management*, et son intérêt est maintenant bien reconnu par toutes les entreprises (Hammer & Champy, 1993). BPM englobe un ensemble d'étapes, l'une d'elles, est "l'optimisation des processus métiers" - BPO, pour *Business Process Optimization* -. En vue d'accroître la rentabilité, hausser la productivité et augmenter les bénéfices, un nombre croissant d'entreprises s'appuie sur l'optimisation des processus métiers, et ainsi améliorer leurs performances, accroître leur efficacité et renforcer leur compétitivité. En mettant l'accent sur l'efficacité des processus, les entreprises développent des processus métiers plus souples, plus rapides et mieux intégrés dans la stratégie globale de l'entreprise.

En mettant en avant l'importance des processus métiers en entreprise, une des dernières tendances est l'automatisation de ces processus métiers (Castellanos et al. (2004)). Les avantages de cette automatisation sont multiples. La plus notable, est la possibilité d'exécuter les processus métiers plus rapidement, à moindre coût (en raison d'une intervention humaine réduite) et d'une manière contrôlée. Afin d'atteindre cette automatisation largement vantée par (Powell et al., 2001) (Grigory et al., 2001, 2004) (Zhou et Chen, 2003a) (Vergidis et al., 2008, 2012, 2015) et tant d'autres, l'optimisation multi objectifs est conseillée. Par optimisation multi objectifs, les auteurs visent les problèmes impliquant soit la minimisation ou la maximisation d'un ensemble d'objectifs, en raison du fait que la plupart des problèmes du monde réel impliquent naturellement des objectifs multiples et variés. Coello Coello (2005) explique clairement que la résolution des problèmes d'optimisation multi objectifs fait de plus en plus appelle aux

techniques de l'informatique évolutionnaire, de façon générale, et les algorithmes évolutionnaires, de façon particulière. La possibilité d'avoir une gamme de solutions optimales ; reflétant un compromis différent entre les différents critères d'optimisation pour des conceptions de processus métiers, est une raison plus que suffisante de penser à utiliser ces algorithmes dans le problème expliqué dans la section suivante.

2.1. Problématique

L'amélioration des processus métiers repose, en partie, sur les recherches sur les processus métiers, et en partie, sur les décisions - subjectives - mêlant l'élément humain. En outre, bon nombre de ces recherches sont sous-utilisées, car il est difficile de vérifier si les gens respectent - réellement - le processus optimisé suggéré. Il est à noter que la recherche sur les processus métiers se concentre principalement sur l'industrie manufacturière. Avec l'explosion de l'industrie des services sur le net, il serait bon de s'intéresser aux processus assistés par ordinateur, dans cette branche de l'industrie. Pour y arriver, il est nécessaire de faciliter le moyen d'automatiser la cartographie des processus métiers et d'automatiser - tout ou en partie - le développement des conceptions optimisées de processus métiers, au sein d'une entreprise. Les défis dans le domaine de l'industrie des services peuvent être brièvement résumés, comme suit :

- Capturer les processus métiers : La saisie automatisée des processus métiers nécessite une certaine forme d'*intelligence*, même dans un environnement de travail entièrement informatisé. En outre, il est difficile d'obtenir un modèle de processus qui correspond à tous (envers les acteurs qui y participent).
- Optimisation des conceptions de processus métiers : Compte tenu de leur nature qualitative, les conceptions de processus métiers sont difficiles à définir d'une manière formelle et adaptable aux méthodes analytiques.
- Difficulté d'atteindre l'optimum : Il est difficile de quantifier la conformité d'une conception de processus métier avec un optimum suggéré. De plus,

l'identification des optimums entre les conceptions de processus actuelles et obtenues est de nature *subjective*.

Mais ces défis - en mettant l'accent sur l'optimisation - peuvent également être considérés comme des arguments solides pour l'application des techniques d'optimisation évolutionnaires. Ces techniques utilisent les principes de l'évolution et offrent des avantages significatifs en termes de flexibilité et d'adaptabilité, avec de fortes performances. Le résultat de cette recherche sera une approche générique capable de :

- Automatiser la capture d'un processus métier.
- Optimiser la génération de conceptions de processus.
- Evaluer qualitativement et quantitativement les conceptions.

Le problème de cette recherche est formulé comme suit "Développer une nouvelle approche, pour l'optimisation multi objectifs des conceptions de processus métiers basée sur les techniques de l'optimisation évolutionnaire, qui est capable de traiter le problème de l'automatisation de la génération des conceptions optimisées de processus métiers".

Le développement d'une telle approche donne comme résultat un Framework d'optimisation des conceptions de processus métiers, qui pose un certain nombre de défis. Ce Framework nécessite une technique de modélisation ou de représentation qui permet de saisir les principales caractéristiques d'un processus métier et de les exprimer de sorte à pouvoir l'utiliser avec des méthodes d'optimisation. Il suppose une approche algorithmique pour composer de nouvelles conceptions de processus métiers et une mesure quantitative de leur évaluation. Enfin, il présente des techniques d'optimisation évolutionnaires – ayant fait leurs preuves dans d'autres problèmes d'optimisation - et les applique dans le contexte des processus métiers.

3.1. Contribution

Ce travail présente une approche nouvelle et originale pour la génération de conceptions de processus métiers réalisables basées sur un modèle de processus métier, dans un véritable environnement d'optimisation multi objectifs, allant jusqu'à trois (03) critères d'optimisation. La génération de conceptions de processus métiers est gérée par un algorithme évolutionnaire amélioré, tout en vérifiant que la faisabilité de ces conceptions est assurée par l'utilisation d'un algorithme efficace. Pour ce faire, cette approche se concentre sur l'utilisation d'un processus métier basé sur les activités, i.e. les tâches qui composent une conception d'un processus métier (pour la modélisation), en raison de l'abondance des travaux qui l'utilisent dans la littérature. En plus des activités, les auteurs ont fait le choix d'utiliser les attributs de ces activités (e.g. le coût en ressources, la durée d'exécution et la satisfaction vis-à-vis d'un client) tout en négligeant d'autres composants caractéristiques des processus métiers, pour l'évaluation des solutions. L'optimisation des processus métiers est une question difficile en raison de la nature non linéaire et des modèles mathématiques souvent discontinus impliqués (Salomie et al., 2012). Pour sa part, Tiwari et al. (2010a) affirment qu'il y a relativement peu de recherches sur l'optimisation des processus métiers travaillant sur les tâches participantes. Les contributions de ce travail sont triples. Tout d'abord, la vérification de la faisabilité des conceptions des processus métiers générées est fournie par l'algorithme de vérification inverse des processus métiers (*ReProVA*). Deuxièmement, ce travail utilise un algorithme évolutionnaire adapté et amélioré (*xNSGAI*), pour la génération de ces conceptions de processus métiers. Troisièmement, les critères d'optimisation utilisés dans l'optimisation multi objectifs sont au nombre de trois (03) : Les coûts en ressources, la durée d'exécution d'une conception et la satisfaction du client envers une conception du processus métier.

4.1. Feuille de route pour cette thèse

La structure de cette thèse suit les principales étapes de cette recherche, comme le montre la figure 1.1. Une brève description de chaque chapitre est fournie ci-dessous.

L'introduction générale traite du contexte de cette recherche, expliquant brièvement son but et ses objectifs. L'énoncé du problème et la motivation de cette recherche sont décrits avec la contribution suggérée.

Le chapitre 1 présente une étude de la littérature dans le domaine de la définition et la modélisation des processus métiers. Par la suite, l'optimisation multi objectifs est expliquée, ainsi que l'importance qu'elle apporte dans les problèmes liés à l'optimisation des processus métiers.

Le chapitre 2 présente une étude de la littérature dans le domaine des métaheuristiques, plus particulièrement, les algorithmes évolutionnaires. Suivi, le détail de l'algorithme génétique, NSGAII, et un ensemble de techniques de sélection - potentiellement – utilisables, dans un problème d'optimisation multi objectifs.

Le chapitre 3 se consacre à présenter l'état de l'art des travaux portant sur l'optimisation multi objectifs des processus métiers avec des métaheuristiques. Une proposition de classification de ces travaux par rapport aux techniques de modélisation des processus métiers est présentée, puis discutée.

Le chapitre 4 introduit l'approche proposée pour l'optimisation des conceptions de processus métiers. La représentation adoptée, pour capturer et exprimer, une conception d'un processus métier, pour l'optimisation, est décrite. Le chapitre propose, également, de décrire l'algorithme de vérification inverse des conceptions. (**ReProVA**) est expliqué, ainsi que les différentes étapes le composant.

Le chapitre 5 présente le Framework proposé pour l'optimisation des conceptions des processus métiers. Le Framework utilise la technique de représentation des processus métiers proposée, l'algorithme de vérification inverse des conceptions et un algorithme évolutionnaire amélioré, *xNSGAI*. L'algorithme est détaillé et utilisé, afin de générer des conceptions optimisées des processus métiers.

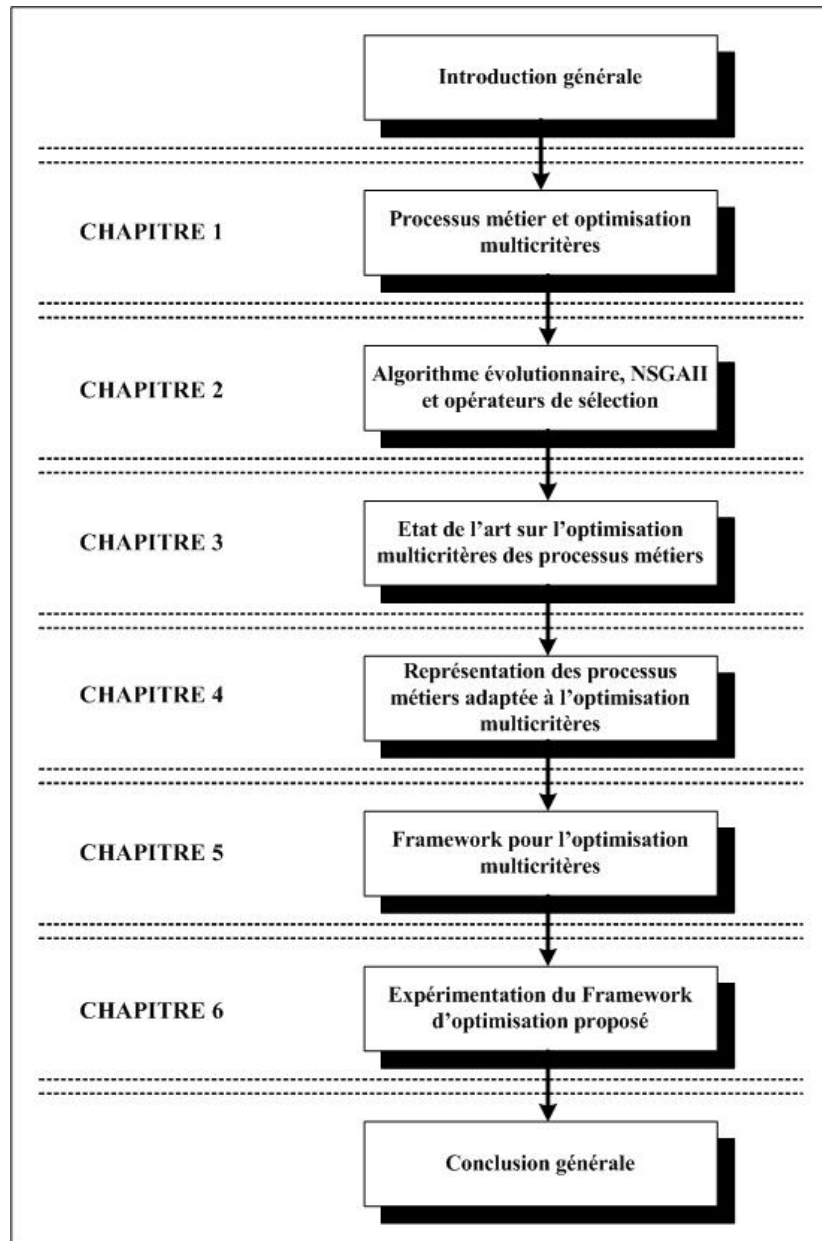


Figure 1.1 Structure de la thèse

Le chapitre 6 valide cette recherche en utilisant un scénario de processus métier réel : Le processus métier de "prévisions de ventes". Le scénario est présenté, avant que le Framework d'optimisation proposé soit appliqué sur ce dernier. Les résultats sont analysés, comparés et discutés, par la suite.

La conclusion générale parachève cette thèse avec une discussion sur la généralité de cette recherche, la contribution et les limites de la méthodologie de la recherche, de la technique de représentation et le Framework d'optimisation proposé. Ce chapitre discute, également, des futures recherches qui découlent de ce travail.



Chapitre 1

Processus métier et optimisation multicritères

Introduction.....	10
1.1. Introduction aux processus métiers	10
1.1.1. Définition des processus métiers	11
1.1.2. Principaux éléments des processus métiers	14
1.1.3. Pourquoi automatiser les processus métiers ?	16
1.2. Optimisation évolutionnaire multicritères	17
1.2.1. Pourquoi l'optimisation multicritères ?.....	17
1.2.2. Introduction à l'optimisation évolutionnaire.....	19
1.2.3. Pourquoi utiliser l'optimisation évolutionnaire multicritères ?	20
Conclusion	21

Introduction

En réponse à des environnements de plus en plus volatils et compétitifs, les entreprises examinent leurs processus métiers afin de les reconstituer, pour améliorer la performance de ces entreprises et la réactivité du marché. La conception et la gestion des processus métiers est un facteur clé permettant aux entreprises de pratiquer une concurrence efficace, dans l'environnement commercial instable d'aujourd'hui. En se concentrant sur l'optimisation et l'amélioration continue des processus métiers, les entreprises peuvent mettre en place un solide avantage concurrentiel en réduisant les coûts, en améliorant la qualité et l'efficacité, et en permettant une meilleure adaptation aux changements des besoins. Cette recherche porte sur la re-conception des instances de processus métiers grâce à une optimisation multicritères.

Ce chapitre traite de la notion de processus métier, et ceci via une étude comparative des multiples définitions des processus métiers présentes dans la littérature. Ainsi, que le pourquoi de la nécessité d'adopter sérieusement l'automatisation de ces processus métiers. Nous continuons notre investigation sur l'optimisation multicritères des processus métiers avec les métaheuristiques, cette fois sous l'angle de l'informatique évolutionnaire. Nous présentons une idée générale de ce que ce concept (l'optimisation multicritères) implique, ses nombreuses orientations et voix de recherche, et l'importance qui en découle, dans la problématique traitée.

1.1. Introduction aux processus métiers

Bien que le concept de processus métier soit assez jeune (Davenport and Short, 1990) (Hammer and Chappy, 1993), les définitions pullulent. Chaque définition confortant la précédente mais, tout en y apportant son lot de nouveautés. Cette section présente le point central de cette recherche : Le processus métier. Elle présente les différentes définitions des processus métiers existant dans

la littérature, en argumentant sur le choix porté sur la définition adoptée, dans la suite de notre recherche.

Avant d’aller plus loin, il faudrait souligner que nous considérons un processus métier comme un processus à part entière, mais il est différent dans le sens où il effectue une opération commerciale (business). Une confusion pourrait subsister entre la nature d’un processus métier et celle d’un workflow. Mieux vaudrait mettre les choses au clair. Un workflow étant un concept similaire, plus vieux que celui de processus métier. Un workflow n’est pas limité au contexte de business, bien que ce soit l’une de ses applications populaires (Basu and Blanning, 2000). Cependant, selon Wang and Wang (2005), les approches de workflow traditionnelles sont trop rigides pour correspondre à des activités commerciales complexes et dynamiques en raison du manque de flexibilité et d’adaptabilité.

1.1.1. Définition des processus métiers

Depuis les années 1990, lorsque les premières définitions des processus métiers sont apparues dans la littérature, de nombreux auteurs ont tenté de présenter leur propre version -améliorée- de la définition des processus métiers, en général avec un seul but ; celui de tenter d’orienter les processus métiers vers une direction particulière en soulignant un aspect spécifique et unique (Vergidis, 2008). Cependant, dans presque toutes les références étudiées, les auteurs citent les définitions de processus métiers présentées par Hammer et Champy (1993) et Davenport (1993). Tableau 1.1 présente un résumé reflétant la variété et la diversité des différentes définitions des processus métiers qui existent dans la littérature.

Bien que la plupart des définitions ont tendance à être similaires dans les concepts utilisés pour décrire un processus métier (activité ou tâche, entrée, sortie), ils ont reçu des critiques pour ne pas mettre en évidence la composante « entreprise » et pas suffisamment distinguer « des procédés de fabrication » ou de « production » (Vergidis and Tiwari, 2008).

Auteur	Définition
Pall (1987)	L'organisation logique des personnes, des matériaux, de l'énergie, de l'équipement et des procédures dans les activités de travail visant à produire un résultat final spécifique.
Davenport and Short (1990)	Un ensemble de tâches logiquement liées exécutées pour atteindre un résultat métier défini.
Davenport (1993)	La chaîne d'activités ayant comme finalité la production d'un produit spécifique pour un client ou un marché particulier.
Hammer and Chappy (1993)	Une collection d'activités qui prend un ou plusieurs types d'entrées et crée une sortie qui ait de la valeur pour le client. Il a un but et est affecté par des événements se produisant dans le monde extérieur ou dans d'autres processus.
Johanson et al. (1993)	Un ensemble d'activités liées qui prend une entrée et la transforme pour créer une sortie. Il doit ajouter de la valeur à l'entrée et créer une sortie qui est plus utile et efficace pour le destinataire.
Jacobson et al. (1994)	L'ensemble des activités internes à valeur ajoutée réalisées pour servir un client.
Soliman (1998)	Un réseau complexe d'activités reliées entre elles.
Agerfalk (1999)	Il se compose d'activités commandées de manière structurée dans le but de fournir des résultats précieux au client.
WMC (1999)	Un ensemble d'activités liées et ordonnées pour atteindre un objectif commercial dans le contexte d'une structure organisationnelle
Volkner and Werners (2000)	Une séquence d'états, qui résultent de l'exécution d'activités dans les organisations pour atteindre un certain objectif.
Dayal et al. (2001)	Une unité de travail de persistance commencée par un événement commercial, e.g. une facture ou une demande de transfert de fonds.
Fan (2001)	Un ensemble d'une ou plusieurs procédures ou activités liées qui réalisent collectivement un objectif commercial ou un objectif politique, normalement dans le contexte d'une structure organisationnelle définissant les rôles et les relations fonctionnels.
Stock and Lambert (2001)	Une structure d'activités conçue pour l'action, axée sur le client final et la gestion dynamique des flux impliquant des produits, des informations, des liquidités, des connaissances et des idées.
Stohr and Zhao (2001)	Une séquence d'activités. Il a des entrées et des sorties distinctes et sert un but significatif au sein d'une organisation ou entre les organisations.
Gunasekaran and Kobu (2002)	Un groupe de tâches connexes qui, ensemble, créent de la valeur pour un client.

Irani <i>et al.</i> (2002)	Un ordre dynamique des activités de travail à travers le temps et le lieu, avec un début, une fin, des entrées et des sorties clairement identifiées.
Zhou and Chen (2002)	Il est soit un processus optionnel, de soutien et de gestion avec des relations sémantiques complexes, y compris des activités séquentielles, concurrentes et des jonctions logiques
Castellanos <i>et al.</i> (2004)	Un ensemble d'activités qui collectivement atteignent un objectif commercial.
Havey (2005)	Un ensemble de règles spécifiques allant étape par étape à la résolution d'un problème d'entreprise.
Merino and Elguezabal (2005)	Une collection d'activités qui sont nécessaires pour atteindre un objectif commercial. Il est représenté avec un flux d'activités qui spécifie l'orchestration nécessaire pour compléter l'objectif.
Wang and Wang (2005)	Un ensemble de règles d'entreprise qui contrôlent les tâches par la représentation explicite de la connaissance du processus.
Papazoglou (2006)	Une ou plusieurs activités connexes qui répondent ensemble à une exigence commerciale pour une action.
Shapira and Forfutdinov (2007)	Décomposable en plusieurs sous-processus, qui ont leurs propres attributs, qui contribuent à atteindre l'objectif du super-processus. Il inclut précisément la cartographie des processus et sous-processus jusqu'au niveau d'activité. Les activités sont des parties du processus qui n'incluent pas de prise de décision
Weske (2010)	Un ensemble d'activités qui sont exécutées dans un environnement organisationnel et technique pour réaliser ensemble un objectif commercial.
Tiwari <i>et al.</i> (2010)	Un ensemble de tâches, qui effectue une opération commerciale lorsqu'elles sont correctement connectées et séquencées. Son but est d'effectuer une opération commerciale, à savoir toute opération liée au service qui produit de la valeur à l'organisation.
Schumm <i>et al.</i> (2011)	Une séquence de collaborations (activités et cycles de dépendances) pour atteindre les objectifs personnels et commerciaux pertinents en fonction des intentions personnelles, des motivations et de la répartition du pouvoir.
Smirnov <i>e al.</i> (2012)	Un système intrinsèquement distribué : Ses activités sont exécutées par différents employés, sur différents sites, en utilisant un ensemble hétérogène de systèmes informatiques.
Dumas <i>et al.</i> (2013)	Une collection d'événements, d'activités et de points de décision inter reliés impliquant un certain nombre d'acteurs et d'objets et conduisant collectivement à un résultat qui est utile pour au moins un client. Les processus métiers sont ce que les entreprises font chaque fois qu'ils livrent un service ou un produit à leurs clients.

Laguna and Marklund (2013)	Un réseau d'activités à travers duquel les unités d'écoulement (flow) doivent passer pour être transformées d'entrées en sorties.
Zhang and Perry (2015)	Des clusters et leurs emplacements, où un cluster est vu comme un ensemble d'activités organisées selon un ordre d'exécution.
Mahammed and Benslimane (2016)	Un graphe orienté valué, où les nœuds symbolisent les tâches du processus métier, les arêtes représentent les ressources utilisées et la valuation les attributs des tâches le composant.

Tableau 1.1 Définitions des processus métiers dans la littérature

Volkner et Werners (2000) affirment qu'aucune définition globale d'un processus métier n'existe pas ; en raison du fait, qu'ils ont été approchés par un certain nombre de disciplines différentes. Lindsay et al. (2003) rapportent que la plupart des définitions de processus métiers sont limitées et les modèles correspondants sont également limités à un point de vue mécanique. Le principal problème avec les définitions de processus métier est double: Soit elles sont trop simplistes (trop générique) pour fournir tout apport palpable ou, elles sont confinées dans une zone très spécifique à leur application. Ce qui les empêche d'avoir une large applicabilité. Dans cette recherche, nous reprenons les définitions de Davenport and Short (1990), Hammer and Chappy (1993) et celle proposée par Mahammed and Benslimane (2016).

« Un processus métier est une collection de tâches logiquement exécutées, prenant des entrées et créant des sorties, ayant comme finalité un service et produit qui ait de la valeur pour le client. Un processus métier est vu comme un graphe orienté valué, où les nœuds symbolisent les tâches (activités) composant le processus métier, les arêtes représentent les ressources utilisées (entrées et sorties) et la pondération des tâches indiquent les valeurs des attributs de ces dernières».

1.1.2. Principaux éléments des processus métiers

La recherche concernant les processus métiers montre que bien qu'il y ait une grande variété en termes de définitions, quand cela traite des éléments structurels d'un processus métier ; il y a un terrain d'entente. Figure 1.2 présente

le schéma d'un processus métier comportant les éléments structurels les plus courants dans la littérature. Ces éléments sont mis ensemble dans un diagramme de classes qui reflète également les relations entre eux.

Les éléments structurels du schéma d'un processus métier présenté sont basés sur ce que de nombreux auteurs (voir le tableau 1.1) considèrent comme les éléments structurels de base d'un processus métier : Les acteurs, les rôles, les ressources, les activités et les événements.

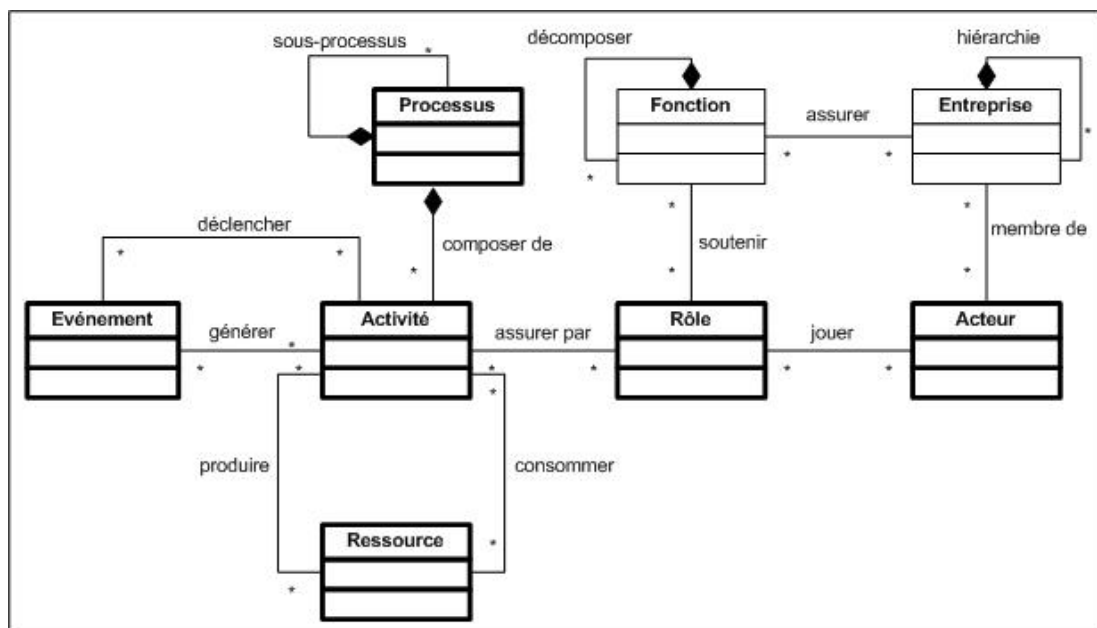


Figure 1.2 Meta modèle d'un processus métier (Mili et al., 2010)

Ce sont les principaux concepts impliqués dans la plupart des définitions d'un processus métier, bien que l'accent soit mis sur les activités et les ressources seulement (dans notre étude). Les acteurs et leurs rôles sont parfois impliqués dans une définition de processus métiers (Lindsay et al., 2003) ou - occasionnellement - perçus comme des entités externes qui adoptent ou exécutent le processus métier. Les activités sont largement acceptées comme éléments centraux qui exécutent les étapes des processus métiers de base, en utilisant les entrées du processus afin de produire les résultats souhaités (sorties). Les ressources sont souvent classées comme des entrées ou des ressources d'entrée et elles sont nécessaires pour

les activités à exécuter. Enfin, les tâches sont perçues comme le plus petit élément analysable d'un processus métier (Orman, 1995). Cependant, elles sont généralement négligées, par la plupart des auteurs ou ont tendance à être considéré comme un autre synonyme pour les activités (Vergidis, 2008).

Le fonctionnement d'un processus métier peut être résumé de la sorte :

"**Les activités** d'un processus sont exécutées par **des acteurs** jouant **des rôles** particuliers, consommant des **ressources** et produisant d'autres. **Les rôles** peuvent soutenir **des fonctions**. **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 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 spécifiques." (Mili et al., 2010)

1.1.3. Pourquoi automatiser les processus métiers ?

Selon Powell et al. (2001), la plus grande partie des coûts de l'entreprise est générée par les processus métiers. Afin d'améliorer l'efficacité organisationnelle, il est nécessaire d'améliorer les processus métiers de l'entreprise. Ils influencent - fortement - la qualité du produit et la satisfaction du client, qui sont tous deux d'une importance fondamentale sur le marché (Grigory et al., 2001). Cela entraîne plusieurs implications : Les processus métiers doivent être correctement conçus, leur exécution doit être prise en charge par un système qui peut répondre aux exigences de la charge du travail, et les ressources du processus doivent être en mesure d'effectuer le travail demandé en un temps adéquat (Grigori et al., 2004). Zhou et Chen (2003a) soulève un point important, il atteste que, pour la planification automatisée des processus métiers, il devrait impérativement y avoir des techniques qui prennent en charge la modélisation, l'analyse et l'optimisation des processus métiers.

Selon Castellanos et al. (2004), une des dernières tendances de la littérature est l'automatisation des processus métiers. Les avantages de cette automatisation sont que les processus peuvent être exécutés plus rapidement, à moindre coût (en raison d'une intervention humaine restreinte) et d'une manière contrôlée. Cependant, comme les processus métiers deviennent de plus en plus automatisés, l'industrie et le milieu universitaire délaissent le déploiement de ces processus ; pour se focaliser sur une modélisation, une analyse et une optimisation efficace des processus métiers (Vergidis and Tiwari, 2008). L'objectif de cette recherche réside dans le domaine de l'optimisation des processus métiers, avec des références à la modélisation et l'analyse. Cette optimisation des processus métiers sera facilitée par les techniques de l'informatique évolutionnaire, comme il est expliqué dans la section suivante.

1.2. Optimisation évolutionnaire multicritères

L'introduction générale de la thèse révèle que l'approche d'optimisation proposée pour les conceptions de processus métiers est construite sur la base d'une approche d'optimisation évolutionnaire multicritères (ou multi-objectifs). Selon Deb (2001), l'optimisation se rapporte à rechercher une ou plusieurs solutions possibles qui correspondent aux valeurs extrêmes d'un ou plusieurs objectifs. Dans la suite de cette section, nous expliquons les raisons qui nous ont poussées à choisir l'optimisation multicritères, ainsi que l'utilisation des algorithmes évolutionnaires. Nous présenterons le type d'algorithme évolutionnaire le plus connu "les algorithmes génétiques", et leurs particularités.

1.2.1. Pourquoi l'optimisation multicritères ?

Les problèmes et algorithmes d'optimisation multicritères ont reçu une grande attention au cours des deux dernières décennies, en raison du fait que la plupart des problèmes du monde réel impliquent naturellement des objectifs multiples et variés. Un problème d'optimisation multicritères peut impliquer soit la minimisation ou la maximisation et peut également être soumis à un certain nombre de contraintes qui restreignent les limites du problème (Deb, 2001).

Le Framework proposé adopte une approche d'optimisation multicritères des conceptions de processus métiers pour trois raisons principales :

1. L'optimisation des processus métiers est en soi un problème d'optimisation multicritères en raison de la diversité des facteurs qui peuvent évaluer un processus métier. Considérer des objectifs multiples ; dans notre recherche, peut la rendre plus captivante et applicable aux problèmes d'optimisation des processus métiers de la vie réelle.
2. L'évaluation des processus métiers basée sur une série de facteurs pertinents garantit que cette recherche soit polyvalente dans le traitement des différents objectifs de l'entreprise (Vergidis, 2008). La capacité de traiter simultanément une série d'objectifs quantifiables personnalisés assure la généralité de la recherche et son application potentielle dans un contexte plus large sur l'amélioration des processus métiers.
3. Au contraire, un Framework d'optimisation monocritère (ou mono-objectif) se concentre sur un critère en particulier (e.g. la réduction des coûts), et ainsi perd sa généralité et ses avantages par rapport aux approches d'amélioration des processus métiers spécifiques au contexte, qui ciblent souvent un aspect spécifique d'un processus métier.

Jaskiewicz (2001) propose une formulation intéressante du problème d'optimisation multicritères :

$$\text{maximiser } \{f_1(x) = z_1, \dots, f_j(x) = z_j\}, x \in D \quad (1.1)$$

Avec : La *solution* $x = [x_1, \dots, x_i]$ est un vecteur de variables de décision, D est l'ensemble des solutions réalisables. Nous verrons par la suite que nous manipulons des variables discrètes (attributs des instances des processus métiers). De la sorte, le problème traité prend l'appellation de « problème d'optimisation combinatoire évolutionnaire multicritères ».

L'image d'une solution x dans l'espace objectif est un point $z^x = [z_1^x, \dots, z_j^x] = f(x)$, tel que $z_j^x = f_j(x)$, $j = 1, \dots, J$.

Le point z^1 domine z^2 , $z^1 > z^2$, si $\forall j \ z_j^1 \geq z_j^2$ et $z_j^1 > z_j^2$ pour au moins une valeur j .

La solution x^1 domine x^2 , si l'image de x^1 domine l'image de x^2 .

Une solution $x \in D$ est un *optimal de Pareto*, s'il n'y a pas de solution $x' \in D$ qui domine x .

Un point est dit *non-dominé*, s'il est une image d'une solution optimale de Pareto. L'ensemble de toutes les solutions optimales de Pareto est appelé *l'ensemble optimale de Pareto*. L'image N de *l'ensemble optimale de Pareto* dans l'espace objectif est appelé *l'ensemble non dominé* ou *le front de Pareto*.

1.2.2. Introduction à l'optimisation évolutionnaire

Dans le monde naturel, l'évolution a créé une gamme incroyablement variée de modèles, ayant beaucoup plus de complexité que l'humanité ne pourrait jamais espérer atteindre. Ainsi, inspirés par cela, les chercheurs ont commencé à adopter les techniques de l'informatique évolutionnaire (Fogel, 1995) qui utilisent les principes de l'évolution pour guider le processus d'optimisation. C'est un sous-domaine de l'intelligence artificielle, plus particulièrement l'intelligence informatique (*Computational Intelligence*), qui implique des problèmes d'optimisation combinatoire (Liu and Hu, 2013). Les techniques de l'intelligence informatique utilisent des progrès itératifs (Vergidis, 2008), tels que la croissance ou le développement d'une population. Cette population est ensuite sélectionnée dans une recherche aléatoire guidée en utilisant un traitement parallèle pour atteindre le but souhaitée. Ces processus sont souvent inspirés par les mécanismes biologiques de l'évolution.

Habituellement, une première population de solutions générées de façon aléatoire comprend la première génération. Une fonction de Fitness est appliquée aux solutions et à toute la descendance suivante. Avec la sélection, les parents sont souvent choisis avec la valeur de Fitness la plus élevée (Skrlec, 2010), pour la prochaine génération. Les parents se reproduisent par l'application d'opérateurs

tels que le croisement et la mutation. Le premier agit - souvent - sur les deux parents sélectionnés et les résultats sont un à deux enfants. Tandis que la mutation agit sur une solution et aboutit à une nouvelle solution. Ces opérateurs créent une nouvelle population de solutions. Ce processus peut être répété jusqu'à ce qu'une population de solutions ayant une qualité suffisante, ou qu'un nombre de générations défini précédemment soit atteint.

Il existe un certain nombre d'avantages à l'optimisation évolutionnaire qui justifient l'effort investi dans ce domaine (Tiwari and Roy, 2004). L'avantage le plus important réside dans le gain de souplesse et d'adaptabilité de la tâche à accomplir, en combinaison avec des performances robustes et des caractéristiques de recherche globale (Coello Coello, 2000). Les techniques d'optimisation basées sur l'évolution utilisent (à chaque itération) une population de solutions au lieu d'une solution unique. Ceci leur permet (en principe) d'identifier de multiples solutions optimales dans leur population finale. Les techniques de l'intelligence informatique proposent souvent des solutions bien adaptées à tous les types de problèmes, car elles ne font - théoriquement - aucune hypothèse sur le paysage de la fonction de Fitness (Vergidis, 2008).

1.2.3. Pourquoi utiliser l'optimisation évolutionnaire multicritères ?

La technique d'optimisation multicritères utilisée par notre recherche est un algorithme évolutionnaire ou évolutif. Ce type d'algorithme imite les principes évolutifs de la nature pour guider le processus d'optimisation vers la découverte de solutions optimales. L'utilisation de cet algorithme est devenue de plus en plus populaires dans les problèmes d'optimisation multicritères (Coello Coello, 2005). Les deux principaux avantages de l'utilisation d'un algorithme évolutionnaire sont selon Vergidis et Tiwari (2008) :

- Après une seule itération, le résultat d'un algorithme évolutionnaire est une population de solutions.
 - Dans l'optimisation multicritères, l'algorithme évolutionnaire traite tous les objectifs comme égaux.
-

La possibilité d'avoir une gamme de solutions optimales reflétant un compromis distinct entre les différents critères d'optimisation est assurée par le manque de préférence vis-à-vis d'un objectif particulier, en rapport avec la population de solutions générée. Cette capacité améliore le processus de sélection d'une seule solution optimale en offrant un choix d'alternatives équivalentes.

Il est clair que le concept d'évolution et d'amélioration de la population des solutions de processus métiers ; dans le processus d'optimisation évolutionnaire, est primordial à cause (i) des conceptions de processus métiers qui risquent d'être négligées par un spécialiste (concepteur humain) peuvent être découvertes par l'algorithme évolutionnaire et, (ii) de la possibilité de faire évoluer une conception d'un processus métier infaisable en une solution faisable. Certains chercheurs suggèrent que la recherche multicritères et l'optimisation peuvent être des problèmes où les algorithmes évolutionnaires font mieux que les autres méthodes de recherche traditionnelles (Fonseca and Fleming, 1995) (Valenzuela-Rendón et al., 1997). N'ayant pas la prétention de présenter tous les algorithmes et autres méthodes impliqués dans la résolution d'un problème multicritères (cela serait dérisoire et franchement inutile !), nous nous contentons ; par la suite, de présenter les algorithmes évolutionnaires, en se focalisant sur les algorithmes génétiques.

Conclusion

Ce chapitre a été consacré, en premier temps, aux définitions et concepts autour des processus métiers. Il peut être résumé ainsi :

- Il y a une abondance de définitions relatives aux processus métiers.
 - Un processus métier est inexorablement un type de processus, mais différent car il peut produire une opération en entreprise.
 - Un processus métier comprend - entre autre - des activités et des ressources.
 - Une activité peut être considérée comme la partie exécutable d'un processus métier. Le terme "tâche" est également utilisé.
-

- Une ressource est utilisée par une activité. Elle peut être classifiée comme une ressource d'entrée ou de sortie, pour un processus métier ou simplement pour une activité.

En deuxième temps, le chapitre s'est intéressé aux concepts de l'optimisation multicritères des processus métiers. Il peut être résumé comme suit :

- Il implique, soit la minimisation ou la maximisation et peut également être soumis à un certain nombre de contraintes qui bornent les limites du problème.
- L'optimisation évolutionnaire multicritères implique l'utilisation des algorithmes évolutionnaires. Après une seule itération, le résultat est une population de solutions.

Le chapitre suivant introduit ce qu'est qu'un algorithme évolutionnaire, avec ses différents types. Nous mettons l'accent sur les algorithmes génétiques.



"I SPENT FIVE HOURS WORKING ON MY REPORT!
ONE HOUR TO GO TO THE MALL FOR AN INK CARTRIDGE,
TWO HOURS ON HOLD WITH TECH SUPPORT, 45 MINUTES
LOOKING FOR A SHEET OF WHITE PAPER, 30 MINUTES
SEARCHING FOR THE PERFECT FONT..."

Chapitre 2

Algorithme évolutionnaire, NSGAI et opérateurs de sélection

Introduction	25
2.1. Algorithmes évolutionnaires	25
2.1.1. Les algorithmes génétiques	26
2.1.2. Les stratégies évolutionnaires	26
2.1.3. La programmation évolutive ou évolutionnaire	27
2.1.4. La programmation génétique	27
2.1.5. L'optimisation stochastique ou les algorithmes à estimation de densité	28
2.2. Algorithmes génétiques et NSGAI.....	29
2.2.1. Qu'est ce qu'un Algorithme génétique ?	29
2.2.2. NSGAI plus qu'un simple algorithme génétique.....	33
2.2.3. Fonctionnement général de NSGAI.....	34
2.3. Sélection, plus qu'un opérateur	38
2.3.1. Importance du choix de la sélection dans un algorithme génétique	38
2.3.2. Techniques de sélection étudiées.....	39
2.3.2.1. Sélection par tournoi.	39
2.3.2.2. <i>Sélection par roulette</i>	40
2.3.2.3. Sélection linéaire par rang.	41
2.3.2.4. Sélection exponentielle par rang.	43
2.3.2.5. Sélection uniforme.	43
2.3.2.6. Sélection par troncation.....	44
2.3.3. Synthèse.....	47
Conclusion	48

Introduction

Le chapitre précédent discute le principe derrière une optimisation évolutionnaire multicritères. Il a clairement expliqué que cette catégorie d'optimisation multicritères implique l'utilisation des algorithmes évolutionnaires, pour la génération de nouvelles solutions. Comme suite logique, ce chapitre discute les différents types d'algorithmes évolutionnaires. Nous nous attarderons par la suite, sur un aspect fondamental de notre travail, les algorithmes génétiques. Sachant que nous nous sommes bornés à utiliser ce type particulier de métaheuristique et de revoir - en profondeur - quelques uns de ses opérateurs. Ce chapitre comporte une partie - assez - importante sur les algorithmes génétiques. NSGAI étant un des algorithmes génétiques les plus prisés dans l'optimisation multicritères, il nous est apparu cruciale de l'exposer. Nous présentons le NSGAI, son fonctionnement général, ses opérateurs génétiques et son paramètre de distance de *crowding*.

