

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de fin d'études Master

Filière : Informatique

Option : Système informatique

Thème :

Conception d'un système d'aide à la décision multicritères

Encadré Par :

Dr. BENHAMZA Karima

Présenté par :

HAMLAOUI Abir

Juin 2018

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de fin d'études Master

Filière : Informatique

Option : Système informatique

Thème :

Conception d'un système d'aide à la décision multicritères

Encadré Par :

Dr. BENHAMZA Karima

Présenté par :

HAMLAOUI Abir

Juin 2018

Abstract

Existing decision-making problems are usually of a multi-objective nature where several criteria are to be taken care of jointly. Solving such problems is often the domain of multi-objective combinatorial optimization.

The work that has been done is the design of a multicriterion decision support system by applying multicriteria optimization to the scheduling problem. The tool used is the genetic algorithm.

In this report, we consider this scheduling problem, which one seeks to optimize by using a meta heuristic approach that is the genetic algorithm. The overall goal of this approach is to find a solution (or solutions) that meets the criteria or better compromise

Keywords: Help with the decision; multi-criteria decision; multi-objective optimization, genetic algorithm.

Résumé

Les problèmes décisionnels existants sont couramment de nature multi-objectifs où plusieurs critères sont à prendre en charge conjointement. Résoudre de tels problèmes relève souvent du domaine de l'optimisation combinatoire multi objectifs.

Dans ce mémoire, le travail établi est la conception d'un système d'aide à la décision multicritère appliquant une approche méta-heuristique qui est l'algorithme génétique. Notre objectif est de montrer les performances de ces méthodes dans ce domaine. Nous considérons pour cela un problème difficile largement exploré qui est le problème d'ordonnancement de tâches.

Mots clé : Aide à la décision; décision multicritère; optimisation multi-objectif, algorithme génétique.

Remerciement

Nous tenons à remercier en premier lieu Allah qui nous a donné vie et santé pour le parachèvement de ce modeste travail.

Nous remercions après de tout coeur notre encadreur « Mme Benhamza Karima » pour son soutien, sa sympathie, ces encouragements, la confiance qu'elle nous témoignée en acceptant de diriger ce travail et pour avoir mis à notre disposition ses conseils pour une meilleure maîtrise du sujet.

Nous remercions nos familles qui nous ont toujours donné la possibilité de faire ce que nous voulions durant nos études et qui ont toujours cru en nous.

Enfin, nous remercions tous ceux qui ont contribué à ce travail par leurs remarques, leurs suggestions et leurs soutiens.

Dédicace

À mes parents qui m'ont soutenu durant mon parcours

À ma sœur Amira et mon frère mustafa Med el-cherif

Abir...

Table de matière

Abstract	i
Résumé	ii
Remerciement	iii
Dédicace	iv
Table de matière	1
Table de Figure	4
List des tableaux	5
Introduction Générale	6
Chapitre 1. La prise de décision	9
1. Introduction	10
2. Concepts de base	10
2.1 Définitions de la décision	10
3. Typologie des décisions	11
3.1 Décisions programmables	11
3.2 Décisions non programmables	11
4. Problématique de décision	12
4.1 Problématique de tri (P, β)	12
4.2 Problématique de rangement (P, γ)	12
4.3 Problématique de choix (P, α)	13
4.4 Problématique de description (P, σ)	13
5. Démarche de modélisation	14
5.1 Démarche descriptive	14
5.2 Démarche axiomatique	15
5.3 Démarche constructive	15
6. Modélisation du processus décisionnel	15
7. Eléments intervenant dans le modèle de décision	16
8. Décision multicritère	17
9. Concepts et terminologie	17
(a) Les alternatives, les variables de décisions et les contraintes Les alternatives sont	17
(b) Les objectifs :	17
10. Typologie des modèles de décision multicritère	18

11. Aide à la décision multicritère	18
12. Conclusion.....	20
Chapitre 2. L'optimisation multicritère.....	21
1. Introduction	22
2. Complexité et théorie de la complexité.....	22
3. Définition de l'optimisation	23
4. Principaux concepts en optimisation.....	23
5. Classification d'optimisation.....	24
6. Formulation du modèle d'optimisation multi objectif	25
7. Méthodes de résolution d'un problème d'optimisation multicritères	26
7.1 Méthodes exactes	26
7.2 Les méthodes heuristiques	27
7.3 Méthodes à base de populations ou distribuées	27
7.4 Exemples de problèmes d'optimisation	28
8. Méthodes de résolution des problèmes d'ordonnancement	28
8.1 Méthodes de résolution approchées	28
8.2 Les méta-heuristiques	28
8.3 Algorithme génétique	29
9. Structure d'un algorithme génétique	30
9.1 Génération de la Population Initiale.....	30
9.2 Sélection	31
9.3 Croisement :	31
9.4 Mutation	31
9.6 Critère d'arrêt	32
10. Conclusion.....	32
Chapitre 3. Conception et Implémentation.....	33
1. Introduction	34
2. Problème d'ordonnancement	34
2.1 Variante des problèmes d'ordonnancement	35
2.2 Objectif du travail	35
3. Base de données	35
4. Modélisation.....	36
4.2 Codification de la tâche.....	Erreur ! Signet non défini.

5. Fonctionnement du système	36
5.2 Scénario du fonctionnement de système	38
6. Implémentation.....	40
6.1 Matériel utilisé.....	40
6.2 Langage utilisé	40
6.3 Langage Orienté objet	40
6.4 Avantages de Java	40
7. Outils utilisé : NetBeans.....	41
8. Présentation du modèle proposé.....	41
9. Lancement de l'application	42
10. Conclusion.....	45
Conclusion générale	46
Bibliographie.....	48

Table de Figure

Chapitre 1

Figure 1.1 : Problématique de tri.....	12
Figure 1.2 : Problématique de rangement	13
Figure 1.3 : Problématique de choix.	13
Figure 1.4 : Problématique de description.	14
Figure 1.5 : Modélisation du processus décisionnel IDC de Simon.	16

Chapitre 2

Figure 2.1 : diagramme d’algorithme génétique.	30
---	----

Chapitre 3

Figure 3.1 : Clin d’œil sur la base de données	36
Figure 3.2 : Schéma du processus général du système proposé.....	37
Figure 3.3 : Croisement.....	39
Figure 3.4 : Mutation.....	39
Figure 3.5 : Bibliothèque java utilisée.	42
Figure 3.6 : Classes utilisées.	42
Figure 3.7: Interface d’accueil.	43
Figure 3.8: menu de la partie mono objective.....	43
Figure 3.9: Interface de mutation.	43
Figure 3.10 Interface de croisement	44
Figure 3.11 : Interface multi objectif	44
Figure 3.12 : Boite de dialogue	Erreur ! Signet non défini.
Figure 3.13 : fenetre des résultats	45

List des tableaux

Chapitre 1

Tableau 1.1 : Méthodes et techniques de prise de décision	11
Tableau 1.2 Les différentes problématiques de décision	14

Introduction Générale

Introduction général

Une discipline qui s'appelle « recherche opérationnelle » consiste en l'application des méthodes scientifique pour résoudre les problèmes complexes. La caractéristique de l'étude est le développement d'un modèle scientifique avec lequel on tente de prévoir et de comparer les résultats de diverses décisions, Le but est d'aider le décideur à déterminer sa politique de manière scientifique.

Aujourd'hui les organisations sont devenues de plus en plus complexes ce qui a compliqué la tâche de la prise de décision tel que cette dernière est un acte pratiqué souvent par des humains. C'est pourquoi les domaines d'applications de l'optimisation multicritère se situent donc autour de l'aide à la décision en opposition avec d'autres branches des mathématiques appliquées ou les modèles à étudier sont formulé à partir d'autre discipline.

Les domaines d'applications de la recherche opérationnelle se situent donc autour de l'aide à la décision pour l'amélioration de la productivité, l'optimisation dans le choix stratégique des organisations, L'objectif essentiel de cette étude est d'évaluer les performances des algorithmes génétiques pour la résolution d'un problème d'ordonnement de taches.

L'ordonnement joue un rôle essentiel dans de nombreux secteurs d'activité tel que l'informatique (ordonnement de processus, ordonnement des réseaux). Les méthodes d'ordonnement se diversifient dans la littérature, elles se différencient par la nature du problème considéré, La nature des contraintes prise en compte, les objectifs à satisfaire et la nature de l'approche de résolution optée. Les problèmes d'ordonnement sont en générale difficiles et d'une grande utilité pratique. Dans ce contexte, nous nous intéressons aux problèmes d'ordonnement de taches.

. La plus part des problèmes d'ordonnement sont d'une telle complexité que le nombre de solutions potentielles croit exponentiellement avec la taille du problème. La réalisation d'un projet suppose l'exécution préalable de multiples opérations soumises à de nombreuses contraintes

Organisation du mémoire

Ce mémoire est organisé en trois chapitres présentant respectivement, l'état de l'art, notre proposition et enfin une réalisation de l'application de l'algorithme génétique.

- Le premier chapitre s'intéresse au domaine de la décision et des systèmes d'aides multicritères.

Introduction général

- Le deuxième chapitre passe en revue les notions liées à l'optimisation multicritère.
- Le troisième chapitre contient les détails du modèle proposé pour une approche basée méta-heuristique, Il contient aussi l'implémentation avec des aperçus montrant les différentes interfaces de l'application.

Nous terminerons notre mémoire par une conclusion générale qui résume l'apport essentiel à notre travail et quelques perspectives.

Chapitre 1.

La prise de décision

Chapitre 1

La prise de décision

1. Introduction

Dans la vie l'être humain est souvent confronté à des problèmes là où il est obligé de prendre une décision optimale, Cette décision n'est pas toujours facile à prendre, car il peut s'agir d'un problème complexe pour lequel la décision recherchée peut engendrer des conséquences relativement importantes. Cependant il devient nécessaire de formaliser ce genre de problème pour garantir une meilleur prise de décision.

La prise de décision est un processus cognitif complexe, Ce processus est activé lorsque nous ressentons le besoin d'agir alors qu'il se présente pour nous un choix important parmi un ensemble d'alternatives.

Dans ce chapitre, nous présentons le domaine de la prise de décision, Nous abordons les concepts de base de la décision avec ses différents types et leurs caractéristiques. Enfin, nous présente la notion d'aide à la décision multicritères.

2. Concepts de base [Nachet, 2014]

Plusieurs définitions de « décision » lui ont été affectées à ce terme qui est très vaste.