Ce chapitre discute, ensuite, une étude comparative des opérateurs de sélection, connus, ayant fait leurs preuves dans divers travaux et utilisables avec un algorithme génétique, pour une - éventuelle - optimisation multicritères des processus métiers, avec NSGAI.

2.1. Algorithmes évolutionnaires

Les algorithmes évolutionnaires ou l'informatique évolutionnaire est un domaine de l'informatique (Martín-Palma and Lakhtakia, 2013). Ils appliquent les principes de recherche heuristique inspirés par l'évolution naturelle inspiré par (Darwin, 1929) à une variété de domaines différents, particulièrement aux problèmes d'optimisation des paramètres, ainsi que des problèmes considérés dans l'Intelligence Artificielle. Les algorithmes évolutionnaires rassemblent des approches différentes dans le nom, l'origine et l'implémentation (Baluja, 1994) (Fogel, 1995) (Bäck, 1996) (Bäck et al., 1997) (Mühlenbein and Mahnig, 2001) (Le Riche et al., 2007) (Banzhaf, 2013).

Par la suite, ces approches sont décrites brièvement. Un intérêt particulier est porté aux algorithmes génétiques.

2.1.1. Les algorithmes génétiques

Ils ont été développés par (Holland, 1975), utilisés comme optimiseurs par (DeJong, 1975) puis revus en profondeur et popularisés par (Goldberg, 1989).

2.1.2. Les stratégies évolutionnaires

Elles ont été développés par (Schwefel, 1981) et (Rechenberg, 1965, 1994). Ils ont été conçu ; à la base, pour résoudre des problèmes d'optimisation de paramètres. Schwefel (1981), Rechenberg (1994) et Bäck (1996) décrivent le fonctionnement d'une stratégie évolutionnaire, comme suit :

1. Une population initiale de μ individus est créée.
2. Une valeur de Fitness est attribuée à chacun des μ individus.
3. Une nouvelle population de λ individus¹ est créée par recombinaison de la population actuelle.
4. L'opérateur de mutation est appliqué sur la nouvelle population.
5. Une valeur de Fitness est attribuée à chacun des individus de la nouvelle population.
6. La sélection des individus est faite, en utilisant soit une sélection (μ, λ) ou $(\mu + \lambda)$.
7. La nouvelle population devient la population actuelle.²
8. Arrêter, si les conditions d'arrêt sont satisfaites, sinon revenir à l'étape 3.

¹ Typiquement, $\lambda = 7\mu$.

² Les meilleurs individus sont choisis à partir des individus de la nouvelle population

2.1.3. La programmation évolutive ou évolutionnaire

Cette technique a été développée par (Fogel, 1962) (Fogel et al., 1966). Elle se limite à l'évolution d'automates à états finis, par la suite généralisée à une représentation quelconque (Fogel, 1992, 1995). La programmation évolutive n'utilise pas d'opérateur de recombinaison, à la différence des stratégies évolutionnaires et des algorithmes génétiques. Fogel et al. (1966) puis Fogel (1995) décrivent le fonctionnement de la programmation évolutive, comme suit :

1. Une population initiale d'individus est créée.
2. Une valeur de Fitness est attribuée à chacun des individus.
3. L'opérateur de mutation est appliqué sur la population actuelle pour produire de nouveaux individus.
4. Une valeur de Fitness est attribuée à chacun des individus de la nouvelle population.
5. Une nouvelle population de taille N est créée à partir des individus de la population initiale et des individus de la nouvelle population avec la sélection par tournoi.
6. Arrêter, si les conditions d'arrêt sont satisfaites, sinon revenir à l'étape 3.

2.1.4. La programmation génétique

Elle est plus récente que les techniques précédemment citées. Elle a vu le jour comme une forme particulière des algorithmes génétiques, qui faisaient évoluer les programmes représentés sous forme d'arbre syntaxiques (Cramer, 1985). La programmation génétique a été popularisée par (Koza, 1992, 1994) (Koza et al., 1999). Elle consiste à faire évoluer une population constituée d'un grand nombre de programmes selon les étapes suivantes (Koza, 1992) :

1. Création d'une population de programmes de façon aléatoire.
2. Evaluation de la qualité (Fitness) de chaque programme (selon une méthode propre au problème posé).

3. Application des opérateurs génétiques à la population, afin d'en créer une nouvelle.
4. Sélection des programmes répondant le mieux au problème posé (les mieux adaptés à leur environnement).
5. Arrêter, si les conditions d'arrêt sont satisfaites, sinon revenir à l'étape 2.

2.1.5. L'optimisation stochastique ou les algorithmes à estimation de densité

Encore plus récente que les techniques présentées. Elle est apparue avec les travaux de (Baluja, 1994) et (Mühlenbein and Mahnig, 2001). Les EDA (pour *Estimation of Distribution Algorithms*) fonctionnent en fouillant l'espace de recherche, en échantillonnant et en mettant à jour itérativement une distribution p^t représentant les meilleurs points connus (Le Riche et al., 2007). Selon (Baluja, 1994) et (Mühlenbein, 1997), le fonctionnement type d'un EDA est le suivant :

1. Initialisation de la densité $p^0, t = 0$.
2. Création de λ points par échantillonnage de p^t .
3. Evaluation de la Fitness pour les λ points.
4. Sélection des μ meilleurs ($\mu + \lambda$).
5. Mettre à jour p^{t+1} par l'intermédiaire des μ meilleurs point et de p^t .
6. Arrêter, si les conditions d'arrêt sont satisfaites, sinon revenir à l'étape 2.

Jones (1998) et Banzhaf (2013) garantissent que chacun de ces algorithmes évolutionnaires a prouvé être capable de fournir des solutions - approximativement - optimales compte tenu de la nature complexe et discontinue des espaces de recherche, avec de moins bons résultats pour les algorithmes à estimation de densité (Kern et al., 2004). La section suivante explique en détail l'origine, le fonctionnement et les différents concepts en rapport avec les algorithmes génétiques.

2.2. Algorithmes génétiques et NSGAI

La sélection naturelle ou le survi de l'individu le plus fort a été défini par le chercheur et naturaliste Charles Darwin dans son livre (Darwin, 1929). Il présente la sélection naturelle comme la préservation des traits qui rendent un individu plus apte à résister quand il se reproduit, et les traits qui le rendent plus faible sont éliminés, dans un processus appelé "évolution" (Coello Coello, 1996). Ces caractéristiques sont contrôlées par des unités appelées "gènes" qui forment des ensembles appelés "chromosomes". Au cours des générations suivantes, les individus les plus résistants survivent, mais aussi leurs gènes les plus adaptés sont transmis à leurs enfants au cours du processus de la recombinaison sexuelle ; appelé "croisement".

John Holland s'est intéressé à l'application de la sélection naturelle à la machine apprenante (*Learning Machine*). Il a développé une technique qui permettait aux programmes informatiques d'imiter le processus d'évolution, à la fin des années 60. Le terme "algorithme génétique" est devenu populaire, après la publication de son livre (Holland, 1975, 1992). Par la suite, (Goldberg, 1989) a publié un livre qui fournissait une solide base de travail pour ce domaine de recherche.

2.2.1. Qu'est ce qu'un Algorithme génétique ?

Ces dernières années, l'intérêt croissant pour les algorithmes génétiques s'est traduit par un nombre sans cesse croissant de travaux, de conférences et de revues scientifiques consacrés à ce sujet. Les définitions sur les algorithmes génétiques pullulent, de son côté (Koza, 1992) fournit une très bonne définition :

"L'algorithme génétique est un algorithme mathématique qui transforme un ensemble d'individus (population d'objets mathématiques), chacun avec une valeur de Fitness associée, en une nouvelle population (la génération suivante) en utilisant des opérations inspirées du principe darwinien de reproduction, et après des opérations génétiques naturelles".

Michalewicz (1994) propose qu'un algorithme génétique doit avoir cinq (05) composantes essentielles, pour résoudre un problème particulier :

1. Une représentation des solutions possibles du problème.
2. Une façon de créer une population (un ensemble initial de solutions).
3. Une fonction d'estimation pour les évaluer.
4. Des opérateurs génétiques qui modifient la composition des descendants (croisement, mutation et sélection).
5. Des paramètres différents que l'algorithme génétique utilise (taille de la population, probabilités des opérateurs génétiques).

Gen et al. (2008) font remarquer que la terminologie utilisée en algorithme génétique est une sorte de mélange entre les notions d'informatique et la génétique. Dans un organisme biologique, la structure qui spécifie sa constitution est appelée "chromosome". L'ensemble de tous les chromosomes est appelé un "génotype", et l'organisme qui en résulte est appelé un "phénotype". Chaque chromosome comporte un nombre de structures individuelles appelées "gènes". Les différentes valeurs d'un gène sont appelées "allèles". Chaque gène code une caractéristique particulière de l'organisme. L'emplacement d'un gène à l'intérieur d'un chromosome est appelé un "locus". Un récapitulatif des termes utilisés dans un algorithme génétique est présenté dans le tableau 2.1.

Terme	Explication
Chromosome	Une structure de données qui contient des gènes.
Gène	Une sous-section d'un chromosome.
Locus	La position d'un gène.
Allèle	La valeur d'un gène.
Phénotype	La signification d'un chromosome particulier.
Génotype	Une solution potentielle à un problème.

Tableau 2.1 Terminologie utilisée avec un algorithme génétique

Pour utiliser un algorithme génétique, un certain nombre d'opérateurs et de concepts doivent être définis. Les plus importants sont :

Représentation (codage des chromosomes) : La première étape de la conception est couramment appelée "représentation". Elle consiste à spécifier une cartographie des phénotypes sur un ensemble de génotypes qui sont censés représenter ces phénotypes (Eiben and Smith, 2003).

Population : Son rôle est de contenir (la représentation) des solutions possibles. Une population est un ensemble (où plusieurs copies d'un même élément sont possibles) de génotypes (Eiben and Smith, 2003). La population initiale est généralement générée au hasard.

Fonction d'évaluation ou fonction de Fitness : Chaque chromosome est évalué et se voit attribuer une valeur selon une fonction d'évaluation, après la création d'une population initiale. Jones (1998) explique que la fonction d'évaluation est la base de la sélection et facilite ainsi les améliorations. Plus précisément, elle définit ce qu'est que cette amélioration. Techniquement, c'est une fonction ou une procédure qui attribue une mesure de qualité aux génotypes. Selon Diaz-Gomez and Hougen (2007), la fonction de Fitness est une boîte noire pour l'algorithme génétique, cela peut être réalisé par un modèle de simulation, une fonction mathématique, ou même un expert humain.

Sélection parentale : Son rôle consiste à faire la distinction entre les individus en fonction de leur qualité (Eiben and Smith, 2003), en particulier pour permettre aux meilleurs individus de devenir parents de la génération suivante. Un individu est un parent, s'il a été choisi pour endurer une variation afin de créer sa descendance.

Elitisme : C'est une méthode de sélection qui permet de garder les meilleurs individus d'une population pour la génération suivante. Mitchell (1999) explique clairement que si les individus ne sont pas conservés, ils risquent de ne pas survivre pour les prochains cycles (après le croisement ou la mutation). L'élitisme consiste à

copier un pourcentage des meilleurs individus dans la nouvelle génération, pour éviter qu'ils soient perdus.

Croisement ou recombinaison : C'est un opérateur génétique qui opère sur deux chromosomes à la fois, et génère la descendance en combinant deux (02) caractéristiques des chromosomes. Le moyen le plus simple d'y arriver est de choisir un point de coupure aléatoire (figure 2.1) et de générer de nouvelles solutions en combinant le segment d'un parent à la gauche du point de coupe avec le segment de l'autre parent à la droite du point de coupe (Gen et al., 2008).

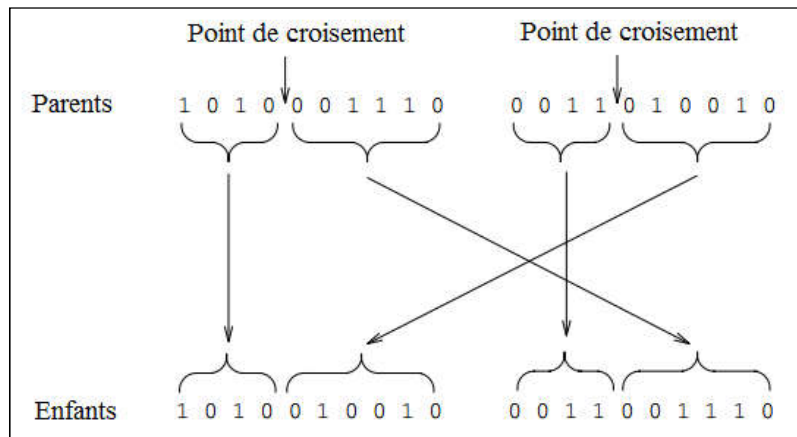


Figure 2.1 Exemple de croisement : Croisement à un point

Taux de croisement : Zhang (2009) le définit comme la probabilité du nombre de descendants produits dans chaque génération par rapport à la taille de la population. Ce taux contrôle le nombre attendu de chromosomes pour subir l'opération de croisement.

Mutation : Shahookar et Mazumder (1990) présente la mutation comme un opérateur qui agit en profondeur, il produit des changements aléatoires et spontanés dans divers chromosomes. Un moyen simple d'obtenir une mutation serait de modifier un ou plusieurs gènes (figure 2.2). Selon Gen et Cheng (1996), la mutation joue un rôle capital dans un algorithme génétique, en (i) remplaçant les gènes perdus au cours de la sélection, pour qu'ils puissent être réutilisés ultérieurement, (ii) en fournissant des gènes absents dans la population initiale.

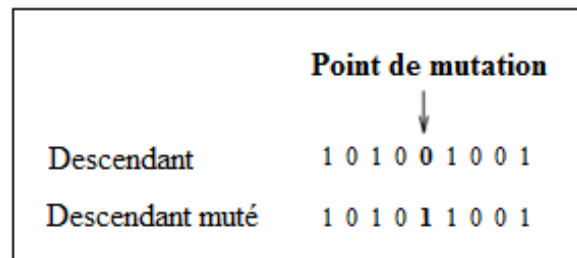


Figure 2.2 Exemple de mutation : Mutation à un point

Taux de mutation : Gen et al. (2008) explique que ce taux contrôle la probabilité avec laquelle de nouveaux gènes sont introduits dans la population pour lors d'une itération. Il contribue à ce que l'algorithme garde la capacité d'apprendre de l'historique de la recherche.

Critères d'arrêt : Les algorithmes génétiques sont des méthodes de recherche stochastique qui pourraient en principe fonctionner à tout jamais (Ilgin and Tunali, 2007). En pratique, un critère d'arrêt est indispensable, plusieurs approches existent tels que fixer un nombre d'itération ou une limite sur le nombre d'évaluations de la fonction de Fitness (Haupt and Haupt, 2004).

2.2.2. NSGAI plus qu'un simple algorithme génétique

Selon Goldberg, (1989) et (Golchha and Qureshi, 2015), les tests de logiciels constituent la plus importante mesure de la qualité, qui consomme au moins 50% des coûts de développement des logiciels. Le processus d'automatisation de la génération des données de test est un moyen de réduire le temps consommé par cette tâche. Les algorithmes génétiques sont utilisés dans ce but. Un certain nombre d'algorithmes évolutionnaires ont été suggérés. L'algorithme génétique de tri non dominé (NSGAI pour *Non-Dominated Sorting Genetic Algorithm*) est un algorithme utilisé pour résoudre des problèmes d'optimisation multicritères. Il a été proposé par Deb et al. (2002), en utilisant et améliorant le concept proposé par Goldberg (1989). NSGAI est l'un des algorithmes les plus utilisés pour résoudre les problèmes d'optimisation multicritères (Golchha and Qureshi, 2015).

Avant d'expliquer le fonctionnement de NSGAII, il est bon de préciser que NSGA a généralement été critiqué pour sa complexité de calcul, son manque d'élitisme et pour la difficulté à choisir la valeur de paramètre optimale pour le partage du paramètre σ_{share} (Seshadri, 2006). Deb et al. (2000, 2002) expliquent que NSGAII diffère de son prédécesseur par (i) l'utilisation d'une procédure de tri plus rapide, basée sur la non dominance ou Pareto optimale, avec une complexité de calcul de $O(mN^2)$, où m est le nombre d'objectifs et N la taille de la population. (ii) Une approche élitiste qui permet de préserver la diversité des populations, en sauvegardant les meilleures solutions trouvées lors des générations précédentes d'une part, et d'autre part un opérateur de comparaison basé sur un calcul de la distance de peuplement (ou *Crowding*) (Belaidi et al., 2009). La mise en œuvre de cet opérateur ne nécessite aucun réglage des paramètres.

2.2.3. Fonctionnement général de NSGAII

NSGAII est un algorithme élitiste qui n'utilise pas de dépôt (ou archive) externe pour stocker les meilleurs éléments. Pour gérer l'élitisme, NSGAII garantit que les meilleurs individus rencontrés soient conservés, à chaque nouvelle génération.

Intéressons nous au fonctionnement de NSGAII, prenons une génération t où deux populations P_t et Q_t de taille N cohabitent (figure 2.3). La population P_t contient les meilleurs individus rencontrés jusqu'à la génération t , tandis que la population Q_t est constituée d'individus issus des phases précédentes du déroulement de NSGAII.

La première étape consiste à créer la population $R_t = P_t \cup Q_t$ et à appliquer une procédure de *ranking*³ (figure 2.4) pour identifier les différents fronts F_i de

³ Goldberg (1989) présente dans ses travaux une procédure itérative pour calculer le *ranking*. Au début, tous les individus non dominés de la population reçoivent le rang (ou front) 1 et sont retirés momentanément de la population. Puis, les nouveaux individus non dominés reçoivent le rang 2 avant d'être à leur tour retirés.

solutions non dominées. Par la suite, les meilleurs individus se retrouvent dans le - ou les - tous premiers fronts).

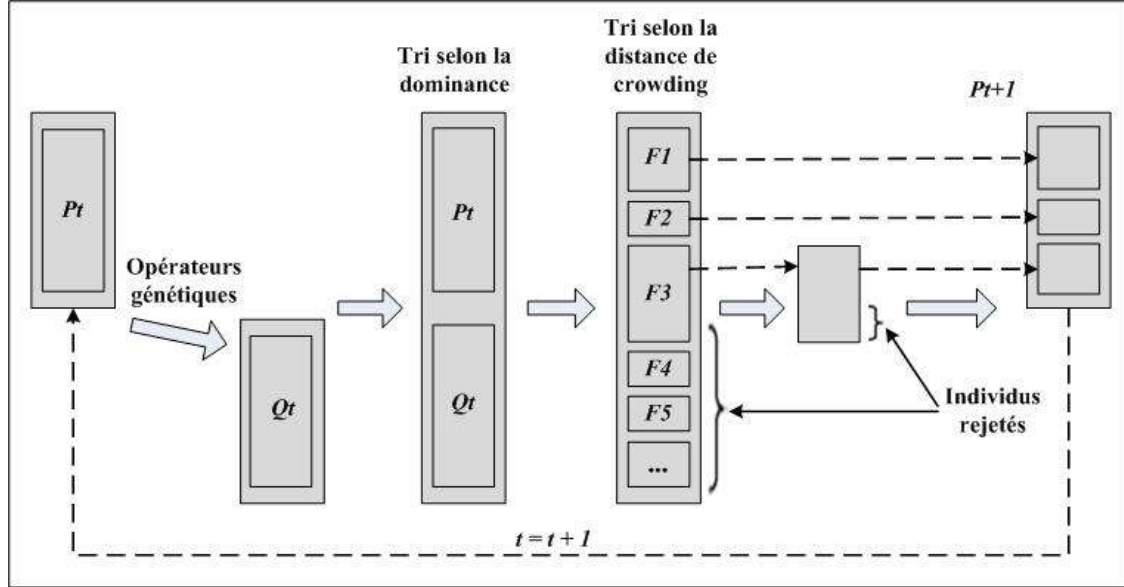


Figure 2.3 Fonctionnement général de NSGAII

La deuxième phase consiste à construire une nouvelle population P_{t+1} contenant les N meilleurs individus de R_t . Pour y arriver, il est nécessaire d'inclure les meilleurs fronts F_i complètement ; en commençant par l'indice 1, et ceci tant que le nombre d'individus présents dans P_{t+1} est inférieur à N . Il reste donc à ce stade $N - |P_{t+1}|$ individus à inclure dans P_{t+1} . Pour cela, une procédure de *peuplement* (ou *crowding*) (voir figure 2.6) est appliquée sur le premier front F_i non inclus. Donc, les $N - |P_{t+1}|$ meilleurs individus en utilisant la procédure de crowding sont insérés dans P_{t+1} .

La troisième phase se résume à emplir la population Q_{t+1} . Il suffit alors d'utiliser les opérateurs génétiques : Sélection, croisement et mutation (nous parlerons en

Le processus se répète tant qu'il reste des individus dans la population. La valeur d'adaptation de chaque individu correspond à son rang dans la population. (Barichard, 2003)

détail des opérateurs utilisés par NSGAI, par la suite) sur les individus de P_{t+1} , puis d'insérer les descendants dans Q_{t+1} .

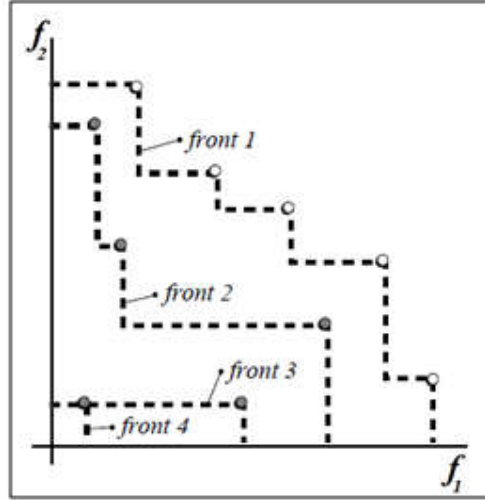


Figure 2.4 Ranking dans NSGAI (Fallahpour et al., 2012)

NSGAI est résumé dans l'algorithme suivant (Barichard, 2003) :

Algorithme NSGAI

Initialiser les populations P_0 et Q_0 de taille N

Tant que critère d'arrêt non rencontré faire

- Créer $R_t = P_t \cup Q_t$
- Calculer les différents fronts F_i de la population R_t (ranking)
- Mettre $P_{t+1} = \emptyset$ et $i = 0$,
- **Tant que** $|P_{t+1}| + |F_i| < N$ faire
 - $P_{t+1} = P_t \cup F_i$
 - $i = i + 1$

FinTantQue

- Inclure dans P_{t+1} les $(N - |P_{t+1}|)$ individus de F_i les mieux répartis selon la distance de crowding
- Sélectionner dans P_{t+1} et créer Q_{t+1} par application des opérateurs de croisement et mutation

FinTantQue

Figure 2.5 Pseudo-code de NSGAI

- i. **Distance de peuplement ou crowding** : Pour la sélection des individus à ajouter dans la population contenant les élites (P_{t+1}), Deb et al. (2000) ont proposé d'utiliser un paramètre particulier : La distance de *crowding*.

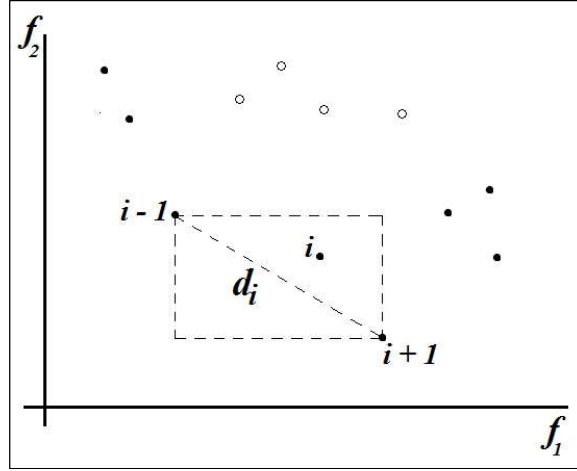


Figure 2.6 Distance de peuplement

Pour calculer la distance de *crowding* d_i (figure 2.6) d'un point particulier i , il faut prendre en considération le périmètre de l'hypercube (ou n-cube) ayant comme sommets les points les plus proches de i sur chaque objectif. Sur la figure 2.6 est représenté l'hypercube en deux dimensions associé au point i . Deb (2001) détaille l'algorithme pour le calcul de la distance de *crowding*, de complexité $O(MN \log(N))$, où M est le nombre d'objectifs du problème traité et N le nombre d'individus en jeu. Une fois tous les d_i calculés, il ne reste plus qu'à les trier par ordre décroissant et à sélectionner les individus possédant la plus grande valeur de crowding.

- ii. **Croisement binaire simulé** : NSGAII utilise le croisement binaire simulé (ou *Simulated Binary Crossover*) (Deb and Agrawal, 1994). Ce type de croisement fonctionne avec deux (02) solutions (deux parents), qui créeront deux (02) autres solutions (deux enfants). Le croisement binaire simulé utilise le principe de fonctionnement d'un opérateur de croisement sur un seul point qui est appliqué sur les chaînes binaires (Deb and Beyer, 2001). Donc, à partir de deux parents $p_1(i)$ et $p_2(i)$, ce croisement génère deux enfants $c_1(i)$ et $c_2(i)$ selon la relation suivante :

$$\begin{cases} c_1(i) = 0.5[(1 + \beta)p_1(i) + (1 - \beta)p_2(i)] \\ c_2(i) = 0.5[(1 - \beta)p_1(i) + (1 + \beta)p_2(i)] \end{cases} \quad (2.1)$$

Avec β représente un facteur de dispersion défini par :

$$\beta = \begin{cases} (2\mu)^{\frac{1}{\eta+1}}, & \text{si } \mu < 0 \\ \left(\frac{1}{2(1-\mu)}\right)^{\frac{1}{\eta+1}}, & \text{ailleurs} \end{cases} \quad (2.2)$$

Avec μ est une variable aléatoire uniformément répartie dans l'intervalle $[0, 1]$ et η un paramètre réel non négatif qui caractérise la forme de la distribution des descendants par rapport aux parents.

2.3. Sélection, plus qu'un opérateur

2.3.1. Importance du choix de la sélection dans un algorithme génétique

Les algorithmes génétiques sont des algorithmes de recherche d'optimisation qui maximisent et/ou minimisent les fonctions données (Shukla et al., 2015). Le travail présenté se concentre sur l'optimisation multicritères des conceptions de processus métiers avec un algorithme génétique. Nous avons opté pour l'utilisation de NSGAI, pour générer de nouvelles instances de processus métiers, comme définie dans la section 4.3.

Cette section compare les différentes techniques de sélection utilisées avec cet algorithme génétique, lors des expérimentations. L'identification de la technique de sélection appropriée est une étape critique dans l'utilisation d'un l'algorithme génétique (Pandey, 2015). L'identification de la méthode de sélection la plus appropriée est considérée comme une étape cruciale dans un algorithme génétique. Pandey (2016) explique que la technique de sélection doit être choisie avec précaution, de sorte que les meilleurs individus ayant une valeur de Fitness élevée aient une plus grande probabilité de sélection. Mais les *pires* individus devraient avoir peu de probabilité de sélection et ne devraient pas être complètement rejetés. Donc, l'utilisation de la méthode de sélection la plus apte à un problème garantit la réduction du risque de convergence prématurée, ce qui risque de se produire en raison du manque de diversité dans la population sélectionnée (Baker, 1985) (Golberg, 1989) (Whitley, L. D. (1989) (Pandey et al., 2014).

2.3.2. Techniques de sélection étudiées

Nous nous proposons de présenter les techniques de sélection les plus utilisées dans un problème d'optimisation multicritères, et les plus référencées dans la littérature portant sur les algorithmes génétiques (Blickle and Thiele, 1995). Le fonctionnement des sélections par tournoi, uniforme, par roulette, par rang et par troncation est présenté, et comparé selon un ensemble de critères (complexité, temps de calcul, taille de la population initial et la moyenne du nombre d'itération pour trouver les solutions optimales).

2.3.2.1. *Sélection par tournoi.*

Cette technique (*Tournament Selection*) est probablement la méthode de sélection la plus populaire avec un algorithme génétique, en raison de la facilité de sa mise en œuvre et de son efficacité (Goldberg and Deb, 1991). Lors de la sélection par tournoi, n individus sont choisis au hasard parmi une large population, puis les individus sélectionnés concourent entre eux. L'individu avec une meilleure valeur de Fitness gagne et fera partie de la prochaine génération (Miller and Goldberg, 1995).

La taille d'un tournoi est représentée par le nombre d'individus concurrents dans chaque tournoi, il est égale à deux pour le tournoi binaire. La sélection par tournoi donne également la chance à tous les individus d'être sélectionnés et donc, de préserver la diversité, même si la diversité peut dégrader la vitesse de convergence (Goldberg and Deb, 1991). La figure 2.7 présente la procédure de sélection par tournoi. Une grande valeur de la taille du tournoi conduit à une plus grande perte de diversité (Goldberg and Deb, 1991) (Blickle and Thiele, 1995). La taille d'un tournoi est grande signifie qu'une plus petite portion de la population contribue réellement à la diversité génétique.

```

Procédure Sélection par tournoi
 $P \leftarrow$  population
 $t \leftarrow$  taille du tournoi,  $t \geq 1$ 
 $\text{Individu}_{\text{meilleur}} \leftarrow$  individu pris au hasard à partir de  $P$ 
Pour tout  $i$  de 2 à  $t$  faire
     $\text{Individu}_{\text{suivant}} \leftarrow$  individu pris au hasard à partir de  $P$ 
    Si  $\text{Fitness}(\text{Individu}_{\text{suivant}}) > \text{Fitness}(\text{Individu}_{\text{meilleur}})$  alors
         $\text{Individu}_{\text{meilleur}} \leftarrow \text{Individu}_{\text{suivant}}$ 
Retourner  $\text{Individu}_{\text{meilleur}}$ 

```

Figure 2.7 Procédure de la sélection par tournoi

La sélection par tournoi présente plusieurs avantages, tel que une complexité temporelle efficace, une faible susceptibilité à la prise de contrôle par des individus dominants et aucune exigence de mise à l'échelle ou de tri (Goldberg and Deb, 1991) (Blickle and Thiele, 1995).

NSGAI utilise la sélection binaire par tournoi ou *Binary Tournament Selection* (Brindle, 1981) (Goldberg and Deb, 1991). Un groupe de n individus est choisi de façon aléatoire à partir d'une population. Le meilleur individu est sélectionné par un tournoi (déterministe) (Bäck, 1992) (Bäck et al., 2000) ou un tournoi probabiliste (ou stochastique) (Angeline and Pollack, 1993).

2.3.2.2. Sélection par roulette.

Appelée aussi *Roulette Wheel Selection* ou *Fitness Proportionate selection*. A ce niveau, les individus sont sélectionnés avec une probabilité qui est directement proportionnelle à leurs Fitness i.e. la sélection d'un individu correspond à une partie d'une roulette (figure 2.8). La probabilité de choisir un parent peut être vue comme faire tourner une roulette, où la taille du segment d'un parent est proportionnelle à sa Fitness. Goldberg (1989) explique clairement que les individus avec les plus grandes tailles de segment ont plus de probabilité d'être choisis. Sur la figure 2.8, la circonférence de la roulette est la somme de toutes les valeurs de Fitness des individus. Lorsque la roue est tournée, elle s'arrête enfin et son pointeur montre l'un des segments

(l'un des plus larges, vraisemblablement !). Cependant, chaque segment a une chance, avec une probabilité qui est proportionnelle à sa largeur. En répétant cela, chaque fois qu'un individu a besoin d'être choisi, les meilleurs individus sont choisis plus souvent que les plus mauvais, remplissant ainsi les exigences de survie du plus apte.

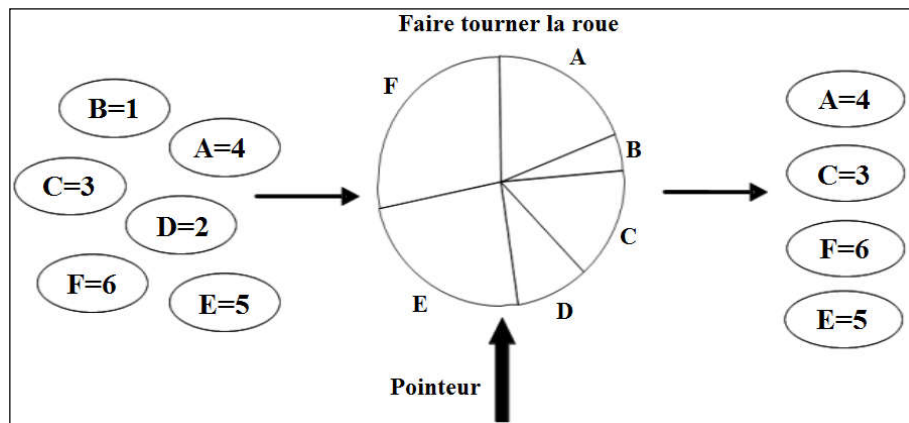


Figure 2.8 Exemple de sélection par roulette

L'avantage de la sélection par roulette est qu'elle n'écarte aucun individu de la population, ainsi elle donne une chance à tous d'être sélectionnés. Par conséquent, la diversité de la population est préservée (Noraini and Geraghty, 2011). Cependant, cette sélection a un défaut majeur. Les individus avec une très grande Fitness introduisent un *biais* au début de la recherche qui peut causer une convergence prématurée et une perte de diversité (Oladele and Sadiku, 2013). D'autre part, si les individus dans une population ont des valeurs de Fitness qui se rapprochent, il sera très difficile pour la population de se déplacer vers les meilleurs éléments (les probabilités de sélection pour les bons et mauvais individus sont très similaires).

2.3.2.3. Sélection linéaire par rang.

Cette technique est appelée aussi *Linear Rank-based Selection* ou simplement *Linear Ranking selection* (Baker, 1985). C'est la technique de sélection où la probabilité qu'un individu soit sélectionné dépend du rang de la Fitness par rapport à la population entière (Noraini and Geraghty, 2011). Elle a été

introduite par (Baker, 1985), pour éliminer les inconvénients de la sélection par tournoi (ou proportionnelle). La sélection par rang commence par classer les individus dans la population en fonction de leur Fitness, puis de calculer les probabilités de sélection en fonction de leurs rangs plutôt que des valeurs de leurs Fitness. Chaque individu de la population obtient un rang différent, même si leurs probabilités sont les mêmes (Shukla et al., 2015).

La technique de sélection par rang se déroule en deux étapes (Baker, 1985). Dans la première étape, la population est triée selon la Fitness de ses individus. En deuxième lieu, le rang est assigné dans selon l'ordre correspondant à la sélection par roulette, où N est le rang donné au meilleur individu, et 1 est affecté au pire. La probabilité de sélection d'un individu I_i est donnée dans l'équation (3) :

$$P_i = \frac{1}{N} \left(n^- + (n^+ - n^-) \frac{i-1}{N-1} \right); i \in \{1, \dots, N\} \quad (2.3)$$

$P_i, \frac{n^+}{N}, \frac{n^-}{N}$ représentent la probabilité de sélection du $i^{ème}$ individu, du meilleur individu et du pire individu, respectivement.

Selon (Baker, 1985), cette technique arrive à maintenir une pression constante dans la recherche évolutive, où elle introduit une échelle uniforme à travers la population et n'est pas influencée par les *super-individus* (avec la meilleure Fitness) comme dans la sélection par roulette. La sélection par rang peut éviter la convergence prématurée, mais peut être coûteuse sur le plan des calculs en raison de la nécessité de trier les populations (Noraini and Geraghty, 2011). La sélection par rang aide à prévenir la convergence prématurée due à des *super* solutions, puisque le meilleur individu a toujours la même probabilité de sélection, quelle que soit sa valeur de Fitness. Cependant, cette méthode peut conduire à une convergence plus lente, parce que les meilleurs individus ne diffèrent pas beaucoup des autres.

2.3.2.4. Sélection exponentielle par rang.

La sélection exponentielle par rang ou de classement (ou *Exponential Ranking Selection*) (Whitley, 1989) diffère de la sélection linéaire dans l'assignement des probabilités (Asco, 2013). Ce dernier se fait d'une manière exponentiellement pondérées (Blickle and Thiele, 1995). La base de l'exposant est le paramètre $0 < c < 1$ de la méthode. Plus la valeur de c est proche de 1, plus l'augmentation exponentielle de la technique de sélection est faible. Tout comme la sélection linéaire, le rang N est attribué au meilleur individu et le rang 1 au plus mauvais individu. Ainsi, les probabilités des individus sont données par l'équation (2.4) (Blickle and Thiele, 1995) :

$$p_i = \frac{c^{\lambda-i}}{\sum_{j=1}^{\lambda} c^{\lambda-j}} \quad \forall i \in [1 \dots \lambda] \text{ et } 0 < c < 1 \quad (2.4)$$

Puisque $\sum_{j=1}^{\lambda} c^{\lambda-j}$ normalise les probabilités pour assurer que $\sum_{i=1}^N p_i = 1$, l'équation (2.4) s'écrit :

$$p_i = \frac{c-1}{c^{\lambda}-1} c^{\lambda-i} \quad \text{pour } i \in [1 \dots \lambda] \text{ et } 0 < c < 1 \quad (2.5)$$

La procédure pour cette sélection est similaire à celle de la sélection linéaire par rang, la seule différence réside dans le calcul des probabilités de sélection (figure 2.9).

2.3.2.5. Sélection uniforme.

La technique de sélection uniforme (ou *Uniform Selection*) reflète le fonctionnement d'une sélection dite *naturelle*, ayant été utilisée - au début - principalement dans les recherches en biologie e.g. (Ehrlich and Raven, 1969) (Levin, 1979) (Cohan, 1984) (Ehrlich and Raven, 1992) (Goldstein and Holsinger, 1992). Par la suite, la sélection uniforme a trouvé sa place dans le domaine de l'informatique évolutionnaire (Hoffman, 1986) (Davis, 1991) (Khare et al., 2003) (Praditwong and Yao, 2006) (Sivanandam and Deepa, 2007) (Yoo et al., 2013) (Thada and Dhaka, 2014).

Procédure Sélection exponentielle par rang
Entrée La population $P(T)$ et la base de classement $c \in]0, 1]$
Sortie La population après la sélection $P(T)'$
Expo_rank($c, J_1, \dots, J_N$)
 $\bar{J} \leftarrow$ Population J triée selon la Fitness avec le pire individu en 1^{ère} place
 $s_0 \leftarrow 0$
Pour tout i de 1 à N **faire**
 $s_i \leftarrow s_{i-1} + p_i$ /* Equation 5 */
Ou
Pour tout i de 1 à N **faire**
 $r \leftarrow \text{random}[0, s_N[$
 $J'_i \leftarrow \bar{J}_l$ tel que $s_{l-1} \leq r < s_l$
Ou
Retourner $\{J'_1, \dots, J'_N\}$

Figure 2.9 Procédure de sélection exponentielle par rang

La sélection uniforme ne fait appelle à aucune valeur d'adaptation, pour biaiser la démarche de sélection des solutions. La sélection uniforme se fait aléatoirement et uniformément où chaque individu de la population a une probabilité de $1/N$ d'être sélectionné, pour le tour suivant, N étant le nombre total d'individus composant la population des solutions. Ce qui donne formellement (Hoffman, 1986) :

$$S_{unif}((g_1, \dots, g_N)) = (g'_1, \dots, g'_L) \quad \text{tel que} \quad Pr(g'_i = g_j) = 1/N \quad (2.6)$$

Pour tout $i \in [L], j \in [N]$

Bien que le principe de la sélection uniforme paraisse simpliste, il n'empêche que cette technique est susceptible d'augmenter le taux de divergence génétique entre les populations (Cohan, 1984) (Hoffman, 1986).

2.3.2.6. Sélection par troncation.

La technique par troncation ou *Truncation Selection* a été introduite par (Muhlenbein and Schlierkamp-Volsen, 1993), dans le domaine des algorithmes génétiques. Dans la technique de sélection par troncation avec un seuil T ,

la fraction T des meilleurs individus est sélectionnée (avec la même probabilité de sélection) (Blickle and Thiele, 1995). Muhlenbein and Schlierkamp-Volsen (1993) expliquent cette technique de la sorte que les meilleurs individus T sont sélectionnés et accouplés de façon aléatoire jusqu'à ce que le nombre de descendants soit égal à la taille de la population. La génération de descendant remplace la population parente (La meilleure personne trouvée jusqu'à présent restera dans la population). L'auto-accouplement est interdit. Cette méthode est équivalente à la stratégie de sélection (μ, λ) utilisée dans les stratégies d'évolution avec $T = \mu/\lambda$ (Schwefel, 1981) (Back, 1991).

La figure 2.10 présente la procédure de la sélection par troncation.

Procédure Sélection par troncation

Entrée Population $P(T)$, seuil de troncation $T \in [0,1]$

Sortie Population après la sélection $P(T)'$

Troncation($T, J_1, \dots, J_N$) :

$\bar{J} \leftarrow$ Population triée J selon la Fitness avec le pire individu en pôle position

Pour tout i de 1 à N **faire**

$r \leftarrow \text{random}\{[(1 - T)N], \dots, N\}$

$J'_i \leftarrow \bar{J}_r$

Fin pour

Retourner $\{J'_1, \dots, J'_N\}$

Figure 2.1 Procédure de la sélection par troncation

Type de sélection	Auteur	Principe	Avantage	Inconvénient	Taille population ⁴	Nombre itérations ⁵	Complexité	Intensité sélection ⁶	Taux reproduction ⁷
Uniforme	$N.C^s$	Faire un choix au hasard.	Facile à implémenter.	Risque de ne pas sélectionner les bons individus.	Peu importe	Moyen	$O(n)$	$\frac{\bar{\alpha}}{M}$	$\frac{f_i}{M}$
Par tournoi	Brindle 1981	Faire des tournois entre les individus.	Chaque individu a une chance d'être sélectionné.	Peut dégrader la vitesse de convergence.	Peu importe	Petit	$O(n)$	$\sqrt{2(\ln t - \ln(\sqrt{4.14 \ln t}))}$	$t(\frac{\bar{S}(f)}{N})^{t-1}$
Linéaire par rang	Baker 1985	Classer les individus par rang.	Chaque individu a une chance d'être sélectionné.	Peut dégrader la vitesse de convergence.	Petit	Petit	$O(n \ln n) + \text{temps de sélection}^9$	$(1 - n^-) \frac{1}{\sqrt{\pi}}$	$n^- + 2 \frac{1 - n^-}{N} \bar{S}(f)$
Par roulette	Goldberg 1989	Principe de la roulette.	Facile à implémenter.	Perte de diversité.	Petit	Petit	$O(n^2)$	$\frac{\bar{\sigma}}{n}$	$\frac{F_i}{\sum_{i=1}^n F_i}$
Exponentielle par rang	Whitkey 1989	Idem que la sélection linéaire sauf pour le calcul de la probabilité de sélection.	Chaque individu a une chance d'être sélectionné.	Peut dégrader la vitesse de convergence.	Petit	Petit	$O(n \ln n) + \text{temps de sélection}$	$0.588 \frac{\ln \ln \frac{\pi}{\alpha}}{3.69^\alpha}$	$\frac{k \ln k}{k-1} k^{-\bar{S}(f)/N}$
Par troncation	Mühlenbein 1993	Utilise un seuil pour la sélection du nombre d'individus.	Convergence rapide.	Les moins bons individus sont condamnés.	Peu importe	Petit	$O(n \ln n)$	$\frac{1}{T} \frac{1}{\sqrt{2\pi}} \exp(\frac{f_i^2}{2})$	$\frac{1}{T}$

Tableau 2.2. Synthèse sur les types de sélection dans la littérature

⁴ Indique une estimation de la taille de la population minimale, pour obtenir des résultats convaincants⁵ Indique une estimation du nombre d'itérations minimal, pour obtenir des résultats convaincants.⁶ Pression ou intensité de la sélection.⁷ Taux ou ratio de reproduction.⁸ Non connu ou non défini dans la littérature scientifique.⁹ Entre $O(n)$ et $O(n^2)$

2.3.3. Synthèse

Le tableau 2.2 présente une synthèse des différentes techniques de sélection étudiées. Plusieurs études ont tenté de comparer les différentes techniques de sélection utilisées avec un algorithme génétique (Bäck and Hoffmeister, 1991) (Goldberg and Deb, 1991) (Blickle and Thiele, 1995) (Miller and Goldberg, 1995) (Bäck, 1996) (Michalewicz and Hartley, 1996) (Alfonso et al., 1998) (Julstrom, 1999) (Zhang and Kim, 2000) (Mashohor et al., 2005) (Zhong et al., 2005) (Al Jadaan et al., 2008) (Razali and Geraghty, 2011) (Filipović, 2012) (Kumar, 2012) (Pandey, 2015) (Shukla et al., 2015) (Pandey, 2016). Pour notre recherche, nous nous sommes contentés de recenser les techniques de sélection les plus mentionnées dans la littérature.

Selon Blickle and Thiele (1995), la performance d'un type de sélection est généralement évaluée en termes de taux de convergence et le nombre de générations pour atteindre les solutions optimales. L'un des travaux les plus marquants est sans conteste celui présenté par (Goldberg and Deb, 1991). Ils ont présenté un travail colossal en comparant les techniques de sélection par rang, par tournoi, et uniforme selon des critères comme le temps d'exécution, la complexité et le taux de convergence. Blickle and Thiele (1995) ont présenté une étude - très - détaillée sur nombreuses techniques de sélection. Ils affirment qu'avec la sélection par tournoi, seules des valeurs discrètes peuvent être utilisées, alors que la sélection linéaire par rang ne permet qu'une plage plus petite pour l'intensité de sélection.

Alors que pour la même intensité de sélection, la technique de sélection par troncature conduit à une variance¹⁰ de sélection beaucoup plus faible que les techniques par rang ou par tournoi (Blickle and Thiele, 1995). Alfonso et al. (1998) ont présenté une étude comparative sur des techniques de sélection avec de

¹⁰ La variance de sélection est la variance normalisée attendue de la distribution de la Fitness de la population, après avoir appliqué une technique de sélection à la distribution de la Fitness.

multiples fonctions de Fitness. Ils affirment que les différentes techniques se valent toutes, et la différence peut se faire selon les paramètres utilisés lors de l'expérimentation. Zhong et al. (2005) ont comparé la technique de sélection par tournoi avec celle par roulette, et ont conclu que la sélection par tournoi est plus efficace en matière de convergence. Al Jadaan et al. (2008) ont présenté une étude comparative entre les techniques de sélection par roulette et linéaire par rang ; en utilisant plusieurs fonctions de Fitness. Il en ressort que la sélection linéaire par rang est meilleure, du point de vue du nombre de générations à utiliser pour atteindre un résultat satisfaisant. Julstrom (1999) a étudié l'efficacité du temps de calcul de deux types de sélections : Par rang et par tournoi. Il souligne la rapidité de la technique de sélection par tournoi comparée à la technique par rang. (Mashohor et al., 2005) ont évalué la performance d'un système d'inspection de circuits imprimés en utilisant un simple algorithme génétique avec la technique de sélection par tournoi, par roulette et par rang. Razali and Geraghty (2011) et Pandey (2016) ont traité le problème du voyageur de commerce avec un algorithme génétique en utilisant trois types de sélections (par tournoi, roulette et rang). Les résultats obtenus sont intéressants dans le sens qu'ils peuvent varier selon la taille de la population utilisée avec une technique de sélection. Un autre travail de recherche - récent - digne d'être cité est celui de (Shukla et al., 2015), dans lequel ils comparent la sélection par tournoi, par rang (linéaire et exponentielle) et par roulette.

De ces différentes et nombreuses études ressort le constat suivant : Affirmer qu'une technique de sélection est la meilleure en toute circonstance, est erronée. Il serait plus judicieux de dire que selon le contexte (le problème d'optimisation à traiter, la taille de la population, le nombre d'itérations) une technique puisse égaler ou surpasser une autre.

Conclusion

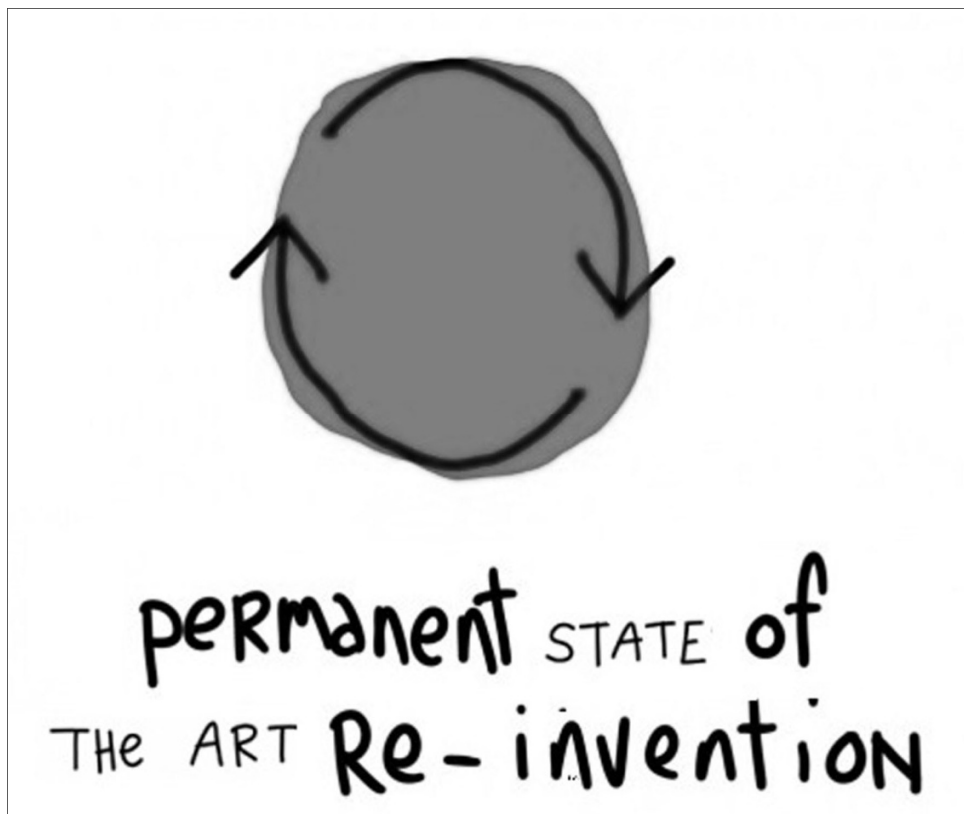
Ce chapitre a été consacré, en premier temps, à la définition du concept d'algorithme évolutionnaire. Il peut être résumé ainsi :

- Un algorithme évolutionnaire est un algorithme stochastique d'optimisation se basant sur un vulgaire parallèle de la théorie de Darwin sur l'évolution biologique des populations.
- Les algorithmes évolutionnaires regroupent un ensemble de méthodes, approches et algorithmes différents.

En deuxième temps, le chapitre s'est intéressé à un type particulier des algorithmes génétiques, le NSGAI :

- Il est fréquemment utilisé pour résoudre les problèmes d'optimisation combinatoires multicritères.
- Il adopte une sélection binaire par tournoi, un croisement binaire simulé et une mutation à un point.
- La distance de peuplement ou *crowding* : Pour la sélection des individus à ajouter dans la population renfermant les élites.

Ce chapitre se termine par la présentation d'un certain nombre d'opérateurs de sélection, en grande partie tirés d'une étude proposée par (Goldberg and Deb, 1991) et (Blickle and Thiele, 1995). Ces techniques de sélection sont - hypothétiquement - utilisables avec l'algorithme évolutionnaire NSGAI. Du moment que le choix de l'opérateur de sélection a une considérable importance, dans un algorithme génétique, il a été évident de présenter ces techniques de sélection, et d'en faire la comparaison.



Chapitre 3

Etat de l'art sur l'optimisation multicritères des processus métiers

Introduction	52
3.1. Classification des méthodes de modélisation des processus métiers.....	52
3.1.1. Les modèles schématiques.....	53
3.1.2. Les modèles formels et/ou mathématiques	55
3.1.3. Les langages des processus métiers	58
3.2. Synthèse	60
Conclusion	64

Introduction

La modélisation des processus métiers est directement liée à la perception et à la compréhension de ces processus métier. Dans la plupart des cas, un processus métier est aussi expressif et aussi communicatif que la technique qui a été utilisée pour le modéliser (Vergidis, 2008). Selon Volkner et Werners (2000) et Aguilar-Saven (2004), la modélisation des processus métiers est essentielle à l'analyse, à l'évaluation et à l'amélioration (en vue d'une optimisation) des processus métiers. Par conséquent, les éléments d'un modèle de processus métier jouent un rôle important dans sa description ainsi que sa compréhension. Même si dans la littérature de la modélisation des processus métiers IDEF¹¹ et les réseaux de Pétri sont fréquemment rencontrés, nous nous proposons de présenter la classification de ces techniques de modélisation selon le travail de Vergidis (2008). Cette classification est utilisée ; par la suite, pour présenter l'état de l'art des différents travaux sur l'optimisation des processus métier avec les métaheuristiques.

3.1. Classification des méthodes de modélisation des processus métiers

Selon Van Der Aalst et al. (2003), la modélisation des processus métiers est utilisée pour expliquer l'identification et la spécification de ces dernières. Elle comprend la modélisation des activités et de leurs relations causales et temporelles (Vergidis and Tiwari, 2008). Alors qu'il existe une abondance de techniques de modélisation des processus métiers e.g. celle de (Kettinger et al., 1997), (Melao and Pidd, 2000) et (Aguilar-Saven, 2004) avec des approches qui capturent les différents aspects

¹¹ Integrated DEFinition Methods

d'un processus métier, chacun ayant des avantages distinctifs et des inconvénients. Nous n'avons aucune intention de présenter les différentes classifications citées plus haut (ce qui sortirait du cadre du présent document). Nous avons opté pour la classification élaborée par Vergidis (2008). La figure 3.1 représente la classification des techniques de modélisation des processus métiers mise à jour par Wibig (2013).

La classification des techniques existantes de modélisation des processus métiers comprend trois (03) ensembles : Des modèles schématiques, des modèles mathématiques et des langages de processus métiers. Cette classification permet de mettre en évidence visuellement un certain nombre d'observations intéressantes, et d'appuyer l'organisation de la suite de ce chapitre.

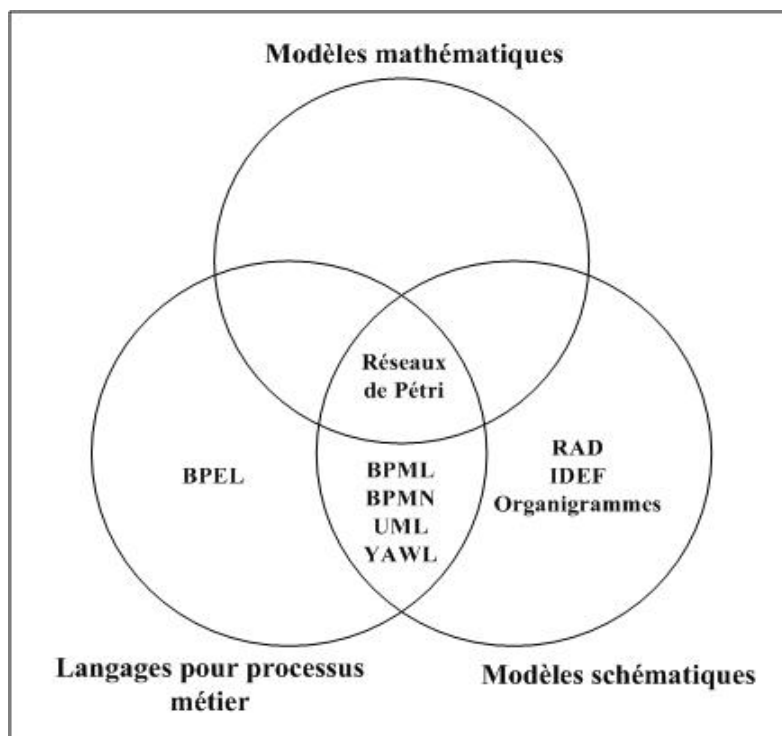


Figure 3.1 Classification des techniques de modélisation des processus métiers

3.1.1. Les modèles schématiques

Les premières techniques utilisées pour la modélisation des processus métiers ont été des représentations graphiques simples (i.e. des organigrammes), développées

initialement pour la spécification logicielle (Knuth, 1963) (Chapin, 1971). L'idée est de représenter un processus métier sans utiliser une notation standardisée (Havey, 2005). Elles sont utiles pour la représentation rapide et informelle d'un processus métier, mais manquent de sémantique pour des représentations plus complexes (Vergidis and Tiwari, 2008). Phalp and Shepperd (2000) notent que toute tentative d'analyse utilisant ces types de modèles consiste fréquemment à l'inspection de diagrammes et les conclusions sont fortement dépendantes des compétences de l'analyste. Parmi ces techniques, nous pouvons citer : IDEF, RAD¹² et les organigrammes.

Vue de notre axe de recherche, les premiers travaux à citer sont ceux de (Zhou and Chen, 2002, 2003). Les auteurs proposent d'utiliser, en premier lieu, IDEF3 pour représenter les processus métiers, puis ils ajoutent une extension sous la forme d'un graphe orienté pondéré. Ils font appel à un simple algorithme génétique (pour (Zhou and Chen, 2002)) puis NSGAI (pour (Zhou and Chen, 2003)), et une fonction de Fitness assez lourde à implémenter. Stelling et al. (2009) tentent de répondre à la question de "comment représenter un processus métier comme un chromosome ?". Pour y arriver, ils modélisent un processus métier sous la forme d'une taxonomie d'une hiérarchie (avec un domaine, des activités et des transactions, comme différents niveaux de cette hiérarchie), avec l'aide de BPMN¹³. Lors de l'implémentation, les auteurs utilisent un simple algorithme génétique avec la programmation à base de règles. De leur côté, Ahmadikatouli and Aboutaleb (2011) abordent le problème de la modélisation d'un processus métier, dans une optimisation multicritères en utilisant les réseaux de Pétri, tout comme (Wibig, 2013) (dont nous reparlerons à la fin du chapitre). Ils utilisent un simple algorithme génétique, tandis que le dernier utilise NSGAI avec de la programmation dynamique, pour générer de nouvelles solutions. Le travail de (Molka et al., 2015) peut être ajouté à ceux usant des modèles schématiques, même

¹² Rapid Application Development

¹³ Business Process Model and Notation

s'il utilise des modèles en BPMN qui seront transformés en réseaux de Pétri, pour permettre de mesurer la qualité des processus métiers (avec ProM de (Van Der Aalst et al., 2009)).

3.1.2. Les modèles formels et/ou mathématiques

La nécessité d'une sémantique formelle pour la modélisation des processus métiers a conduit à une deuxième génération de modèles formels. Ils permettent de définir les concepts de processus métier avec rigueur et précision, de sorte que les mathématiques peuvent être utilisées pour les analyser, en extraire des connaissances et de raisonner à partir d'eux (Vergidis and Tiwari, 2008). Un des avantages des modèles formels est qu'ils peuvent être vérifiés mathématiquement (Koubarakis and Plexousakis, 2002). Hofacker et Vetschera (2001) expliquent que la construction d'un modèle formel de processus métier peut s'avérer beaucoup plus complexe et - assez - exigeant par rapport aux techniques traditionnelles où un diagramme de processus est suffisant. En plus de la complexité, voir de l'impossibilité à représenter des processus métiers de la vie réelle à l'aide de modèles mathématiques (Hofacker and Vetschera, 2001).

Le premier travail cité est celui de (Hofacker and Vetschera, 2001). Ils proposent de modéliser un processus métier à l'aide d'un cube. Il est illustré à la figure 3.2. Un processus métier $\vec{P}$ est représenté sous la forme d'un cube $\vec{C}$, qui peut être interprété comme suit : Une ressource b_1 appartient à l'ensemble des entrées et est consommée par l'activité a_1 , ce qui conduit à $\vec{C}_{0.1.1} = 1$. En outre, une ressource b_1 est produite par l'activité a_2 , et consommée par a_4 , $\vec{C}_{2.4.1} = 1$. La ressource b_3 est produite par l'activité a_1 et consommée par a_∞ , i.e. qu'elle est une sortie globale du processus métier respectif. Les auteurs implémentent un simple algorithme génétique. Même si les résultats obtenus ne sont pas *concluant* (Vergidis et al., 2006). Ce premier travail montre le chemin, pour ce qui sera le racine de travaux marquants dans la voie de l'optimisation des processus métiers avec les algorithmes évolutionnaires (les travaux de Vergidis et Tiwari peuvent en témoigner).

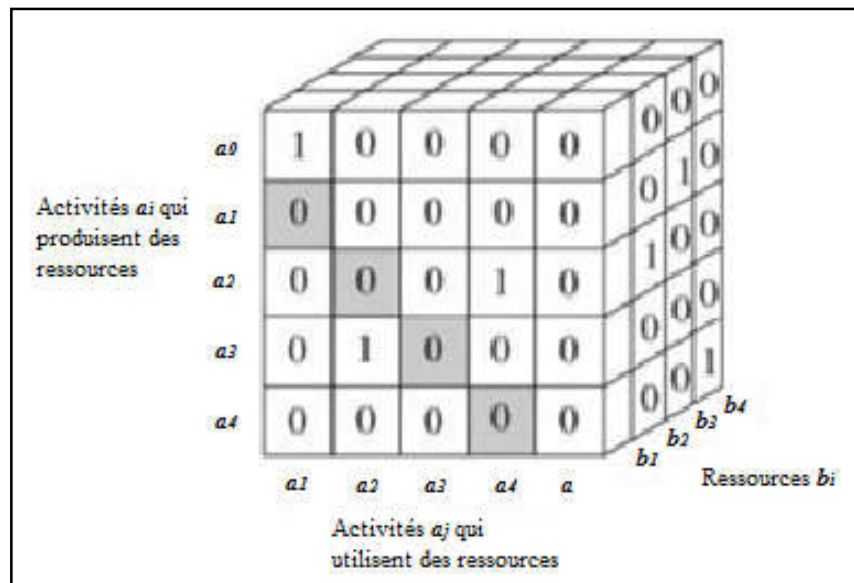


Figure 3.2 Représentation d'un processus métier en cube

Le travail de (Vergidis et al., 2006) reprend le postulat qu' "une conception de processus métier est réalisable, si une ressource est générée par une certaine activité (ou une tâche)", proposé par (Hofacker and Vetschera, 2001). Ils proposent un Framework, pour automatiser la création de processus métiers alternatifs selon un modèle défini. Une conception d'un processus métier est présentée comme un graphe orienté pondéré et deux matrices : Une matrice tâche/ressource et une matrice tâche/attribut. Les auteurs utilisent deux fonctions de Fitness : Deux fonctions d'addition (une pour le coût et une autre pour le calcul de la durée). L'algorithme NSGAI est utilisé par le Framework proposé. Ce travail donnera suite à une série de travaux des mêmes auteurs (Vergidis et al., 2007a, 2007b, 2008, 2012) et (Tiwari et al., 2006, 2010a, 2010b). Ces travaux de recherche reprennent et améliorent celui de (Vergidis et al., 2006), soit en mettant à jour l'algorithme utilisé pour la vérification de la structure des instances de processus métiers générées. Soit en utilisant NSGAI pour (Vergidis et al., 2006, 2007b, 2008) et (Tiwari et al., 2006), ou différents algorithmes évolutionnaires : SPEA2, MOPSO pour (Vergidis et al., 2007a), PAES pour (Tiwari et al., 2010a), PAES, PESA2, SPEA2 pour (Tiwari et al., 2010b) en plus de NSGAI, pour comparer les résultats. Dans (Vergidis et al., 2015), ils présentent la dernière version de leur Framework

pour l'optimisation multicritères des processus métiers, avec le coût d'un processus métier et la satisfaction vis-à-vis d'un processus métier comme critères à optimiser. Ils ajoutent la proposition qu' "une tâche d'un processus métier peut être considérée comme un service web". Ce qui permettra par la suite de présenter une partie expérimentation très riche, (11 pages, sur les 19 pages de l'article). Le dernier travail en date de Vergidis est (Georgoulakos et al., 2017). Il y propose le même Framework d'optimisation des conceptions de processus métiers en y ajoutant ; ce qu'ils appellent, un prétraitement de la librairie des tâches composants les processus métiers. Ce qui permet de gagner en temps d'exécution.

D'autres travaux méritent d'être cités. (Wang et al., 2009) qui proposent d'optimiser la création des instances de processus métiers (avec un simple algorithme génétique), en se concentrant sur les ressources de ces derniers, ainsi que la stratégie de planification des tâches les composant. La représentation des processus métiers fait appelle à un modèle mathématique. (Salomie et al., 2012) et (Pop et al., 2012) sont les seuls à utiliser des algorithmes évolutionnaires, autres que les algorithmes génétiques. Salomie et al. (2012) utilisent une matrice (activité*entrée*sortie*ressource), pour représenter fidèlement un processus métier. Le Framework qui résulte de cette étude fait appel à l'algorithme des lucioles, avec une fonction de Fitness intéressante (qui change des fonctions d'addition proposées, par le passé). Pour sa part, Pop et al. (2012) reprennent le travail de (Salomie et al., 2012) (même équipe de recherche), en y ajoutant la vérification de la structure d'une instance d'un processus métier, mais cette fois, en commençant par le résultat (produit ou service final) et de revenir vers le début du processus (les ressources d'entrées). Autre distinction avec le travail de (Salomie et al., 2012) est le fait de recourir à deux algorithmes bio-inspirés (un algorithme à la fois, pas d'hybridation) autres que les algorithmes génétiques : L'algorithme d'optimisation basé sur les fourmis, et l'algorithme d'optimisation basé sur les abeilles. (Farsani et al., 2013) (Figure 3.3) présentent le seul travail où un ou plusieurs opérateurs d'un algorithme génétique sont modifiés. Ils reprennent le travail de (Vergidis and Tiwari, 2008). Pour enfin arriver au travail de (Bae et al., 2014), ils présentent une méthodologie pour intégrer la sémantique des processus métiers dans la gestion

des processus métiers (BPM). L'approche consiste en une formulation du plan d'exécution d'un processus métier avec la programmation en nombres entiers et un simple algorithme génétique. Les auteurs utilisent une représentation mathématique, pour modéliser les processus métiers.

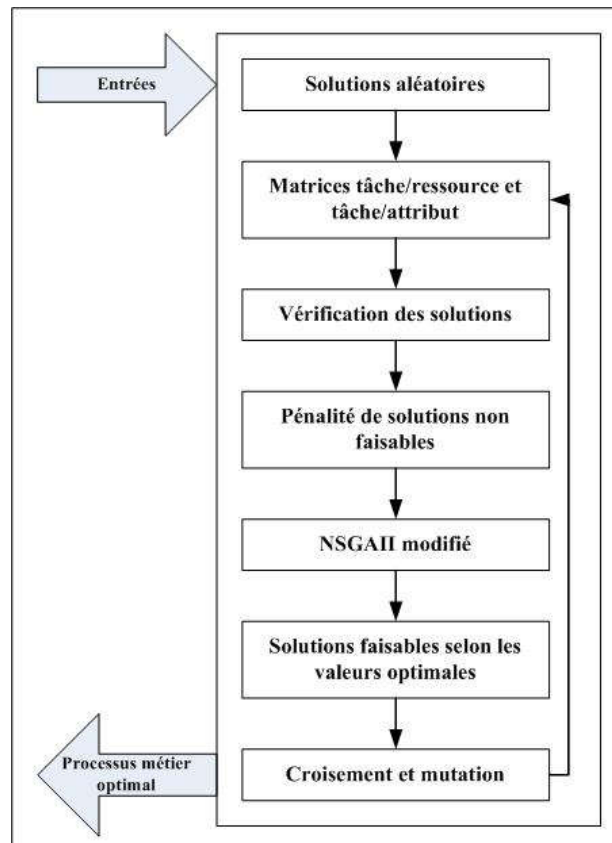


Figure 3.3 Framework proposé par (Farsani et al., 2013)

3.1.3. Les langages des processus métiers

La troisième génération - et la plus récente - des techniques de modélisation des processus métiers provient de la complexité des modèles formels, mais conserve leur cohérence et leur potentiel d'analyse (Vergidis and Tiwari, 2008). Cet ensemble de techniques utilisent - habituellement - des langages de processus basés sur XML, pour modéliser et exécuter un processus métier. Van der Aalst et al. (2003) remarquent que les langages pour processus métier (avec une sémantique

claire) sont utiles, pour exprimer des modèles de processus métiers et contribuer à l'analyse de leurs structures.

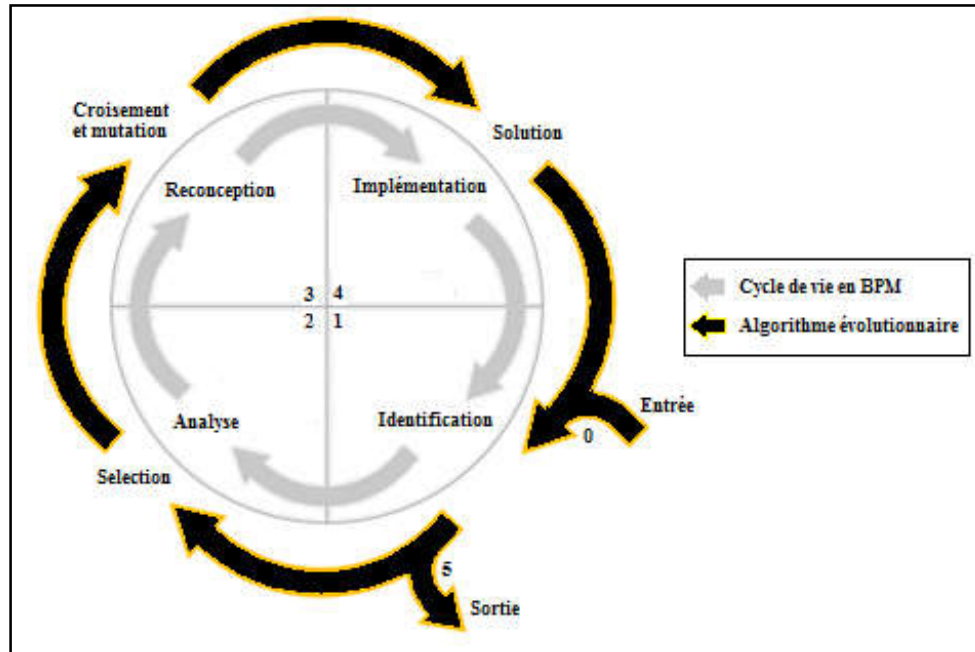


Figure 3.4 Correspondance entre le cycle de vie en BPM et un algorithme évolutionnaire¹⁴

(Afflerbach et al. 2016) tentent d'optimiser la structure des processus métiers en utilisant BPMN. Ils s'aident d'un ensemble de matrices (activité/attribut, objet/attribut, activité/entrée, activité/sortie, activité/processus/attribut), pour la représentation d'un processus métier. L'optimisation implémentée utilise un simple algorithme génétique, et fait un intéressant parallèle de son fonctionnement avec celui du cycle vie en BPM (Figure 3.4).

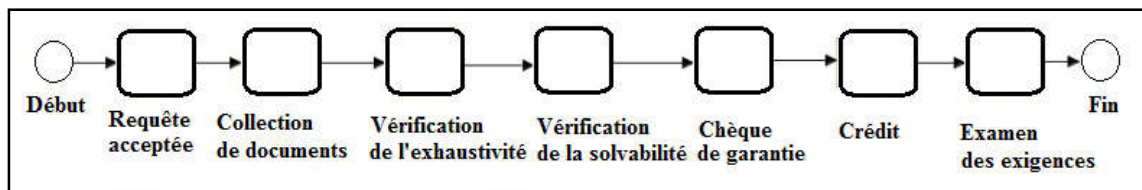


Figure 3.5 Processus métier de la gestion des exigences de crédit (Djedović et al. 2016)

¹⁴ Adapté de (Afflerbach et al. 2016)

(Djedović et al. 2016) suggèrent une méthode capable d'optimiser l'allocation des ressources à la suite de tâches composant un processus métier donné. La modélisation des processus métier se fait avec BPMN, puis implémenter sous MATLAB. Les auteurs implémentent un simple algorithme génétique, et une fonction de Fitness qui utilise le temps d'attente du processus métier et son coût en ressources, comme critères à optimiser. Il est bon de préciser que ce travail se contente d'effectuer des expérimentations sur un modèle de processus métier avec une structure linéaire (voir Figure 3.5).

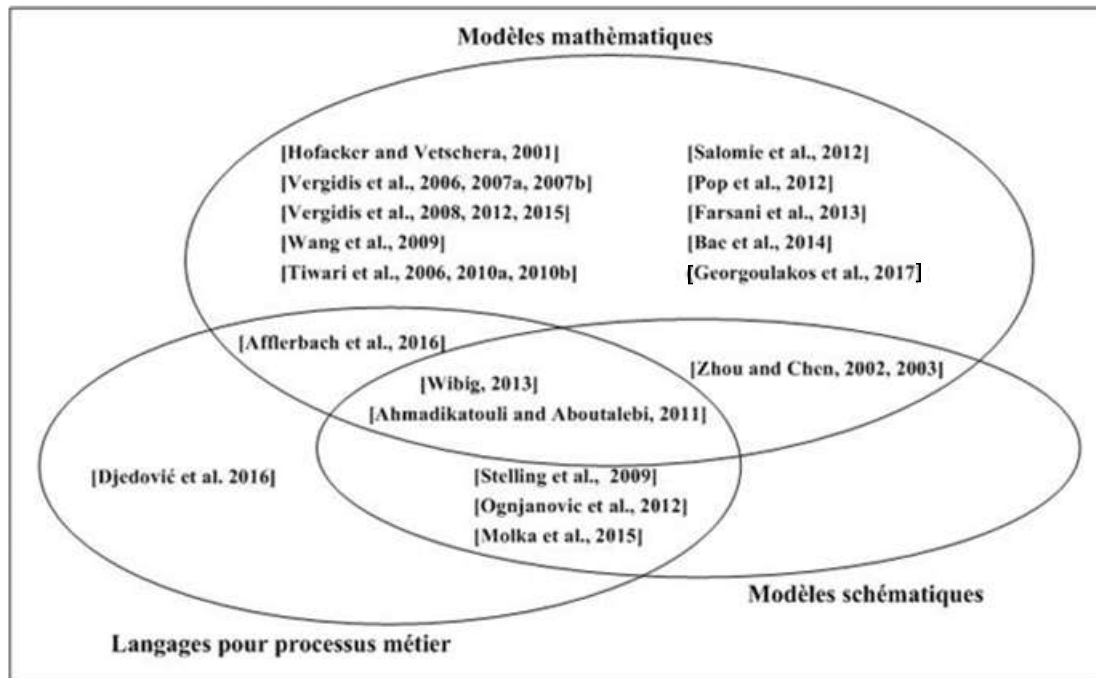


Figure 3.6 Classification des travaux selon les techniques de modélisation pour processus métier

3.2. Synthèse

Pour mettre en avant la diversité des travaux sur l'optimisation multicritères des processus métiers avec des métaheuristiques, nous nous sommes penchés sur les techniques de modélisation utilisées pour représenter un processus métier, le plus fidèlement possible. Vergidis et Tiwari (2008) affirment que toute technique de modélisation des processus métiers devrait être en mesure de supporter les différentes spécificités de ces processus métiers.

La figure 3.6 présente la classification proposée par (Vergidis, 2008), reprise et mise au goût du jour par (Wibig, 2013). Ce dernier met les réseaux de Pétri comme élément commun entre les différentes techniques de modélisation des processus métiers. Il appuie sa proposition par l'affirmation que l'utilisation des réseaux de Pétri arrive à reprendre la simplicité proposée par les techniques utilisant les schémas, avec la sémantiques des modèles mathématiques et la standardisation des langages pour processus métiers. Ce qui nous ramène à la figure 3.6 qui révèlent - clairement - que les modèles mathématiques sont les plus prisés par les chercheurs dans l'optimisation des processus métiers. Cela peut s'expliquer en partie par la maniabilité et la souplesse qu'apporte un modèle mathématique à être conçu, compris, implémenté, amélioré et personnalisable à volonté. Tout le contraire des recherches se basant sur les modèles schématiques, assez rigides. Cette rigidité a obligé les chercheurs à utiliser d'autres techniques de représentation des processus en plus des modèles schématiques, e.g. (Stelling et al., 2009) et Molka et al., 2015) en demandant assistance à BPMN, ou encore (Zhou and Chen, 2002, 2003) avec un modèle mathématique en plus d'utiliser IDEF. Autre point intéressant, les travaux avec un unique langage de modélisation de processus métiers sont rares. (Djedović et al. 2016) utilisent BPMN pour représenter un processus métier, mais ils se contentent d'utiliser une structure linéaire pour ce dernier (contre une représentation en graphe orienté pondéré utilisée dans les travaux de Vergidis). Ce choix de représentation réduit grandement la diversité des conceptions des processus métiers qui doivent être générés avec un simple algorithme génétique, sachant le potentiel qu'offrent les langages de modélisation des processus métiers à représenter des structures complexes.

Avec le tableau 3.1, nous nous proposons de comparer les travaux étudiés selon des critères autres que les techniques de modélisation des processus métiers. En premier temps, nous remarquons que le nombre de critères utilisés dans la problématique de l'optimisation multicritères des processus métiers est souvent égal à deux, mise à part avec les travaux de (Molka et al., 2015) et (Afflerbach et al., 2016) où le nombre de critères d'optimisation n'est pas nettement spécifié.

Il faut savoir qu'une optimisation multicritères se limite à deux ou trois critères à optimiser, alors que plus de critères impliquerait une optimisation *many* critères (Farina and Amato, 2004). Le tableau 3.1 montre clairement que les algorithmes évolutionnaires sont dominants dans le problème d'optimisation étudié. Deb et al., (2000, 2002) recommandent l'utilisation de NSGAII (et les algorithmes génétiques indirectement), pour la simplicité de l'implémentation qui leur est propre et la rapidité avec laquelle ils arrivent à générer de nouvelles solutions.

Autre constat est que plus de la moitié des travaux présentés utilise NSGAII, comme algorithme évolutionnaire d'optimisation, suivi de très près de l'algorithme génétique canonique (12 travaux pour NSGAII, contre 09 pour l'algorithme génétique). Même si les algorithmes génétiques, en général, et NSGAII ; en particulier, semblent les métaheuristiques de prédilection des travaux sur l'optimisation multicritères des processus métiers, quelques travaux sortent du lot. A titre d'exemple, (Salomie et al., 2012) et (Pop et al., 2012) qui implémentent un algorithme évolutionnaire basé sur les lucioles, pour le premier. Puis, deux (02) autres algorithmes évolutionnaires basés sur les colonies de fourmis, et les abeilles, pour le second travail. De son côté, (Farsani et al., 2013) sont les seuls à revoir les opérateurs (croisement et mutation) de NSGAII, alors qu'aucune recherche ne s'est aventurée à modifier, repenser ou proposer un opérateur de sélection de l'algorithme évolutionnaire adopté.

Le tableau 3.1 offre une vue exhaustive sur le types de fonction de Fitness utilisée. Entre un modèle mathématique (08 études) pour évaluer la qualité d'un processus métier en s'aidant d'une métaheuristique. Puis, 12 études où la fonction de Fitness se résume à l'utilisation d'une simple fonction d'addition par critère d'optimisation, e.g. pour calculer la valeur du critère d'optimisation C_i d'un processus métier PM_k composé de n tâches, il suffit d'utiliser la fonction d'addition $\sum_{j=1}^n c_j$ où c_j représente la valeur du critère d'optimisation d'une tâche $t_i \in PM_k$. Ce qui implique de se retrouver avec deux fonctions de Fitness (une fonction par objectif à optimiser), dans le cas d'une optimisation avec deux (02) critères. La fonction de Fitness occupe un rôle important dans la phase d'évaluation des solutions générées par le métaheuristique (Machado et al., 2014), et cette façon de procéder peut

induire à de mauvaises évaluations, appréciations et - enfin - comparaisons des solutions.