2.1 Définitions de la décision

Selon [Levin & Pomeroi, 1989] « Une décision est une action qui est prise pour faire face à une difficulté ou répondre à une modification de l'environnement, c'est à dire, pour résoudre un problème qui se pose à l'individu ou à l'organisation ».

[Holtzman, 1989] la définit comme : « Le résultat d'un processus mental qui choisit une parmi plusieurs alternatives mutuellement exclusives»

Et d'après [Schneider, 1994] une décision est aussi définie comme "un choix entre plusieurs alternatives", ou encore par "le fait qu'elle concerne aussi le processus de sélection de buts et d'alternatives".

[Roy, 2000] trouve qu'une décision est souvent vue comme le fait d'un individu isolé (« le décideur ») exerçant un choix entre plusieurs possibilités d'actions à un moment donné

En général, la décision est définie comme étant l'action de décider après délibération, où l'acteur exerce un rôle important. Ce n'est donc pas un acte élémentaire et simple, mais c'est l'aboutissement d'un processus complexe de décision.

3. Typologie des décisions [Nachet, 2014]

Dans sa classification, [Simon, 1977] a proposé deux types de décisions : « les décisions programmées » et « les décisions non programmées ».

3.1 Décisions programmables

Ce sont les décisions routinières dont une procédure peut être définie pour les effectuer, Pour cela des techniques d'optimisation via programmation mathématique et programmation linéaire sont utilisées.

3.2 Décisions non programmables

Ce sont les décisions pour lesquelles aucune procédure spécifique n'est définie pour les effectuer parce qu'elles sont non structurées. On présente ci-dessous les méthodes de prise de décision :

	Traditionnelles	Modernes
Décisions programmables	<ul style="list-style-type: none"> • L'habitude. • La routine. • Procédure Opérationnelle/ standardisée. 	<ul style="list-style-type: none"> • Recherche opérationnelle <ul style="list-style-type: none"> - Les modèles, - L'analyse mathématique, - La simulation par ordinateur. • Le traitement informatique des données par les programmes (les algorithmes).
Décisions non programmables	<ul style="list-style-type: none"> • Le jugement. • L'intuition, la créativité. • Les règles empiriques. • La sélection et la formation des décideurs 	<ul style="list-style-type: none"> • Les techniques heuristiques de résolution de problème et leur informatisation (intelligence, artificielle, système experts, programmation sous contraintes, etc). • Le traitement informatique de connaissance à partir de données (entrepôt et fouille de données)

Tableau 1.1 : Méthodes et techniques de prise de décision [Espinasse, 2009].

Et donc, la recherche d'une solution pour ce genre de problème implique un mélange de recherche d'information, de formulation du problème, de calcul et de manipulation de données.

Dans la décision programmable, l'homme prend l'avantage sur la machine, et le décideur peut adopter une stratégie progressive avec des retours en arrière. Cependant la situation de décision sera complexe si elle possède les caractéristiques suivantes : on ne peut pas modéliser tout le processus de décision, la phase de recherche d'information est difficile, l'identification du problème nécessite une réelle expertise.

4. Problématique de décision [Othmani, 1998]

La problématique de décision se voit comme une orientation de l'investigation d'un problème de décision donnée, elle exprime la terminologie dont laquelle le décideur pose le problème et traduit le type suivi de prescription qu'il veut obtenir. [Roy, 1985] pose quatre problématique et indique tout problème de décision multicritère de se ramener nécessairement à l'une d'entre elle.

4.1 Problématique de tri (P,β)

Elle consiste à effectuer chaque action à un ensemble de catégorie prédéfini dont l'objectif est de proposer une recommandation parmi un ensemble spécifiées en avance, telle que chaque recommandation peut être associé avec une catégorie. Cette formulation est adéquate lorsque le problème de décision consiste à examiner chaque action indépendamment des autres. Le problème est alors vu comme trier les actions potentielles aux différentes catégories. Formellement, une prescription consiste à une partition de Λ [Mousseau, 1993]

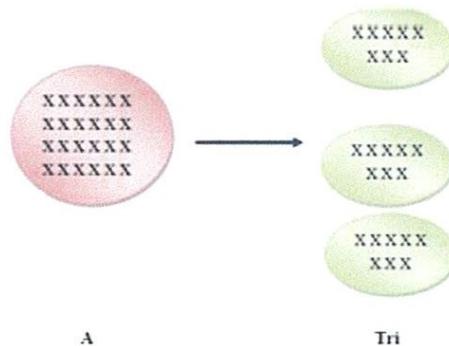


Figure 1.1 : Problématique de tri [mousseau .1993].

4.2 Problématique de rangement (P, γ)

Consiste à ranger les différentes actions en allant de la meilleur à la moins bonne action, elle importante quand les actions sont à différencier selon un intérêt relatif. Obtenir un ordre complet c'est le cas idéal. Formellement, la prescription est un ordre partiel, une relation transitive est définie sur A.

Au niveau de la décision multicritère, Cette problématique vise à ranger les alternatives suivant un ordre de préférence en établissant un pré-ordre ou ordre complet ou partielle sur ces alternatives. Ils existent deux autres problématiques multicritères proposées par Bernard Roy [Roy, 1985] qui sont celles de la classification et de la description.

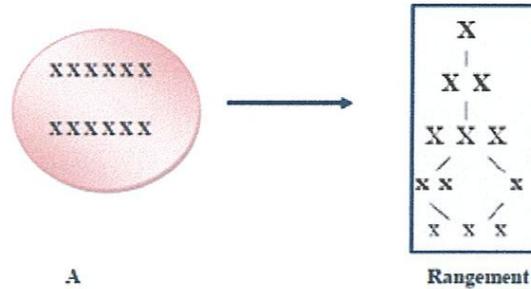


Figure 1.2 : Problématique de rangement [Mousseau.1993].

4.3 Problématique de choix (P,α)

Consiste à sélectionner un sous ensemble des actions A qui contiennent les meilleures actions et dans ce cas-là l'idéal est d'obtenir une seule et la meilleure action mais la nature conflictuelle des critères pousse le décideur à fournir quelque action qui représente différentes variantes de la meilleure action.

Lors de la décision multicritère, Ce type de problématique vise à identifier un ensemble réduit d'alternatives (éventuellement ou alternative) meilleur que le reste, entre lesquelles il est difficile de choisir. L'optimisation est un exemple de cette problématique.

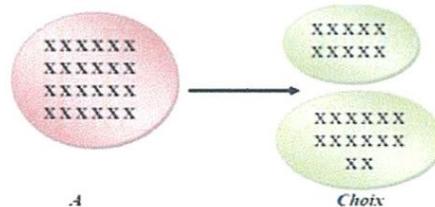


Figure 1.3 : Problématique de choix [Mousseau.1993].

4.4 Problématique de description (P,σ)

Cette problématique consiste à décrire les actions et leurs conséquences et non pas à les comparer comme c'est le cas par rapport aux trois autres problématiques précédentes.

Dans ce cas il n'existe pas une prescription et la procédure d'investigation est cognitive [Roy, 1985].

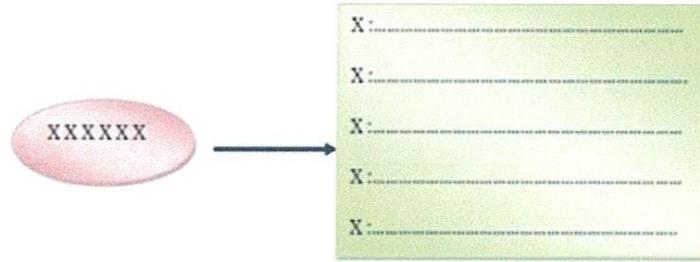


Figure 1.4 : Problématique de description [Roy. 1985].

Le tableau ci-dessous résume les quatre problématiques :

Problématique	Procédure	Prescription	Résultat	objectif
(P, α)	Sélection	Sous ensemble de A	Un choix ou une procédure de sélection	Sélectionner un sous ensemble aussi restreint que possible de A contenant les meilleurs actions.
(P, β)	Segmentation	Partition de A	Un tri ou une procédure d'affectation	Affecter chaque action de A à une seule des catégories définies selon de normes préétablies
(P, γ)	Classement	Ordre partiel sur A	Un rangement ou une procédure de classement	Ordonner les actions de A de la meilleur à la moins bonne.
(P, δ)	Cognitive	Aucune prescription	Une description ou une procédure cognitive	Décrire les actions et leurs conséquences.

Tableau 1.2 : Les différentes problématiques de décision [Roy, 1985].

5. Démarche de modélisation [Nachet 2014]

D'après [Roy, 1992] l'aboutissement d'un raisonnement qui conduit à construire une préférence à partir des performances d'un groupe de critère, cela fait référence à la phase de modélisation du problème décisionnel. Il existe trois démarches pour élaborer un système d'aide à la décision dont la différence principale consiste dans la manière d'aborder la réalité.

5.1 Démarche descriptive

[Bouyssou, 1990] vise à décrire et à expliquer le problème décisionnel, à partir du système de préférence des intervenants de la manière la plus fidèle, sans le perturber

5.2 Démarche axiomatique

L'approche axiomatique est presque inexistante dans le domaine de l'optimisation multicritère, exception faite pour la théorie de la valeur multi attribut. Ce sont plutôt les méthodes d'essai-erreur et l'approche par apprentissage qui sont prônées par les chercheurs dans ce domaine. Selon [Mousseau, 1993], Sur la base d'un système d'axiomes caractérisant le comportement rationnel en matière de décision, il y'a un ensemble de principes et de règles à suivre prescrit aux intervenants pour modéliser le problème décisionnel.

5.3 Démarche constructive

D'après [Bouyssou, 1990], C'est l'élaboration du modèle d'aide à la décision à partir de « la partie stable de la prescription du problème ».

Dans l'attitude adopter pour la modélisation des préférences, les démarches descriptive et constructive sont opposées. L'approche constructive considère que les systèmes de préférence est conflictuel moins structuré et appelé à être évalué au cour du processus d'aide à la décision alors que l'approche descriptive considère l'existence d'un système de préférence réel et objectif.

6. Modélisation du processus décisionnel [Simon, 1977] [Nachet 2014]

Le processus de décision est la détermination des étapes à suivre par un décideur pour arriver à opter une décision comme solution à un problème posé. En 1960 Simon a proposé le modèle IDC qui décompose ce processus en trois étapes (Intelligence, Designe, Choice) ou en français (Information, Conception, Choix). En 1977 ce même chercheur à modifier son modèle en lui ajoutant une quatrième étape (Review) ou (évaluation), elle permet de valider la décision à appliquer. Ce modèle est à ce jour une référence pour toute modélisation de la décision.

Recherche d'information (intelligence) : c'est la phase dans laquelle on identifie le problème. Donc il est nécessaire de rechercher les informations pertinentes en fonction des préoccupations du décideur. Il s'agit aussi d'identifier aussi les objectifs ou buts du décideur et de définir le problème à résoudre.

Conception (design) : c'est la phase de modélisation là où le décideur construit des solutions, elle aboutit aux différents chemins possibles à la résolution du problème.

Choix (choice) : c'est la phase qui se base sur la sélection d'un mode d'action particulière et donc c'est la phase de la prise de décision.

Contrôle ou évaluation (review) : phase ajoutée permettant d'évaluer la solution choisie qui est la décision prise, elle permet le retour en arrière vers l'une des trois phases précédentes ou la validation de la solution.

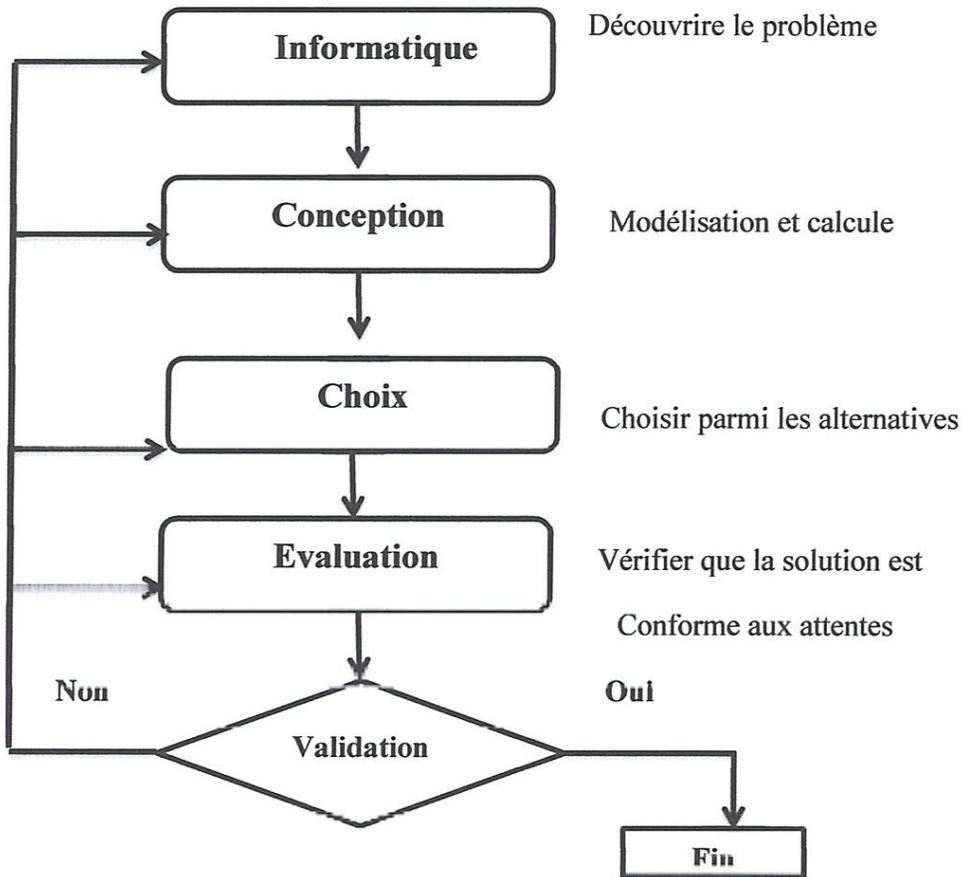


Figure 1.5 : Modélisation du processus décisionnel IDC de Simon [Simon, 1977].

7. Éléments intervenant dans le modèle de décision [Adla, 2007] [Othmani,1998]

La décision ne se présente pas après avoir posé le problème ou collecté les informations mais progressivement durant un long processus d'action et de planification, Pour n'importe quel processus décisionnel, on développe un modèle de décision qui est composé généralement de cinq éléments :

- **Le décideur :** un ou plusieurs individus chargé de prendre une décision particulière.
- **Un ensemble d'entrées au processus de décision :** ce sont des modèles numériques ou qualitatives.
- **Le processus de décision lui-même :** ensemble des étapes qui sert à transformer les entrées en sorties sous forme de décisions.

- Un ensemble de sorties produit par le processus de décision.
- Un ensemble de critères pour évaluer les décisions produites par le processus par rapport à l'ensemble des besoins, des problèmes ou des objectifs de la décision.

8. Décision multicritère [Othmani, 1998]

La décision multicritères s'intéresse aux problèmes de prise de décision en présence de critères multiples éventuellement coordination, Selon [Bouyoussou, 1984][Fishburn, 1968] ces problèmes se rencontre a tous les niveaux et dans tous les domaines, par exemple : la sélection des candidats à un concours selon les critères requis par les postes, d'adopter un nouveau processus de production dans une usine pour l'amélioration des critères d'investissement.

La majorité des décisions multicritère sont de nature complexe et leurs résultats sont important et stratégique, C'est pourquoi l'aspect conflictuel des critères, l'indétermination et le manque d'informations liée au problème sont souvent de nature complexe.

9. Concepts et terminologie [Othmani, 1998]

Dans cette partie nous allons présenter la terminologie couramment utilisée dans la littérature de la théorie de décision multicritères. D'après [Othmani, 1998], Les termes qui permettent de décrire ce problème sont : les alternatives, les objectifs, les attributs, les critères, le décideur et l'analyste.

(a) Les alternatives, les variables de décisions et les contraintes Les alternatives sont l'ensemble des scénarios sur lesquels porte la décision, L'identification de cet ensemble de solutions réalisables est une tâche primordiale dans la définition du problème défini explicitement par un ensemble fini d'alternative de cardinal relativement faible ou implicitement par un ensemble de propriétés ou de condition que les alternatives doivent vérifier.

(b) Les objectifs : Un objectif indique le sens de l'amélioration qu'un décideur souhaite apporter à un système lors d'un changement d'état. Pour poursuivre un objectif il existe trois manières qui sont : maximiser, minimiser ou le maintenir dans un état. D'autres types d'objectifs souvent être affronté comme : près d'une cible(le but), plus grand ou plus petit d'un certain seuil, dans un intervalle,...etc

(c) Les attributs : ce sont les caractéristiques des alternatives, ils permettent d'évaluer les niveaux des objectifs.

(d) **Les critères** : En théorie de la décision, il correspond à un attribut ou à un objectif. Dans ce sens, un problème de décision désigne soit un problème de décision multi attribut, soit multi objectif, ou les deux.

(e) **Décideur** : Est un individu (ou même groupe des individus) qui face la situation de décision, il prend la responsabilité de l'évaluation des alternatives pour proposer ou mettre une ou plusieurs solutions. La plus importante tâche du décideur est de se dévoiler ses jugements personnel, de s'en convaincre lui-même et de décider, Nous parlons donc de l'auto conviction qui peut se faire à l'aide d'une analyse plus formelle et plus structurée ou d'une façon totalement intuitive.

10. Typologie des modèles de décision multicritère [Othmani, 1998]

Les problèmes multicritères sont généralement classifiés selon la nature des conséquences de décision, la nature de l'ensemble des alternatives, le contexte de la prise de décision et le nombre de décideurs. Selon [Vansnick, 1986], la principale composante du processus décisionnel multicritère est les importances attribuées aux différents critères considérés, Ces importances sont représentées par deux expressions quantitatives : poids ou biais ou priorité. Dans la majorité de processus de décision, décrire la structure de préférence entre les différentes alternatives revient à l'évaluation des poids ou des priorités.

Les modèles de décision multicritères utilisent généralement la notion fondamentale de structure ou de préférence. Classer les solutions d'un problème multicritère revient à comparer des vecteurs de nombre réels, les relations d'ordre servant de critères de décision pour la comparaison et le classement des solutions s'appelle relation de dominance.

11. Aide à la décision multicritère [Kolokotsa & al, 2009]

[Roy, 2000] a défini l'aide à la décision comme suit : « L'activité de celui qui, prenant appui sur des modèles, clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponse aux questions que se pose un acteur dans un processus de décision. Eléments concourant à éclairer la décision et normalement à recommander, ou simplement à favoriser, un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service desquels ces acteurs se trouvent placés d'autre part ».

Une autre définition pour les systèmes d'aide à la décision est celle de [Keen, & Scott-Morton, 1978] «Les systèmes d'aide à la décision font coupler les ressources intellectuelles des individus avec les capacités de l'ordinateur pour améliorer la qualité des

décisions. C'est un système informatique d'aide aux décideurs qui traite des problèmes semi-structurés ».

Aussi, Selon [Turban, 1993], un système d'aide à la décision est « un système d'information automatisé, interactif, flexible, adaptable et spécifiquement développé pour aider à la résolution d'un problème de décision non structuré et améliorer la prise de décision. Il utilise des données, fournit une interface utilisateur simple et autorise l'utilisateur à développer ses propres idées ou points de vue. Il peut utiliser des modèles standards ou spécifiques, supporter les différentes phases du processus de la prise de décision et inclure une base de connaissances ». Cette définition porte à la fois sur les fonctionnalités du système ainsi que sur son architecture

La décision multicritère est un problème délicat. On parle d'ailleurs plus volontiers "d'aide multicritère pour montrer que la décision finale appartient à l'humain. Toutes les méthodes ont des avantages et des inconvénients : il n'y a pas une "meilleure" méthode les méthodes apparemment naturelles possède toutes de fort biais structurels.

D'après ces définitions, on peut classer les systèmes d'aide à la décision selon plusieurs critères, dans cette partie nous allons en présenter quatre classifications :

- En fonction de la quantité des informations manipulées.
- En fonction du niveau de la décision impliqué.
- En fonction de l'envergure de la décision.

En fonction du niveau conceptuel du système.

Les développements des systèmes d'aide à la décision multicritère s'intéressent à l'état de l'art des méthodologies d'aide à la décision pour l'identification de stratégie efficaces.