Travail	Technique de modélisation			Nombre critères à optimiser	Type de fonction de Fitness		Métaheuristique		
	Langage pour pm ¹⁵	Mathématique	Schématique		Fonction d'addiction	Modèle mathématique	AG ¹⁶	NSGAII	Autre
[Hofacker and Vetschera, 2001]		✓		02	✓		✓		
[Zhou and Chen, 2002]		✓	✓	02		✓	✓		
[Zhou and Chen, 2003]		✓	✓	02		✓		✓	
[Vergidis et al., 2006]		✓		02	✓			✓	
[Vergidis et al., 2007a]		✓		02	✓			✓	SPEA2, MOPSO
[Tiware et al., 2006]		✓		02	✓			✓	
[Vergidis et al., 2007b]		✓		02	✓			✓	
[Vergidis and Tiware, 2008]		✓		02	✓			✓	
[Stelling et al., 2009]	✓		✓	02	-			✓	
[Wang et al., 2009]		✓		02		✓	✓		
[Tiware et al., 2010a]		✓		02	✓			✓	PAES
[Tiware et al., 2010b, 2012]		✓		02	✓			✓	PAES, PESA2, SPEA2
[Ahmadikatoouli and Aboutalebi, 2011]	✓	✓	✓	02		✓	✓		
[Ognjanovic et al., 2012]	✓		✓	02	✓		✓		
[Salomie et al.,		✓		02		✓			Lucioles

¹⁵ Processus métier

¹⁶ Algorithme génétique

2012]									
[Pop et al., 2012]		✓		02		✓			ACO, BCO
[Wibig, 2013]	✓	✓	✓	02	-			✓	
[Farsani et al., 2013]		✓		02	✓				NSGAI modifié
[Bae et al., 2014]		✓		02	✓		✓		
[Molka et al., 2015]	✓	✓	✓	-		✓	✓		
[Vergidis et al., 2015]		✓		02	✓			✓	
[Djedović et al. 2016]	✓			02	✓		✓		
[Afflerbach et al., 2016]	✓	✓		-		✓	✓		
[Georgoulakos et al., 2017]		✓		02	✓			✓	

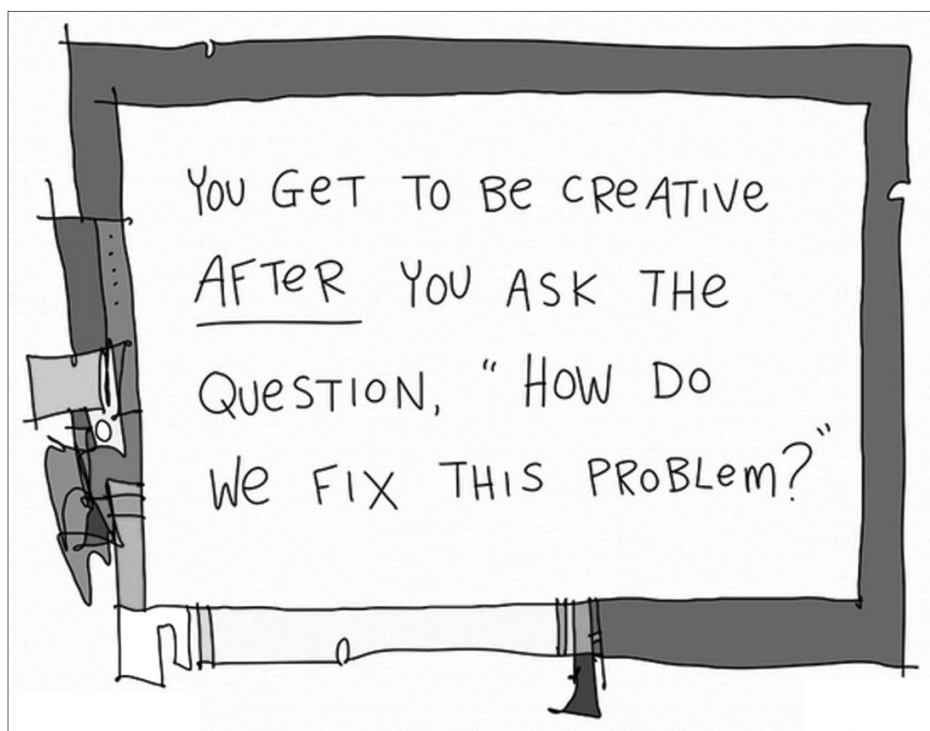
Tableau 3.1 Synthèse des travaux sur l'optimisation multicritères des processus métiers

Conclusion

Ce chapitre a été consacré à dresser un tableau descriptif et comparatif des différents travaux portant sur l'optimisation multicritères des processus métiers, avec les métaheuristiques. Il en ressort que :

- La représentation mathématique des processus métiers fait légion.
- L'optimisation multicritères des processus métiers est combinatoire.
- Les algorithmes génétiques sont généralement utilisés, et le NSGAI en particulier.
- La réduction du coût et de la durée d'un processus métier (fabrication et/ou exécution) sont souvent les critères d'optimisation à viser.
- La fonction de Fitness n'a pas l'importance qu'elle mérite.
- Un travail important reste à faire sur la modification des opérateurs génétiques à utiliser avec un algorithme évolutionnaire.

Même si les remarques présentées semblent ouvrir d'innombrables voies de recherche, quelques unes (de remarques) nous serviront de boussole, dans la suite de nos recherches.



Chapitre 4

Représentation des processus métiers adaptée à l'optimisation multicritères

Introduction	68
4.1. Spécification des processus métiers.....	68
4.1.1. Définition adoptée pour une conception de processus métier.....	68
4.1.1.1. Eléments d'un processus métier.....	69
4.2. Représentation proposée pour notre approche.....	70
4.2.1. But et objectifs de l'approche proposée	70
4.2.2. Représentation visuelle d'une conception de processus métier.....	71
4.2.3. Représentation quantitative d'une conception de processus métier	72
4.2.4. Matrice tâches/ressources	74
4.2.5. Librairie des tâches.....	76
2.4.5. Perspective visuelle contre perspective quantitative.....	77
4.3. Algorithme de vérification inverse des conceptions de processus métiers	78
4.3.1. Objectif de l'algorithme	79
4.3.2. Exigences et résultats de ReProVA	81
4.3.3. Principales étapes de ReProVA	82
4.3.4. Algorithme pour "Elaborer un niveau fils"	86
4.3.5. Algorithme pour "Mettre à jour le graphe et TRM"	87
Conclusion	89

Introduction

Ce chapitre présente la représentation des conceptions des processus métiers adoptée à l'optimisation évolutionnaire multicritères. Avant que la représentation soit détaillée, le concept de "conception de processus métier" est spécifié dans le contexte de cette recherche. La spécification de ce qu'est qu'une conception est considérée comme nécessaire en raison de la diversité des définitions et des approches identifiées dans la littérature. La représentation proposée est ensuite présentée en détail. Vergidis et Tiwari (2008) explique que le but et le choix d'une représentation, est de rendre les conceptions de processus métiers adaptées à l'optimisation évolutionnaire multicritères. Dans ce chapitre, les différents aspects de la représentation proposée tels que les paramètres mathématiques sont élaborés et discutés. La représentation proposée comporte également un algorithme efficace (*ReProVA*), pour assurer la correction de la conception de chaque instance d'un processus métier. Cet algorithme est également détaillé.

4.1. Spécification des processus métiers

Cette section spécifie la notion de "conception d'un processus métier" dans le contexte de notre recherche. Nous nous proposons de suivre la spécification proposée par Vergidis et al. (2006) et de Mahammed et Benslimane (2016). Le but de cette spécification est de fournir une définition des processus métiers, d'identifier leurs caractéristiques, et de préciser les éléments importants du processus métier, pour note recherche.

4.1.1. Définition adoptée pour une conception de processus métier

La définition adoptée pour une conception de processus métier, pour notre recherche, est la suivante : Un conception de processus métier est perçue comme un ensemble de tâches qui, une fois correctement connecté et ordonné effectue une opération commerciale, i.e. toute opération liée au service qui produit de la valeur pour l'organisation (Vergidis et al., 2006).

Il serait de bonne guerre de définir ce qu'est que nous appelons la conception ou l'instance de processus métier : Une conception/instance de processus métier est la représentation d'un processus métier décrivant les tâches participantes qui déterminent le flux du processus. Le but de la conception/instance est de capturer, visualiser et communiquer un processus métier.

4.1.1.1. *Eléments d'un processus métier*

Dans notre recherche, les éléments qui sont impliqués dans un processus métier et - par conséquent - représentés dans la conception du processus métier sont :

- Les tâches.
- Les ressources d'une tâche / conception de processus métier.
- Les attributs d'une tâche / conception de processus métier.

Essentiellement, les principaux éléments impliqués sont les tâches et les ressources des conceptions de processus métiers. Les attributs des tâches et des conceptions de processus métiers sont également pris en compte, afin de fournir la capacité d'évaluer chaque conception. Chacun de ces éléments est discuté plus en détail.

Tâches de participation : Ce sont les éléments principaux du processus métier.

Chaque tâche représente une fonctionnalité spécifique. Les tâches sont considérées comme des fonctionnalités "boîte noire" (Vergidis, 2008). Elles sont similaires dans la nature et les caractéristiques, mais différents en termes d'opération de base qu'elles exécutent.

Ressources d'une tâche / conception de processus métier : Elles sont considérées comme les produits d'entrée et de sortie des tâches participantes et des conceptions de processus métiers. Les ressources sont les éléments d'une conception qui circulent, et sont transformés à travers cette conception, pour produire le résultat final. Une distinction pour les ressources d'une conception de processus métier qui sont considérées comme des exigences du processus métier. Dans cette situation, elles sont classées comme l'entrée requise du processus et les ressources de sortie attendues par le processus métier.

Attributs d'une tâche / conception de processus métier : Ils sont considérés comme les caractéristiques mesurables (quantitatives) des tâches participantes et des conceptions de processus métiers. Nous estimons que les attributs sont communs à toutes les tâches participantes, e.g. le coût de la tâche. Vergidis et al. (2006) proposent de calculer les attributs correspondants à une conception de processus métier en utilisant une fonction d'agrégation appropriée. Les attributs sont utilisés pour évaluer une conception de processus métier, et ainsi le comparer avec d'autres et l'optimiser.

4.2. Représentation proposée pour notre approche

Cette section détaille la représentation des conceptions de processus métiers qui sera utilisée dans notre recherche. La section commence par énoncer le but et les objectifs que l'approche doit satisfaire. La représentation proposée comprend deux perspectives : **Visuelle** et **quantitative**.

- La représentation visuelle communique la conception du processus métier sous forme de diagramme.
- La représentation quantitative capte la conception de manière à pouvoir être analysée.

4.2.1. But et objectifs de l'approche proposée

Le but et les objectifs de la démarche de représentation proposée sont les suivants : Le but de la représentation proposée est de capturer, visualiser et exprimer une conception de processus métier de manière quantitative qui permet à l'optimisation évolutionnaire multicritères utilisée de générer une série de conceptions alternatives optimisées.

L'approche proposée tente d'atteindre les objectifs suivant :

- Fournir une communication visuelle des conceptions de processus métiers.
- Formuler les éléments d'une conception en utilisant des paramètres mathématiques.

- Fournir des moyens quantitatifs pour évaluer les conceptions.
- Constituer une base pour de nouvelles solutions à partir des solutions existantes.

4.2.2. Représentation visuelle d'une conception de processus métier

Vergidis et al., (2006) propose qu'une représentation visuelle d'une conception de processus métier doit reposer sur les principes d'un organigramme simple (figure 4.1). La raison en est que :

- Un organigramme est largement reconnu et familier entre les chercheurs et les praticiens de l'industrie liés aux processus métiers (Vergidis and Tiwari, 2008).
- Un organigramme peut accueillir visuellement tous les éléments de processus identifiés (tâches et ressources), à l'exception des attributs des tâches.
- Un organigramme peut communiquer le flux de base d'une conception de processus métier.

La notation proposée pour un organigramme décrivant une conception de processus métier est :

- Deux boîtes arrondies marquées comme "DEBUT" et "FIN" apparaissent dans chaque conception et indiquent le début et la fin du processus métier.
- Les tâches participantes sont esquissées sous forme de cases/rectangles.
- Les ressources sont les flèches de connexion qui relient les tâches.

La figure 4.1 illustre un exemple d'organigramme pour une conception de processus métier basée sur la notation proposée. La démonstration visuelle d'une conception est une partie essentielle de la représentation adoptée. Elle fournit les moyens de communiquer les résultats du Framework proposé (conceptions optimisées de processus métiers) en exposant leurs principales tâches et la façon dont elles sont interconnectées. Le défi suivant est la conversion de la représentation visuelle en représentation quantitative équivalente à l'aide de constructions mathématiques.

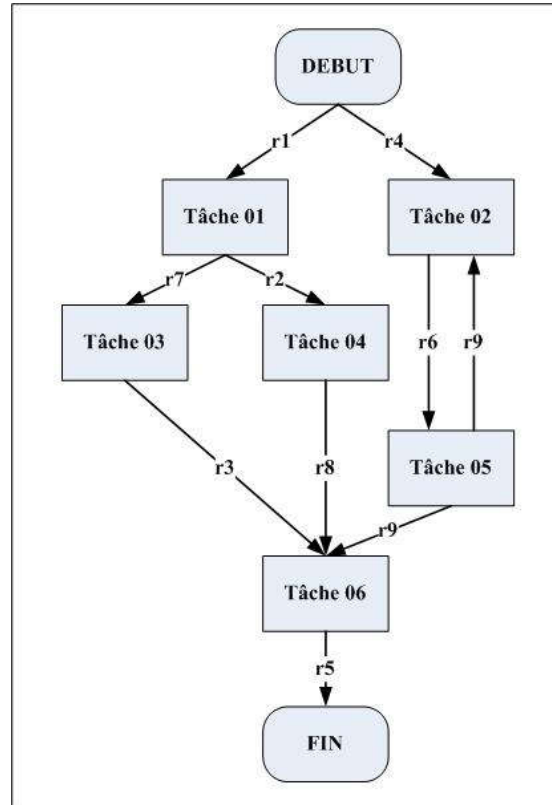


Figure 4.1 Exemple de représentation visuelle d'une conception de processus métier

4.2.3. Représentation quantitative d'une conception de processus métier

Cette sous-section présente les paramètres mathématiques des principaux éléments d'une conception de processus métier tels qu'identifiés. L'expression de ces concepts en utilisant la notation mathématique aide à introduire des constructions plus complexes qui peuvent représenter une conception de processus métier de manière quantitative. Le tableau 4.1 montre le codage des principaux paramètres d'une conception de processus métier.

L'ensemble des n_d tâches qui appartiennent à une conception de processus métier particulière est $N_d = \{t_1, t_2, \dots, t_{n_d}\}$. L'ensemble des r_d ressources dans la conception $R_d = \{r_1, r_2, \dots, r_{n_d}\}$ englobe les sous-ensembles R_{in} et R_{out} qui stockent ; respectivement, les ressources d'entrée et les ressources de sortie du processus métier. La conception d'un processus métier utilise toutes les ressources de R_{in} et produit toutes les ressources en R_{out} . Chaque tâche i dans la conception a

des ressources d'entrée stockées dans $I_i \subseteq R_d$ et toutes les ressources de sortie stockées dans $O_i \subseteq R_d$. Chaque tâche i possède p valeurs d'attributs mémorisées dans l'ensemble TA_i et les p attributs du processus métier correspondants sont stockés dans l'ensemble PA .

Paramètre	Description	Paramètre	Description
n	Nombre de tâches dans la librairie	N	Ensemble de n tâches
n_d	Nombre de tâches dans une conception	N_d	Ensemble des n_d tâches
r_d	Nombre de ressources dans une conception	R_d	Ensemble des r_d ressources
t_{in}	Numéro de ressources d'entrée d'une tâche	I_i	Ensemble des t_{in} ressources d'une tâche i
t_{out}	Numéro de ressources de sortie d'une tâche	O_i	Ensemble des t_{out} ressources d'une tâche i
r_{in}	Numéro d'une ressource en entrée	R_{in}	Ensemble des r_{in} ressources
r_{out}	Numéro d'une ressource en sortie	R_{out}	Ensemble des r_{out} ressources
p	Nombre d'attributs d'une tâche/conception	TA_i	Ensemble d'attributs pour une tâche i
		PA	Ensemble d'attributs pour une conception de processus métier

Tableau 4.1 Principaux paramètres mathématiques d'une conception de processus métier

La figure 4.2 porte sur le détail de la "Tâche 01" de la figure 4.1 et montre comment les paramètres du tableau 4.1 se rapportent à son représentation visuelle. La *Tâche 01* possède une ressource d'entrée $\{r1\}$ et deux ressources de sortie $\{r7, r2\}$. Les seuls éléments liés à la tâche qui ne sont pas affichés sont les valeurs d'attributs de la tâche.

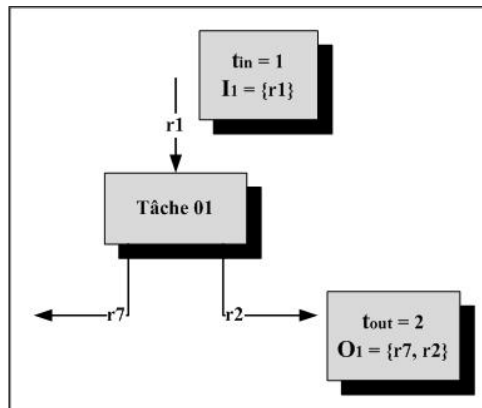


Figure 4.2 Représentation proposée pour une tâche

La figure 4.3 montre la conception du processus métier de la figure 4.1 relative aux paramètres du tableau 4.1. Les ressources qui résultent du nœud "DEBUT" sont les exigences d'entrée de processus et les ressources qui finissent dans le nœud "FIN" sont les exigences de sortie du processus métier. La conception du processus comporte 06 tâches participantes et 09 ressources différentes qui transitent par les tâches. Les valeurs d'attribut de la conception du processus métier qui sont calculées à partir des valeurs d'attributs des tâches participantes ne sont pas affichées dans la représentation visuelle de la conception du processus métier.

4.2.4. Matrice tâches/ressources

Après avoir défini les paramètres des principaux éléments d'une conception de processus métier, des constructions plus complexes peuvent être élaborées. La première construction est une matrice qui vise à capturer la séquence des tâches et les modèles formulés dans une conception de processus métier. Pour ce faire, la matrice mappe les ressources d'entrée et de sortie des tâches dans la conception. Cette matrice est appelée la matrice tâches/ressources ou *Tasks Resources Matrix* (*TRM*). C'est une matrice bidimensionnelle de taille $n_d \times r_d$. Les lignes de *TRM* contiennent les tâches n_d et les colonnes comportent les ressources r_d , dans une conception de processus métier. Chaque cellule dans *TRM* présente la relation entre la tâche et la ressource.

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

- Si $r_j \in I_i$ 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 $r_j \in O_i$ 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'est ni dans les ressources d'entrée ni dans les ressources de sortie de la tâche, alors sa relation est marquée par "0").

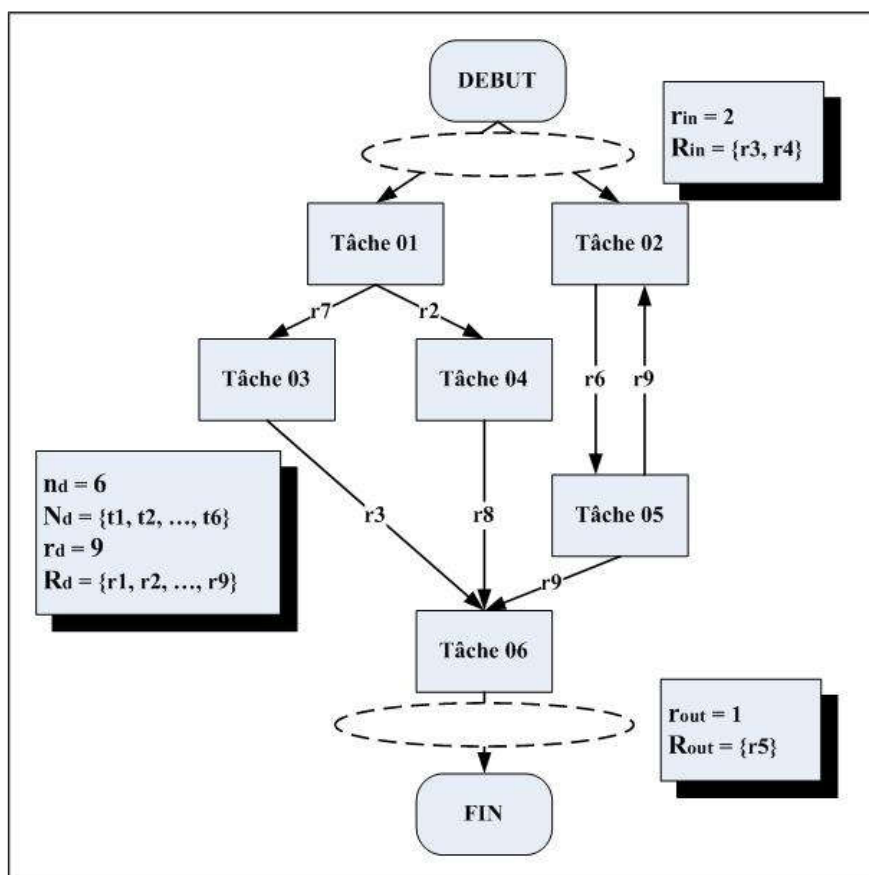


Figure 4.3 Représentation détaillée proposée pour une conception de processus métier

La figure 4.4 montre la cartographie de *TRM* pour la "Tâche 01" et les neuf (09) ressources dans la conception du processus métier. Le tableau 4.2 montre le *TRM* pour toutes les tâches dans la conception du processus basé sur la cartographie proposée de la relation entre les tâches et les ressources.

Représentation visuelle	TRM correspondante
<pre> graph TD r1[r1] --> T01[Tâche 01] T01 --> r7[r7] T01 --> r2[r2] </pre>	$TRM_{11} = 1$ $TRM_{12} = 2$ $TRM_{13} = 0$ $TRM_{14} = 0$ $TRM_{15} = 0$ $TRM_{16} = 0$ $TRM_{17} = 2$ $TRM_{18} = 0$ $TRM_{19} = 0$

Figure 4.4 Exemple de TRM pour représenter la Tâche 01

Ressource Tâche	r₁	r₂	r₃	r₄	r₅	r₆	r₇	r₈	r₉
Tâche 01	1	2	0	0	0	0	2	0	0
Tâche 02	0	0	0	1	0	2	0	0	1
Tâche 03	0	0	2	0	0	0	1	0	0
Tâche 04	0	1	0	0	0	0	0	2	0
Tâche 05	0	0	0	0	0	1	0	0	2
Tâche 06	0	0	1	0	2	0	0	1	1

Tableau 4.2 Exemple de matrice tâches/ressources

TRM peut capturer la suite des tâches d'une conception de processus métier, et permet de fournir une base pour reproduire cette conception sur la base des exigences du processus métier.

4.2.5 Librairie des tâches

Le but de cette recherche est la génération de conceptions alternatives optimisées de processus métiers. Cela ne peut se produire sans avoir une gamme de tâches disponibles qui, dans différentes combinaisons peuvent façonner une variété de conceptions de processus métiers équivalentes. La deuxième construction est également une matrice qui vise à présenter l'ensemble de tâches à partir duquel les conceptions de processus métiers sont réalisées. Le tableau 4.3 présente une librairie des tâches sous la forme de sept (07) colonnes, le nombre de lignes dépend du nombre de tâches existantes (le paramètre n de la représentation quantitative d'une conception de processus métier). Le nombre de colonnes dépend du nombre d'attributs des tâches.

Le tableau 4.3 fournit un exemple d'une librairie des tâches utilisables pour les conceptions de processus métiers (figure 4.1) en supposant trois (03) attributs (Attribut 01, Attribut 02 et Attribut 03). Les attributs d'une conception de processus métier sont calculés sur la base de l'équation d'agrégation 4.1. La librairie des tâches ne stocke pas les attributs d'une conception mais, uniquement les valeurs d'attributs des tâches.

N° tâche	Nom tâche	Ressources entrée	Ressources sortie	Attribut 01	Attribut 02	Attribut 03
01	Tâche 01	{r1}	{r2, r7}	208	113	354
02	Tâche 02	{r4, r9}	{r6}	219	109	322
03	Tâche 03	{r7}	{r3}	229	115	367
04	Tâche 04	{r2, r8}	{r4}	202	109	324
05	Tâche 05	{r6, r9}	{r4}	203	108	370
06	Tâche 06	{r3, r8, r9}	{r6}	211	101	342

Une conception de processus métier	1272	655	2079
---	-------------	------------	-------------

Tableau 4.3 Exemple d'une librairie des tâches avec trois attributs

La librairie des tâches contribue au calcul des valeurs d'attributs pour une conception de processus spécifique, en supposant qu'un attribut d'une conception est calculé en fonction des valeurs d'attributs des tâches participantes.

En utilisant la librairie des tâches, un attribut de processus $j(PA_j)$ peut être calculé comme un agrégat des attributs correspondants des tâches, pour toutes les tâches n_d dans la conception du processus métier :

$$PA_j = \sum_{i=1}^{n_d} \text{Attribut}(\alpha)_{ij} \quad (4.1)$$

$\text{Attribut}(\alpha)$ représente l'attribut numéro α de la librairie des tâches.

2.4.5. Perspective visuelle contre perspective quantitative

Les sous-sections précédentes ont présenté les deux perspectives de la représentation : La perspective visuelle et la perspective quantitative. La perspective visuelle décrit une conception du processus métier au moyen d'un organigramme. Tandis que la perspective quantitative mappe la conception de processus métier pour la rendre appropriée pour la méthode d'optimisation multicritères de notre recherche. Une fois que la perspective visuelle est à portée de main, sa transformation vers l'aspect quantitatif de la représentation est directe.

Les attributs des tâches sont enregistrés dans la librairie des tâches, alors que les relations entre les tâches et les ressources sont cartographiées dans la matrice tâches/ressources de chaque conception de processus métier.

Toutefois, dans le cas où une conception de processus métier est exprimée en fonction de la perspective quantitative, sa transformation en diagramme visuel pose problème. Ceci parce que la perspective quantitative ne garantit pas qu'une conception de processus métier est faisable et donc, un processus algorithmique est nécessaire. Vergidis et Tiwari (2008) propose qu'un algorithme de construction d'un diagramme de processus métier puisse être basé sur (i) les informations stockées dans TRM et (ii) les exigences d'entrée et de sortie du processus métier, afin que le point de départ et le point final du processus soient connus.

Ce qui n'est pas suffisant, il se peut que toutes les tâches de TRM ne soient pas reliées entre elles, jusqu'à arriver à une conception terminée. Par conséquent, pour que la représentation soit complète, il est - fortement - nécessaire qu'un algorithme puisse composer la perspective visuelle d'une conception de processus métier, et assurer sa faisabilité sur la base de la perspective quantitative. Cet algorithme est appelé "Algorithme de vérification inverse de conceptions de processus métiers", et est présenté dans la section suivante.

4.3. Algorithme de vérification inverse des conceptions de processus métiers

Cette section présente l'algorithme de vérification utilisé avec la représentation de l'approche proposée, l'Algorithme de vérification inverse des conceptions de processus métiers ou *Reverse Process Verification Algorithm (ReProVA)*. L'algorithme décrit est une partie essentielle de la représentation proposée, puisqu'elle fournit le pont entre la perspective visuelle et la perspective quantitative. En outre, l'algorithme en vérifiant une conception assure que la conception capturée par les deux perspectives de représentation est faisable.

4.3.1. Objectif de l'algorithme

L'un des principaux objectifs de la représentation proposée est de fournir la capacité de générer des conceptions alternatives. Sur la base de la représentation, une conception de processus métier peut être sous la forme de *TRM* et vérifié si elle correspond à une conception de processus métier faisable. Le but de l'algorithme introduit dans cette section est double :

- Produire la représentation visuelle d'une conception de processus métier en prenant en compte la représentation quantitative.
- Vérifier si la conception de processus métier capturée correspond à une conception réalisable.

L'algorithme proposé porte l'appellation de "Algorithme de vérification inverse des conceptions de processus métiers" (*ReProVA*). Cet algorithme tente de composer une conception de processus métier sous forme de diagramme, tout en prenant en compte sa représentation quantitative, et de vérifier si le résultat final correspond à celui d'une conception réalisable (faisable). Le concept de faisabilité concernant une conception de processus métier est défini comme suit :

- $TRM_{ij} = 1, \forall j: r_j \in R_{in}, i: t_i \in N_d$: Toutes les ressources d'entrée du processus sont utilisées par une ou plusieurs tâches qui participent à la conception de processus métier.
- $TRM_{ij} = 2, \forall j: r_j \in R_{out}, i: t_i \in N_d$: Toutes les ressources de sortie du processus sont produites par une ou plusieurs tâches qui participent à la conception de processus métier.
- Chaque tâche dans la conception est connectée soit avec les entrées du processus, soit les sorties du processus ou une autre tâche dans la conception.

La figure 4.5 montre comment une conception de processus métier peut être élaborée. Elle est dite réalisable, si elle commence avec les ressources dans R_{in} et en connectant correctement les tâches dans *TRM* produit les ressources R_{out} demandées. Le défi dans la production d'une conception réalisable est d'obtenir

une matrice TRM avec les tâches de la librairie qui puisse satisfaire les contraintes de faisabilité. Cependant, les contraintes de faisabilité donnent un nombre significatif de cas infaisables :

- Une ou plusieurs ressources d'entrée du processus ne peuvent pas être utilisées à partir des tâches dans TRM .
- Une ou plusieurs ressources de sortie du processus ne peuvent pas être produites à partir des tâches dans TRM .
- Il y a un lien brisé dans la conception : Il n'y a aucune tâche dans TRM qui puisse être attachée au diagramme de la conception basé sur ses ressources d'entrée et de sortie.

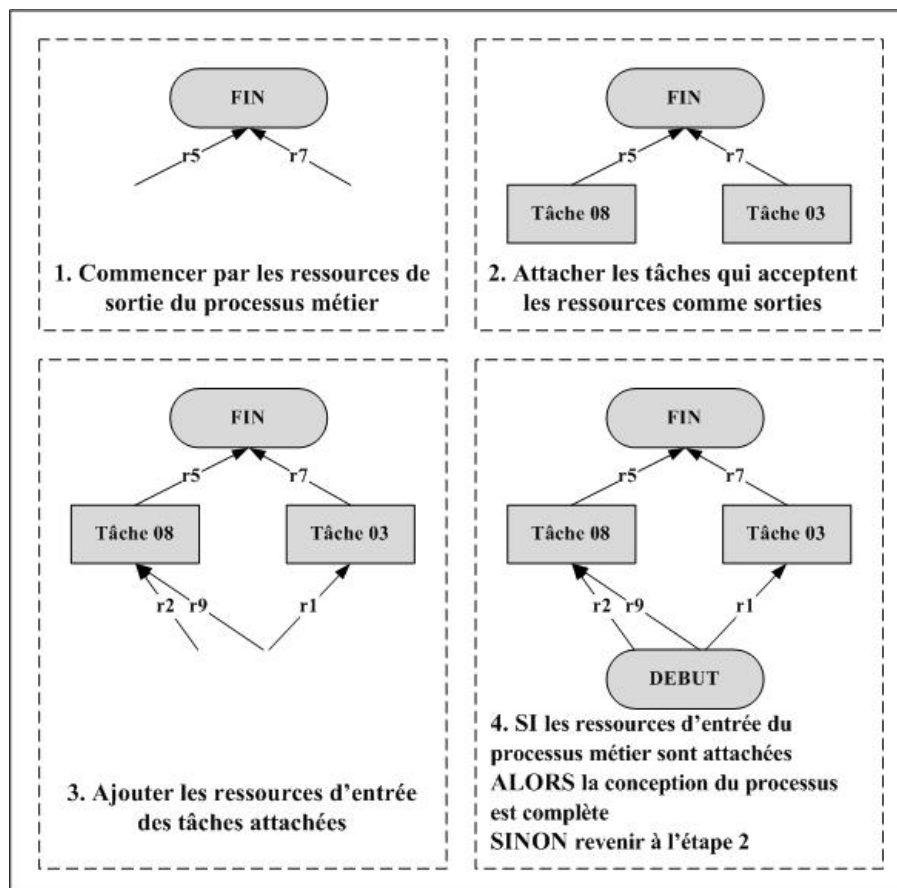


Figure 4.5 Exemple d'élaboration du graphe d'une conception de processus métier

Ces situations d'infaisabilités impliquent que, dans le cas où R_{out} n'est pas produit, une conception de processus est considérée comme impossible. Même dans le cas où R_{in} et R_{out} sont utilisés et produits, il y a une faible probabilité d'assurer que toutes les tâches intermédiaires soient interconnectées par leurs ressources d'entrée et de sortie.

4.3.2. Exigences et résultats de ReProVA

ReProVA a comme mission de s'occuper des cas d'infaisabilité d'une conception et de construire un diagramme de conception de processus réalisable. Les exigences (ou entrées) de l'algorithme sont :

- Les ressources d'entrée et de sortie du processus métier (R_{in} et R_{out}).
- Les tâches participantes dans la conception (TRM).
- La librairie des tâches (N).

Les exigences du processus métier (sous la forme des ressources d'entrée et de sortie du processus) sont requises comme conditions d'arrêt. *ReProVA* ajoute des tâches à une conception de processus jusqu'à ce que les entrées du processus métier soient utilisées et que les sorties de processus métier soient produites. La deuxième exigence est TRM qui contient les tâches qui forment la conception. *ReProVA* ajoute des tâches à la conception à partir de cette matrice et vérifie si elles correspondent à une conception réalisable. Enfin, la librairie des tâches est essentielle pour modifier ou réparer une conception. Par conséquent, les tâches dans la librairie peuvent aider à réparer une conception et ainsi la rendre possible avec des changements mineurs. De plus, la librairie des tâches peut être utilisée pour améliorer une conception de processus réalisable en remplaçant les tâches par d'autres avec de meilleures valeurs d'attributs.

Les résultats (ou sorties) de l'algorithme sont :

- La conception de processus métier (graphe de la conception réalisable).
- L'ensemble des tâches mis à jour dans la conception (N_d).

Le résultat principal de *ReProVA* est la conception du processus métier. Une conception est composée et représentée sous la forme d'un graphe orienté. Les nœuds du graphe sont les tâches de la conception du processus métier et les arêtes représentent les ressources utilisées pour la connexion. Le graphique est orienté i.e. les arêtes sont dirigées pour montrer le flux des ressources entre les tâches.

Le deuxième résultat de *ReProVA* est la mise à jour des tâches dans la conception (N_d). Sur la base de la représentation proposée, *ReProVA* traduit *TRM* en un diagramme d'une conception de processus métier. Cependant, il est nécessaire de mettre à jour *TRM* elle-même, pendant l'exécution de *ReProVA* pour les deux raisons suivantes :

- Dans *TRM*, l'élimination de toutes les tâches qui n'ont pas été ajoutées au diagramme d'une conception en cours de sa composition, i.e. ces tâches ne contribuent donc pas à remplir les exigences du processus.
- Le remplacement de toute tâche de *TRM* par des tâches à partir de la librairie qui assurent la faisabilité de la conception traitée.

Sur la base des modifications apportées à *TRM* au cours de l'exécution de *ReProVA*, le deuxième résultat du *ReProVA* est le jeu de tâches mis à jour (N_d) qui participe à la conception du processus métier basée sur l'exécution de l'algorithme.

4.3.3. Principales étapes de ReProVA

La figure 4.6 présente les principales étapes de l'algorithme de vérification inverse des conceptions de processus métiers. *ReProVA* construit un graphe d'une conception de processus et le parcourt pour s'assurer qu'il répond aux exigences du processus. Dans le graphe, chaque tâche est représentée comme un nœud, avec deux nœuds artificiels, le nœud "DEBUT" avec les ressources d'entrée du processus métier et le nœud "FIN" avec les ressources de sortie du processus métier. Ces nœuds facilitent la connexion des ressources d'entrée et de sortie du processus métier avec les tâches participantes, afin de produire

une conception de processus qui répond à ses exigences. Le graphe est élaboré à l'aide de l'algorithme de parcours en largeur, en utilisant les notions de niveaux "parent" et "fils". Le niveau "parent" se compose des nœuds déjà insérés dans le graphe. Le niveau "fils" est celui où les nouvelles tâches sont ajoutées dans la conception en fonction des ressources d'entrée des tâches au niveau "parent". Une fois l'élaboration de toutes les tâches au niveau "fils" est terminée, elle est définie comme le nouveau niveau "parent", pour l'élaboration du graphe.

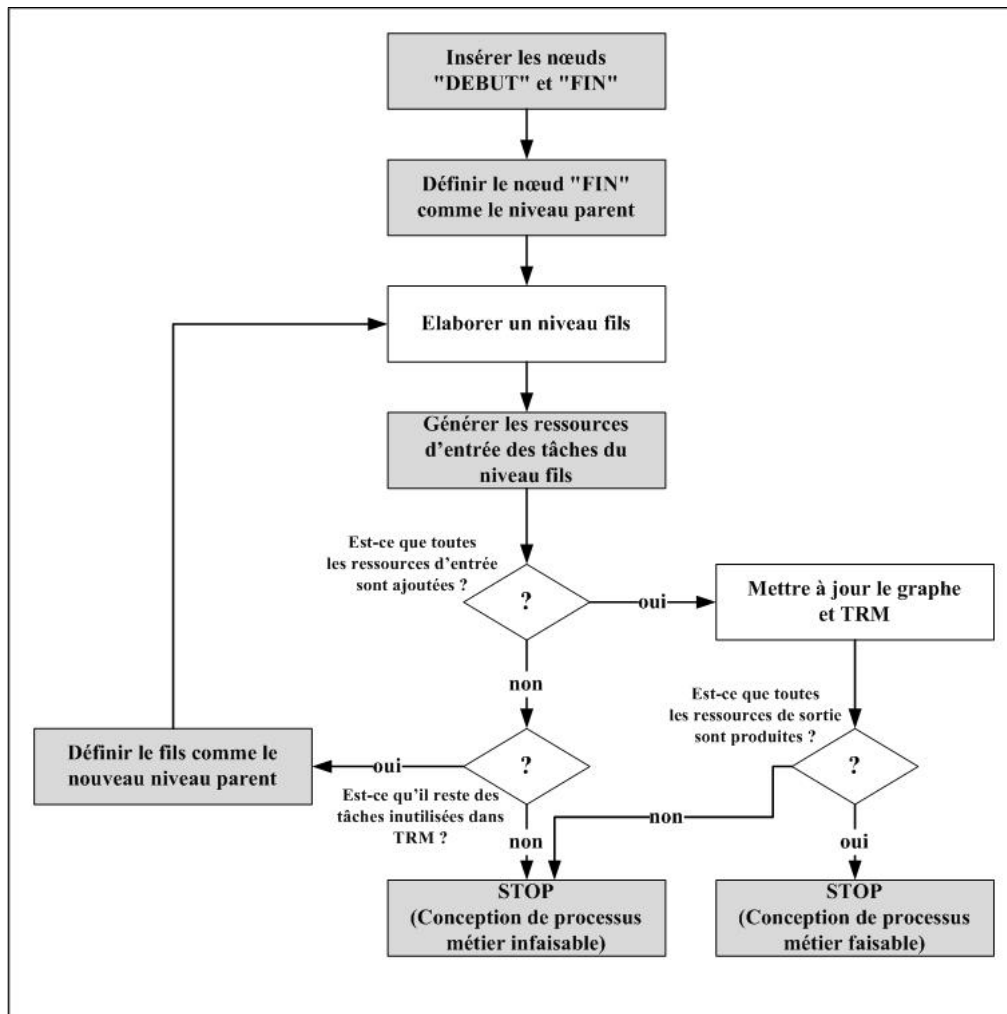


Figure 4.6 Principales étapes de ReProVA

ReProVA commence par insérer les nœuds artificiels "DEBUT" et "FIN" sur un graphe vide. Le nœud "FIN" est initialement marqué comme un "niveau parent".

Ensuite, l'algorithme visite tous les nœuds au niveau parent un par un afin d'élaborer le niveau fils. Une fois l'élaboration du niveau fils terminée, les ressources d'entrée des tâches récemment jointes ainsi que les ressources d'entrée non liées des tâches précédentes sont vérifiées pour savoir si elles contiennent les ressources d'entrée du processus. Dans le cas où toutes les ressources d'entrée ne sont pas produites et qu'il y a des tâches inutilisées dans *TRM*, les tâches en niveau "fils" deviennent le nouveau niveau "parent" et le processus d'élaboration est répété..

Dans le cas où - à un certain stade du processus d'élaboration - toutes les ressources d'entrée du processus métier sont produites, *TRM* et le graphique sont mis à jour. Le processus de mise à jour comprend deux parties : (i) L'élimination de toutes les tâches qui n'ont pas été insérées dans une conception de processus de *TRM* et (ii) l'élimination des nœuds (tâches) qui ne contribuent pas à la production des ressources d'entrée du processus métier. Après la mise à jour, *ReProVA* vérifie si toutes les ressources de sortie du processus sont produites. Certaines des tâches qui utilisaient les sorties du processus n'avaient peut-être pas contribué aux entrées du processus et ont donc été supprimées de la conception. Dans le cas où toutes les sorties du processus sont produites, la conception est marquée comme faisable.

La figure 4.7 présente les principales étapes de la composition d'une conception de processus métier basée sur les principales étapes de *ReProVA*. Initialement, les nœuds "DEBUT" et "FIN" sont insérés (figure 4.7.a). Dans la phase d'élaboration du niveau fils, les tâches sont insérées dans le graphe et attachées aux ressources d'entrée des nœuds de niveau parent (figure 4.7.b). Il peut y avoir des ressources pour lesquelles il n'y a pas de tâche correspondante. Lorsque toutes les ressources de sortie du processus métier sont produites, *ReProVA* supprime du graphe tous les nœuds qui n'ont pas contribué au résultat du processus (figure 4.7.c).

Le résultat de *ReProVA* est, dans le meilleur des cas, une conception de processus métier réalisable (faisable) dans laquelle toutes les tâches de la conception sont

liées entre elles en utilisant les ressources de sortie du processus et en produisant les ressources d'entrée du processus. *ReProVA* ne supprime pas les solutions infaisables (ou irréalisables) mais les répare (en utilisant la librairie des tâches). Les auteurs jugent que les solutions non réalisables peuvent mener à des solutions réalisables à mesure qu'elles évoluent au cours des générations au cours de l'optimisation.