Les problèmes d'aide la décision peuvent être basée sur différentes approches telle que l'approche d'optimisation multicritère et l'approche d'aide multicritère à la décision (MCDA). Ces approches sont généralement basées sur la simulation ou l'optimisation multicritères. Ainsi, certaines méthodes sont pour la génération ou pour l'optimisation de solution optimale.

12. Conclusion

Dans ce chapitre, une vue détaillée est donnée sur un domaine de recherche opérationnelle très actif. L'approche de concept de décision, d'aide à la décision, et de système d'aide à la décision multicritère, ainsi que le processus de développement de ce type de système.

Dans le prochain chapitre, on va décrire les approches de résolution et de l'optimisation multicritère pour la prise décision

Chapitre 2.

L'optimisation multicritère

Chapitre 2 :

L'optimisation multicritère

1. Introduction

Depuis une trentaine d'années, le domaine de l'optimisation multicritère connaît une évolution importante. Cette évolution s'est traduite par le développement d'un grand nombre de méthodes. La multitude des méthodes d'optimisation multicritère est perçue comme une richesse incontestable de ce domaine. D'ailleurs, certains la justifient par la diversité des problèmes ainsi que par l'existence de différentes approches de résolution possibles et légitimes de ces problèmes. Le travail présenté dans ce chapitre expose un aperçu sur l'optimisation multicritère et ses différentes méthodes de résolution.

2. Complexité et théorie de la complexité [Othmani, 1998]

Pour pouvoir dire qu'un tel problème est plus facile à résoudre que l'autre et afin de mesurer la difficulté d'un problème donnée et la comparer avec celle des autres problèmes, d'après [Gherboudj, 2013] nous pouvons calculer la complexité algorithmique de chaque un d'entre eux, telle que cette complexité est étudiée sur deux côtés qui sont le côté temporel et le côté spatial qui dépendent de l'évaluation du temps d'exécution nécessaire et l'espace mémoire pour la résolution d'un problème donnée.

L'estimation de la complexité est en fonction du nombre d'instruction permettant d'aboutir à la solution du problème posé. En effet elle est influencée par la taille du problème en question c'est-à-dire elle exprime le rapport entre la taille du problème, le temps de calcul nécessaire et l'espace mémoire requis.

Donc la théorie de la complexité s'intéresse à l'étude et l'évaluation du taux de la difficulté des problèmes via l'étude de la complexité de solution algorithmique proposée.

Bien que la théorie de complexité se concentre des problèmes de décision, elle peut s'étendre au problème d'optimisation [Garey & Johnson, 1979], en classant les problèmes selon leurs complexités principales : la classe P Polynomial time et la classe NP (non déterministe polynomial time).

De plus, elle partage les problèmes de la classe NP en deux sous classes : NP_complet et NP_difficile

3. Définition de l'optimisation

Selon [Paschos, 2005] une définition générale a été affectée à ce terme : « L'optimisation c'est l'art de comprendre un problème réel, de pouvoir le transformer en un modèle mathématique que l'on peut étudier afin d'en extraire les propriétés structurelles et de caractériser les solutions du problème. Enfin, c'est l'art d'exploiter cette caractérisation afin de déterminer des algorithmes qui les calculent mais aussi de mettre en évidence les limites sur l'efficience et l'efficacité de ces algorithmes ».

En optimisation mono objective, l'ensemble des solutions réalisable est totalement ordonnée selon une fonction objective, le but est de trouver la solution (ou les solutions) optimale.

Cependant, lorsque plusieurs objectifs sont concernés, la situation change, l'ensemble des solutions est en générale non totalement ordonnée mais partiellement ordonnée et donc la solution représentée est la meilleur par rapport aux autres, Elle fournit une meilleur performance avec un cout plus faible. Dans le but d'exprimer cette situation mathématiquement l'optimisation multi objectifs l'ensemble des solutions réalisable est un ensemble souvent numérique pour les quels des valeurs sont à choisir, L'ensemble de ces valeurs est généralement regroupé dans un vecteur représentant une solution.

Les différentes valeurs prise par les variables du vecteur constituent des solutions envisageables.

Pour cela on distingue deux classes de problèmes d'optimisation : des problèmes continus : les variables composant une solution donnée sont de type réel, des problèmes discrets : dans cette classe les variables composant une solution donnée peut être de type naturel, entier ou binaire et puis la principale différence entre l'optimisation mono objective et multi objectif vient de la définition d'optimalité.

4. Principaux concepts en optimisation [smairi, 2013]

Nous allons définir les notions communes à n'importe quelles méthodes d'optimisation multicritère :

(a) **Fonction objective** : c'est la fonction qui modélise le but à atteindre dans le problème d'optimisation sur l'ensemble des critères, elle est appelé critère d'optimisation ou fonction d'adaptation. Il s'agit de la fonction qui doit être optimisé.

(b) Paramètre : est une variable qui exprime les données quantitative et qualitative sur un problème. Ces paramètres correspondent aux variables de la fonction objectif, ils sont ajustés pendant le processus d'optimisation afin d'obtenir les solutions optimales.

Ce sont les variables de conception ou du projet appelé aussi variable d'optimisation.

(c) Vecteur de décision : c'est le vecteur qui correspond à l'ensemble des variable du problème.

(d) Critère de décision : peut-être une variable du problème ou une combinaison de variable.

C'est le critère sur lequel sont jugés les vecteurs de décision pour déterminer le meilleur.

(e) Contrainte : c'est la condition que doivent respecter les vecteurs de décisions du problème.

(f) Espace de recherche : c'est l'ensemble des valeurs pouvant être prise par les variables.

(g) Espace réalisable : correspond à l'ensemble des valeurs des variables satisfaisant les contraintes.

(h) Espace objectif : représente les ensembles des images de l'espace recherche déterminer par les valeurs des fonctions objectifs.

5. Classification d'optimisation

Selon [Xin, 2010] plusieurs critères sont employés pour classifier les problèmes d'optimisation :

- 1- Classification basé sur nombre d'objectifs : cette classification consiste à définir deux catégories de problème d'optimisation : ceux a un seul objectif et ceux a plusieurs objectifs mais la majorité des problèmes réels sont multi objectifs en terme de la fonction.
- 2- Classification en termes de fonction : si les fonctions de contraintes et la fonction objective sont linéaires donc on est devant un problème d'optimisation linéaire sinon on parle d'un problème non linéaire.
- 3- Classification selon les contraintes : dans cette classification on constate deux catégories qui sont : les problèmes sans contraintes et les problèmes avec contraintes d'égalité ou inégalité ou sa peut être les deux à la fois.
- 4- Classification selon l'allure de la fonction objective qui peut admettre un seul optimum local qui est aussi l'optimum global, et donc ici on parle d'un problème

d'optimisation uni modal tant dis que si la fonction objectifs admis plusieurs on parle alors d'un problème multimodal.

- 5- Classification selon les types de variables de décision : lorsque les variables de décision sont discrètes on parle de problème d'optimisation combinatoire (discret).
- 6- Et en dernier on a le problème d'optimisation déterministe qui se reflète dans la définition exacte des variables de décision et de fonctions objectifs. Le problème alors devient stochastique.

6. Formulation du modèle d'optimisation multi objectif [Chelloui & kaidi, 2016]

Auparavant le problème d'optimisation multicritère a été étudié par deux communautés qui sont : la communauté de recherche opérationnelle et la communauté de contrôle.

Les problèmes réels prennent en charge les multiples mesures de performance qui doivent être optimisé. En pratique ce n'est pas toujours possible parce que les objectifs pouvant être conflictuels. Dans ce que cas, la qualité d'un individu décrite par un vecteur.

Un problème d'optimisation multicritère consiste à trouver le vecteur de décision idéale tel que la fonction objectif est optimal. Ces objectifs multiples son souvent concurrente ou lors de l'amélioration de l'un entraîne la détérioration de l'autre ou des autres.

Dans les problèmes multi objectifs, l'optimum n'est plus une simple valeur mais un ensemble de point représentant les meilleurs compromis et ce n'est ce qui fait la différence entre les problèmes mono et multi objectifs.

Selon [Yang, Chen & Yeo, 1999] et [Oliver, 2000] la formulation d'un modèle d'optimisation multi objectif se fait comme suit :

1- Fonction objective

- L'accélération de la nacelle est la 1^{ère} fonction objective qui doit être maximisée.
- Pour déterminer la valeur maximale de l'accélération il faut minimiser alors le terme
- La 2^{ème} fonction objective prise en considération est le conditionnement de la matrice jacobienne (ou indice d'isotropie).

2- Espace de travail et variable de conception

L'espace de travail c'est l'espace de notre recherche et les variables de conception se sont les variables qui seront optimisées par la suite.

3- Contraintes.

7. Méthodes de résolution d'un problème d'optimisation multicritères

La résolution de différents types de problème dans la vie quotidienne à pousser les chercheurs à proposer des méthodes de résolution en fournissant un grand effort pour améliorer les performances et la qualité de la solution proposée, c'est pourquoi de nombreuses méthodes de résolution de problème de différente complexité ont été proposées.

Selon [Gherboudj, 2013], Cette variété et ces différences ont permis de regrouper les différentes méthodes de résolution de différents problèmes en deux sous-classes principales qui sont :

La classe des méthodes exacte et la classe des méthodes approchées et l'hybridation de ces deux classes donne naissance à une nouvelle méthode qui s'appelle la méthode hybride.

Les méthodes exactes garantissent l'optimalité de la solution mais elles sont très gourmandes en terme de temps calcule et de l'espace mémoire et surtout utilisé dans les problèmes faciles.

C'est la raison pour laquelle les chercheurs ont proposé un autre type de méthode approché, de nombreuses méthodes de ce type ont été mise, elles sont pratiques pour la résolution de problèmes difficiles ou de problèmes dont la recherche des solutions en bref délai. Ces méthodes sont classées en deux catégories : heuristique et méta heuristique.

Les méthodes heuristiques sont destinées à un problème particulier, ce sont des règles empiriques qui se basent sur l'expérience et qui nécessite une connaissance sur le domaine du problème traité ainsi que les résultats acquises afin d'améliorer les recherches

7.1 Méthodes exactes [kaidi & chelouti, 2016]

Il y a très peu de travaux sur les méthodes exactes dans le contexte de la résolution des problèmes d'optimisation multi-objectif, sans doute, à cause de la grande difficulté de ce type de problème. Les références existantes et qui présentent la plupart des méthodes exactes sont Ulungu et Ehr Gott et Gandibleux 2000.

- **L'agrégation linéaire**

Cette méthode populaire transforme le problème multi-objectif en un problème mono objectif en combinant linéairement les différents objectifs. Ainsi, le nouveau problème obtenu, car il s'agit alors d'un problème différent, consiste à optimiser $P_i \lambda_i$. Le théorème de Geoffrion indique qu'en utilisant différentes valeurs pour le vecteur λ , il est possible d'obtenir toutes les solutions supportées du problème multi-objectif initial. Par contre, aucune solution non supportée ne peut être trouvée par cette méthode. La méthode d'agrégation linéaire a donc ses limites. Toutefois, elle est intéressante pour des problèmes ayant de nombreux objectifs et/ou un grand nombre de solutions supportées bien réparties. Dans ce contexte, il peut être suffisant de générer les solutions supportées.

- **Méthode e contrainte**

Cette méthode permet de transformer le problème d'optimisation multiobjectif en un problème mono objectif. La méthode consiste à convertir $(m - 1)$ des m objectifs du problème en contraintes et d'optimiser séparément l'objectif restant.

7.2 Les méthodes heuristiques

Règles empiriques simples basées sur l'expérience (résultats déjà obtenus) et sur l'analogie. Généralement, on n'obtient pas la solution optimale mais une solution approchée.

- **L'algorithme glouton** : construction d'une solution réalisable en se ramenant à une suite de décisions qu'on prend à chaque fois au mieux en fonction d'un critère d'optimisation local sans remettre en question les décisions déjà prises. Généralement, la solution obtenue est approchée.

Intérêt : algorithmes simples à implémenter.

Défauts : solutions approchées obtenues plus ou moins bonnes, critère local

- **Recuit simulé** : Donne généralement de bonnes solutions par rapport aux algorithmes de recherche classiques, peut être utilisé dans la plupart des problèmes d'optimisation, il converge vers un optimum global (lorsque le nombre d'itérations tend vers l'infini).

7.3 Méthodes à base de populations ou distribuées

- **Algorithmes hybrides évolutionnistes** : résolution des problèmes complexe, parallélisation, générique et adaptable
- **Algorithme de colonies de fourmis** : Dû à M. Dorigo dans les années 90, s'inspire du comportement collectif des fourmis pour la recherche d'un plus court chemin de la fourmilière à un point de nourriture

7.4 Exemples de problèmes d'optimisation [Gherboudj,A, 13]

Les problèmes d'optimisations peuvent être des problèmes académiques ou industriels. En fait, sur le plan pratique les problèmes industriels sont des projections des problèmes académiques.

Parmi les problèmes d'optimisation académique on a le problème de satisfiabilité booléenne Max-Sat, le problème d'ordonnement, le problème de coloriage de graphe.

En outre, parmi les problèmes industriels, nous citons alors les problèmes de design dans les domaines d'ingénierie, les problèmes d'investissement financier, la planification de trajectoire de robots mobiles ...etc

8. Méthodes de résolution des problèmes d'ordonnement [groupe Gotha, 1993]

Les méthodes de résolution des problèmes d'ordonnement puisent dans toutes les techniques d'optimisation multicritère, ces méthodes garantissent l'optimisation de la solution fournie.

Les algorithmes dans la complexité n'est pas polynomiale ne pouvant pas être utilisé pour les problèmes de grandes tailles, d'où la nécessité d'utiliser ces méthodes approchée efficace pour ces problème NP-difficile.

8.1 Méthodes de résolution approchées

Ce sont des méthodes qui sacrifient le caractère optimale de la solution pour obtenir des solutions optimale de bonne qualité en un temps de calcul raisonnable, Ces méthodes reposent généralement sur un mécanisme de déplacement aléatoire ou non elles ne sont pas exactes mais permettent en général d'obtenir des solutions proche de l'optimum.

8.2 Les méta-heuristiques [Oumar, 2009]

Une méta-heuristique désigne un schéma algorithmique général qui peut s'appliquer à différents problèmes d'optimisation combinatoire. Plus précisément, elle utilise des stratégies qui guident la recherche dans l'espace des solutions, ces stratégies étant indépendantes du problème auquel on les applique. Le but est d'explorer le plus efficacement possible l'espace des solutions afin de ne pas rester bloqué dans les minima locaux et de se diriger rapidement vers les régions les plus prometteuses. Il existe un grand nombre de méta-heuristiques allant de schémas très simples (qui mettent en œuvre des processus de recherche basiques, comme

la descente), à des schémas beaucoup plus complexes (avec des processus de recherche élaborés comme les colonies de fourmis).

Il existe plusieurs méta-heuristique comme la recherche locale, le recuit simulé, la recherche tabou, l'optimisation par colonies de fourmis, les algorithmes génétique.

8.3 Algorithme génétique [Merhoum, Djeghaba, 2006]

Les Algorithmes génétiques s'attachent à reproduire l'évolution naturelle d'individus en respectant la loi de survie énoncée par Darwin. Les principes de base des Algorithmes génétiques développés initialement par Holland pour répondre à des besoins spécifiques en biologie, ont été rapidement appliqués pour résoudre et avec succès les problèmes d'optimisation multicritère en recherche opérationnelle et les problèmes d'apprentissage dans le domaine de l'intelligence artificielle. Portmann, introduit et explique comment appliquer les algorithmes génétiques aux problèmes d'ordonnancement. Dans le cadre de l'application des algorithmes génétique dans un problème d'optimisation multicritère, une analogie est développée entre un individu dans une population et une solution d'un problème dans l'espace de solutions globale.

Dans le cadre de l'optimisation et par analogie, une solution (individu) est codée par une structure de données qui représente son empreinte génétique, on associe à chaque individu une valeur de la fonction critère, celle-ci permet de lui attribuer sa force d'adaptation « fitness » pour simuler la sélection naturelle, de sorte que les solutions pour lesquelles la valeur du critère est plus éloignée de la valeur optimale soient l'exploration de l'espace de solution s'effectue grâce à l'application des mécanismes des opérateurs génétique le croisement et la mutation sur une population initiale définit aléatoirement. Leur déroulement sur plusieurs générations dont le nombre est fixé par le critère d'arrêt, nous permet d'espérer obtenir une solution proche de la solution optimale, ou même une solution optimale le déroulement de l'AG standard peut être résumé comme suit :

- Génération de la population initiale.
- Sélection.
- Reproduction (croisement et mutation).
- Remplacement par la nouvelle population.

9. Structure d'un algorithme génétique

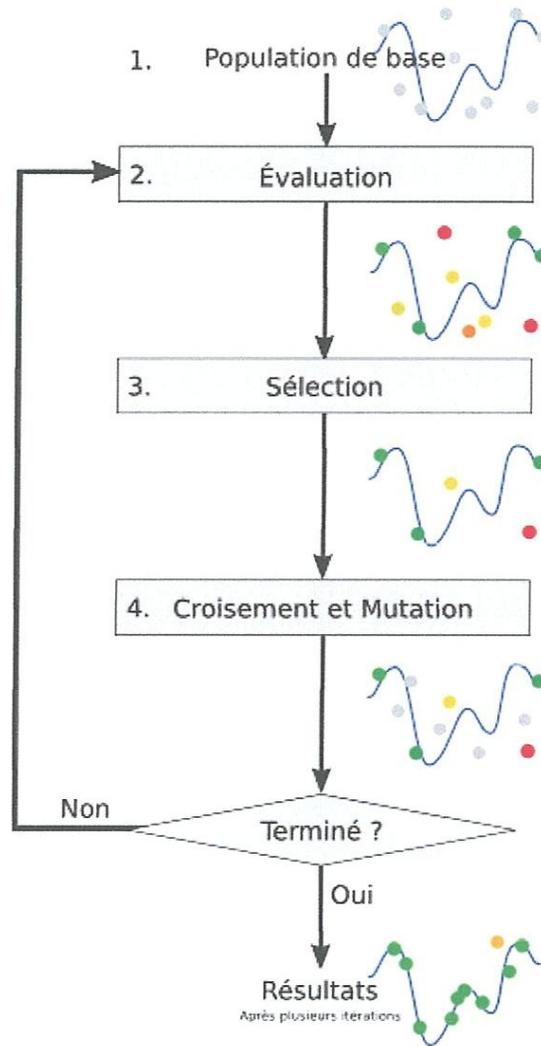


Figure 2.1 : Diagramme d'algorithme génétique.

9.1 Génération de la Population Initiale

Plusieurs méthodes existent pour définir le mécanisme de la génération de la population initiale qui représente le point de départ pour la constitution des générations futures : le tirage aléatoire, les heuristiques ou une combinaison de solution heuristique et aléatoire. Une solution ou l'individu i qui compose cette population est codée par une matrice de séquence d'opérations qui définit l'ordre des opérations que doit effectuer chaque machine comme le montre la table I. Les solutions constituant la population initiale sont choisies aléatoirement dans cette étude.

9.2 Sélection

La phase de sélection consiste à choisir parmi les N individus i de la population courante les plus forts individus à partir desquels la génération suivante sera créée. Soit pour un ordonnancement i réalisable, on calcul la valeur de la fonction objectif pour notre problème c'est effectuer un ordonnancement des tâches. On associe à chaque solution une fonction d'évaluation pour calculer sa force d'adaptation, dans notre cas le problème a pour but de minimiser la fonction objectif, Les individu ainsi sélectionnés constituent une population intermédiaires.

Il existe différente stratégie de sélection, tel que la sélection par la Roue de Loterie (Roulette Wheel Selection) ou La sélection par la méthode de Tournois.

9.3 Croisement :

L'opérateur de croisement est le plus important dans les AGs, il permet d'explorer efficacement l'espace de recherche. L'opérateur de croisement appliquer sur deux individu parent1 et parent2 choisis parmi la population sélectionnée, permet de générer deux nouvelles solutions enfant1 et enfant2 par combinaison des propriétés des parents 1 et 2

9.4 Mutation

Le deuxième opérateur génétique important est la mutation, elle vient en deuxième place sur le plan d'importance par rapport au croisement. Une mutation est une perturbation introduite sur la composante de l'individu afin de garantir la diversité et élargir le champ d'exploration. La démarche suivie dans ce travail consiste à choisir aléatoirement un individu et la machine qui doit subir cette perturbation, ensuite on choisit aléatoirement deux opérations, l'opération consiste à permuter l'ordre entre eux.

9.5 Remplacement

Cette étape constitue la population de la génération suivante à partir des parents et des enfants de la génération courante. Une fraction de la population est remplacée par sa descendance à chaque génération. L'écart entre les générations indique la proportionnel de parents remplacés par des enfants.