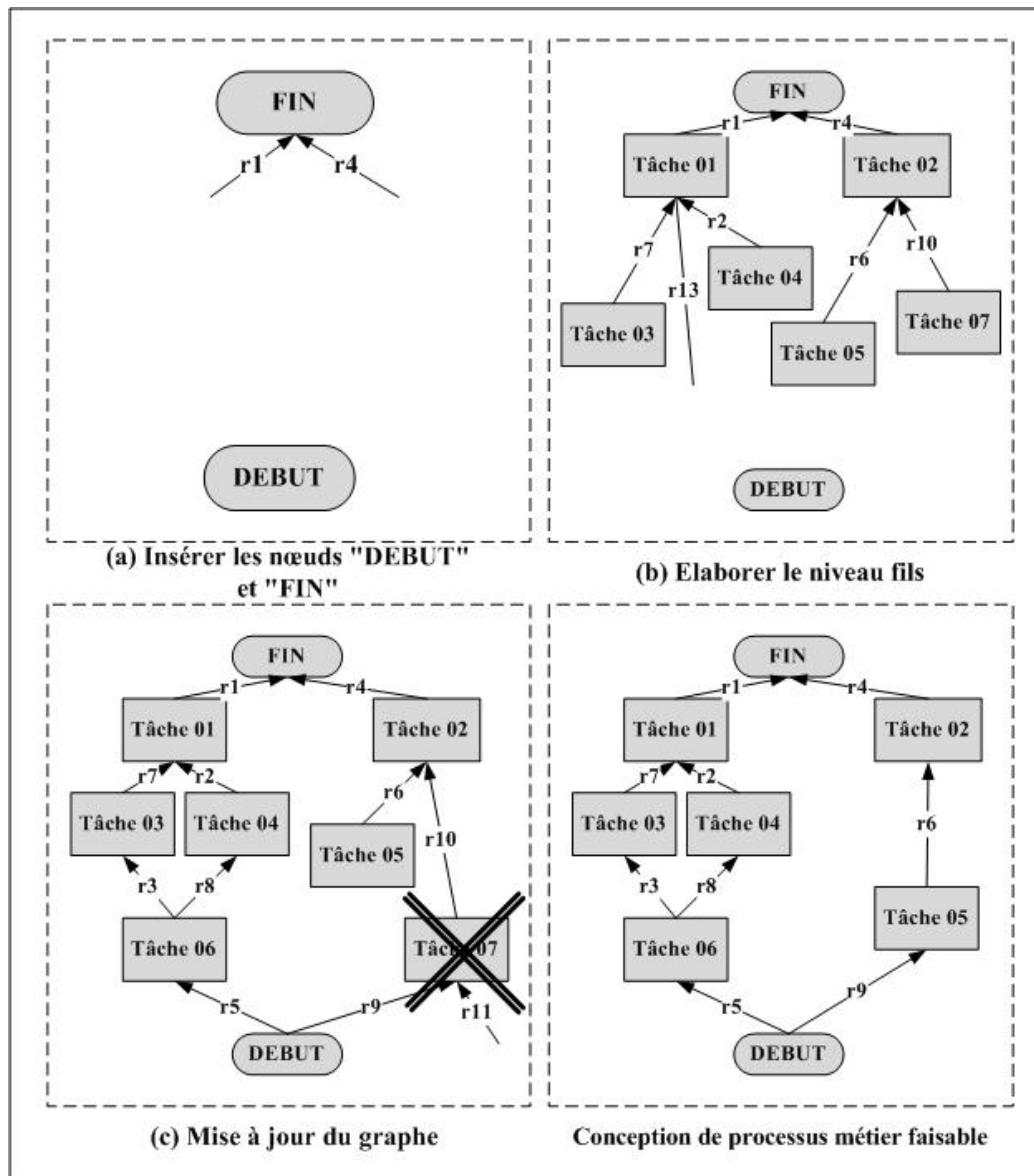


Figure 4.7 Principales étapes pour la composition et la vérification d'une conception de processus métier

4.3.4. Algorithme pour "Elaborer un niveau fils"

La figure 4.8 illustre l'algorithme pour l'étape "Elaborer un niveau fils" de *ReProVA*. Le but de cet algorithme est d'élaborer le graphe d'une conception de processus métier en ajoutant des tâches au niveau "fils" qui peuvent être rattachées à d'autres tâches à partir du niveau "parent".

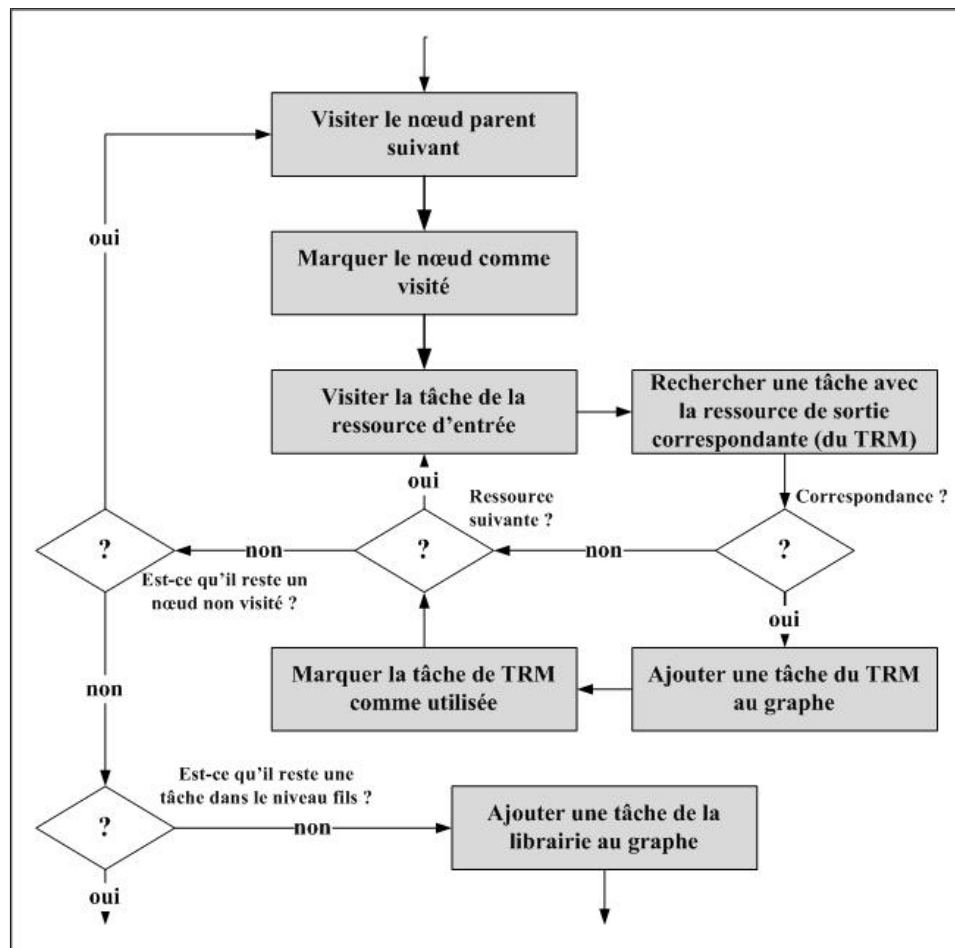


Figure 4.8 Algorithme pour "Elaborer un niveau fils"

Cet algorithme marque également les tâches de *TRM* qui sont ajoutées au graphe de la conception de processus comme "utilisées". Ceci est utile plus tard dans l'exécution de *ReProVA*, pour déterminer s'il faut poursuivre l'élaboration du graphe. Dans le cas où toutes les tâches de *TRM* sont marquées comme "utilisées"

et les ressources d'entrée du processus métier ne sont pas produites, alors la conception est infaisable.

Les étapes de base de cet algorithme sont décrites comme suit : Pour chaque nœud du niveau "parent", toutes les ressources d'entrée sont visitées. Pour chaque ressource d'entrée, l'algorithme vérifie dans *TRM* s'il existe une tâche avec au moins une ressource de sortie correspondante. Si une tâche avec une ressource commune est trouvée, elle est insérée dans le graphe, liée à la tâche parent et ajoutée au jeu de niveau "fils".

Dans le cas où il n'y a pas de tâche correspondante, l'algorithme passe à la ressource d'entrée suivante du niveau parent. Lorsque l'algorithme atteint la dernière ressource d'entrée de la dernière tâche de niveau "parent", il vérifie s'il existe des tâches attachées au niveau "fils". Dans le cas où aucune tâche n'est insérée dans le niveau "fils", l'algorithme associe une tâche de correspondance de la librairie des tâches, afin de poursuivre le processus d'élaboration du graphe. Chaque tâche ajoutée à la conception est liée non seulement à la tâche du niveau parent mais également à toute tâche avec laquelle elle possède une ressource correspondante.

4.3.5. Algorithme pour "Mettre à jour le graphe et *TRM*"

La figure 4.9 montre les étapes de l'opération "Mettre à jour le graphe et *TRM*" de *ReProVA*. Il s'agit d'une opération importante dans *ReProVA* qui ne se déroule que dans le cas où toutes les ressources d'entrée du processus métier sont produites pendant l'élaboration du graphe. Le but de cette étape est double : (i) Créer une *TRM* mise à jour et N_d qui contiennent les tâches réelles dans le graphe qui commence à partir des ressources de sortie du processus et se termine avec des ressources d'entrée du processus métier, et (ii) à mettre à jour le graphe basé sur N_d mis à jour en supprimant les tâches qui ne contribuent pas aux entrées du processus métier. L'élaboration du graphe suit une approche descendante du nœud artificiel "FIN". L'élaboration du graphe suit la stratégie de parcours en largeur, avec le risque que tous les nœuds ajoutés ne contribuent pas nécessairement à

la production des entrées du processus. Par conséquent, le premier critère d'arrêt est la vérification des ressources d'entrée.

En partant du postulat que les ressources d'entrée du processus métier soient produites, cet algorithme parcourt le graphe à l'envers, suivant la stratégie de parcours en profondeur. À partir de chaque ressource d'entrée, l'algorithme localise le chemin le plus court entre la ressource d'entrée du processus et les ressources de sortie de la conception de processus métier. Les tâches qui sont visitées au cours de ce processus sont insérées dans *TRM*. Lorsque cette opération est répétée pour toutes les ressources d'entrée, les tâches du graphe de la conception du processus métier qui ne sont pas visitées sont éliminées, car elles ne relient pas les sorties du processus métier à ses entrées. Le résultat de cet algorithme est un *TRM* et N_d mis à jour, qui ne contiennent que les tâches nécessaires et un graphe de conception de processus métier mis à jour avec la suppression des nœuds redondants.

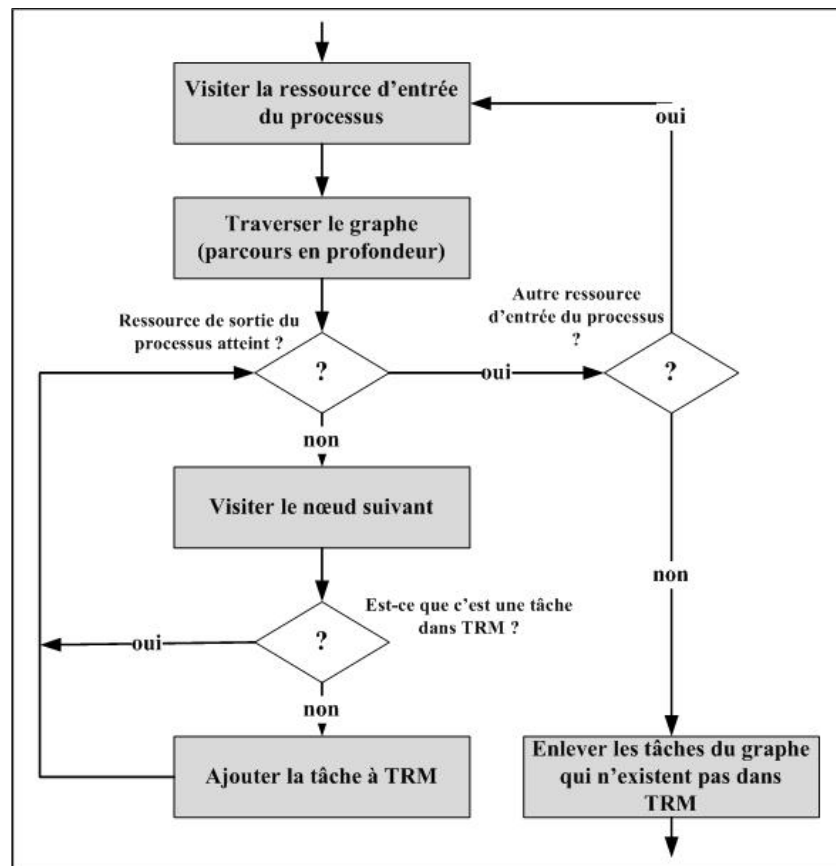


Figure 4.9 Etapes pour "Mettre à jour le graphe et TRM"

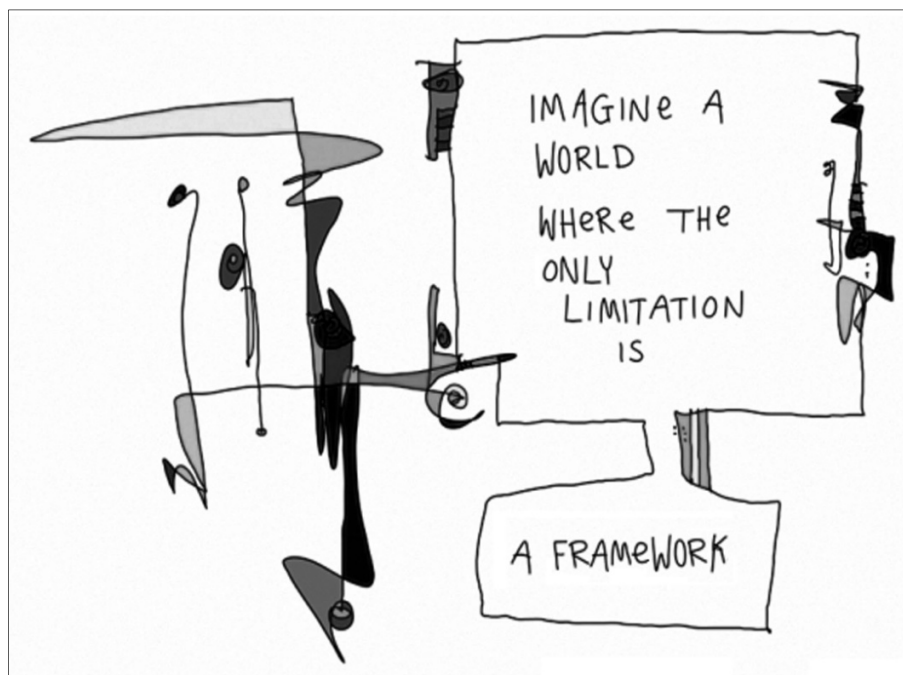
Conclusion

Ce chapitre présente la représentation proposée pour la capture, la visualisation et la quantification des conceptions de processus métiers. Elle se compose de trois (03) structures principales :

- La perspective visuelle : Une représentation schématique des conceptions de processus métiers utilisant des organigrammes.
- La perspective quantitative : Une représentation mathématique qui capture les éléments des conceptions de processus métiers à l'aide de la matrice TRM et de la librairie des tâches.
- L'Algorithme de vérification inverse *ReProVA* : Un algorithme qui fournit un pont entre les deux perspectives et assure la faisabilité des conceptions de processus métiers générées.

Dans ce chapitre, les conceptions de processus métiers ont été spécifiés en fournissant des définitions basées sur le contexte de cette recherche. Le but de la représentation est de rendre une conception de processus métier souple et facile à manipuler par un algorithme évolutionnaire.

Le chapitre suivant traite de la façon dont l'algorithme évolutionnaire choisie pour l'optimisation multicritères utilise la représentation proposée, pour produire d'autres conceptions de processus métiers.



Chapitre 5

Framework pour l'optimisation multicritères des conceptions de processus métiers

Introduction	92
5.1. Formulation du problème	92
5.2. Challenges et défis de l'optimisation	94
5.3. Framework d'optimisation proposé.....	96
5.3.1. Fonctionnement du Framework.....	96
5.3.2. Principales étapes du Framework	98
5.3.2.1. Générer une population aléatoire.....	98
5.3.2.2. Vérifier les contraintes.....	98
5.3.2.3. Evaluer la solution.....	99
5.3.2.4. Exécuter l'algorithme évolutionnaire.....	99
5.3.3. Proposition de croisement.....	100
5.3.3.1. Pourquoi penser à changer de croisement ?.....	100
5.3.3.2. Représentation en clés aléatoires.....	101
5.3.3.3. Croisement sélectif.....	104
5.3.4. Représentation d'une solution.....	108
Conclusion	109

Introduction

Ce chapitre présente le Framework pour l'optimisation multicritères des conceptions de processus métiers proposée. Le chapitre commence par présenter la formulation mathématique du problème d'optimisation des conceptions de processus métiers et de détailler les défis que le Framework proposé doit aborder. Le Framework est construit en utilisant deux (02) composantes principales : La représentation des conceptions de processus métiers proposée (détaillée dans le chapitre 4) et une version modifiée de l'algorithme évolutionnaire NSGAI, appelé *xNSGAI* dédié à une optimisation combinatoire multicritères. Les entrées, les sorties et les principales étapes du Framework sont discutées. Le chapitre conclut avec la mise en œuvre du Framework, afin de générer des résultats.

5.1. Formulation du problème

Cette section présente la formulation du problème de l'optimisation des conceptions de processus métiers. La formulation du problème suppose que les exigences des conceptions de processus métiers sont saisies en fonction de la représentation proposée au chapitre 4. Le tableau 5.1 présente les paramètres du problème en fonction de la représentation de processus métier proposée.

La formulation du problème de l'optimisation multicritères des conceptions de processus métiers est la suivante :

Pour une conception de processus métier avec un ensemble n_d tâches et p attributs :

Minimiser/maximiser $(PA_1, PA_2, \dots, PA_p)^T$

Tel que :

- | | |
|--|---------------------------|
| 1. $n \geq n_d > 0$ | a. $n_d \geq n_{min} > 0$ |
| 2. $r \geq r_{in}, r_{out}, t_{in}, t_{out} > 0$ | b. $n_d \in S_d$ |
| 3. $p \geq 2$ | |

Paramètre	Description	Paramètre	Description
n	Nombre de tâches dans la librairie	n_d	Nombre de tâches dans une conception
N	Ensemble de n tâches	N_d	Ensemble des n_d tâches, $N_d \subset N$
n_{min}	Nombre minimal de tâches dans une conception	n_{max}	Nombre maximal de tâches dans une conception
t_{in}	Nombre de ressources d'entrée d'une tâche	t_{out}	Nombre de ressources de sortie d'une tâche
r_{in}	Nombres de ressources d'entrée d'une conception	r_{out}	Nombres de ressources de sortie d'une conception
r	Nombre de ressources disponibles	p	Nombre d'attributs d'une tâche/conception
S_d	Ensemble des différentes tailles d'un processus	PA	Ensemble de p attributs pour une conception
TA_i	Ensemble d'attributs pour une tâche i		

Tableau 5.1 Paramètres du problème d'optimisation des conceptions de processus métiers

Nous supposons que les attributs d'une conception de processus métier sont utilisés comme critères d'optimisation. Un attribut d'une conception (PA_j) peut être calculé comme un agrégat des attributs de tâches correspondants stockés dans la librairie, pour toutes les tâches n_d dans une conception de processus métier, selon l'équation (4.1).

La formulation du problème suppose qu'il existe plus d'un attribut d'une conception utilisé comme critère d'optimisation, ce qui conduit à un problème d'optimisation multicritères. La formulation du problème comporte également cinq (05) contraintes, 03 obligatoires et 02 facultatives. La 1^{ère} contrainte garantit que les tâches disponibles dans la librairie (n) sont plus ou au moins égales aux tâches requises, pour composer une conception (n_d) et que (n, n_d) sont supérieures à zéro. La 2^{ème} contrainte garantit que tous les paramètres relatifs aux ressources sont supérieurs à zéro et que les ressources disponibles (r) sont supérieures à celles requises par les entrées et sorties d'une conception et d'une tâche. Enfin, la 3^{ème} contrainte suppose qu'il existe au moins deux attributs par tâche / conception, et que le problème est donc multicritères (au moins bi critères).

Chacune des contraintes qui appartiennent au deuxième ensemble (a-d) est facultative. Elles sont fournies pour rendre le problème plus souple en termes de conceptions de processus métiers générées. La contrainte (a) définit une limite

inférieure (n_{min}) au nombre de tâches qui peuvent former une conception. Dans le cas où $n_d = n_{min}$, une solution acceptable contient exactement n_d tâches dans sa conception. Une extension de la contrainte (a) est la contrainte (b). Elle considère une conception comme acceptable seulement si sa taille appartient à une plage spécifiée de tailles de processus ($\mathcal{S}_d$).

5.2. Challenges et défis de l'optimisation

La formulation décrite du problème dans la section précédente soulève certaines questions concernant la génération de conceptions optimisées de processus métiers. Les principaux défis pour cette optimisation peuvent être résumés comme suit :

- *La nature du problème.* Basé sur la formulation du problème, l'optimisation des conceptions de processus métiers est un problème discret, car la variable principale est un ensemble de tâches (N_d) qui forment des conceptions de processus métiers. Un problème discret est plus difficile et moins souple à optimiser, en grande partie à cause des contraintes sur les valeurs que peuvent prendre les variables. En outre, dans un problème discret, un changement mineur dans l'une des variables peut avoir un effet inattendu sur le processus d'optimisation.
- *Les sorties du Framework.* Sur la base de la formulation proposée pour la problématique traitée, la difficulté consiste à identifier un ensemble de tâches avec des valeurs d'attributs optimales de conception. Le Framework proposé devrait être capable de composer et produire des conceptions de processus métiers faisables. Le processus de conception doit être intégré dans le processus d'optimisation.
- *La formulation du problème multicritères.* En plus de sa nature discrète, le problème d'optimisation des conceptions de processus métiers est formulé comme multicritères. En supposant que les objectifs (ou critères) proposés sont contradictoires et que chaque solution représente un compromis entre

les différents objectifs, la découverte du front optimal de Pareto pour tous les objectifs est un autre défi majeur pour le Framework d'optimisation.

- *La représentation des solutions.* La formulation du problème requiert différents aspects dans l'élaboration des conceptions. Par exemple, l'évaluation des objectifs nécessiterait l'utilisation de la librairie des tâches. Tandis que pour vérifier la contrainte de faisabilité, il faudrait *TRM* et l'exécution de *ReProVA*. De plus, l'application d'un algorithme évolutionnaire par le Framework nécessite une représentation appropriée de la solution, pour chacun des opérateurs génétiques (sélection, croisement et mutation).
- *La sélection des solutions.* Il existe une série de contraintes obligatoires et facultatives dans le problème. Les contraintes doivent être gérées de manière à pouvoir générer et préserver des solutions réalisables, pendant le processus d'optimisation. Le Framework doit gérer les contraintes avec souplesse, afin de découvrir des solutions réalisables et les faire évoluer vers les solutions optimales.
- *La taille d'une solution.* L'optimisation des conceptions de processus métiers requiert des solutions de taille variable. L'algorithme *ReProVA* compose une conception de processus métier qui peut avoir un maximum de n_d tâches ou moins, à condition que les contraintes de conception soient satisfaites. Le fait d'avoir un nombre fixe de tâches dans la conception constituerait un obstacle majeur à la composition de la conception et est directement en conflit avec l'objectif du Framework. Par conséquent, le Framework doit être capable de traiter des solutions de taille variable, pour les mêmes contraintes de conception.
- *L'évolution d'une conception.* Comme pour le challenge précédent, le processus d'optimisation doit permettre de générer des conceptions de processus métiers diverses. Le Framework ne doit en aucun cas restreindre la conception d'une solution. Au contraire, il doit favoriser l'évolution des solutions.

La section suivante présente le Framework d'optimisation proposé pour répondre aux défis évoqués, afin de générer des résultats optimaux sous la forme d'autres conceptions de processus métiers.

5.3. Framework d'optimisation proposé

Cette section fournit une description détaillée du Framework d'optimisation évolutionnaire multi objectifs proposé pour les conceptions de processus métiers. Les principaux éléments du Framework proposé sont au nombre de deux (02), comme indiqué sur la figure 5.1.

- La représentation proposée pour les conceptions de processus métiers.
- L'algorithme évolutionnaire d'optimisation multicritères sélectionné.

Le Framework d'optimisation applique un algorithme évolutionnaire sur des conceptions de processus métiers capturées avec l'aide de la représentation proposée. Le résultat est une série de conceptions optimisées alternatives.

5.3.1. Fonctionnement du Framework

L'objectif du Framework d'optimisation évolutionnaire multicritères proposé pour les conceptions de processus métiers est d'appliquer un algorithme évolutionnaire aux exigences d'un processus métier donnés, afin de générer une série de nouvelles conceptions optimisées.

La figure 5.2 présente les entrées et sorties de Framework.

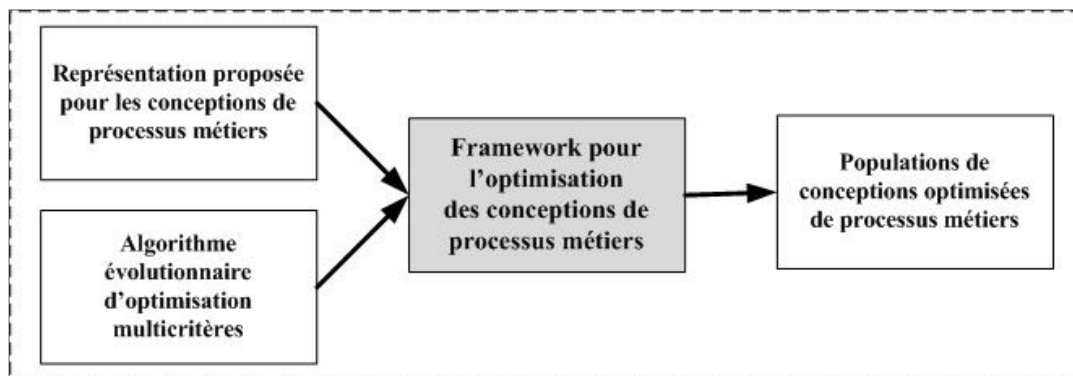


Figure 5.1 Composants du Framework d'optimisation proposé

- *Les exigences du processus métier.* Elles se présentent sous la forme des ressources d'entrée du processus métier (R_{in}) et des ressources de sortie du

processus nécessaires (R_{out}). Toutes les conceptions générées doivent partir des mêmes ressources d'entrée et conclure aux mêmes ressources de sortie.

- *La taille du processus (n_d)*. La taille du processus métier indique le nombre maximum de tâches dans les conceptions. Au cours du processus d'optimisation, *ReProVA* est autorisé à générer des conceptions avec moins de tâches.
- *La librairie des tâches (N)*. Cet ensemble contient toutes les tâches susceptibles de participer dans l'élaboration de conceptions de processus métiers. *TRM* est formé avec n_d tâches de la librairie des tâches, pour créer une solution faisable.
- *Les attributs d'une conception de processus métier*. Le Framework proposé utilise ces attributs comme objectifs (ou critères) d'optimisation. Les attributs d'une conception dépendent toujours des attributs des tâches le composant.

Le Framework proposé génère une population de conceptions de processus métiers optimisées, en utilisant les entrées (figure 5.2) avec l'algorithme évolutionnaire choisi. Pour chaque conception, le Framework produit :

- Les tâches dans chaque conception, stockées dans l'ensemble N_d .
- Le graphe de chaque conception (sa représentation schématique).
- Les valeurs des attributs de chaque conception.

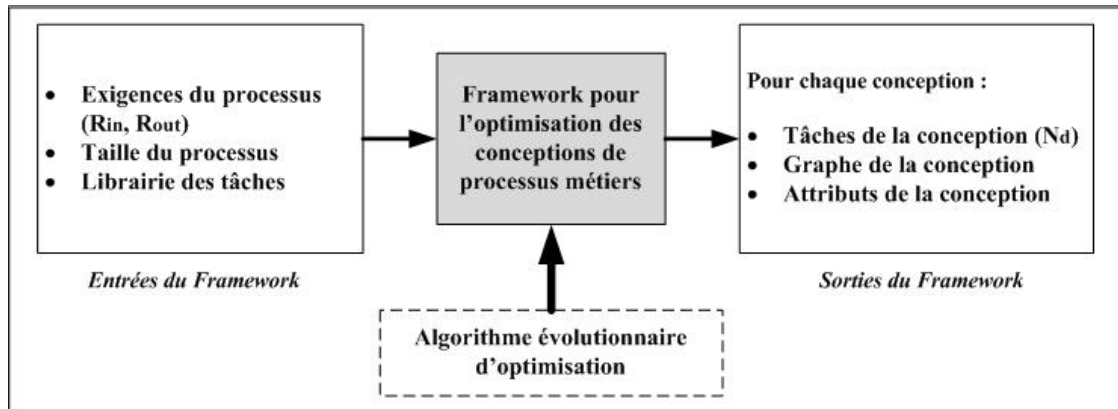


Figure 5.2 Entrées et sorties du Framework proposé

Le résultat du Framework implique la génération du graphe de chaque conception de processus métier. Les résultats (les tâches dans la conception et le graphe de

la conception) du Framework sont les résultats de l'exécution de *ReProVA*. Seules les valeurs des attributs de chaque conception de processus métier sont le résultat d'un calcul (équation 4.1), au sein du Framework.

5.3.2. Principales étapes du Framework

Les principales étapes et la structure du Framework d'optimisation des conceptions de processus métiers proposé sont présentées à la figure 5.3. Le Framework emploie une structure d'optimisation générique (boîte entourée de pointillés) qui est manipulée par l'algorithme évolutionnaire utilisé. Chacune de ces étapes d'optimisation est cependant ajustée, pour refléter le problème de l'élaboration des conceptions de processus métiers et s'assurer que le Framework utilise les ressources entrées et produit les sorties comme indiqué sur la figure 5.2.

5.3.2.1. *Générer une population aléatoire.*

Cette étape se produit une seule fois dans le processus d'optimisation alors que la population évolue vers un nombre défini de générations. La génération de population aléatoire crée un nombre fixe d'ensembles de tâches n_d . Chacun des ensembles de population contient n_d tâches allouées de N (la librairie des tâches). Cependant, pour chacun des ensembles, il existe une contrainte dans l'attribution aléatoire des tâches. La contrainte est qu'une tâche ne doit apparaître qu'une seule fois dans le même ensemble. Cette contrainte évite d'avoir des tâches en double dans un ensemble et dans une conception de processus métier. Une fois la population aléatoire générée, les étapes 2 à 4 sont répétées, pour un nombre prédéfini de générations.

5.3.2.2. *Vérifier les contraintes.*

Pour chaque solution de la population, les contraintes (exigences) liées à la formulation du problème sont vérifiées. Contrairement à la plupart des procédures d'optimisation où la solution est évaluée en premier, le Framework proposé vérifie les contraintes avant l'évaluation de la solution,

pour la simple raison que les contraintes modifient la solution. Pour ce faire, deux actions sont déclenchées : *TRM* est formé et *ReProVA* est exécuté. *TRM* reflète les relations entre les tâches de l'ensemble N_d et les ressources du problème et *ReProVA* utilise ces informations pour composer une conception de processus métier basée sur les exigences du processus. Le résultat de *ReProVA* est la version schématique de la conception du processus métier, puis N_d est mis à jour, pour deux raisons :

- Une conception peut être composée de moins de n_d tâches, donc les tâches restantes sont supprimées de l'ensemble N_d .
- Une conception aurait pu être réparée pendant la composition et la vérification, donc certaines tâches dans N_d ont pu être remplacées.

La dernière partie de cette étape est de vérifier la solution (mise à jour), si elle viole l'une des contraintes facultatives du problème.

5.3.2.3. *Evaluer la solution.*

L'évaluation de la solution implique une étape importante basée sur la représentation proposée : Le calcul des valeurs des attributs de la conception du processus métier. La raison de l'évaluation de la solution après la vérification des contraintes est que seules les tâches qui participent à une conception sont effectivement prises en compte dans le processus d'évaluation.

5.3.2.4. *Exécuter l'algorithme évolutionnaire.*

Après l'évaluation, les solutions passent par un algorithme évolutionnaire, afin de subir les différents opérateurs génétiques (sélection, croisement et mutation).

Les auteurs proposent un nouveau algorithme évolutionnaire, le *xNSGAI*. C'est un algorithme évolutionnaire d'optimisation multi objectifs basé sur NSGAI. Ce dernier est connu pour ne pas donner de bonnes performances, dans les problèmes avec de multiples fronts locaux. Cette situation empire en raison de l'utilisation de l'élitisme (convergence prématurée). Le Framework proposé tente d'y palier en proposant d'utiliser une autre forme de croisement

(représentation en clés aléatoires avec un croisement sélectif), ainsi que d'autres types de sélection (par tournoi binaire, uniforme, par roulette, par rang et par troncation). Le dernier opérateur du processus d'optimisation est la mutation. Il est appliqué sur l'ensemble de tâches N_d de chaque solution particulière. L'utilisation de *xNSGAI* permet d'examiner la capacité du Framework à découvrir et à optimiser des solutions de tailles variables.

La sous-section 3.3 présente en détail *xNSGAI* et la proposition de croisement utilisée avec cet algorithme évolutionnaire.

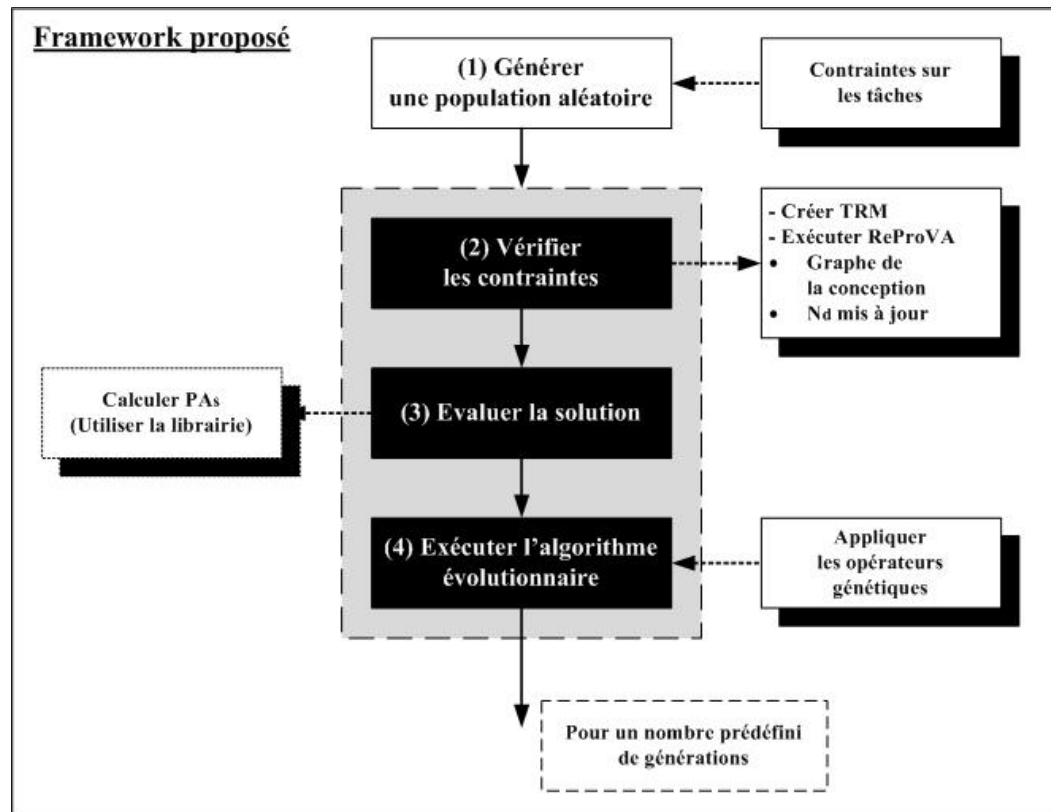


Figure 5.3 Principales étapes du Framework proposé

5.3.3. Proposition de croisement

5.3.3.1. Pourquoi penser à changer de croisement ?

Sachant que Deb and Agrawal (1994) utilise le croisement binaire simulé (ou *Simulated Binary Crossover*) avec la version canonique de NSGAI, nous avons

pris le pari d'utiliser un nouveau type de croisement. Certes, annoncer un nouveau type de croisement serait prétentieux. Il serait de bonne guerre d'expliquer que, nous nous sommes attelés à utiliser deux techniques distinctes pour arriver à un croisement ; n'ayant pas été employé auparavant dans un problème d'optimisation multicritères des conceptions de processus métiers (Preux and Talbi, 1995) (Duvivier, 2000). Le croisement proposé pourrait être pris pour une forme *hybridation séquentielle de croisement*. Selon Preux et Tabli (1995), une hybridation séquentielle (de croisement utilisée dans notre étude) consiste à exécuter séquentiellement différentes méthodes de telle manière que le résultat d'une méthode serve de solution initiale à la méthode suivante. Cette technique d'hybridation est la plus simple, elle ne nécessite pas de modification des méthodes de recherche utilisées.

Nous avons opté pour une hybridation de différentes techniques de croisement, en voici certaines des raisons possibles favorisant cette décision (Sinha and Goldberg, 2003) :

- Améliorer la performance de l'opérateur.
- Améliorer la qualité des solutions obtenues par chaque croisement.
- Améliorer l'exploration de l'espace de recherche par l'algorithme évolutionnaire.

Dans notre recherche, nous avons choisi d'utiliser, comme algorithme évolutionnaire ; NSGAI, d'y appliquer l'hybridation séquentielle de croisement proposée, et de le renommer *xNSGAI*. L'hybridation séquentielle de croisement proposé utilise (i) la représentation en clés aléatoires des solutions et (ii) la technique de croisement sélectif.

5.3.3.2. *Représentation en clés aléatoires.*

La représentation en clés aléatoires (ou *Random Keys Representation*) d'un individu (ou chromosome) a été proposée par Bean (1994), et fréquemment utilisées pour les problèmes d'optimisation combinatoire (Mason et al., 2007).

Une des particularités de ce codage est l'utilisation de nombres aléatoires comme tags, pour représenter les solutions. Ces nombres aléatoires sont prélevés d'un espace $[0, 1]^n$. L'algorithme de recherche (*xNSGAI*, dans notre étude) perçoit cet espace comme un échantillon de l'espace des solutions (Figure 5.4). Pour l'évaluation, les points dans l'espace des clés aléatoires sont mappés aux points de l'espace des solutions. Une autre particularité est la faisabilité des enfants résultant d'un croisement l'utilisant (Bean, 1994).

Nos lectures nous ont conduites à affirmer qu'aucune étude sérieuse n'a été faite sur l'utilisation de cette représentation, avec un problème d'optimisation multicritères touchant aux processus métiers. Autre point à souligner, les travaux de la représentation en clés aléatoires impliquent plus son utilisation avec un algorithme génétique simple, et rarement avec NSGAI (Mason et al., 2007) (Fan et al., 2009) (Hao and Gen, 2011) (Roque et al., 2012) (Gu et al., 2013) (Yang et al., 2013) (Zheng et al., 2014).

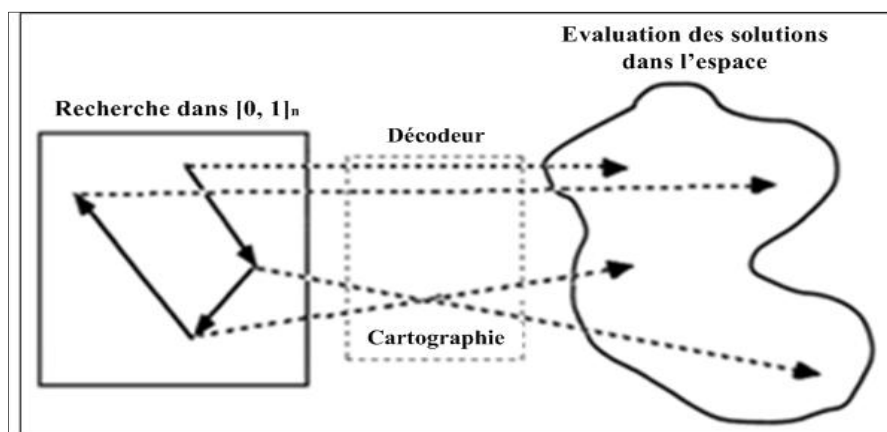


Figure 5.4 Processus de la représentation en clés aléatoires¹⁷

En adoptant la terminologie de la génétique proposée par (Bean, 1994) : Un chromosome est le codage d'une solution, prenant la forme d'un vecteur dans R^n . Un gène est un élément de ce chromosome, et un allèle est la valeur

¹⁷ Adapté de (Bean, 1994)

donnée à ce gène. La représentation en clés aléatoires sert de tags que le croisement utilise pour réorganiser la solution.