9.6 Critère d'arrêt

Le processus est stoppé au bout d'un nombre fixé de génération, ou lorsque les individus ont convergé vers une ou plusieurs solutions satisfaisantes, dans notre étude nous avons opté pour la première démarche.

Les meilleurs individus de la population sont alors retenus comme solutions au problème.

10. Conclusion

Au témoignage du nombre important d'article qui ont été publiés sur les algorithmes génétique, Ces derniers sont très bien adaptés aux problèmes d'optimisation multi objectif. Ce domaine est très dynamique et ne cesse de se développer.

Chapitre 3.

Conception et Implémentation

Chapitre 3

Conception et Implémentation

1. Introduction

Dans ce chapitre nous allons nous focaliser sur un problème classique et connu dans la recherche opérationnelle qui est le problème d'ordonnancement.

Dans notre travail, nous nous limitons aux algorithmes génétiques pour démontrer l'objectif essentiel de cette étude qui concerne l'application de ce type d'algorithme évolutionnistes pour la résolution d'un problème de décision multicritères.

Ainsi, nous présentons les différentes étapes réalisées durant la modélisation et l'implémentation de notre application ensuite nous définissons les fonctionnalités de notre système et enfin nous exposons sur les résultats obtenus.

2. Problème d'ordonnancement [EL Hillal & al, 2009]

Le problème d'ordonnancement consiste à ordonner dans le temps un ensemble de tâches compte tenu de contraintes temporelles tel que les contraintes de délais, contraintes d'enchaînement et les contraintes portant sur la disponibilité et l'utilisation des ressources requises. L'objectif est de minimiser la durée de réalisation compte tenu des contraintes reliant les différentes tâches.

Une tâche : est une entité élémentaire localisée dans le temps, par une durée d'exécution, une date limite d'exécution et une priorité sa réalisation est constituée d'un ensemble d'opérations qui requiert pour son exécution certaines ressources dans laquelle il est nécessaire de la programmer de façon à optimiser un certain objectif.

Les contraintes : expriment des restrictions sur les valeurs que peuvent prendre simultanément les variables de décision. On distingue :

- Des contraintes temporelles.
- Des contraintes de ressources.

Les problèmes d'ordonnancement touchent tous les domaines de l'économie.

Les méthodes de résolutions des problèmes d'ordonnancement puisent dans toutes les techniques d'optimisation multicritère, ces méthodes garantissent l'optimisation de la solution fournie. Notre objectif est de montrer les performances des algorithmes génétiques (métaheuristique) dans la résolution de ce genre de problème

2.1 Variante des problèmes d'ordonnancement

Selon la nature des variables et des contraintes mise en jeu, plusieurs variantes des problèmes d'ordonnancement sont proposées dans la littérature

- Utilisation des réseaux de pétri pour l'ordonnancement hors-ligne optimal des systèmes temps réel.[Grolleau.E et Choquet-Geniet.A,2000].
- Nouvelle méthodes pour les problèmes d'ordonnancement cyclique : [touria chafaquan ben rahho, 2013].
- Algorithmique parallèle hétérogène et technique d'ordonnancement :[olivier.B et yves.R,2003].
- Ordonnancement des ressources humaines : étude du cas d'une entreprise d'injection plastique :[Laur.P et Pière.B, 2001].
- Contribution au raisonnement progressif et temps réel dans un univers multi-agents : [abdel-illah.M,1993].
- Un algorithme génétique pour l'ordonnancement robuste : application au problème du flow shop hybride [Tarek Chaari, 2011].

2.2 Objectif du travail :

Les méthodes de résolutions des problèmes d'ordonnancement puisent dans toutes les techniques d'optimisation multicritère. Ces méthodes cherchent à garantir l'optimisation de la solution fournie.

La méthode optée dans notre travail est l'application des algorithmes génétiques. Ces derniers appartiennent à la famille des algorithmes évolutionnistes. Notre objectif est de montrer les performances de ces méthodes pour la résolution de problème d'ordonnancement de tâches. Le nombre de ces dernières étant très important, notre système doit trouver un ordre d'exécution des différentes tâches de manière à ce qu'il minimise la somme des priorités des tâches qu'ils ne pourront pas être exécutés à temps

3. Base de données

Notre système a une base de données dont l'utilité est de stocker les données nécessaire relié au fonctionnement de notre système. Cette base de données contient un ensemble de tâches à ordonner qui sont généralement conflictuels avec des critères multiples qui sont : temps d'exécution, date limite d'exécution et la priorité.

Les paramètres définissant les tâches (durée d'exécution, une date limite d'exécution et une priorité) sont définis aléatoirement et peuvent donner une solution « toutes les tâches

sont exécutées à temps » pour une quantité de l'ordre de 30 tâches, il est toujours possible de remplir ces paramètres soi-même.

NomTache	TempsExecution	DateLimite	Priorité
Tache1	5	57	2
Tache2	1	140	9
Tache3	4	118	1

Figure 3.1 : Clin d'œil sur la base de données

4. Modélisation

Nous intéressons à un problème d'ordonnancement, notre application doit trouver un ordre d'exécution des différentes taches de manière à ce qu'il minimise la somme des priorités des taches qu'ils ne pourront pas être exécutées à temps, le nombre de taches étant très important on cherche ordre qui ne sois pas forcément le plus adapté mais qui soit parmi les mieux adaptés.

4.1 Modélisation de l'individu

Comme tout système d'aide à la décision celui-ci comporte une solution représenté sous forme d'individu tel que ce dernier est un vecteur de tache trié.

Tri des taches : -> tache3 -> tache5 -> tache2 -> tache4 -> tache6 -> tache1

Codage pour individus : 011000000110011010001110010111

Figure 3.2 : représentation de l'individu

5. Fonctionnement du système

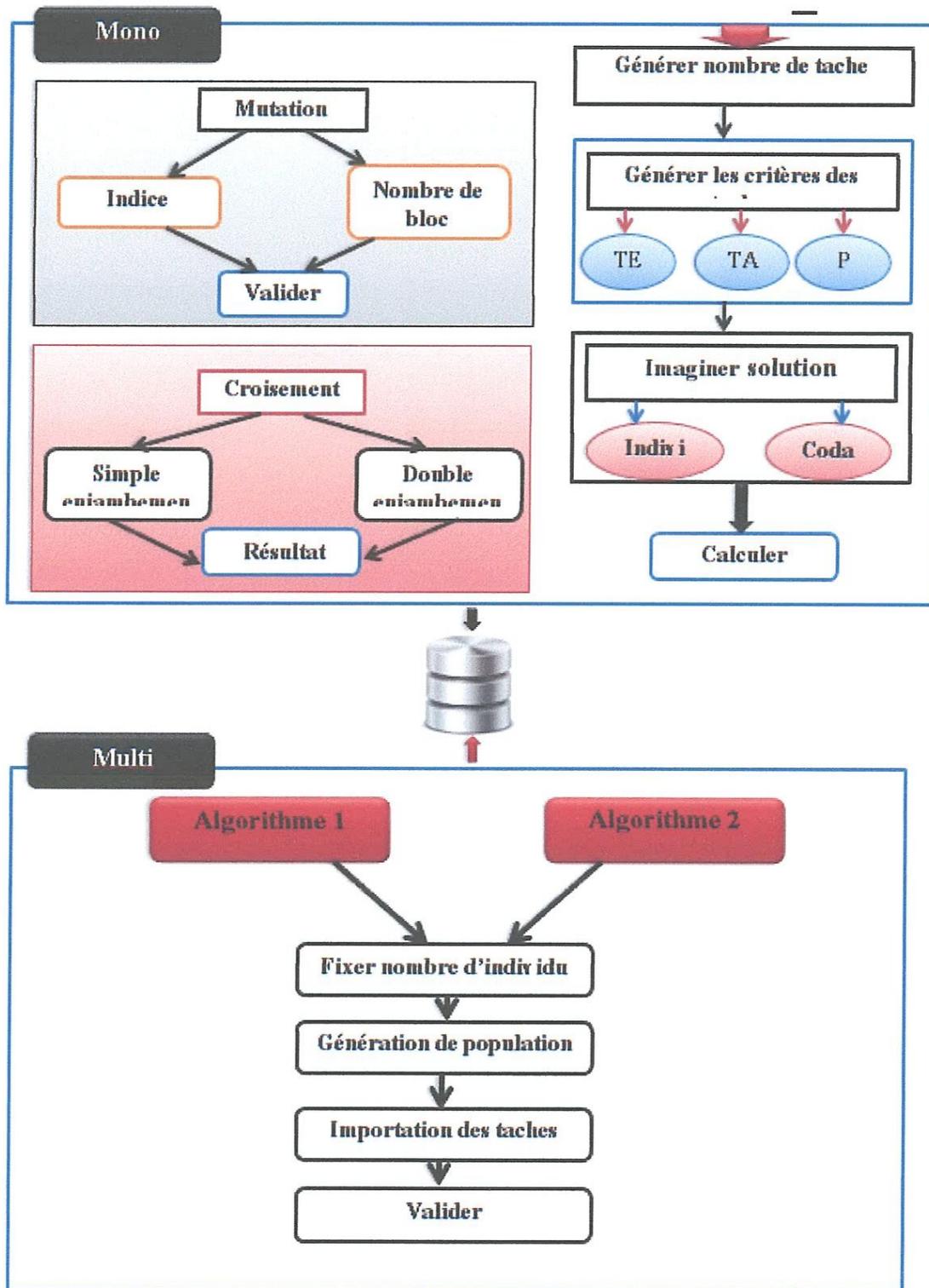


Figure 3.2 : Schéma du processus général du système proposé.

5.2 Scénario du fonctionnement de système

On considère une population d'individus, dans notre cas un individu correspondra à un ordre d'exécution de tâches. Du point de vue génétique l'ordre d'exécution est à chromosome (individu) que la tâche est au gène. En génétique les individus sont des combinaisons de gènes appartenant à un « alphabet » génétique de quatre gènes, dans notre cas l'alphabet génétique correspond aux différentes tâches. Une solution potentielle sera donc un ordre (ou individu).

En utilisant le pseudo code suivant :

Début

Générer population_initiale.

Calculer fitness_individu .

Tant que « critère d'arrêt non satisfait » **Faire**

Générer Population_Sélectionner.

Générer Population_Croisement.