Etape	Solution proposée	Représentation en clés aléatoires
Former un chromosome en générant des nombres réels aléatoires entre $[0, 1]^n$.	$PM01 = (t_3, t_5, t_1, t_2, t_4)$	(2, 4, 5, 1, 3)
Dériver une solution en triant les nombres aléatoires affectés.	$PM02 = (t_2, t_3, t_4, t_5, t_1)$	(1, 2, 3, 4, 5)

Tableau 5.2 Représentation d'une conception de processus métier en clés aléatoires

Bean (1994) présente la structure générale de la représentation en clés aléatoires comme suit (tableau 5.2) :

1. Former un chromosome, pour chaque solution en générant des nombres réels aléatoires entre $[0, 1]^n$. Par exemple, une solution proposée avec cinq (05) tâches $(t_3, t_5, t_1, t_2, t_4)$, représentée par ses attributs (49, 39, 58, 32, 63) donne la séquence **2** → **4** → **5** → **1** → **3**.
2. A partir d'un chromosome donné, dériver une solution en triant les nombres aléatoires affectés. La suite de l'exemple donne la séquence triée selon les valeurs affectées : **1** → **2** → **3** → **4** → **5** (*i.e.* la nouvelle structure du chromosome est $(t_2, t_3, t_4, t_5, t_1)$).

Bean (1994) conseille d'appliquer le croisement sur la représentation en clés aléatoires, et non pas sur la solution elle-même. Dans ce cas, nous comptons appliquer successivement le croisement sélectif (Verkaria, 1998).

Une précision de taille mérite d'être formulée. L'association d'un chromosome avec une solution du problème d'optimisation présent fait appelle à un algorithme déterministe appelé "décodeur" (Gonçalves and Resende, 2011). Pour notre travail, nous reprenons l'idée proposée par (Bean, 1994), celle où le décodeur trie les clés aléatoires du vecteur, et utilise les indices des clés triées pour représenter une séquence.

5.3.3.3. Croisement sélectif

Après avoir présenté ce qu'est que la représentation d'une solution en clés aléatoires, nous proposons de combiner cette représentation avec le croisement sélectif prôné par (Verkaria and Clack, 1998). Démarche qui n'a pas été appliquée - auparavant - dans une démarche d'optimisation multicritères des conceptions de processus métiers. Le croisement sélectif peut être résumé en l'utilisation d'un vecteur supplémentaire qui accompagne une solution. Ce vecteur a comme objectif d'accumuler les connaissances de ce qui s'est produit (les échanges de gènes) dans les générations précédentes, et les utiliser afin de promouvoir l'utilisation de meilleurs gènes au cours des croisements sur les générations avenir. Un tel procédé permet d'éviter une stagnation au niveau de l'espace de recherche et ainsi pousser la recherche vers des régions particulières de cet espace (Verkaria, 2000).

Pour appliquer le croisement sélectif, Verkaria and Clack (1998) propose d'utiliser la notion de *dominance* des gènes. Le tableau 5.3 présente un exemple de l'utilisation de la technique de croisement sélectif :

- A chaque chromosome est associé un vecteur de valeurs réelles, et une valeur de dominance est associée pour chaque gène (tableau 5.3).

Vecteur de dominance	0.4	0.3	0.01	0.9	0.1	0.2
Chromosome	t_2	t_3	t_0	t_1	t_5	t_0

Tableau 5.3 Représentation d'un individu avec son vecteur de dominance

- Deux parents interviennent pour créer deux descendants, après la comparaison des valeurs de leurs Fitness. Le "Parent 1" est appelé *le contributeur*, du moment que sa Fitness est supérieure à celle du Parent 2 (figure 5.5).

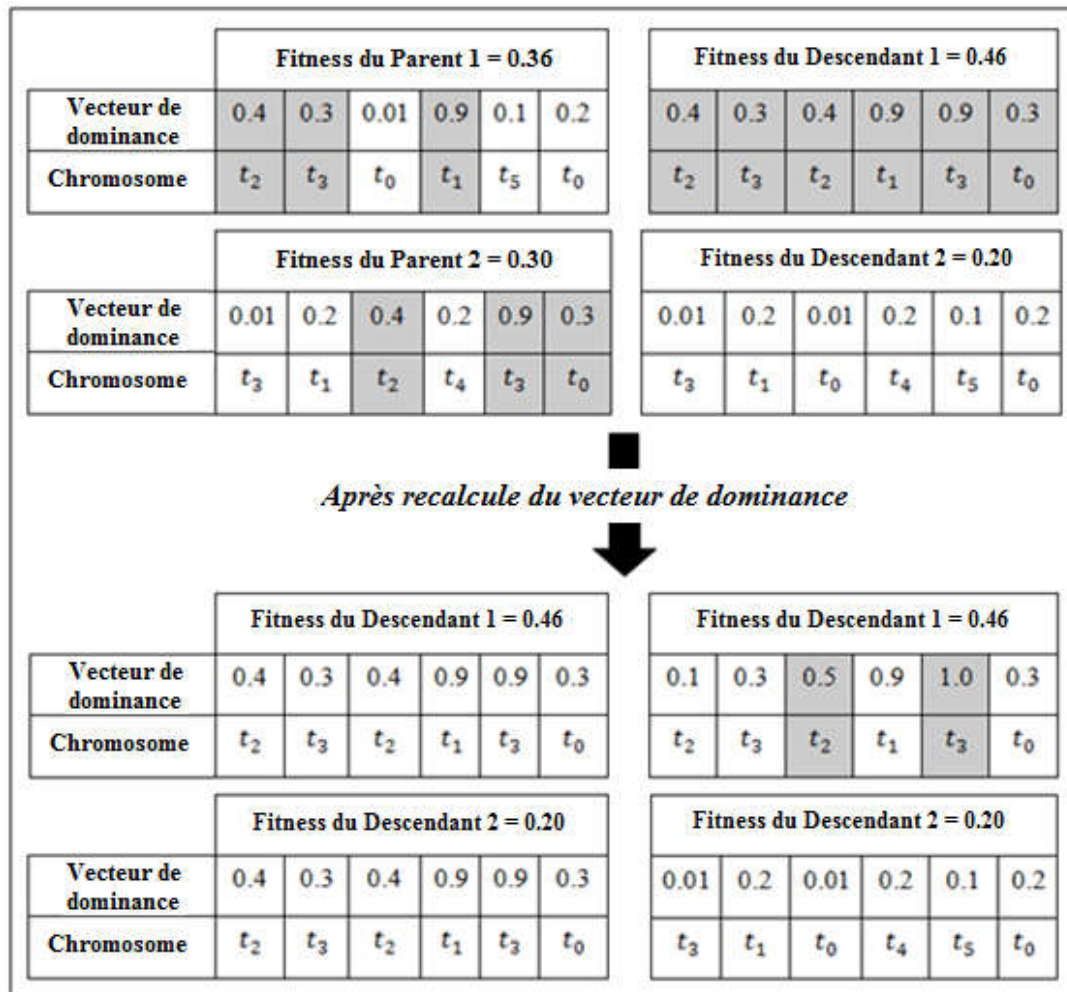


Figure 5.5 Croisement sélectif et mise à jour de la dominance

- Le descendant 1 récolte le gène avec la valeur de dominance la plus élevée (cases grises), après une comparaison linéaire des valeurs de dominance de chaque gène des deux parents (pas de croisement, si les valeurs de dominance sont égales (figure 5.6)).
- Le descendant 2 hérite des gènes non dominants. Et ceci pour maintenir la diversité de la population, afin que la diversité génétique ne se perde pas dans les premières générations.

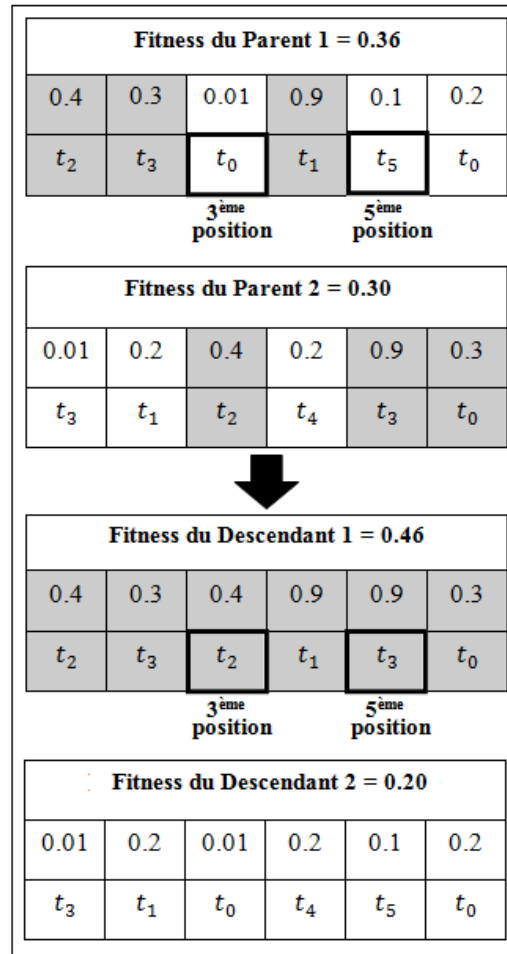


Figure 5.6 Recombinaison avec le croisement sélectif

- Après le croisement, les vecteurs de dominance des descendants doivent être mis à jours. Lors de la comparaison de la Fitness du descendant 1 avec celle d'un parent, cela donne les cas suivants (revenir à (Verkaria, 2000) pour plus de détails) :
 - (i) Fitness du Descendant 1 $>$ Fitness du Parent i , la valeur de dominance des gènes du Descendant (échangés lors du croisement) est augmentée proportionnellement avec la différence liée au Parent 1 (figure 5.7). Nous remarquons que la Fitness du Descendant 1 (0.46) est supérieur à la Fitness du Parent 1 (0.36). Les valeurs de dominance des gènes qui ont été échangés au cours du croisement seront augmentées de 0.1

(i.e. $0.46 - 0.36 = 0.1$), pour le Descendant 1. Les valeurs du 3^{ème} et 5^{ème} loci (position) du Descendant 1 diffèrent des positions correspondantes avec le Parent 1. Comme résultat, nous obtenons la valeur de dominance de 3^{ème} position du Descendant 1 égale à 0.5 (i.e. $0.4 + 0.1 = 0.5$) et la valeur de dominance de 5^{ème} position du descendant 1.0 (i.e. $0.9 + 0.1 = 1.0$).

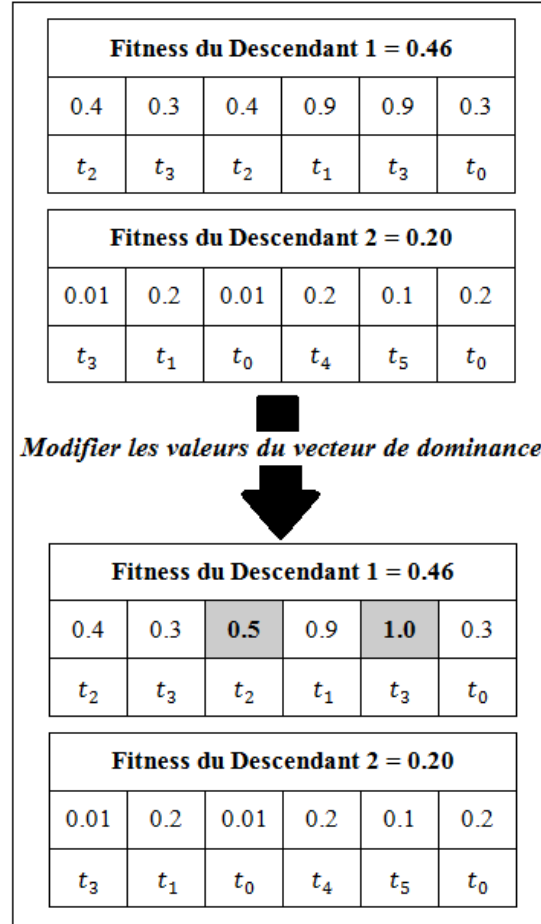


Figure 5.7 Calcule du vecteur de dominance

- (ii) Fitness du Descendant 1 $>$ Fitness du Parent 1 et Fitness du Descendant 1 $\leq$ Fitness du Parent 2, idem que le cas (i).
- (iii) Fitness du Descendant 1 $\leq$ Fitness du Parent 1 et Fitness du Descendant 1 $>$ Fitness du Parent 2, la valeur de dominance des gènes du Descendant 1 (échangés lors du croisement) est augmentée proportionnellement avec la différence liée au Parent 2.

- (iv) Fitness du Descendant 1 $\leq$ Fitness du Parent 1 et Fitness du Descendant 1 $\leq$ Fitness du Parent 2, La valeur de dominance des gènes du Descendant 1 reste inchangée.

5.3.4. Représentation d'une solution

Le Framework répond à un ensemble de défis, en rapport avec la représentation et la taille des solutions générées, en transformant et en actualisant la solution en fonction des différentes exigences du processus d'optimisation. Au cours de ce processus, le Framework s'assure que la transformation de chaque solution soit cohérente (via *ReProVA*). Les différentes formes qu'une solution peut prendre sont basées sur la représentation du processus métier proposée. La figure 5.8 montre les différentes formes d'une solution (conception) à travers les différentes étapes du Framework.

La forme initiale d'une conception est l'ensemble de tâches participantes N_d . La population aléatoire est créée sous cette forme. La séquence des tâches dans l'ensemble n'a pas d'importance, car c'est *ReProVA* qui les organise dans une conception de processus. La deuxième étape (la vérification des contraintes) est celle où la solution change le plus. Au début, la solution prend la forme de *TRM*, afin de refléter la relation entre les tâches dans la solution avec les ressources disponibles. Ceci est nécessaire pour l'exécution de *ReProVA*. *TRM* est transformée en un graphe de processus métier, afin de mettre à jour l'ensemble N_d avec les tâches qui participent réellement à la conception. *ReProVA* est autorisé à créer un graphe avec moins de tâches que celui de la solution d'origine, à condition que le graphe réponde aux exigences du processus (les ressources d'entrée et de sortie).

L'opérateur de sélection à l'étape 3 évalue la solution en utilisant les valeurs des attributs à partir de la librairie des tâches, en se basant sur l'ensemble N_d mis à jour. Les attributs des conceptions de processus métiers sont calculés, pour aider à choisir les solutions les plus adaptées. *xNSGAI* est exécuté avec le croisement sélectif qui s'applique sur l'ensemble N_d de chaque solution. Par la suite, la mutation est appliquée sur l'ensemble N_d .

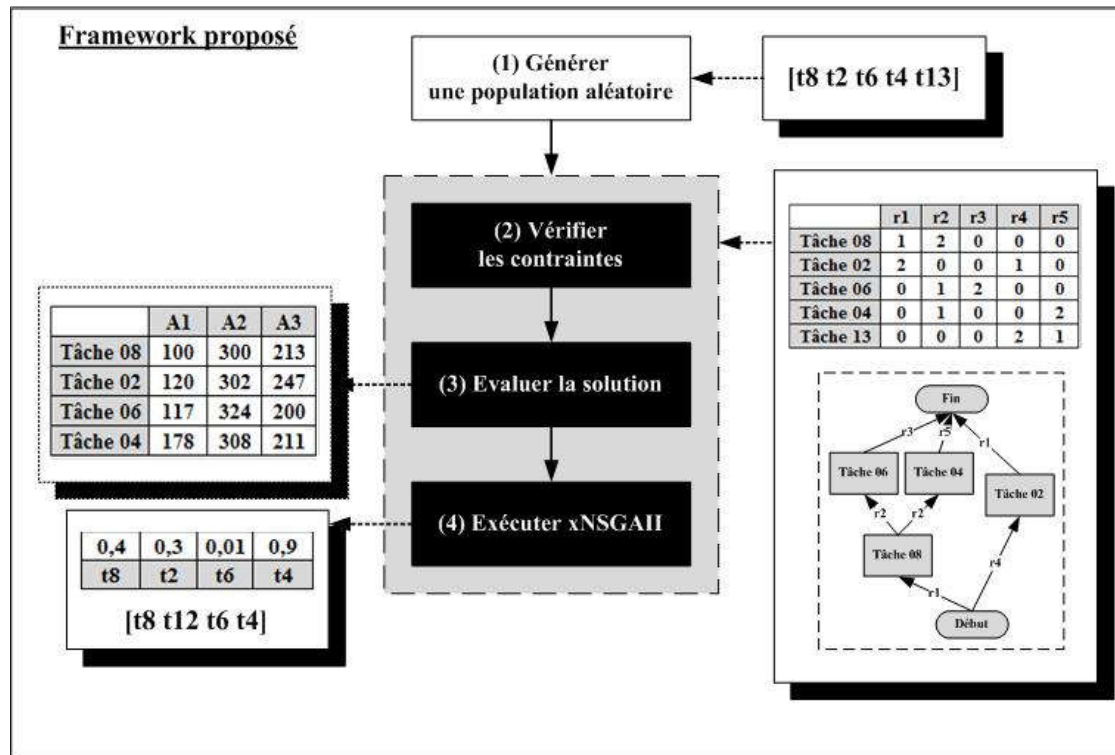


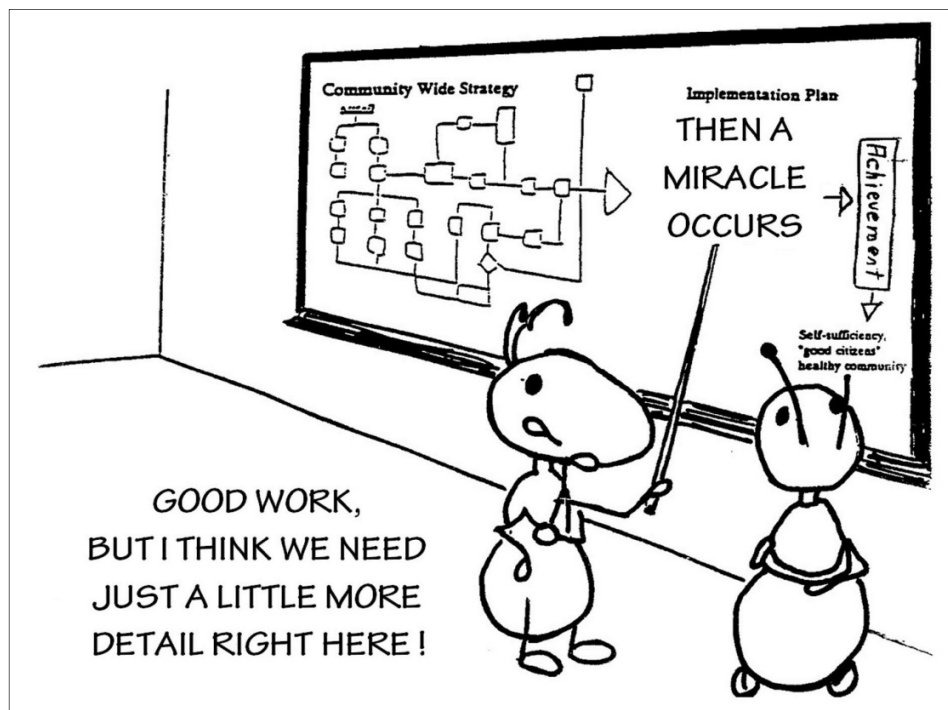
Figure 5.8 Différentes formes d'une solution durant les étapes du Framework

La transformation de la solution à travers les différentes étapes du processus d'optimisation contribue à relever plusieurs défis énoncés auparavant, puisque chaque solution est transformée pour répondre aux exigences de chaque étape : *TRM* et le graphe de la conception pour les contraintes, N_d pour la population aléatoire et l'exécution de *xNSGAI*. Toutes ces transformations sont basées sur la représentation proposée pour les conceptions de processus métiers.

Conclusion

Ce chapitre a présenté le Framework d'optimisation évolutionnaire multi objectifs proposé, pour les conceptions de processus métiers. L'approche d'optimisation proposée utilise *xNSGAI*, une version modifiée de l'algorithme évolutionnaire NSGAI. *xNSGAI* se propose d'utiliser un type de croisement. L'association de la représentation en clés aléatoires d'une solution, avec un opérateur de croisement (croisement sélectif) utilisant autre chose que la fonction de Fitness pour procéder.

La sélection de cet algorithme évolutionnaire d'optimisation s'est justifiée par les exigences posées par les conceptions de processus métiers. La formulation du problème a soulevé un certain nombre de défis pour le Framework d'optimisation proposé, tels que la représentation de la solution et la gestion des contraintes. Pour que le Framework proposé soit correctement évalué, le chapitre 6 présente un scénario test (inspiré de la vie réelle) avec lequel les résultats expérimentaux permettent de juger de ses performances.



Chapitre 6

Expérimentation du Framework d'optimisation proposé

Introduction.....	113
6.1. But du scénario réel de processus métier.....	113
6.1.1. Contexte du scénario de la vie réelle.....	113
6.1.1.1. Processus métier en tant que système de services.	114
6.1.1.2. Caractéristiques des processus métiers de la vie réelle.....	115
6.1.1.3. Scénario de la vie réelle.	116
6.2. Scénario test : La prévision des ventes	117
6.2.1. Conception de processus métier initial.....	117
6.2.2. Librairie des tâches.....	118
6.2.3. Scénario de processus métier complet	119
6.3. Fonction de Fitness proposée	120
6.3.1. Pourquoi une - nouvelle - fonction de Fitness ?.....	120
6.2.3. Fonction de Fitness utilisée par le Framework proposé.....	121
6.3. Expérimentation du Framework proposé.....	122
6.3.3. Paramètres du Framework proposé avec le scénario test.....	123
6.3.4. Evaluation et interprétation des résultats.....	123
Conclusion	133

Introduction

Ce chapitre valide la capacité du Framework d'optimisation proposé à traiter des conceptions de processus métiers qui appellent des éléments réels. Pour ce faire, un scénario – tiré de l'industrie des services - est sélectionné et testé par le Framework. La procédure d'adaptation d'un scénario tiré de la vie réelle, dans le Framework d'optimisation proposé, est décrite en détail. Ce chapitre présente également les résultats d'une série d'expérimentations visant à comparer le rendement du Framework avec la pratique actuelle en matière d'optimisation des conceptions de processus métiers.

6.1. But du scénario réel de processus métier

Cette section justifie les raisons pour lesquelles le Framework d'optimisation des conceptions de processus métiers proposé est testé avec un scénario réel. Voici une définition de ce qui est perçu comme un scénario de processus métier réel, par notre recherche :

"Un scénario de processus métier réel est un ensemble de paramètres qui sont extraits d'un modèle de processus métier réel rapporté dans la littérature ou pris en compte dans la pratique de l'industrie." (Vergidis, 2008).

Tester le Framework proposé avec un scénario réel vise à valider sa capacité en matière de composition et d'optimisation des conceptions de processus métiers.

6.1.1. Contexte du scénario de la vie réelle

Cette sous-section définit le contexte de sélection d'un scénario de la vie réelle en présentant deux perspectives d'un processus métier (i) L'automatisation des processus métiers et (ii) définir les processus métiers en tant que système de services. Ces deux perspectives sont choisies par les chercheurs car elles fournissent les caractéristiques du scénario de processus métier avec lesquelles le Framework peut travailler.

6.1.1.1. *Processus métier en tant que système de services.*

Un système de service est une configuration de réseaux technologiques et organisationnels conçus pour fournir des services qui satisfont le client (Reijers and Hee, 2002). Nous avons choisi d'orienter cette recherche sur les conceptions de processus métiers, vers le contexte de l'industrie des services. Ainsi, une conception de processus métier peut être perçue comme un type *générique* d'un système de services. Les problèmes de conception d'un système de services ont donné naissance à *SOA*¹⁸ qui est un style architectural pour la création et l'utilisation de processus métiers emballés comme des services tout au long de leur cycle de vie. *SOA* considère les tâches dans un processus métier comme des unités distinctes qui peuvent être réparties sur un réseau et peuvent être combinées et réutilisées. Ces unités sont appelées des services web. Un service web est un ensemble défini de fonctionnalités commerciales ou techniques contiguës et autonomes mises en œuvre sur un réseau (Vergidis and Tiwari, 2008).

Donc, la considération d'un processus métier comme un système de services permet aux tâches d'une conception de processus métier d'être appliqués (exécutés) par des services web.

Un autre avantage de l'adoption de la perspective du système de services pour les conceptions de processus métiers est les attributs de spécification de services qui peuvent être utilisés comme attributs des conceptions. Ce qui peut conduire à les considérer comme des objectifs (critères) d'optimisation par le Framework proposé.

Selon Lakin et al., (1996), tout service peut être clairement et complètement spécifié au moyen des douze (12) attributs de spécification de services suivants :

1. Prestation (s) du consommateur de services (*Service Consumer Benefit(s)*).

¹⁸ Service-Oriented Architecture

2. Paramètre (s) fonctionnel (s) (*Service-specific Functional Parameter(s)*).
3. Point de livraison des services (*Service Delivery Point*).
4. Nombre de consommateurs de service (*Service Consumer Count*).
5. Temps de préparation du service (s) (*Service Readiness Time(s)*).
6. Durée du service (s) (*Service Support Time(s)*).
7. Langue (s) du support technique (*Service Support Language(s)*).
8. Objectif de la réalisation du service (*Service Fulfilment Target* ou **SFT**).
9. Durée maximale de l'incapacité par incident (*Maximum Impairment Duration per Incident*).
10. Délai de livraison du service (*Service Delivering Duration* ou **SDD**).
11. Unité de prestation de services (*Service Delivery Unit*).
12. Prix de livraison de service (*Service Delivering Price* ou **SDP**).

6.1.1.2. *Caractéristiques des processus métiers de la vie réelle.*

Le contexte d'un scénario réel aide à aligner le scénario de la vie réelle avec les éléments du processus métier suggéré par le Framework proposé. Les caractéristiques spécifiques d'un scénario réel et leur lien avec les éléments proposés par cette recherche sont ainsi présentés :

- Librairie des tâches (librairie des services web) : Un service Web est un système logiciel qui exécute une tâche et est conçu pour interagir sur un réseau. Compte tenu du fait qu'il existe des librairies de services web disponibles sur Internet et que le Framework proposé repose largement sur une librairie des tâches, l'adoption de la perspective du "service web" peut être l'occasion de (i) démontrer les capacités d'automatisation et d'optimisation du Framework, en utilisant des exemples concrets de conceptions créées par les services web disponibles. Et, (ii) pour souligner l'importance d'adopter les services web comme un élément crucial pour l'avenir de l'automatisation et

de l'amélioration des conceptions de processus métiers. Ainsi, le scénario de processus métier réel est composé de services web stockés dans des librairies en ligne.

- Objectifs (ou critères) d'optimisation (attributs des spécifications des services web) : Le Framework proposé est orienté vers l'optimisation multi objectifs (avec trois (03) objectifs à traiter). Il est également focalisé sur des problèmes de min-max, i.e. la minimisation du premier objectif, la maximisation du second et minimisation du troisième. Sur la base des attributs de spécification de service, la minimisation du prix de livraison de service (*SDP*) est sélectionnée comme premier objectif. *SDP* spécifie le coût d'utilisation du service. Le deuxième objectif est la maximisation de la cible de réalisation du service (*SFT*). *SFT* spécifie le taux de satisfaction d'un client demandant le service, par rapport à la promesse du prestataire de services d'une livraison efficace. Le troisième objectif est la minimisation du délai de livraison du service (*SDD*). *SDD* spécifie la durée que prennent la réalisation et la remise du service demandé par le client. Ces trois attributs de services sont utilisés comme objectifs d'optimisation par le scénario de la vie réelle.

6.1.1.3. *Scénario de la vie réelle.*

Le scénario sélectionné a comme objectif de montrer la polyvalence et la capacité du Framework d'automatiser et d'optimiser les conceptions de processus métiers. Ce scénario est tiré de la littérature et est adapté de façon appropriée en fonction de la représentation d'une conception de processus métier proposée.

Pour le choix du scénario de la vie réelle, trois (03) critères ont été appliqués (i) être reconnu comme pratique courante dans la littérature des processus métiers, (ii) être en mesure de s'accorder avec le Framework proposé et (iii) être en mesure de démontrer que l'utilisation des services web autant que tâches participantes dans des conceptions de processus métier rapprochent ces

conceptions de la technologie SOA. Le scénario sélectionné est : "La prévision des ventes".

6.2. Scénario test : La prévision des ventes

Ce scénario décrit le processus métier de la prévision des ventes (Grigori et al., 2004). Le Framework doit automatiser complètement le processus en sélectionnant et en mettant en œuvre des services web pertinents et en proposant un ensemble de conceptions optimisées qui répondent aux exigences du processus métier avec des valeurs d'attributs optimales.

6.2.1. Conception de processus métier initial

La figure 6.1 illustre la conception générique du processus métier pour le scénario test. Elle implique 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 que le service web puisse extraire des données pertinentes sur la société spécifiée. La deuxième ressource est une demande de mise à jour du marché qui doit être prise en compte pour les prévisions des ventes.

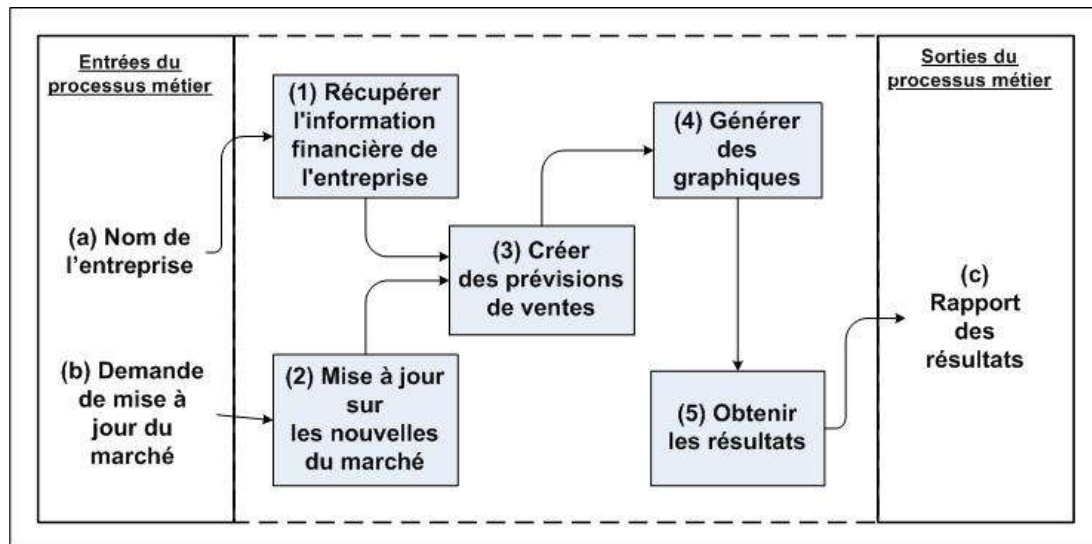


Figure 6.1 Conception initiale du processus métier "La prévision des ventes"

La conception initiale du processus se compose de cinq étapes :

La récupération des informations financières adéquates pour les prévisions pertinentes. Les derniers niveaux de marché (e.g le niveau des stocks) doivent être mis à jour, car ils sont pris en compte pour des prévisions justes. Le résultat de ces deux étapes est ensuite transmis à une simulation de Monte-Carlo qui génère la prévision de vente. Cette prévision est ensuite tracée dans un graphe. Puis communiquée à la personne qui la demande. Le résultat de ce processus est un rapport contenant les résultats de la prévision.

6.2.2. Librairie des tâches

La deuxième étape est la compilation de la librairie des services web basée sur les principales étapes du processus métier. Des recherches pertinentes sur les librairies en ligne de services web ont permis de sélectionner 20 services web de différents fournisseurs (Vergidis and Tiwari, 2008). Les entrées et les sorties de chaque service web ont été fournies soit par leur source, soit par leur description. Dans les deux cas, les noms des ressources ont été modifiés, afin que ces services soient considérés comme des alternatives par le Framework proposé.

N°	Nom de la ressource
0	Détails d'entreprise
1	Requête d'entreprise
2	Graphique
3	Nom de l'entreprise
4	Fax (en ligne)
5	Données financières
6	Demande de mise à jour du marché
7	Tendances récentes du marché
8	Prévisions de séries chronologiques

Tableau 6.1 Ressources disponibles pour le scénario test

N°	Nom de tâche ¹⁹	Entrée(s)	Sortie(s)	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	{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	{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 6.2 Librairie des tâches pour le scénario test

6.2.3. Scénario de processus métier complet

Après avoir rassemblé toutes les informations nécessaires (librairie des services web et des ressources disponibles), les paramètres du problème peuvent être définis. Le tableau 6.2 résume la librairie des tâches, avec vingt (20) services web sélectionnés parmi les librairies en ligne. Pour chaque tâche, ses ressources d'entrée

¹⁹ Les noms des tâches sont gardés en anglais.

et de sortie ainsi que les valeurs SDP, SFT et SDD se trouvent dans les colonnes adjacentes. Le tableau 6.1 présente les 9 ressources différentes parmi les services web dans la librairie. Sur la base de ces données, le tableau 6.3 résume les valeurs des paramètres du scénario test.

Désignation	Valeur
n	20
n_d	5
n_{min}	4
n_{max}	6
r	9
r_{in} / r_{out}	2/1
p	3
SDP	200 - 230
SFT	100 - 115
SDD	285 - 375

Tableau 6.3 Paramètres utilisés avec le scénario test

6.3. Fonction de Fitness proposée

Après avoir présenté le scénario test du processus métier et avoir défini les critères d'optimisation utilisables par le Framework proposé, il est important de préciser comment ces derniers sont utilisés. Cette section introduit et explique la fonction de Fitness que le Framework adopte pour l'optimisation multicritères des conceptions de processus métiers.

6.3.1. Pourquoi une - nouvelle - fonction de Fitness ?

Nelson et al. (2009) affirment que la fonction de Fitness est au cœur d'une application informatique évolutive. La fonction de Fitness a pour rôle d'évaluer - qualitativement - les solutions, afin de résoudre un problème particulier.

Lors de la phase d'expérimentation, le Framework d'optimisation proposé utilise le scénario test présenté, avec *xNSGAI* comme algorithme évolutionnaire. Sachant

que le travail présenté s'inspire de celui présenté par (Vergidis et al., 2015), le Framework compare les résultats obtenus avec ceux proposés par ce dernier.

L'étude comparative sur les travaux d'optimisation des processus métiers (chapitre 03) a fait paraître les points suivants (i) la grande majorité des travaux étudiés utilisent une simple fonction d'agrégation des objectifs d'optimisation comme fonction de Fitness. (ii) (Vergidis et al., 2015) utilisent deux fonctions d'agrégation, pour calculer la fonction de Fitness (une pour chaque objectif défini). (iii) L'importance de la fonction de Fitness dans l'algorithme évolutionnaire choisi (*xNSGAI*). Les auteurs ont jugé bon de proposer une fonction de Fitness autre que celle utilisée couramment dans un processus d'optimisation multi objectifs. Il paraît évident que réécrire la fonction de Fitness, ou d'en proposer une nouvelle capable de traiter jusqu'à trois (03) critères d'optimisation : Minimiser SDP et SDD et maximiser SFT, soit un choix qui s'impose aux auteurs.

6.2.3. Fonction de Fitness utilisée par le Framework proposé

Afin de présenter la fonction de Fitness utilisé avec le Framework d'optimisation, les auteurs se proposent de l'expliquer et de la représenter géométriquement. Pour faciliter la lecture, les lettres $\mathcal{C}, \mathcal{T}, \mathcal{S}$ sont utilisées, pour représenter les trois (03) critères d'optimisation SDP, SDD et SFT, respectivement. $F(\mathbf{x})$ est la fonction de Fitness pour l'optimisation multicritères utilisée dans le Framework proposé.

Pour représenter correctement $F(\mathbf{x})$, les auteurs font appel à la loi des sinus (voir (McKeague, 1983)). $F(\mathbf{x})$ peut être symbolisée géométriquement par le rayon R du cercle circonscrit du triangle $\mathcal{CTS}$ (Figure 6.2), où $\mathcal{CTS}$ est un triangle. Le type du triangle importe peu, pour simplifier le calcul de R . Chaque sommet du triangle est situé dans un des axes du plan XYZ . Le choix de l'utilisation de R comme fonction de Fitness s'est fait, par l'absolue nécessité de ne favoriser aucun objectif en particulier et spécialement dans une optimisation multicritères (Fonseca & Fleming, 1995). La fonction de Fitness se présente comme suit :

$$F(\mathbf{x}) = R = SC / \sqrt{1 - \{(|TS|^2 + |CT|^2 - |SC|^2) / [2(TS)(CT)]\}^2} \quad (6.1)$$

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

c_i : Prix de livraison d'une tâche i appartenant à une conception.

s_i : Taux de satisfaction dû à la réalisation d'une tâche i d'une conception.

t_i : Durée de livraison d'une tâche i faisant partie d'une conception.

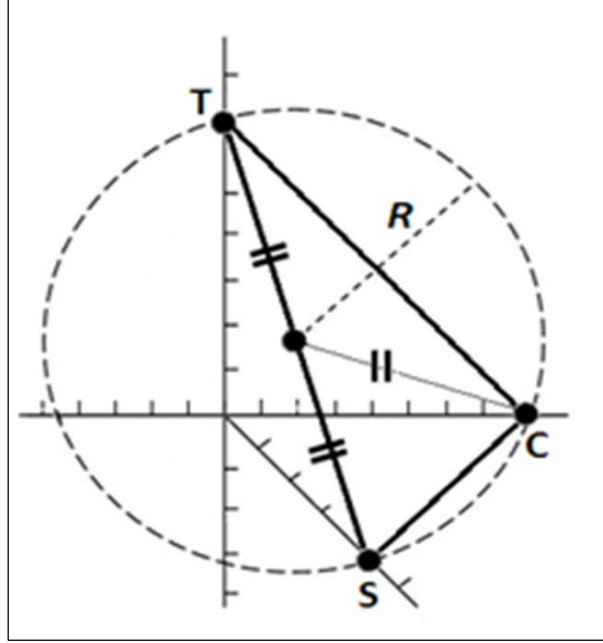


Figure 6.2 Représentation du rayon du cercle circonscrit du triangle CTS

6.3. Expérimentation du Framework proposé

Ce chapitre présente les résultats des expérimentations du Framework proposé en utilisant le scénario test présenté ; tiré de la vie réelle. Comme expliqué dans le chapitre 4, le Framework d'optimisation utilise une représentation proposée pour les conceptions de processus métiers, avec un nouvel algorithme évolutionnaire *xNSGAI* (chapitre 5). Ce dernier faisant appel à une fonction de Fitness, capable d'agir dans une optimisation évolutionnaire combinatoire et multicritères, avec (03) objectifs d'optimisation.

Les auteurs proposent de faire passer une série d'expérimentation au Framework proposé, en modifiant :

- La taille de la population initiale utilisée par le Framework.
- Le nombre d'itérations du Framework.
- Le type de techniques de croisement avec *xNSGAI*.
- Le type de techniques de sélection utilisé avec *xNSGAI*.

6.3.3. Paramètres du Framework proposé avec le scénario test

Le but du Framework proposé est de trouver des solutions satisfaisantes, i.e. des conceptions de processus métiers faisables avec des valeurs d'attributs optimales (au vu de l'optimisation multicritères traitée). Pour y parvenir, le Framework doit (i) garantir l'obtention de conceptions optimales (convergentes vers le front de Pareto), (ii) respecter la diversité de tailles des conceptions (conserver la diversité de la population au cours des différentes itérations), et (iii) utiliser la diversité des critères que comprend une tâche (utiliser les attributs des tâches, pour évaluer chaque conception de processus métier).

Le tableau 6.4 présente les paramètres de l'algorithme évolutionnaire *xNSGAI* utilisé par le Framework proposé, pour tester ce dernier avec le scénario test. Les valeurs proposées pour les caractéristiques d'une conception de processus métier avec $n_{\min}$, $n_{\max}$ - la taille maximale et la taille minimale d'une conception, respectivement -, la taille de la population initiale, le nombre d'itérations du Framework, la probabilité de mutation sont repris de Vergidis et al. (2015). La taille de la librairie (n), les ressources d'entrée et de sortie du processus métier du scénario test (R_{in} , R_{out}) et le nombre de ressources (r_d) sont tirés de la description du scénario d'expérimentation.

6.3.4 Evaluation et interprétation des résultats

Comme mentionné précédemment, pour évaluer l'efficacité du Framework, il est recommandé de comparer les résultats obtenus avec ceux publiés par (Vergidis et al., 2015), pour le scénario test. Une telle comparaison devrait mettre en évidence

la manière dont *xNSGAI* génère les solutions en termes de minimisation de coût (SDP), maximisation de la satisfaction (SFT) et minimisation de la durée (SDD).

Paramètre	Valeur
n	20
$n_{\min}$	4
$n_{\max}$	6
r_d	9
R_{in}	{r3, r6}
R_{out}	{r8}
p	3
<i>SDP</i>	200 - 230
<i>SFT</i>	100 - 115
<i>SDD</i>	285- 375
Population initiale	500
Itération	25000
Probabilité mutation	0,2

Tableau 6.4 Paramètres du Framework pour l'expérimentation du scénario test

En utilisant le scénario test, les résultats obtenus sont indiqués dans tableau 6.5 et 6.9. Le tableau 6.5 affiche les différences entre les résultats obtenus et ceux de (Vergidis and Tiwari, 2008) et (Vergidis et al., 2015), pour le scénario test. Le tableau 6.6 reprend les résultats du tableau 6.5 et les compare avec ceux obtenus par le Framework avec *xNSGAI*, mais en changeant la technique de sélection employée.

Le tableau 6.5 compare les résultats de (Vergidis and Tiwari, 2008) et (Vergidis et al., 2015) et ceux du Framework d'optimisation proposé, selon trois critères: SDP, SFT et SDD. la comparaison se fait - en premier temps - selon les critères d'optimisation SDP et SFT, sachant que les travaux de Vergidis se contentent d'utiliser ces deux critères.

	Vergidis et Tiwari (2008)			Vergidis et al., (2015)			Framework proposé		
Sélection binaire par tournoi	Croisement binaire simulé								
Taille conception (tâche)	4	5	6	4	5	6	4	5	6
SDP	853	1056	1264	853	1056	1261	844	1053	1257
SFT	446	553	662	446	553	663	447	554	663
SDD	-	-	-	-	-	-	1457	1740	2181

Tableau 6.5 Comparaison des résultats avec ceux de (Vergidis et al., 2008, 2015)

Les résultats affichés par le tableau 6.5, pour le Framework proposé, ont été obtenus en utilisant le Framework avec le croisement binaire simulé comme opérateur de croisement, et la sélection binaire par tournoi comme technique de sélection. Cette comparaison reprend les opérateurs qu'utilise - habituellement - l'algorithme évolutionnaire NSGAI, en mettant l'accent sur les tailles obtenus pour les solutions : 4, 5 puis 6 tâches (selon les paramètres du tableau 6.3). Il en ressort clairement que les résultats obtenus, par le Framework d'optimisation, sont meilleurs que ceux de Vergidis (Les résultats en gras indiquent les améliorations réalisées par le Framework proposé). Même si les opérateurs génétiques utilisés (sélection, croisement et mutation) restent les mêmes, sur un scénario test identique, l'algorithme chargé de composer et d'optimiser (*ReProVA*) de nouvelles solutions et la fonction de Fitness proposée font la différence.

Le tableau 6.6 présente les résultats du Framework proposé, selon trois objectifs d'optimisation, en utilisant différents opérateurs génétiques :

- Comme opérateur de croisement : Le croisement binaire simulé puis, le croisement proposé : Le croisement sélectif combiné avec la représentation par clés aléatoires.
- Comme opérateur de sélection : La sélection binaire par tournoi, uniforme, par roulette, par rang, puis par troncation.

	Croisement binaire simulé			Croisement sélectif + clés aléatoires		
Taille conception (tâche)	4	5	6	4	5	6
Sélect. binaire par tournoi						
SDP	844	1053	1257	843	1045	1248
SFT	447	554	662	448	555	665
SDD	1442	1740	2181	1395	1752	2120
F(x)	843	1073	1257	823	1030	1242
Sélection uniforme						
SDP	838	1048	1260	843	1045	1256
SFT	446	558	662	448	555	659
SDD	1420	1767	2122	1395	1752	2144
F(x)	832	1037	1254	823	1030	1246
Sélection par roulette						
SDP	847	1053	1248	843	1045	1256
SFT	448	554	645	448	555	659
SDD	1411	1740	2140	1395	1752	2144
F(x)	831	1073	1249	823	1030	1246
Sélection par rang						
SDP	852	1050	1252	843	1052	1251
SFT	448	550	642	448	553	663
SDD	1456	1828	2176	1395	1729	2135
F(x)	851	1063	1265	823	1053	1249
Sélection par troncation						
SDP	852	1053	1261	843	1045	1251
SFT	448	554	662	448	555	663
SDD	1456	1740	2120	1395	1752	2135
F(x)	851	1073	1252	823	1030	1249

Tableau 6.6 Comparaison des résultats pour le scénario test

Les résultats présentés (tableau 6.6) sont structurés de la manière suivante : Chaque type d'opérateur de croisement est utilisé avec les différents opérateurs de sélection. Les solutions optimisées ont été produites en exécutant chaque opérateur dans le Framework avec les valeurs du tableau 6.3. Les solutions générées

représentent des conceptions de processus métiers réalisables, avec SDP et SDD minimisé et SFT maximisé, selon le processus métier représentant le scénario test.

Le tableau 6.6 montre clairement que l'utilisation d'un opérateur de croisement différent que le croisement binaire simulé (utilisé habituellement avec NSGAI) avec d'autres techniques de sélection, améliore les résultats présentés dans le tableau 6.5. Cependant, à la suite de la conduite des expérimentations, il s'avère que ces mêmes résultats peuvent être obtenus avec une population initiale de près de 100 Individus et 50 itérations, comparé aux paramètres du tableau 6.4 (repris de (Vergidis et al., 2015)). Une telle modification conduit à un énorme gain de temps d'exécution. La cause première de ce constat est l'utilisation de l'algorithme *ReProVA*.

Le tableau 6.6 reprend en partie les résultats du tableau 6.5. Deux conclusions importantes ressortent. Tout d'abord, les auteurs notent que l'utilisation d'un croisement sélectif combiné avec une représentation en clés aléatoires améliore les solutions obtenues (voir tableau 6.5). Ces résultats peuvent être expliqués par le fait que l'utilisation du croisement sélectif explore de nouvelles zones de l'espace des solutions (comme affirmé par (Verkaria, 2000)). Deuxièmement, après avoir obtenu des solutions améliorées par rapport à celles présentées dans le tableau 6.5 - d'un point de vue quantitatif -, il semble intéressant aux auteurs de comparer les solutions qualitativement.

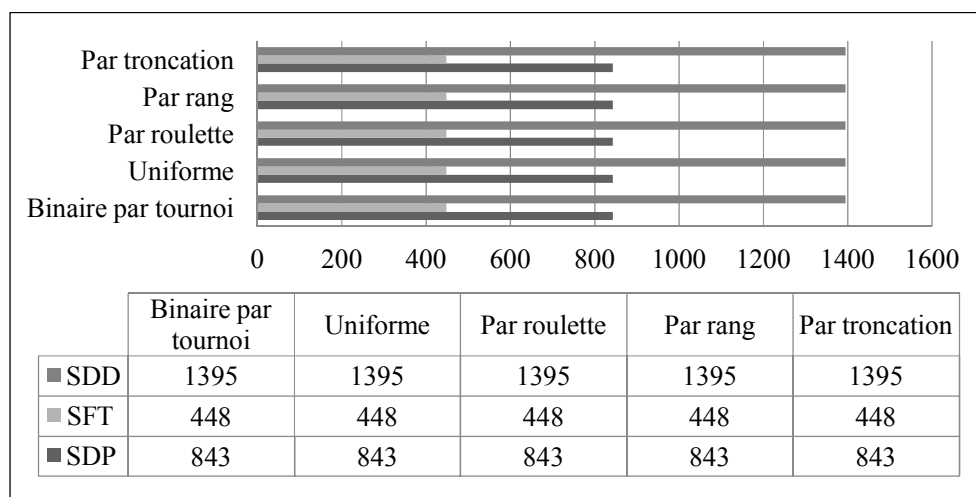


Figure 6.3 Comparaison des solutions avec une taille = 4 tâches / scénario test

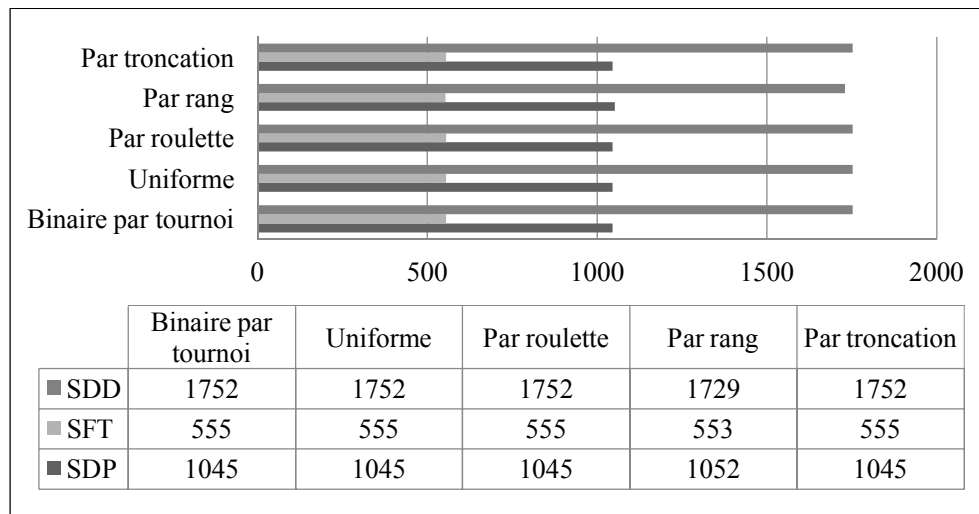


Figure 6.4 Comparaison des solutions avec une taille = 5 tâches / scénario test

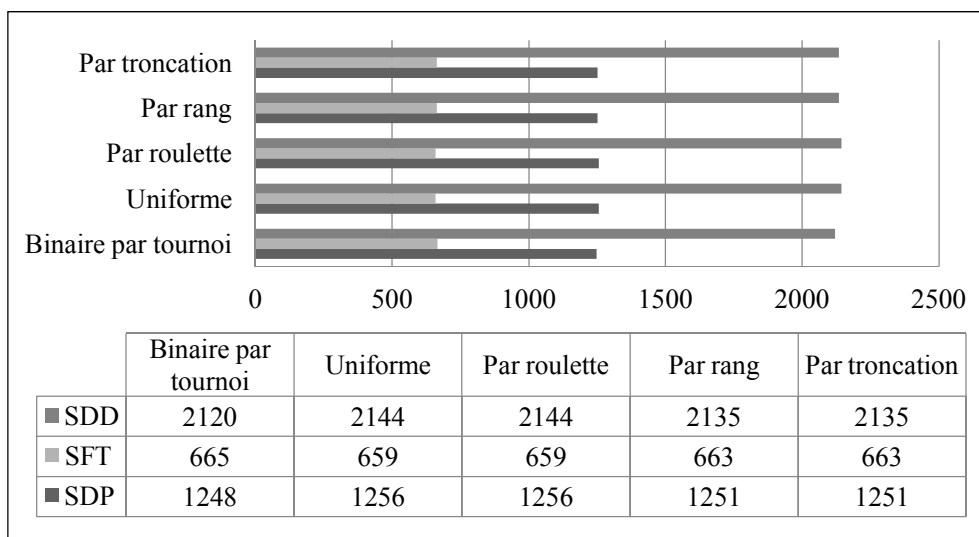


Figure 6.5 Comparaison des solutions avec une taille = 6 tâches / scénario test

Les auteurs ont opté pour l'utilisation de la fonction de Fitness comme critère de comparaison qualitative sur les résultats du tableau 6.6. Il semble que le croisement proposé dans cette étude donne les meilleurs résultats avec la technique de sélection binaire par tournoi (les solutions sélectionnées avec cette configuration, ont la valeur de Fitness la plus petite). Il s'avère que les résultats des différentes expériences (en utilisant les autres techniques de sélection) avec comme taille de conceptions de processus métiers égale à 4 tâches (Figure 6.3) ou 5 tâches (Figure 6.4) convergent vers les solutions (843, 448, 1395), (1045, 555, 1752) - pour

(SDP, SFT, SDD) -. A l'exception de l'expérimentation avec la sélection binaire par tournoi et une taille de conceptions de processus métiers égale à 6 tâches (Figure 6.5) ; donne le résultat le plus convaincant (1248, 665, 2120). Ce constat est dû en grande partie à la taille réduite de la librairie des tâches (20 services web) et à la taille des conceptions (entre 4 et 6 tâches), générées par le Framework. Il faut tenir compte du fait qu'après un nombre élevé d'itérations, le croisement sélectif devient destructif, par rapport aux solutions générées.

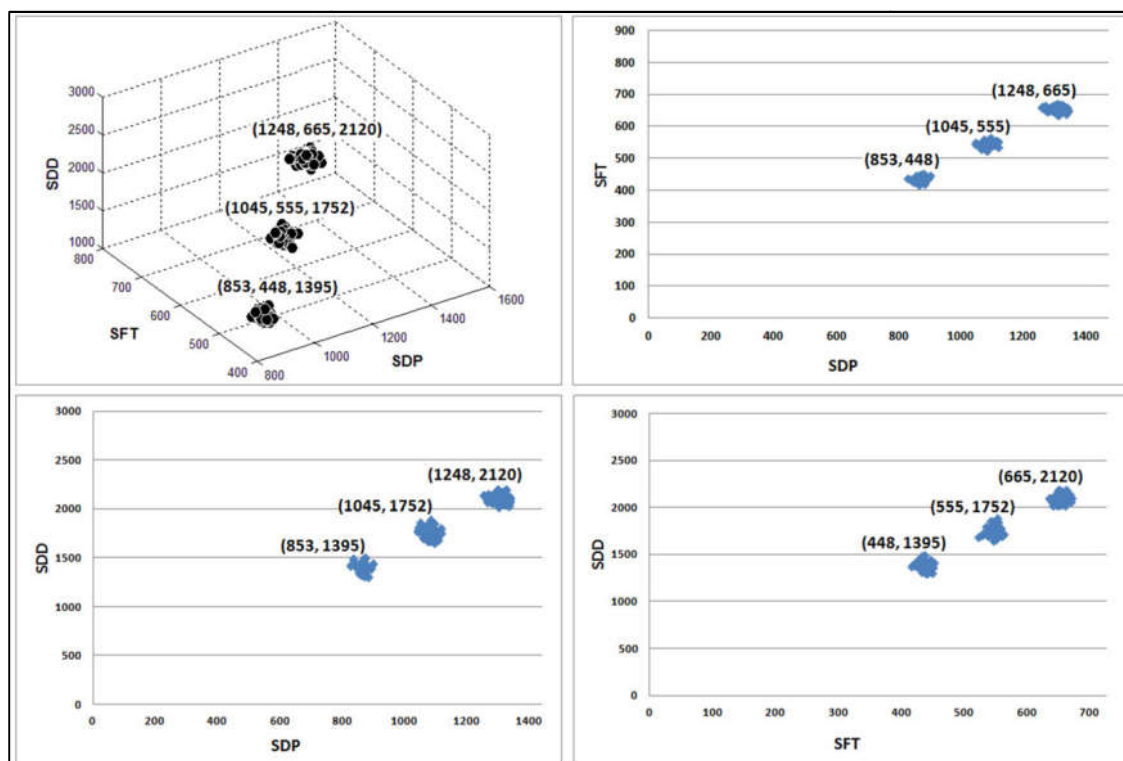


Figure 6.6 Résultats obtenus avec la sélection binaire par tournoi et le croisement proposé

La Figure 6.6 présente les solutions obtenues par le Framework d'optimisation avec la sélection binaire par tournoi, et le croisement proposé (utilisé dans *xNSGAI*) avec trois critères d'optimisation. Trois clusters de solutions sont générés.

Chaque cluster correspond à la taille moyenne d'une solution, i.e. la taille de chaque conception de processus métier faisable doit être comprise entre 4 et 6 tâches, ce qui donne un cluster par taille (4 tâches, 5 tâches 5 puis 6 tâches). Bien que les résultats semblent meilleurs que ceux des travaux de Vergidis,

il semble évident que l'opérateur de croisement autant que l'opérateur de sélection ont un effet (certes !) mais modéré sur la qualité des solutions générées. Ce qui fait la différence est la phase de composition et d'optimisation des nouvelles solutions, i.e. l'algorithme *ReProVA*. Ainsi, que la fonction de Fitness utilisée par *xNSGAI*, lors de l'évaluation et la sélection des solutions.

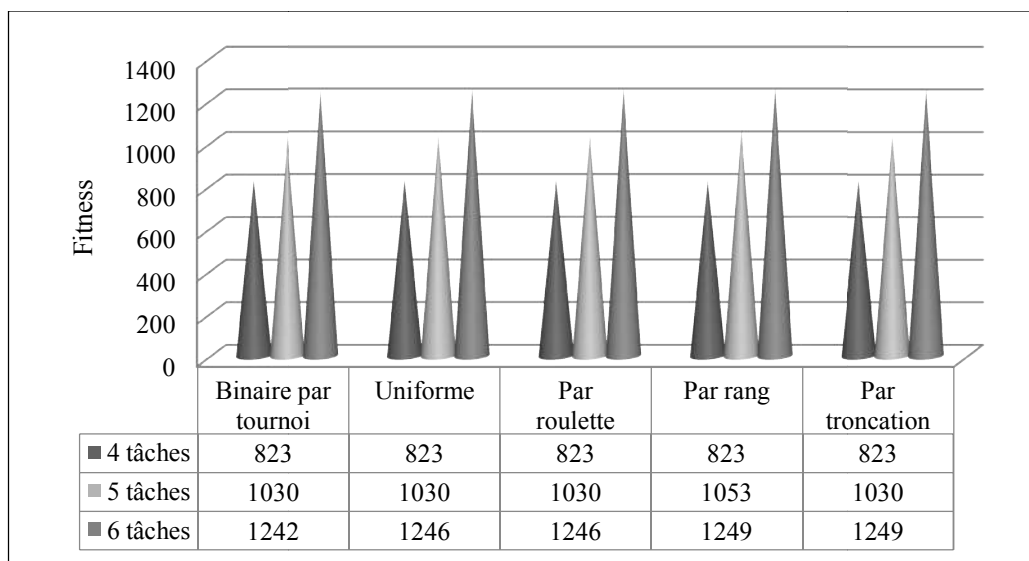


Figure 6.7 Comparaison des techniques de sélection employées avec *xNSGAI* selon la valeur de Fitness

La Figure 6.7 et Figure 6.8 comparent les résultats obtenus par le Framework d'optimisation multicritères proposé, par rapport aux valeurs de la fonction de Fitness, avec les différentes techniques de sélection utilisées. La Figure 6.7 présente les valeurs de Fitness des solutions optimales (de taille différente) obtenues par le Framework proposé en utilisant *xNSGAI* (i.e. le croisement sélectif avec la représentation en clés aléatoires), en fonction de techniques de sélection différentes. Les valeurs de Fitness pour les conceptions optimales sont sensiblement différentes, ce qui s'explique par l'effet relatif des opérateurs génétiques sur l'optimisation proposé. Ce que la Figure 6.8 - qui décrit l'évolution des valeurs de la Fitness des solutions toute taille confondue, en changeant la technique de sélection utilisée par *xNSGAI* - confirme, par la suite.

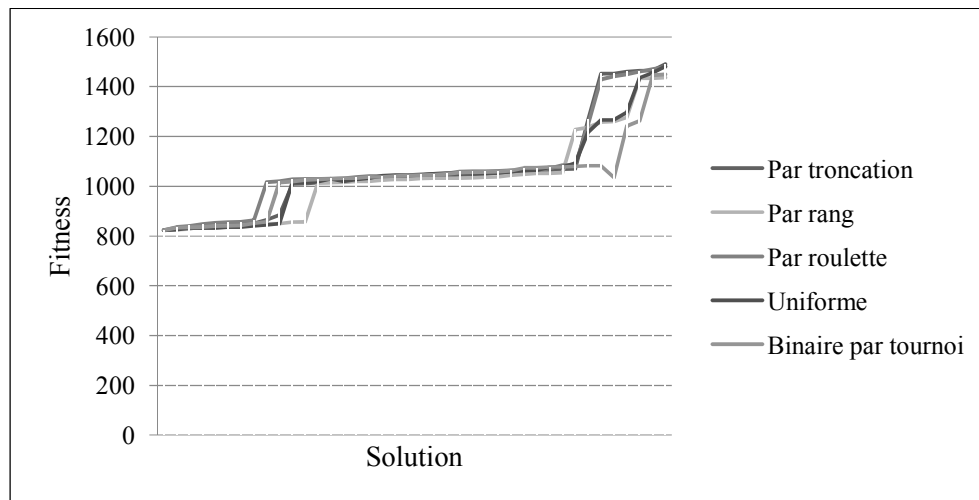


Figure 6.8 Evolution de la Fitness des solutions en fonction des techniques de sélection utilisées

Le tableau 6.7 montre les résultats de l'étude sur l'importance de la fonction de Fitness avec le Framework proposé, par rapport à une fonction de Fitness habituellement utilisée, dans le problème de l'optimisation multicritères des conceptions de processus métiers. Il serait bon de rappeler que les différentes études discutées (chapitre 3) utilisent ; pour leur majorité, une fonction de Fitness se basant sur l'agrégation des attributs d'une conception de processus métier.

		Agrégation		Fitness proposée	
		Temps d'exécution	Nombre de solutions	Temps d'exécution	Nombre de solutions
NSGA2	Par tournoi	6554	33	5258	37
	Uniforme	7023	26	5821	31
	Roulette	6936	30	5554	34
	Rang	6933	29	5577	35
	Troncation	6783	27	5788	34
xNSGA2	Par tournoi	6517	36	4610	40
	Uniforme	6902	30	4985	34
	Roulette	6758	31	4688	34
	Rang	6875	30	4623	35
	Troncation	6676	31	4671	35

Tableau 6.7 Améliorations apportées par la fonction de Fitness proposée

Sur le tableau 6.7 présente les résultats du Framework testé avec les différentes techniques de sélection présentées (chapitre 02), tout en variant l'algorithme évolutionnaire utilisé i.e. NSGAII, puis xNSGAII. Chaque algorithme génétique est testé avec une mutation à un point. Les techniques de croisements utilisés sont : Le croisement binaire simulé (avec NSGA2) et le croisement sélectif avec la représentation en clés aléatoires (pour le xNSGA2). Tous les algorithmes évolutionnaires effectuent 50 itérations, avec une population initiale de 100 individus, à chaque expérimentation. Le nombre peut sembler excessivement bas, mais des expériences initiales ont montré qu'il produisait des résultats de meilleure qualité en comparaison avec des nombres plus grands et en un temps raisonnable. Les paramètres restants sont ceux généralement utilisés dans notre problème (tableau 6.4). Pour chaque expérience, chaque algorithme est exécuté au nombre de 50 expérimentations indépendantes.

		Agrégation		Fitness proposée	
		Réduction Temps d'exécution (%)	Augmentation Nb de solutions (%)	Réduction Temps d'exécution (%)	Augmentation Nb de solutions (%)
NSGA2	Par tournoi	-	-	19,78	12,12
	Uniforme	-	-	17,12	19,23
	Roulette	-	-	19,93	13,33
	Rang	-	-	19,56	20,69
	Troncation	-	-	14,67	25,93
xNSGA2	Par tournoi	0,56	9,10	29,67	21,21
	Uniforme	1,72	15,40	29,02	30,77
	Roulette	2,57	3,33	32,41	13,33
	Rang	0,84	3,45	33,39	20,69
	Troncation	1,58	14,81	31,14	29,63

Tableau 6.8 Taux d'amélioration des résultats du Framework proposé

Les résultats du tableau 6.7 présentent les meilleurs résultats (après 20 expérimentations) d'une expérimentation, selon (i) le nombre de solutions non-dominées obtenues, (ii) le temps d'exécution par le Framework en milliseconde. Il

ressort en clairement, que l'utilisation de la fonction de Fitness proposée améliore le fonctionnement du Framework. Cette amélioration se fait progressivement.

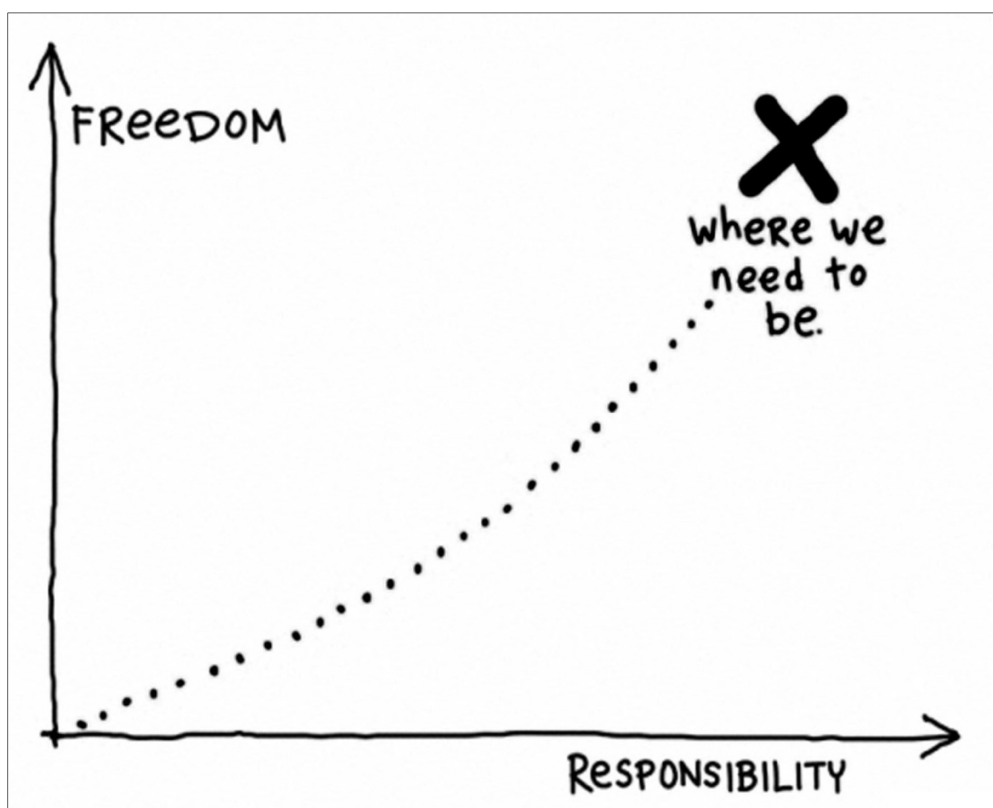
Les améliorations des résultats des expériences sont résumées dans le tableau 6.8 qui présente le taux de réduction du temps d'exécution, et le taux d'augmentation des solutions non dominées générées par le Framework proposé. Ces deux taux représentent les - indéniables - avantages d'utiliser une fonction de Fitness adaptée au problème d'optimisation multicritères des conceptions de processus métiers. En premier temps, l'utilisation de xNSGAI avec une fonction de Fitness à base d'agrégation d'attributs accroît le nombre de solutions non-dominées. Par la suite, l'utilisation du canonique NSGAI avec la fonction de Fitness proposée améliore ; à son tour, les résultats obtenus par xNSGAI, que ce soit vis-à-vis du temps d'exécution ou du nombre de solutions. Pour enfin arriver, au meilleur calcul de temps et de nombres de solutions, à l'aide de l'utilisation de la fonction de Fitness proposée avec xNSGAI.

Conclusion

Ce chapitre a présenté l'expérimentation du Framework d'optimisation évolutionnaire multicritères proposé, sur un scénario test tiré de la vie réelle, pour les conceptions de processus métiers. Le scénario test a comme contexte l'utilisation des services web comme composantes des conceptions de processus métiers.

L'expérimentation avec le Framework proposé a utilisé (i) la représentation des conceptions de processus métiers proposée (chapitre 4), (ii) l'algorithme évolutionnaire proposé *xNSGAI* (chapitre 5) avec différents techniques de sélection (chapitre 2). Et (iii) une proposition de fonction de Fitness, pour une optimisation multi objectifs (pour 03 objectifs). Les résultats de l'expérimentation ont démontré - clairement - la capacité du Framework proposé à automatiser le processus de composition, d'identification, de vérification et d'optimisation des conceptions de processus métiers avec des valeurs d'attributs optimisées.

Le chapitre suivant fournira une vue générale sur la recherche présentée, des critiques sur la contribution, les limitations constatées ainsi que de potentiels voies de recherche avenir.



Conclusion générale

Ce chapitre conclut notre travail de recherche avec une discussion sur l'ensemble du travail présenté. La discussion suivante porte sur les principales observations tout au long de cette recherche et identifie les principales contributions. Ce chapitre souligne également les limites de l'approche proposée en matière d'optimisation multicritères des conceptions de processus métiers avec des algorithmes évolutionnaires, et met en évidence les voies de recherche susceptibles de faire progresser la recherche dans ce domaine.

7.1. Principales observations

Cette section résume les principales observations de cette recherche faites durant ses principales étapes.

7.1.1. Etude bibliographique

L'étude de la littérature menée a porté sur la définition, la modélisation et l'optimisation des processus métiers. L'examen des approches de modélisation et d'optimisation a été fondé sur une classification proposée basée sur les types de techniques de modélisation des processus métiers (diagrammes, langages mathématiques et langages réservés aux processus métiers). La classification proposée a permis de mettre en évidence un certain nombre d'observations intéressantes, dont le manque d'approches d'optimisation multicritères des processus métier avec des algorithmes évolutionnaires.

L'optimisation des processus métiers avec les algorithmes évolutionnaires n'a pas reçu autant d'attention que les techniques de modélisation des processus métiers. Cette dernière a attiré l'attention de chercheurs de divers domaines qui ont abouti à diverses approches de modélisation. Chacune de ces approches de modélisation

présente des avantages incontestés, mais il manque encore une approche - globale - qui implique à la fois des constructions visuelles et mathématiques, pour capturer fidèlement une conception de processus métier. Malgré l'existence de nombreuses notations formelles de modélisation, la communauté travaillant sur les processus métiers utilise encore des techniques de modélisation dites schématique. Certes, simple à comprendre et à utiliser, mais qui manquent cruellement de potentiel pour l'analyse des performances et l'optimisation des processus métiers. Il faut souligner l'importance de l'analyse de la performance qui peut être directement utilisée pour l'aide à la décision et l'optimisation des processus métiers.

Cependant, il existe très peu de tentatives rapportées dans la littérature pour évaluer l'optimisation des conceptions de processus métiers suite à une implémentation de Framework utilisant un algorithme évolutionnaire. Certes, il y a eu quelques tentatives réussies signalées, mais elles sont assez compliquées et ne concernent le plus souvent que des processus métiers simples.

7.1.2. Représentation des processus métiers

La première étape vers l'élaboration d'un Framework pour l'optimisation multi objectifs des conceptions de processus métiers passe par la proposition d'une représentation pour ces conceptions. Cette représentation permet de capturer, visualiser et quantifier toute nouvelle conception de processus métier. Les tâches et les ressources qui composent une conception, et les attributs utilisés par ses tâches sont les éléments traités par la représentation proposée. Le but de la représentation proposée est de rendre une conception de processus métier accessible à toute tentative d'optimisation par/avec un algorithme évolutionnaire.

Le premier objectif de la représentation proposée est de fournir des moyens visuels, pour décrire une conception de processus métier. Cet objectif est couvert par la perspective visuelle de la représentation (un simple organigramme) qui fournit une représentation schématique du processus métier. Le deuxième objectif est de permettre de capturer mathématiquement et d'exprimer les mêmes éléments de la conception capturés par la perspective visuelle. La perspective quantitative

exprime tous les éléments d'une conception de processus métier, en utilisant des constructions mathématiques et introduit l'utilisation de deux (02) matrices pour (i) capturer (matrice tâches/ressources) et (ii) composer et évaluer (la librairie des tâches) toute conception de processus métier. La librairie des tâches permet le calcul des valeurs des attributs (objectifs d'optimisation) des conceptions de processus métiers, afin de les évaluer puis de les comparer. Ainsi, le troisième objectif de la représentation est satisfait. Le quatrième objectif de la représentation proposée concerne la composition (ou génération) et l'évaluation des conceptions alternatives de processus métiers. Cet objectif peut être réalisé avec l'algorithme *ReProVA*. Avec cet algorithme, une conception de processus métier sous la forme d'une perspective quantitative peut être (i) transformée en perspective visuelle équivalente et (ii) évaluée en fonction qu'elle soit acceptée comme une conception réalisable (selon un nombre de contraintes).

7.1.3. Framework pour l'optimisation multicritères des processus métiers

La présente recherche aboutit à l'implémentation d'un Framework capable de générer et d'évaluer des conceptions de processus métiers, dans un environnement d'optimisation multi objectifs, avec un algorithme évolutionnaire. Le Framework comprend deux composantes principales (i) la représentation proposée pour une conception de processus métier et (ii) l'algorithme évolutionnaire *xNSGAI* comme technique d'optimisation multicritères utilisée. Le Framework utilise la représentation proposée, pour répondre aux différents défis induit par l'optimisation multicritères. Le Framework (avec l'aide de *ReProVA*) est capable de générer des solutions, de les vérifier et de les mettre à jour (si nécessaire), ce qui garantit la découverte et la faisabilité des conceptions de processus métiers.

L'algorithme évolutionnaire employé par le Framework est également un défi. La nature discrète du problème en conjonction avec sa formulation multicritères peut rendre le processus de la découverte de solutions réalisables belliqueux, pour l'algorithme employé.

7.1.4. Algorithme évolutionnaire proposé

Le fonctionnement du Framework, pour générer des conceptions de processus métiers faisables et optimales, implique l'utilisation d'une structure capable de générer une série de conceptions alternatives de processus métiers. Le bon fonctionnement du Framework proposé implique l'utilisation de l'algorithme évolutionnaire le plus indiqué au vue de l'optimisation multicritères présentée. L'étude bibliographique a clairement montrée que, l'algorithme génétique NSGAI soit le plus utilisé et le mieux exploité dans un problème d'optimisation de conceptions processus métiers. Afin de tester systématiquement le Framework, une proposition d'utiliser une version modifiée a été adoptée. Cette proposition a donné lieu à *xNSGAI*.

L'algorithme évolutionnaire *xNSGAI* reprend le principe de tri non dominé de son aïeul, en utilisant une fonction de Fitness ; autre qu'une fonction d'agrégation des attributs d'une conception de processus métier. Cette fonction de Fitness est définie par un modèle mathématiques capable de traiter jusqu'à (03) critères d'optimisation. Ce qui permet - par la suite - de rendre des résultats de meilleures qualités que ceux obtenus avec un canonique NSGAI. Entre autre, *xNSGAI* utilise des opérateurs génétiques différents de ceux de NSGAI. L'algorithme évolutionnaire utilisé avec le Framework fait appel à une série de techniques de sélection tirée de la littérature (la sélection uniforme, par roulette, par rang et par troncation), en plus de la sélection binaire par tournoi, préférée par NSGAI. Autre spécificité de *xNSGAI*, est l'intégration d'un croisement hybride séquentielle. Pour ce faire, cet opérateur génétique utilise la représentation en clés aléatoires des tâches d'une conception d'un processus métier, avec le croisement sélectif du résultat. Par la suite, le Framework est testé avec un scénario de test particulier qui donnera suite à une série de remarques pertinentes.

7.1.5. Validation avec un scénario de processus métier réel

Le Framework proposé a été testé avec un scénario de processus métier, tiré de la vie réelle. L'objectif de la validation est de démontrer la polyvalence du

Framework d'optimisation et de mettre en évidence sa force, autant que sa faiblesse, lorsqu'il est appliqué à un exemple de processus métier qui englobe des éléments réels. Dans le scénario test, la librairie des tâches consistait en des services web réels, avec un ensemble de ressources limité. Lors de l'expérimentation, des conceptions optimales de processus métiers ont été générées, par le Framework. Ce qui a pu démontrer la capacité du Framework à automatiser et optimiser des conceptions, pour la recherche proposée. Les performances de l'algorithme évolutionnaire proposé employé - *xNSGAI* -, dans le Framework, sont jugées très bonnes. Avec les différentes configurations de *xNSGAI*, les résultats obtenus surpassés ceux publiés par les travaux de Vergidis. Cela démontre clairement que le Framework - avec son algorithme évolutionnaire - peut être utilisé avec confiance et fournir des résultats de qualité, avec un scénario test comprenant des éléments réels (e.g. de petits procédés).

Selon les auteurs, le principal résultat de la recherche proposée est qu'elle constitue une continuation du changement - ce que les travaux de Hofacker et Vergidis ont entamé - dans le domaine de l'amélioration et l'automatisation des processus métiers, en entreprise. Cependant, il y a beaucoup de voies à suivre, de recherches à commencer et de fonctionnalités à implémenter, dans le domaine de l'optimisation multicritères des conceptions de processus métiers.

7.2. Limites de la recherche

Bien que cette recherche ait abouti à des résultats améliorant ce qui a été fait par le passé, ce travail a aussi quelques limitations, comme avec toute autre recherche. Les sections suivantes regroupent et identifient certaines de ces limites.

7.2.1. Limites de la méthodologie de la recherche

Les principales limites de la méthodologie utilisée dans cette recherche sont les suivantes :

- L'étude de la littérature a identifié un certain nombre de questions liées à la définition, la modélisation, l'analyse et l'optimisation des processus métiers.

Cependant, cette recherche est principalement axée sur l'optimisation des conceptions de processus métiers et, sur la base de cette orientation, elle suggère une approche de spécification et de modélisation des "conceptions" de processus métiers.

- L'approche d'optimisation utilisée dans le Framework proposé repose sur un algorithme évolutionnaire d'optimisation multi objectifs nouvellement proposé et n'a pas exploré ou envisagé d'autres techniques d'optimisation. Ceci est du à la formulation multi objectifs du problème d'optimisation des conceptions de processus métiers et la nécessité de faire évoluer les solutions irréalisables vers des solutions réalisables (ou faisables).
- L'algorithme évolutionnaire proposé a été adapté de NSGAI, l'algorithme le plus cité, référencé et utilisé dans un problème d'optimisation multi objectifs de processus métiers. Ce qui a permis d'améliorer les résultats des travaux pris comme références, pour cette recherche.
- L'étude de cas réel qui est utilisée dans cette recherche est empruntée à la littérature. Cela a permis de mieux comprendre comment intégrer et tester un processus réel de l'industrie des services au Framework d'optimisation proposé.

7.2.2. Limites de la représentation proposée

Voici les principales limites constatées avec la représentation proposée :

- Les éléments sélectionnés dans la représentation proposée d'une conception de processus métier sont un sous-ensemble du schéma global d'un processus métier. Les acteurs, par exemple, ne sont pas impliqués dans la représentation proposée, car cette recherche est orientée vers l'optimisation de la conception des processus métiers, et non pour l'élaboration des processus métiers, à proprement parler.
- La représentation proposée suppose le concept de librairie des tâches, une forme de référentiel de tâches pouvant être potentiellement utilisées pour la création de conceptions de processus métiers. Pour un processus métier tiré de la vie réelle,

il se peut qu'une librairie des tâches ne soit pas facilement envisageable et/ou disponible.

- Les valeurs des attributs tâche / conception sont statiques, par opposition à une distribution dynamiques. Bien que cela facilite le calcul des objectifs et l'évaluation des conceptions de processus métiers, ces valeurs peuvent changer dynamiquement pendant l'exécution, dans un contexte réel.

7.2.3. Limites du Framework d'optimisation proposé

Les principales limites du Framework d'optimisation proposé sont les suivantes :

- Le Framework proposé est étroitement lié à la représentation. Cela signifie que seul une conception de processus métier exprimée dans la représentation proposée peut être soumise à une optimisation évolutionnaire multicritères (tel qu'expliqué dans cette recherche).
- Le Framework d'optimisation proposé permet une optimisation multicritères des conceptions de processus métiers. Cependant, dans toutes les expériences présentées, le Framework n'est testé que pour trois (03) objectifs. Même si la pratique courante dans l'évaluation des problèmes d'optimisation multicritères soit réalisée avec deux (02) objectifs. Il serait intéressant d'évaluer le fonctionnement du Framework avec plus d'objectifs, même si cela implique une optimisation *many* objectifs.
- L'approche d'optimisation utilisée par le Framework proposé est basée ; exclusivement, sur un algorithme évolutionnaire, ce qui indique - clairement - l'influence que subit le Framework par ce choix.
- La proposition de revoir les opérateurs génétiques employés par l'algorithme évolutionnaire utilisé par le Framework proposé porte sur la sélection et le croisement. La mutation étant laissée pour compte, vu le nombre limité de recherche sur le sujet. Des tests approfondis en utilisant des variantes de cet opérateur pourraient inspirer de recherches futures.

7.3. Perspectives et recherches futures

Malgré le potentiel d'une approche d'optimisation multicritères pour les conceptions de processus métiers et les avantages qu'elle pourrait offrir au secteur des services, par exemple, cette recherche reste dans le cadre de la théorie. Afin d'y remédier, des recherches supplémentaires sont nécessaires, pour l'appliquer dans un contexte réel (et exploitable) de l'industrie.