Générer Population_Muté.

Générer Population_Remplacement.

Calculer fitness_individu.

Fin tant que .

Fin.

- A l'initialisation on **génère une population de N individus**, chaque individu est créé aléatoirement de manière à ce que la population soit répartie uniformément sur l'ensemble de recherche. Ensuite trois opérateurs vont être appliqués pour générer la population suivante.
- **L'opérateur de sélection** : Dans un premier temps on va sélectionner les individus suivant leur adaptation au problème, à chaque individu est associé une fitness qui est une probabilité fonction de P : un individu ayant une fitness élevée (P faible) aura plus de chance d'être sélectionné qu'un individu de fitness moins élevée.
- **L'opérateur de croisement** : Il permet de croiser deux individus entre eux : création d'un individu fils à partir de deux individus parents. Cet opérateur retournera une population fille à une population mère. Lors de cette opération, deux individus s'échangent des parties de leurs chaînes binaire, pour donner de nouveaux individus. Ces enjambements peuvent être simples ou multiples. Dans le premier cas, les deux individus se croisent et s'échangent des portions d'ADN en un seul point. Dans le deuxième cas, il y a deux points de croisement. Pour les algorithmes génétiques.

Simple enjambement

Population avant Croisement

```
0100111001011101011010011100010100000111
0111000010111011011011111110111000100011
```

Resultat du Croisement

```
0110111001011101011010011100010100000111
0101000010111011011011111110111000100011
```

Double enjambement

Population avant Croisement

```
0100111001011101011010011100010100000111
0111000010111011011011111110111000100011
```

Resultat du Croisement

```
0110111001011101011010011100010100000111
0101000010111011011011111110111000100111
```

Figure 3.3 Croisement

- **L'opérateur de mutation** : Il modifie faiblement un individu de manière aléatoire. Cet opérateur retournera une population mutée à une population à muter.

L'individu avant

```
0111000010111011011011111110111000100011
```

L'individu après

```
0111011100111011011011111110111000100011
```

Figure 3.4 : Mutation

- **Remplacement**

Cette étape constitue la population de la génération suivante à partir des parents et des enfants de la génération courante. Une fraction de la population est remplacée par sa descendance à chaque génération. L'écart entre les générations indique la proportionnel de parents remplacés par des enfants.

Variante 1

Dans cette version la probabilité de mutation (P_m) n'a aucune importance. La probabilité de croisement (P_c) définit le pourcentage d'individus qui seront issus d'un croisement, le reste de la population sera issue de mutations.

Variante 2

Cette version quant à elle prend en compte la totalité des paramètres, comme dans l'algorithme 1 la probabilité de croisement définie ici aussi le pourcentage d'individus qui seront issus d'un croisement, mais la nuance est que la probabilité de mutation définie quand à elle le pourcentage d'individus issus de mutations, le reste de la population ne sera issu ni de mutation ni de croisement mais sera pioché au hasard dans le vivier de sélection.

6. Implémentation

6.1 Matériel utilisé

Nous avons utilisé un ordinateur Intel (R) celeron (R) CPU N3060 @1.60 GHz avec une RAM de 4 Go et un système Windows 8.1 édition intégral.

6.2 Langage utilisé

Java est un langage de programmation orienté objet et un environnement d'exécution, développé par Sun Microsystems. Il fut présenté officiellement en 1995. Le Java était à la base un langage pour Internet, pour pouvoir rendre plus dynamiques les pages (tout comme le JavaScript aujourd'hui). Mais le Java a beaucoup évolué et est devenu un langage de programmation très puissant permettant de presque tout faire, je dis bien presque car nous verrons pourquoi il ne permet pas de tout faire. Java est aujourd'hui officiellement supportée par Sun, mais certaines entreprises comme IBM font beaucoup pour Java.

6.3 Langage Orienté objet

Java est complètement orienté objet. Java vous permet et vous pousse même à développer vos applications d'une façon orientée objet et vous permet d'avoir une application bien structurée, modulable, maintenable beaucoup plus facilement et efficace. Cela augmente une fois de plus votre productivité.

6.4 Avantages de Java

- Portabilité excellente
- Langage puissant
- Langage orienté objet
- Langage de haut niveau
- JDK très riche
- Nombreuses bibliothèques tierces

- Très grande productivité
- Applications plus sûres et stables
- Nombreuses implémentations, JVM et compilateurs, libres ou non
- IDE de très bonne qualité et libres : Eclipse et Netbeans par exemple
- Supporté par de nombreuses entreprises telles que Sun ou encore IBM et des projets comme Apache

7. Outils utilisé : NetBeans

On a utilisé NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

8. Présentation du modèle proposé

Le programme peut être exécuté en tant qu'application indépendante « java interface ». Dans notre application on va présenter les interfaces graphiques associées aux différentes fonctionnalités du système.

Bibliothèque java utilisée :

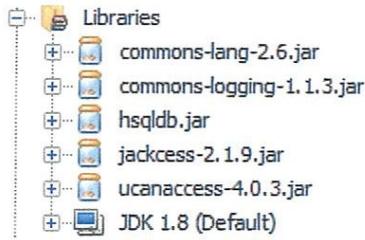


Figure 3.5 : Bibliothèque java utilisée.

Classes utilisées :

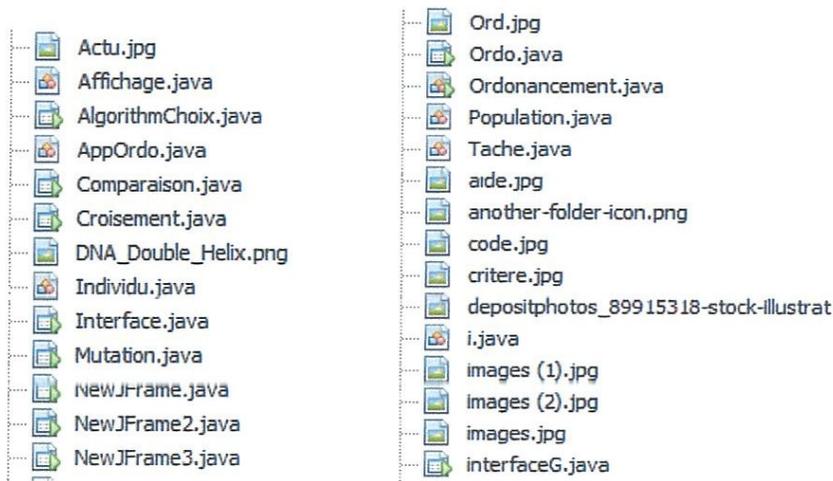


Figure 3.6 : Classes utilisées.

9. Lancement de l'application

- Fenêtre d'accès au système

Après avoir lancé l'application, une fenêtre d'accueil apparaîtra qui demande à l'utilisateur de choisir entre « Mono objectif » et « multi objectif » en cliquant sur l'un des boutons



Figure 3.7: Interface d'accueil.

Afin d'accéder au menu principal de la partie mono objectif de l'application

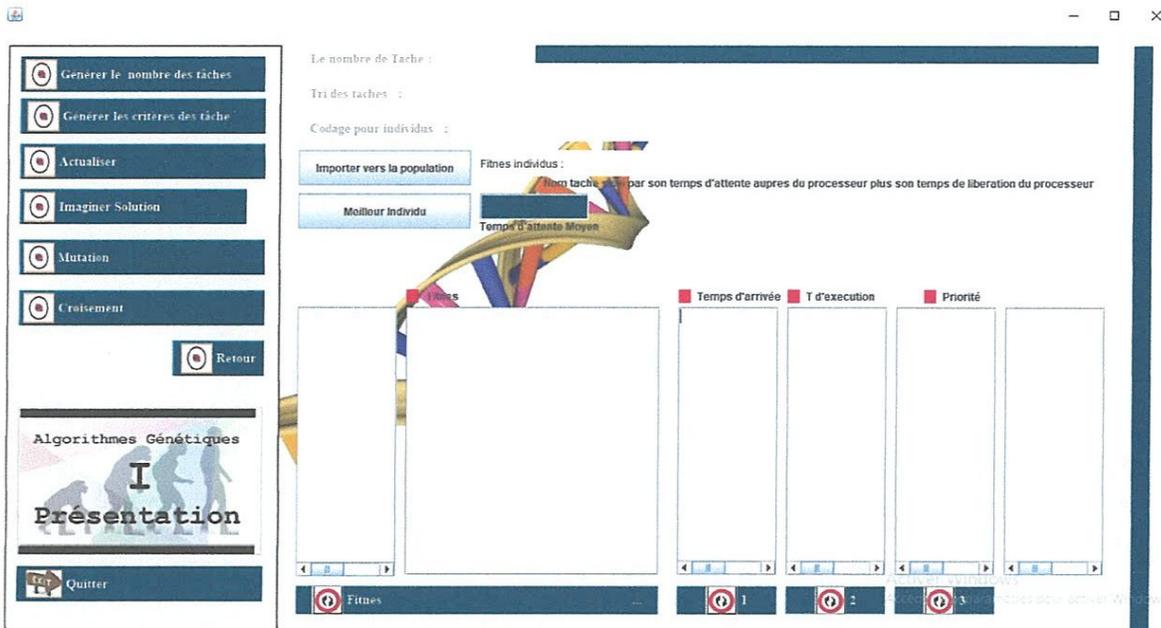


Figure 3.8: menu de la partie mono objective.

Après, on génère un nombre de tâches et les leurs critère à partir de la base de la base de donnée puis on affecte un code binaire différent pour chacune de ces tâches afin de codifier l'individu.

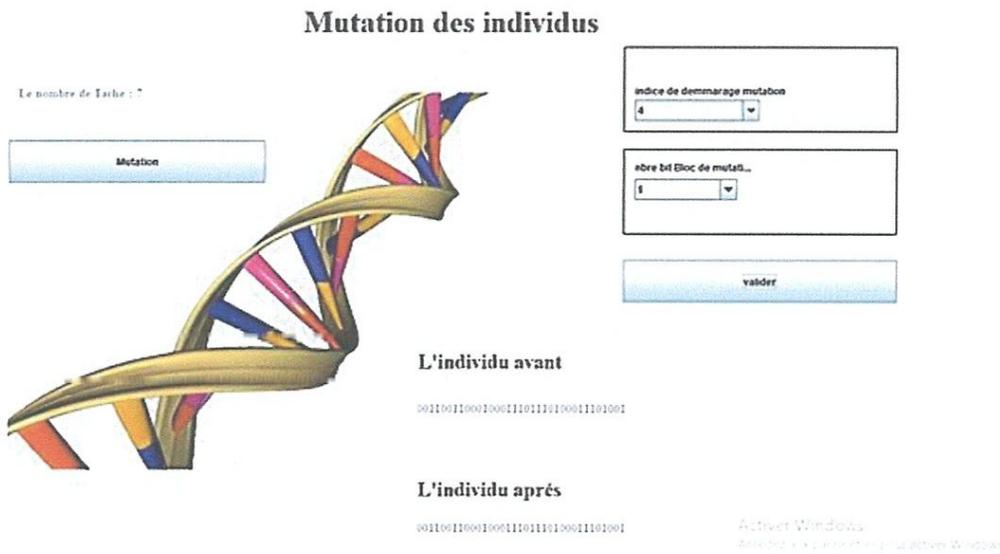


Figure 3.9: Interface de mutation.

Ensuite, on applique l'opérateur Mutation sur l'individu obtenu en choisissant un indice de démarrage et un nombre de bloc de mutation. Cette opération sert à changer le 0 à 1 et le 1 à 0.

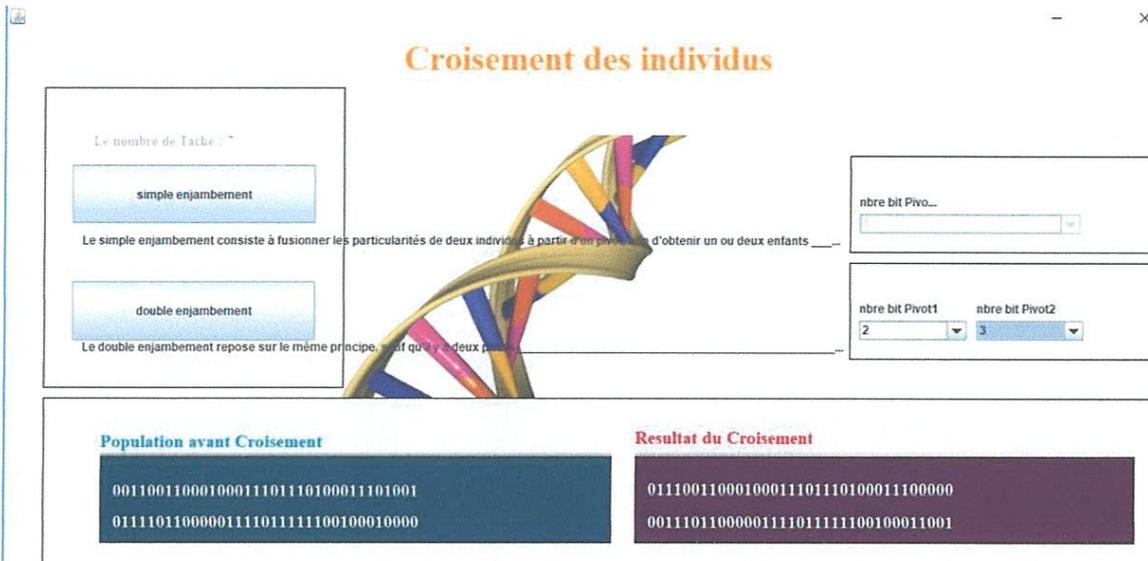


Figure 3.10 Interface de croisement



Figure 3.11 : Interface multi objectif

Ensuite, on aura cette fenêtre de visualisation qui sert à manipuler les différentes fonctionnalités du système qui commence par la génération de la population

Résultats obtenues :

```
Generation:
299
Moyenne des Priorite:4
meilleur Individu:
[ |t17|, |t5|, |t24|, |t9|, |t23|, |t14|, |t8|, |t4|, |t13|, |t11|, |t12|, |t16|, |t22|, |t19|, |t6|, |t3|, |t7|, |t21|, |t18|, |t20|, |t
Plus mauvais Individu:
[ |t16|, |t6|, |t23|, |t4|, |t2|, |t18|, |t17|, |t20|, |t13|, |t5|, |t7|, |t8|, |t12|, |t11|, |t24|, |t15|, |t9|, |t22|, |t21|, |t19|, |t
```

Fenetre de visualisation

Figure 3.13 : Fenêtre des résultats

Les résultats obtenus présente la solution optimale représenter en : Meilleur individu et formule aussi le plus mauvais individu calculer à partir de la fitness.

10. Conclusion

Dans ce chapitre, nous avons essayé d'exposer l'ensemble des idées caractérisant le système proposé en se concentrant sur la conception et l'implémentation de l'application. Nous avons présenté les interfaces et les captures d'écrans de notre application. Les résultats obtenus en appliquant l'algorithme génétique sur le problème d'ordonnancement de tâche

Conclusion générale

Conclusion générale

Les problèmes d'ordonnement se retrouvent souvent dans la vie courante au travers de très nombreuses applications, et suscitent donc un intérêt croissant. Leur but est de déterminer la meilleure manière d'effectuer un ordre sur un ensemble de solutions aux membres d'une population afin d'optimiser un objectif donné. Dans ce travail, nous avons proposé un cadre pour l'optimisation multicritère pour un problème d'ordonnement des tâches utilisant l'algorithme génétique. Nous avons appliqué le principe de recherche d'une meilleure solution.

L'application de l'algorithme génétique a permis de construire une solution de bonne qualité avec un temps raisonnable pour le problème d'ordonnement avec un nombre élevé de tâches. Les algorithmes génétiques semble être une technologie prometteuse et s'adapte mieux aux applications complexes.

En perspective nous proposons d'hybrider les méthodes heuristiques telles que le recuit simulé pour l'amélioration de la solution retrouvée par l'algorithme génétique et aussi comparer les résultats obtenus avec d'autre méthode d'optimisation.

Ce mémoire présente un cadre de travail conceptuel et architectural pour la mise en place d'une bonne optimisation par un système d'aide à la décision multicritère.

Bibliographie

- [Adla 2007] «Architecture Coopérative pour L'Aide à la Décision de groupe Distribuée», Thèse de Doctorat d'Etat. Université d'Oran Es-Sénia, 2007.
- [Espinasse 2009] « Introduction aux Systèmes Interactifs d'Aide à la Décision(SIAD) », 2009.
- [Garlatti 1996] "Tutorial: Multimédia et systèmes d'aide à la décision en situation complexe", in 43th meeting of the european working group "Multicriteria Aid for Decisions", Brest, 1996.
- [Keen & Scott-Morton 1978] "Decision Support Systems: an organizational perspective", Addison-Wesley Publishing, 1978.
- [Le bars 2003], « Un Simulateur Multi-Agent pour l'Aide à la Décision d'un Collectif : Application à la gestion d'une Ressource Limitée Agro-environnementale », Thèse de doctorat. Université Paris IX-Dauphine, France, 2003.
- [Roy 2000] "Réflexions sur le thème : quête de l'optimum et aide à la décision", Cahier du Lamsade n° 167. Université Paris-Dauphine, France.
- [Schneider 1994] " Modélisation de la démarche du décideur politique dans la perspective de l'intelligence artificielle", thèse de l'université de Genève, Suisse, 2000.
- [Simon 1977] "The new science of management decision", Prentice Hall, New Jersey, Etats Unis, 1977.
- [Turban 1993] "Decision Support and Expert Systems", Macmillan Publishing Company, 1993.
- [ROY 2000] Roy, B. (2000). Réflexions sur le thème quête de l'optimum et aide à la décision. *Décision Prospective Auto-organisation*, 2000.
- [Paschos 2005] V. Th. Paschos, Optimisation combinatoire : concepts fondamentaux , Hermès cience publication, pp. 17-60, Lavoisier, 2005.
- [Garey & Johnson, 1979] R. M. Garey et D. S. Johnson. Computers and Intractability. A guide to the Theory of NP-Completness. W.II.Freeman and CO, San Francisco. 1979.
- [Xin 2010] Xin-SheYang, Engineering Optimization: An Introduction with Metaheuristic Applications - Published by John Wiley & Sons, Inc., Hoboken, New Jersey,ISBN 978-0-470-58246-6, 2010.
- [Gherboudj 2013] Méthodes de résolution de problèmes difficiles acad émiq ue s : *Amira Gherboudj*, 2013
- [Yang, Chen & Yeo 1999] Yang G., Chen I-M., and Yeo S-H., Design consideration and kinematic modeling for modular reconfigurable parallel robots, In 10th World Congress on the Theory of Machines and Mechanisms, pages 1079-1084, Oulu, 1999.
- [Oliver 2000] Oliver C., Machines-outils rapides à structure parallèle, Méthodologie de conception applications et nouveaux concepts. Thèse de doctorat, Université de Montpellier II, 2000.
- [Feigenbaum & Feldman 1963] E. A. Feigenbaum and J. Feldman. (Edirors). Computers and thought. McGraw-Hill Inc. pp.192. New York, 1963.

- [Slagle 1971] J. R. Slagle. Artificial intelligence: The heuristic programming approach. McGraw-Hill. pp. 3. New York, 1971
- [Newella 1980] Newella. The heuristic of George Polya and its relation to artificial intelligence. A paper given at The International Symposium on the Methods of Heuristic. University of Bern, Switzerland, Sept. 15-18, pp.16. (Published in Groner et al. (1983), pp. 195-244. 1980.
- [Solso 1979] R. L. Solso. Cognitive psychology. Harcourt Brace Jovanovich, Inc., New York. pp. 436, 1979.
- [Pearl 1984] J. Pearl. Heuristics: intelligent search strategies for computer problem solving. pp. 3. Addison-Wesley Publ. Co, London, 1984.
- [Osman et Laporte, 1996] I.H. Osman, G. Laporte. Metaheuristics: A bibliography. Ann.
- [EL Hilali et al 2009] « Initiation à la recherche opérationnelle », 2009.
- [ZOUAOUI 2011] .Mèmoire de magistère : Approche multicritère pour le problème de sac à dos. USTHB,Algerie,alger, 2011.
- [Zgaya 2007] Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine: Une approche multi-agent pour la recherche et la composition des services liés au transport (Doctoral dissertation, Ecole Centrale de Lille), 2007.
- [smairi 2013] Optimisation par essaim particulaire: adaptation de tribes à l'optimisation multiobjectif (Doctoral dissertation, Université Paris-Est),2013
- [Nachet 2014] Modèle Multi-agents pour la conception d'un système d'aide à la décision collective