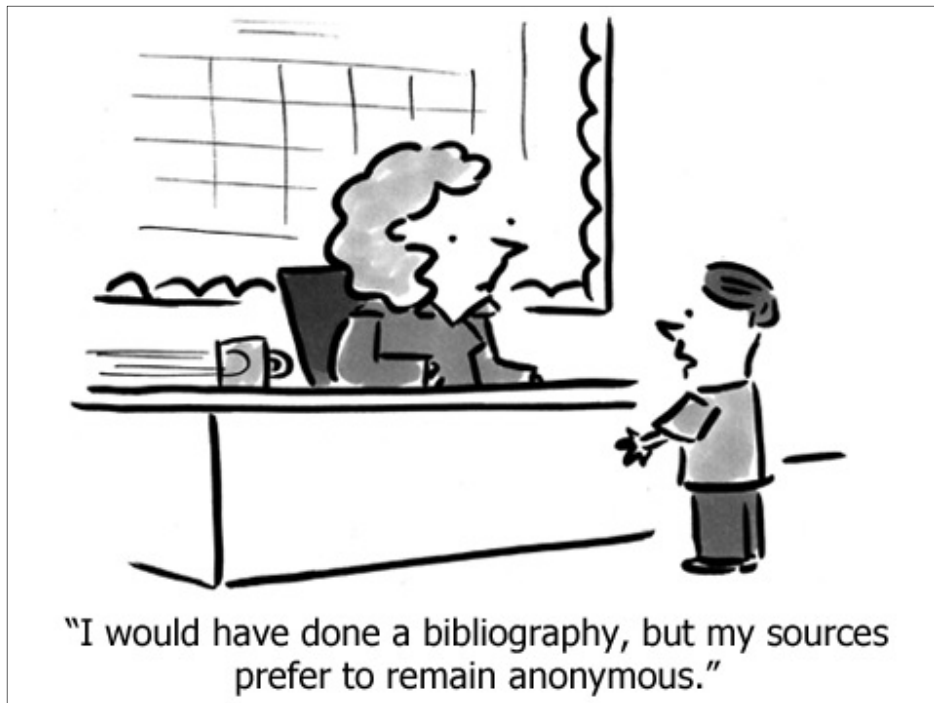
La recherche proposée suggère un Framework pour l'optimisation multicritères des conceptions de processus métiers en utilisant un - nouvel - algorithme évolutionnaire. Avec le Framework proposé, cette recherche a proposé une spécification et une représentation des conceptions de processus métiers. Il est nécessaire d'étendre cette représentation et d'améliorer le Framework, afin de mieux répondre aux exigences de l'optimisation des processus métiers tirés de la vie réelle.

Les activités de recherche pour l'élaboration future de la spécification de processus métiers proposée et du Framework d'optimisation peuvent être résumées comme suit :

- Inclure d'autres éléments dans la spécification d'une conception de processus métier (i) faire participer les acteurs responsables de la mise en œuvre d'une tâche / conception. Cela va beaucoup aider à obtenir une exécution plus fiable - réaliste - des conceptions de processus métiers au sein de l'entreprise.
- Les attributs des tâches stockés dans la librairie des tâches peuvent être dynamiques, au lieu de statiques. Ils peuvent également être des valeurs réelles au lieu d'être estimées.
- Ajouter le contexte qualité aux attributs de la tâche / conception peut également être introduits, pour une représentation plus précise des caractéristiques d'une conception du processus métier.

- Revoir la structure de la fonction de Fitness, et l'étendre à un nombre plus important d'objectifs d'optimisation, participera efficacement à l'évaluation des conceptions.
- Tester le Framework d'optimisation proposé avec plus de critères d'optimisation.
- Le Framework pourrait intégrer la rétroaction de l'exécution (e.g. l'historique des données) des conceptions de processus métiers, afin d'affiner et/ou optimiser les conceptions façonnées.

L'optimisation des conceptions de processus métiers a une croissance potentielle avec un avantage direct pour la communauté des processus métiers, et il y a encore un nombre ; non négligeable, de voies de recherche à aborder.



Bibliographie

- Afflerbach, P., Hohendorf, M., & Manderscheid, J. (2016), Design It Like Darwin- A Value-Based Application Of Evolutionary Algorithms for Proper and Unambiguous Business Process Redesign, *Information Systems Frontiers*, pp. 1-21.
- Agerfalk, P., Goldkuhl, G. and Cronholm, S. (1999), Information Systems Actability Engineering - Integrating Analysis of Business Process and Usability Requirements, in *Proceedings of the 4th International Workshop on the Language Action Perspective on Communication Modelling* Copenhagen, Denmark, pp. 1245-1252.
- Aguilar-Saven, R. S. (2004), Business Process Modelling: Review And Framework, *International Journal of production economics*, Vol. 90(2), pp. 129-149.
- Ahmadikatouli, A., & Aboutalebi, M. (2011), New Evolutionary Approach to Business Process Model Optimization, In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 2.
- Al Jadaan, O., Rajamani, L., & Rao, C. R. (2008), Improved Selection Operator For Ga, *Journal of Theoretical & Applied Information Technology*, Vol. 4(4).
- Angeline, P. J., & Pollack, J. B. (1993). Competitive Environments Evolve Better Solutions for Complex Tasks. In *ICGA*, pp. 264-270.
- Asco, A. (2013), *Constructive And Evolutionary Algorithms For Airport Baggage Sorting Station And Gate Assignment Problems* (Doctoral dissertation, University of Nottingham).
- Bäck, T., Hoffmeister, F., & Schwefel, H. P. (1991). A Survey Of Evolution Strategies, In *Proceedings of the 4th international conference on genetic algorithms*, pp. 2-9.

- Bäck, T. (1992), The Interaction of Mutation Rate, Selection, and Self-Adaptation Within a Genetic Algorithm. In *PPSN*, pp. 87-96.
- Bäck, T. (1996), *Evolutionary Algorithms In Theory And Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford university press.
- Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997), Handbook Of Evolutionary Computation. *Release, 97(1)*, B1.
- Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (2000), *Evolutionary computation 1: basic algorithms and operators* (Vol. 1). CRC Press.
- Bae, H., Lee, S., & Moon, I. (2014), Planning Of Business Process Execution In Business Process Management Environments, *Information Sciences*, Vol. 268, pp. 357-369.
- Baker, J. E. (1985), Adaptive Selection Methods For Genetic Algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications* (pp. 101-111).
- Baluja, S. (1994). *Population-Based Incremental Learning. A Method For Integrating Genetic Search Based Function Optimization And Competitive Learning* (No. CMU-CS-94-163). Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science.
- Banzhaf, W. (2013), *Engineered Biomimicry: Chapter 17. Evolutionary Computation and Genetic Programming*. Elsevier Inc. Chapters.
- Barichard, V. (2003), *Approches Hybrides pour les Problèmes Multi Objectifs* (Doctoral dissertation, Université d'Angers).
- Basu, A., & Blanning, R. W. (2000), A Formal Approach to Workflow Analysis. *Information Systems Research*, Vol. 11(1), pp. 17-36.
- Bean, J. C. (1994), Genetic algorithms and random keys for sequencing and optimization, *ORSA journal on computing*, Vol. 6(2), pp. 154-160.

- Belaidi, I., Mohammedi, K., & Brachemi, B. (2009), Mise en Ouvre de l'Algorithme NSGAII pour l'Optimisation Multi-Objectif des Paramètres de Coupe en Fraisage en Bout, 19^{ème} Congrès Français de Mécanique, I-revue. *Edit.: AFM-Maison de la mécanique-Courbevoie, Paris.*
- Blickle, T., & Thiele, L. (1995), A comparison of selection schemes used in genetic algorithms. *TIK-Report, Zurich.*
- Brindle, A. (1981), *Genetic algorithms for function optimization*. PhD thesis, University of Alberta.
- Busetti, F. (2007), Genetic algorithms overview. *Retrieved on December, 1.*
- Castellanos, M., Casati, F., Umeshwar, D. and Ming-Chien, S. (2004), A Comprehensive and Automated Approach to Intelligent Business Processes Execution Analysis, Distributed and Parallel Databases, Vol. 16, pp. 1-35.
- Chapin, N. (1971), *Flowcharts*, Auerbach Publishers, Princeton.
- Coello, C. A. C. (1996), *An empirical study of evolutionary techniques for multiobjective optimization in engineering design* (Doctoral dissertation, PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA).
- Coello Coello, C.A. (2000), Handling Preferences in Evolutionary Multi-objective Optimization: A Survey, in *Proceedings of the Congress on Evolutionary Computation (CEC'2000)* San Diego, USA, pp. 30-37.
- Coello Coello, C.A. (2005), Recent Trends in Evolutionary Multi-objective Optimization, in Abraham, A., Jain, L. and Goldberg, R. (editors), *Evolutionary Multi-objective Optimization: Theoretical Advances and Applications*, Springer-Verlag, London, pp. 7-53.
- Cohan, F. M. (1984), Can Uniform Selection Retard Random Genetic Divergence Between Isolated Conspecific Populations?. *Evolution*, pp. 495-504.
- Cramer, N. L. (1985), A representation for the adaptive generation of simple sequential programs. In *Proceedings of the First International Conference on Genetic Algorithms*, pp. 183-187.

- Davis, L. (1991), *Handbook of genetic algorithms*, New York, USA: Van Nostrand Reinhold.
- Duvivier, D. (2000), Etude de l'hybridation des méta-heuristiques, application à un problème d'ordonnancement de type jobshop, Université du Littoral Côte d'Opale.
- Darwin C. (1929), *The Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*. The Book League of America. Originally published in 1859.
- Davenport, T.H. and Short, J.E. (1990), The New Industrial Engineering: Information Technology and Business Process Redesign, *Sloan Management Review*, pp. 11-27.
- Davenport, T.H. (1993), *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press , Boston.
- Davis, L.H. (1991), *Handbook Of Genetic Algorithms*, Van Nostrand Reinhold.
- Dayal, U., Hsu, M., & Ladin, R. (2001). *Business Process Coordination: State of the Art, Trends, and Open Issues*. In *VLDB* (Vol. 1, pp. 3-13).
- Deb, K., & Agrawal, R. B. (1994), Simulated Binary Crossover For Continuous Search Space. *Complex Systems*, 9(3), 1-15.
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm For Multi-Objective Optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature* (pp. 849-858). Springer Berlin Heidelberg.
- Deb, K., & Beyer, H. G. (2001), Self-Adaptive Genetic Algorithms With Simulated Binary Crossover. *Evolutionary Computation*, 9(2), 197-221.
- Deb, K. (2001), *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York.

- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002), A Fast And Elitist Multi Objective Genetic Algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.
- DeJong, K. (1975), An Analysis Of The Behavior Of A Class Of Genetic Adaptive Systems, *PhD. Thesis, University of Michigan*.
- Diaz-Gomez, P. A., & Hougen, D. F. (2007), Initial Population for Genetic Algorithms: A Metric Approach. In *GEM* (pp. 43-49).
- Djedović, A., Žunić, E., Avdagić, Z., & Karabegović, A. (2016), Optimization Of Business Processes By Automatic Reallocation Of Resources Using The Genetic Algorithm, In *Telecommunications (BIHTEL), 2016 XI International Symposium on*, pp. 1-7, IEEE.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013), *Fundamentals Of Business Process Management*, Vol. 1, pp. 2, Heidelberg: Springer.
- Eiben, A.E., & Smith, J.E. (2003), *Introduction To Evolutionary Computing*. Berlin.
- Ehrlich, P., & Raven, P. (1969), Differentiation Of Populations, *New Series, Vol. 165, No. 3899*, pp. 1228-1232.
- Ehrlich, P., & Raven, P. (1992). Differentiation Of Populations. *The Units of Evolution. Essays on the Nature of Species. MIT Press, Cambridge, Massachusetts*, pp. 57-68.
- Fallahpour, M. B., Hemmati, K. D., & Pourmohammad, A. (2012), Optimization of a LNA Using Genetic Algorithm. *Electrical and Electronic Engineering*, 2(2), 38-42. Springer.
- Fan, Y.S. (2001), *Fundamental of Workflow Management Technology*, Springer-Verlag, New York.
- Fan, W., Xu, H., & Xu, X. (2009), Simulation On Vehicle Routing Problems In Logistics Distribution, *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, Vol. 28(6), pp. 1516-1531.

- Farina, M., & Amato, P. (2004), A Fuzzy Definition of "Optimality" for Many-Criteria Optimization Problems, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 34(3), pp. 315-326.
- Farsani, S.T., Aboutalebi, M., Motameni, H. (2013), Customizing NSGAI to Optimize Business Processes Designs, *Research Journal of Recent Sciences*, Vol.2, pp. 74-79.
- Filipović, V. (2012), Fine-grained tournament selection operator in genetic algorithms, *Computing and Informatics*, Vol. 22(2), pp. 143-161.
- Fogel, L. J. (1962). Autonomous automata. *Industrial Research*, 4(2), 14-19.
- Fogel L. J., Owens A. J., & Walsh M. J. (1966), *Artificial Intelligence through Simulated Evolution*, Wiley, New York.
- Fogel, D. B. (1992). Evolving artificial intelligence. *PhD thesis*, University of California, San diego, CA.
- Fogel, D. B. (1995), *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (Piscataway, NJ: IEEE).
- Fonseca, C. M., & Fleming, P. J. (1995), An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 3(1), 1-16.
- Gen, M., & Cheng, R. (1996). *Genetic Algorithms and Manufacturing Systems Design*. New York: John Wiley & Sons, Inc.
- Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization: Multiobjective genetic algorithm approach*. Springer Science & Business Media.
- Georgoulakos, K., Vergidis, K., Tsakalidis, G., & Samaras, N. (2017, June). Evolutionary Multi-Objective Optimization of business process designs with pre-processing. In *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 897-904). IEEE.
- Golchha, A. and Qureshi, S. G. (2015), *Non-Dominated Sorting Genetic Algorithm-II-A Succinct Survey*. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (1), pp. 252-255

- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Mass: Addison-Wesley Publishing Co.
- Goldberg, D. E., & Deb, K. (1991), A comparative analysis of selection schemes used in genetic algorithms, *Foundations of genetic algorithms*, Vol. 1, pp. 69-93.
- Goldstein, D. B., & Holsinger, K. E. (1992), Maintenance of polygenic variation in spatially structured populations: roles for local mating and genetic redundancy. *Evolution*, pp. 412-429.
- Gonçalves, J. F., & Resende, M. G. (2011), Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, Vol. 17(5), pp. 487-525.
- Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M. and Shan, M.-C. (2004), Business Process Intelligence, Computers in Industry, Vol. 53, pp. 321-343
- Grigori, D., Casati, F., Dayal, U. and Shan, M.-C. (2001), Improving Business Process Quality through Exception Understanding, Prediction and Prevention, in Proceedings of the 27th Very Large Data Base (VLDB) Conference, Los Angeles, pp. 872-888.
- Gu, X., Sun, G., Li, G., Mao, L., & Li, Q. (2013), A comparative study on multi objective reliable and robust optimization for crashworthiness design of vehicle structure, *Structural and Multidisciplinary Optimization*, Vol. 48(3), pp. 669-684.
- Gunasekaran, A. and Kobu, B. (2002), Modelling and Analysis of Business Process Reengineering, *International Journal of Production Research*, Vol. 40, No. 11, pp. 2521-2546.
- Hao, X., & Gen, M. (2011), Multi-objective Job Shop Rescheduling with Evolutionary Algorithm, *IEEJ Transactions on Electronics, Information and Systems*, Vol. 131, pp. 674-681.
- Haupt, R.L., Haupt S.E. (2004), *Practical Genetic Algorithms, 2nd Edition*. Wiley, ISBN 0-471-45565-2, Hoboken, N.Y.

- Hammer, M. and Champy, J. (1993), *Reengineering the Corporation: a Manifesto for Business Revolution*, N. Brealey, London.
- Havey, M. (2005), *Essential Business Process Modelling*, O'Reilly, U.S.A.
- Hofacker, I., & Vetschera, R. (2001), Algorithmical approaches to business process design. *Computers & Operations Research*, Vol. 28(13), pp. 1253-1275.
- Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*. Ann Harbor: University of Michigan Press.
- Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, Massachusetts: MIT Press.
- Ilgin, M. A., & Tunali, S. (2007). Joint optimization of spare parts inventory and maintenance policies using genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, 34(5-6), 594-604.
- Irani, Z., Hlupic, V. and Giaglis, G.M. (2002), Business Process Reengineering: An Analysis Perspective, *International Journal of Flexible Manufacturing Systems*, Vol. 14, pp. 5-10.
- Jacobson, I., Ericsson, M., & Jacobson, A. (1994). The Object Advantage-Business Process Reengineering With Object Technology.
- Jaszkiewicz, A. (2001). *Multiple objective metaheuristic algorithms for combinatorial optimization*. Wydawn. Politechniki Poznańskiej.
- Johanson, H.J., McHugh, P., Pendlebury, A.J. and Wheeler, W.A.I. (1993), *Business Process Reengineering - Breakpoint Strategies for Market Dominance*, Wiley, Chichester.
- Jones, G. (1998), Genetic and evolutionary algorithms. *Encyclopedia of Computational Chemistry*, 2, 1127-1136.
- Julstrom, B. A. (1999), It's all the same to me: Revisiting rank-based probabilities and tournaments. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, Vol. 2*.

- Kern, S., Müller, S. D., Hansen, N., Büche, D., Ocenasek, J., & Koumoutsakos, P. (2004). Learning probability distributions in continuous evolutionary algorithms—a comparative review. *Natural Computing*, 3(1), 77-112.
- Kettinger, W. J., Teng, J. T., & Guha, S. (1997), Business process change: a study of methodologies, techniques, and tools, *MIS quarterly*, pp. 55-80.
- Khare, V., Yao, X., & Deb, K. (2003), Performance scaling of multi-objective evolutionary algorithms, In *International Conference on Evolutionary Multi-Criterion Optimization*, Springer Berlin Heidelberg, pp. 376-390.
- Knuth, D. E. (1963), Computer-drawn flowcharts, *Communications of the ACM*, Vol. 6(9), pp. 555-563.
- Koubarakis, M., & Plexousakis, D. (2002), A formal framework for business process modeling and design, *Information systems*, Vol. 27(5), pp. 299-319.
- Koza, J. R. (1992), Genetic programming: on the programming of computers by means of natural selection, Vol. 1, MIT press.
- Koza, J. R. (1994), Genetic programming II: Automatic discovery of reusable subprograms. *Cambridge, MA, USA*.
- Koza, J. R., Bennet, F. H., Andre, D., & Keane, M. A. (1999), Genetic programming III: Automatic synthesis of analog circuits. *MIT Press, Massachusetts*.
- Kumar, R. (2012), Blending Roulette Wheel Selection & Rank Selection In Genetic Algorithms. *International Journal of Machine Learning and Computing*, Vol. 2(4), pp. 365.
- Laguna, M., & Marklund, J. (2013), *Business process modeling, simulation and design*. CRC Press.
- Lakin, R., Capon, N., & Botten, N. (1996). BPR enabling software for the financial services industry. *Management services*, 40(3), 18-20.
- Le Riche, R., Schoenauer, M., & Sebag, M. (2007). Un état des lieux de l'optimisation évolutionnaire et de ses implications en sciences pour l'ingénieur.

- Modélisation numérique. 2, Défis et perspectives. Traité MIM, série Méthodes numériques et éléments finis*, 187-259.
- Levin, D. A. (1979), The nature of plant species. *Science*, Vol. 204, No. 4391, pp. 381-384.
- Lindsay, A., Downs, D. and Lunn, K. (2003), Business Processes - Attempts to Find a Definition, *Information and Software Technology*, Vol. 45, pp. 1015-1019.
- Liu M., & Hu H. (2013), Optimization of Image Zernike Moments Shape Feature Based on Evolutionary Computation, *Intelligent Image and Video Interpretation: Algorithms and Applications*, pp. 199-216.
- Machado, P., Martins, T., Amaro, H., & Abreu, P. H. (2014), An Interface for Fitness Function Design, In *International Conference on Evolutionary and Biologically Inspired Music and Art*, pp. 13-25, Springer Berlin Heidelberg.
- Mahammed, N., & Benslimane, S. M. (2016, September). Toward Multi Criteria Optimization of Business Processes Design. In *International Conference on Model and Data Engineering - MEDI 2016* (pp. 98-107). Springer International Publishing.
- Martín-Palma, R. J., & Lakhtakia, A. (2013). Engineered Biomimicry For Harvesting Solar Energy: A Bird's Eye View. *International Journal of Smart and Nano Materials*, 4(2), 83-90.
- Mason, S. J., Kurz, M. E., Pfund, M. E., Fowler, J. W., & Pohl, L. M. (2007), Multi-Objective Semiconductor Manufacturing Scheduling: A Random Keys Implementation Of NSGA-II, In *2007 IEEE Symposium on Computational Intelligence in Scheduling*, pp. 159-164.
- McKeague, C. (1983). *Intermediate Algebra with Trigonometry*. Academic Press, 565-596.
- Melao, N., & Pidd, M. (2000), A Conceptual Framework For Understanding Business Processes And Business Process Modeling, *Information systems journal*, Vol. 10(2), pp. 105-129.

- Merino, L. B., & Elguezabal, G. B. (2005). Business Process Definition Languages Versus Traditional Methods Towards Interoperability. In *International Conference on COTS-Based Software Systems* (pp. 25-35). Springer Berlin Heidelberg.
- Michalewicz, Z. (1994). Gas: What Are They?. In *Genetic algorithms+ data structures= evolution programs* (pp. 13-30). Springer Berlin Heidelberg.
- Michalewicz, Z., & Hartley, S. J. (1996), Genetic Algorithms + Data Structures = Evolution Programs. *Mathematical Intelligencer*, Vol. 18(3).
- Mili, H., Tremblay, G., Jaoude, G. B., Lefebvre, E., Elabed, L., & Boussaidi, G. E. (2010), Business process modeling languages: Sorting through the alphabet soup. *ACM Computing Surveys (CSUR)*, 43(1), 4.
- Miller, B. L., & Goldberg, D. E. (1995). Genetic Algorithms, Tournament Selection, And The Effects of Noise. *Complex systems*, 9(3), 193-212.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Molka, T., Redlich, D., Gilani, W., Zeng, X. J., & Drobek, M. (2015), Evolutionary Computation Based Discovery Of Hierarchical Business Process Models, In *International Conference on Business Information Systems*, pp. 191-204. Springer International Publishing.
- Mühlenbein, H. & Schlierkamp-Voosen, D. (1993), Predictive Models For The Breeder Genetic Algorithm. *Evolutionary Computation*, Vol. 1(1), pp. 25-49.
- Mühlenbein, H. (1997). The Equation For Response To Selection And Its Use For Prediction. *Evolutionary Computation*, 5(3), 303-346.
- Mühlenbein, H., & Mahnig, T. (2001), Evolutionary Algorithms: From Recombination to Search Distributions. In *Theoretical Aspects of Evolutionary Computing* (pp. 135-173). Springer Berlin Heidelberg.
- Nelson, A. L., Barlow, G. J., & Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4), 345-370.

- Noraini, M. R., & Geraghty, J. (2011), Genetic Algorithm Performance With Different Selection Strategies In Solving TSP, *Proceedings of the World Congress on Engineering 2011, Vol. 2. London*.
- Ognjanovic, I., Mohabbati, B., Gaevic, D., Bagheri, E., & Bokovic, M. (2012), A Metaheuristic Approach For The Configuration Of Business Process Families, In *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, pp. 25-32, IEEE.
- Oladele, R. O., & Sadiku, J. S. (2013), Genetic Algorithm Performance with Different Selection Methods in Solving Multi-Objective Network Design Problem. *International Journal of Computer Applications*, Vol. 70(12).
- Orman, L.V. (1995), A Model Management Approach to Business Process Reengineering, in *Proceedings of the 1995 American Conference on Information Systems* Pittsburgh, PA, USA, pp. 238-247.
- Pall, G.A. (1987), *Quality Process Management*, Prentice-Hall, Englewood Cliffs, NJ, USA.
- Pandey, H. M., Chaudhary, A., & Mehrotra, D. (2014), A comparative review of approaches to prevent premature convergence in GA. *Applied Soft Computing*, 24, 1047-1077.
- Pandey, H. M. (2015), Exploring Genetic Algorithm's Selection Approaches: Theory, Implementations and Statistical Analysis. *Scholar's Press*.
- Pandey, H. M. (2016). Performance Evaluation of Selection Methods of Genetic Algorithm and Network Security Concerns. *Procedia Computer Science*, 78, 13-18.
- Parazoglou, M. P. (2006), *E-Business Organisational & Technical Foundations*. John Wiley & Sons.
- Phalp, K., & Shepperd, M. (2000), Quantitative analysis of static models of processes, *Journal of Systems and Software*, Vol. 52(2), pp. 105-112.

- Pop, C. B., Chifu, V. R., Salomie, I., Kovacs, T., Niculici, A. N., & Suia, D. S. (2012), Business Process Optimization Using Bio-Inspired Methods—Ants Or Bees Intelligence?. In *Intelligent Computer Communication and Processing, 2012 IEEE International Conference on*, pp. 65-71. IEEE.
- Powell, S.G., Schwaninger, M. and Trimble, C. (2001), Measurement and Control of Business Processes, *System Dynamics Review*, Vol. 17, 1, pp. 63-91.
- Praditwong, K., & Yao, X. (2006), A New Multi-Objective Evolutionary Optimisation Algorithm: The Two-Archive Algorithm. In *2006 International Conference on Computational Intelligence and Security*, IEEE, Vol. 1, pp. 286-291.
- Preux, P., & Talbi, E. G. (1995), Assessing the evolutionary algorithm paradigm to solve hard problems, In *Constraint Processing, workshop on really hard problem solving*.
- Rechenberg, I. (1965), *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment, Farnborough, Library Translation 1122, 1965.
- Rechenberg, I. (1994), *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien Der Biologischen Evolution*. Frommann-Holzbog, Stuttgart, 1973. *Step-Size Adaptation Based on Non-Local Use of Selection Information*. In *Parallel Problem Solving from Nature (PPSN3)*.
- Reijers, H. A., & Hee, K. M. (2002). Product-based design of business processes applied within the financial services. *Journal of Research and Practice in Information Technology*, 34(2), 110-122.
- Roque, L. A., Fontes, D. B., & Fontes, F. A. (2012), BRKGA Adapted to Multi objective Unit Commitment-Solving Pareto Frontier for UC Multi objective Problem using BRKGA SPEA2 NPGA and NSGA II Techniques, In *International Conference on Operations Research and Enterprise Systems*, pp. 64-72.

- Salomie, I., Chifu, V. R., Pop, C. B., & Suci, R. (2012), Firefly-Based Business Process Optimization, In *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on*, pp. 49-56, IEEE.
- Schwefel, H. P. (1981), *Numerical optimization of computer models*. John Wiley & Sons, Inc.
- Schumm, D., Karastoyanova, D., Kopp, O., Leymann, F., Sonntag, M., & Strauch, S. (2011), Process Fragment Libraries For Easier And Faster Development Of Process-Based Applications. *Journal of Systems Integration*, 2(1), 39.
- Seshadri, A. (2006), A Fast Elitist Multi Objective Genetic Algorithm: NSGA-II. *MATLAB Central*, 182.
- Shahookar, K., & Mazumder, P. (1990), A Genetic Approach To Standard Cell Placement Using Meta-Genetic Parameter Optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(5), 500-511.
- Shapira, B., and Forfutdinov, V. (2007), *U.S. Patent Application No. 12/377,341*.
- Shukla, A., Pandey, H. M., & Mehrotra, D. (2015), Comparative Review Of Selection Techniques In Genetic Algorithm. In *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on* (pp. 515-519). IEEE.
- Sinha, A., & Goldberg, D. E. (2003), A Survey of Hybrid Genetic and Evolutionary Algorithms. *IlligAL report, 2003004*.
- Sivanandam, S. N., & Deepa, S. N. (2007), *Introduction To Genetic Algorithms*, Springer Science & Business Media.
- Skrlec, D. (2010), Smart Long-Term Planning For Smartgrids. In *Energy Conference and Exhibition (EnergyCon), 2010 IEEE International* (pp. 682-687). IEEE.
- Smirnov, S., Reijers, H. A., Weske, M., & Nugteren, T. (2012), Business Process Model Abstraction: A Definition, Catalog, And Survey. *Distributed and Parallel Databases*, 30(1), 63-99.

- Soliman, F. (1998), 'Optimum Level of Process Mapping and Least Cost Business Process ReEngineering', *International Journal of Operations and Production Management*, Vol. 18, No. 9/10, pp. 810-816.
- Stelling, M., Roy, R., & Tiwari, A. (2009), A Novel Modelling And Optimisation Technique For Business Processes: An Evolutionary Computing Based Approach, In *Applications of Soft Computing*, pp. 75-85, Springer Berlin Heidelberg.
- Stock, J.R. and Lambert, D.M. (2001), *Strategic Logistics Management*, McGraw-Hill, Irwin.
- Stohr, E.A. and Zhao, J.L. (2001), Workflow Automation: Overview and Research Issues, *Information Systems Frontiers*, Vol. 3, No. 3, pp. 281-296.
- Thada, V., & Dhaka, S. (2014), Genetic Algorithm based Approach to Solve Non Fractional (0/1) Knapsack Optimization Problem. *International Journal of Computer Applications*, 100, Vol. 15, pp. 21-26.
- Tiwari, A., & Roy, R. (2004). Challenges In Real World Optimisation Using Evolutionary Computing.
- Tiwari, A., Vergidis, K., & Majeed, B. (2006), Evolutionary Multi-Objective Optimization of Business Processes, In *Evolutionary Computation, 2006, IEEE Congress on*, pp. 3091-3097, IEEE.
- Tiwari, A., Vergidis, K., & Turner, C. (2010a), Evolutionary Multi-Objective Optimisation Of Business Processes, In *Soft Computing in Industrial Applications*, pp. 293-301, Springer Berlin Heidelberg.
- Tiwari, A., Turner, C., Ball, P., & Vergidis, K. (2010b), Multi-Objective Optimisation Of Web Business Processes, In *Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 573-577, Springer Berlin Heidelberg.
- Valenzuela-Rendón, M., Uresti-Charre, E., & Monterrey, I. (1997), A non-generational genetic algorithm for multiobjective optimization. In *in Proceedings of the Seventh International Conference on Genetic Algorithms*.

- Van Der Aalst, W. M., Ter Hofstede, A. H., & Weske, M. (2003), Business process management: A survey, In *International conference on business process management*, Springer Berlin Heidelberg, pp.1-12.
- Van Der Aalst, W. M., van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E., & Weijters, T. (2009), ProM: The process mining toolkit, *BPM (Demos)*, Vol. 489(31), pp. 2.
- Vergidis, K., Tiwari, A., & Majeed, B. (2006), Business Process Improvement Using Multi-Objective Optimisation, *BT Technology Journal*, Vol. 24(2), pp. 229-235.
- Vergidis, K., Tiwari, A., & Majeed, B. (2007a), Composite Business Processes: An Evolutionary Multi-Objective Optimization Approach, In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 2672-2678, IEEE.
- Vergidis, K., Tiwari, A., Majeed, B., & Roy, R. (2007b), Optimisation Of Business Process Designs: An Algorithmic Approach With Multiple Objectives, *International Journal of Production Economics*, Vol. 109(1), pp. 105-121.
- Vergidis, K., & Tiwari, A. (2008), Business Process Design And Attribute Optimization Within An Evolutionary Framework. In *Evolutionary Computation, 2008, IEEE World Congress on Computational Intelligence, IEEE Congress on*, pp. 668-675. IEEE.
- Vergidis, K. (2008). *Business Process Optimization Using An Evolutionary Multi Objective Framework: A thesis in computer science* (phd thesis). Cranfield university, School of applied sciences, department of manufacturing, UK.
- Vergidis, K., Saxena, D., & Tiwari, A. (2012), An Evolutionary Multi-Objective Framework For Business Process Optimisation. *Applied Soft Computing*, Vol. 12(8), pp. 2638-2653.
- Vergidis, K., Turner, C., Alechnovic, A., & Tiwari, A. (2015), An Automated Optimisation Framework For The Development Of Re-Configurable Business Processes: A Web Services Approach, *International Journal of Computer Integrated Manufacturing*, Vol. 28(1), pp. 41-58.

- Verkaria, K., & Clack, C. (1998), Selective crossover in genetic algorithms: An empirical study, In *International Conference on Parallel Problem Solving from Nature*, Springer Berlin Heidelberg, pp. 438-447.
- Verkaria, K. P. (2000), *Selective crossover as an adaptive strategy for genetic algorithms* (Doctoral dissertation), University College London.
- Volkner, P. and Werners, B. (2000), A Decision Support System for Business Process Planning, *European Journal of Operational Research*, Vol. 125, pp. 633-647.
- Wang, M. and Wang, H. (2005), Intelligent Agent Supported Business Process Management, in *Proceedings of the 38th Hawaii International Conference on System Sciences*, Hawaii, pp. 567-577.
- Wang, B., Zhang, L., & Tian, Y. (2009), Multi-Objective Parameter Optimization Technology For Business Process Based On Genetic Algorithm, In *Computer Engineering and Technology, 2009, International Conference on*, Vol. 1, pp. 308-311, IEEE.
- Weske, M. (2010). *Business process management: concepts, languages, architectures*. Springer Publishing Company, Incorporated.
- Whitley, L. D. (1989), The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. In *ICGA*, Vol. 89, pp. 116-123.
- Whitley, L. D. (1994), A Genetic Algorithm Tutorial, *Statistics and computing*, Vol. 4(2), pp. 65-85.
- Wibig, M. (2013), Dynamic Programming and Genetic Algorithm for Business Processes Optimisation, *International Journal of Intelligent Systems and Applications*, Vol. 5(1), pp. 44.
- Workflow Management Coalition: Terminology & Glossary (1999), Technical Report WFMCTC-1011.

- Yang, Y., Fritzsche, K. J., & Hong, M. (2013), Resonance assignment of the NMR spectra of disordered proteins using a multi-objective non-dominated sorting genetic algorithm, *Journal of biomolecular NMR*, Vol. 57(3), pp. 281-296.
- Yoo, S., Harman, M., & Ur, S. (2013), GPGPU Test Suite Minimisation: Search Based Software Engineering Performance Improvement Using Graphics Cards, *Empirical Software Engineering*, Vol. 18(3), pp. 550-593.
- Zhang, D. (2009), *Parallel robotic machine tools*. Springer Science & Business Media.
- Zhang, Y., & Perry, D. E. (2015), Generating Symbolic Business Processes in Support of Evaluating Process Optimization. In *12th IEEE International Conference on Services Computing (SCC 2015)*, New York NY.
- Zheng, F., Simpson, A. R., & Zecchin, A. C. (2014), An efficient hybrid approach for multiobjective optimization of water distribution systems. *Water Resources Research*, Vol. 50(5), pp. 3650-3671.
- Zhong, J., Hu, X., Zhang, J., & Gu, M. (2005), Comparison of performance between different selection strategies on simple genetic algorithms. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vol. 2, pp. 1115-1121.
- Zhou, Y. and Chen, Y. (2002), Business Process Assignment Optimisation, in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics Vol. 3*, pp. 540-545
- Zhou, Y. and Chen, Y. (2003), The Methodology for Business Process Optimised Design, in *IECON Proceedings (Industrial Electronics Conference)* Vol. 2, pp. 1819-1824.

Résumé

En réponse à un environnement de plus en plus volatile et compétitif, les entreprises examinent comment leurs processus métiers peuvent être repensés, afin d'améliorer leurs performances et leurs réactivités vis-à-vis du marché. Cependant, il y a un manque - terrible - d'approches globales pour l'identification, l'analyse et l'optimisation des conceptions de processus métier. Le but de cette recherche est de développer un Framework d'optimisation évolutionnaire multi objectifs (ou multicritères) des conceptions (ou instances) de processus métiers capable de (i) représenter des conceptions de processus métiers d'une manière quantitative, (ii) composer d'une manière algorithmique des conceptions de processus métiers en fonction d'exigences spécifiques. (iii) Identifier les conceptions de processus métiers optimales en utilisant une proposition d'un nouveau algorithme évolutionnaire. Enfin, (iv) évaluer la qualité des nouvelles conceptions optimisées de processus métiers en usant d'une fonction de Fitness capable de traiter jusqu'à trois (03) critères d'optimisation. Cette recherche propose une représentation technique pour les conceptions de processus métiers en utilisant, à la fois, une perspective visuelle et quantitative. Elle propose également l'algorithme de vérification inverse des processus métiers - **Rear Process Verification Algorithm** - (**ReProVA**) -, qui est un algorithme pour la vérification et la recombinaison de nouvelles conceptions de processus métiers. L'évaluation de ces conceptions se fait par l'intermédiaire d'une fonction de Fitness adaptée à l'optimisation multi objectifs proposée. Le Framework d'optimisation proposé est au cœur de cette recherche. Il emploie la représentation technique citée, **ReProVA**, la fonction de Fitness et un algorithme évolutionnaire amélioré, pour l'optimisation multi objectifs. Le Framework est capable de générer une série de conceptions alternatives optimisées de processus métier, en fonction des exigences données, le tout se basant sur un modèle de processus métier proposé. Enfin, un scénario de processus métier inspiré de la vie réelle est présenté et utilisé. Afin d'évaluer, à la fois, la capacité du Framework dans l'optimisation évolutionnaire multicritères d'un scénario de processus métier tiré de la vie réelle, et la performance du dit algorithme évolutionnaire amélioré. Les résultats de l'expérimentation ont démontré la capacité du Framework proposé à automatiser le processus de composition, d'identification, de vérification et d'optimisation des conceptions de processus métier avec des valeurs d'attributs optimisées.

Abstract

In response to an increasingly volatile and competitive environment, companies are examining how their business processes can be redesigned to improve their performance and responsiveness to the market. However, there is a terrible lack of holistic approaches to the identification, analysis and optimization of business process designs. The aim of this research is to develop a multi objective (or multi criteria) evolutionary optimization framework for business process designs (or instances) capable of (i) representing business process designs in a quantitative way, (ii) composing in an algorithmic way business process designs according to specific requirements. (iii) Identify optimal business process designs using a proposal for a new evolutionary algorithm. Finally, (iv) evaluate the quality of new, optimized business process designs using a Fitness function capable of processing up to three (03) optimization criteria. This research provides a technical representation for business process designs using both a visual and a quantitative perspective. It also offers the Rear Process Verification Algorithm (**ReProVA**) algorithm, which is an algorithm for the verification and recombination of new business process designs. The evaluation of these designs is done through a Fitness function adapted to the proposed multi objective optimization. The proposed optimization framework is at the heart of this research. It uses the cited technical representation, **ReProVA**, the Fitness function and an improved evolutionary algorithm, for multi objective optimization. The framework is able to generate a series of optimized alternative business process designs, based on given requirements and a proposed business process model. Finally, a business process scenario inspired from real life is presented and used. In order to evaluate both the framework's capability in multi-criteria evolutionary optimization of a real-life business process scenario and the performance of the enhanced evolutionary algorithm. The results of the experiment demonstrated the ability of the proposed framework to automate the process of identifying, verifying and optimizing business process designs with optimized attribute values.

ملخص

للاستجابة إلى بيئة متقلبة وتنافسية متزايدة، تدرس الشركات كيفية إعادة تصميم عملياتها التجارية لتحسين أدائها واستجابتها للسوق. ومع ذلك، هناك نقص رهيب في النهج الشاملة لتحديد وتحليل وتحسين التصميمات العملية التجارية. والهدف من هذا البحث هو وضع إطار للتحسين التطوري متعدد الأهداف (أو متعدد المعايير) لتصاميم العمليات التجارية (أو الحالات) قادر على (1) تمثيل تصاميم العمليات التجارية بطريقة كمية، (2) إنشاء الأعمال بطريقة خوارزمية عملية التصميمات وفقاً لمتطلبات محددة (3) تحديد التصميمات المثلى لعملية التجارية باستخدام اقتراح لخوارزمية تطويرية جديدة. وأخيراً، (4) تقييم نوعية تصاميم العمليات التجارية الجديدة والأمثلة باستخدام وظيفة اللياقة البدنية قادرة على معالجة ما يصل إلى ثلاثة (03) معايير الأمثلة. ويوفر هذا البحث تمثيلاً فنياً لتصاميم العمليات التجارية باستخدام منظور مرئي وكمي. كما يقدم خوارزمية التحقق من العملية الخلفية (**ReProVA**)، وهو خوارزمية للتحقق وإعادة تشكيل تصاميم عملية التجارية الجديدة. ويتم تقييم هذه التصميمات من خلال وظيفة اللياقة البدنية تتكيف مع التحسين المتعدد الهدف المقترح. إطار التحسين الأمثل هو في صميم هذا البحث. ويستخدم التمثيل الفني استشهد، **ReProVA**، وظيفة اللياقة البدنية وخوارزمية التطويرية المحسنة، لتحسين موضوع متعددة. الإطار قادر على توليد سلسلة من التصميمات عملية بديلة الأمثلة، استناداً إلى متطلبات معينة ونموذج عملية الأعمال المقترحة. وأخيراً، يتم عرض سيناريو عملية التجارية مستوحاة من واقع الحياة واستخدامها. من أجل تقييم كل من قدرة الإطار في متعددة المعايير الأمثلة التطويري لسيناريو العملية التجارية واقع الحياة وأداء خوارزمية التطوير المعزز. وأظهرت نتائج التجربة قدرة الإطار المقترح على أتمتة عملية تحديد التصميمات العملية للأعمال والتحقق منها وتحسينها مع قيم السمات المحسنة.