

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université M'hamed Bougara de Boumerdès

Faculté des Sciences

Département des Mathématiques



En vue de l'obtention du diplôme de Master en Recherche Opérationnelle

Option : Recherche Opérationnelle, Optimisation et Management Stratégique

THÈME

**Optimisation et planification en matière d'affectation
du matériel**

Présenté par :

M^r. BOUAZZOUNI Abdelkrim & M^{me}. BACHI Louiza

Devant le jury composé de :

Présidente	M ^{me}	N.RAGGAS	M.A.A	U.M.B.B.
Promotrice	M ^{me}	K.KHODJA	M.A.A	U.M.B.B.
	M ^{me}	S.OUATIKI	M.C.B	U.M.B.B.
Examinatrice	M ^{me}	BARAKA	M.A.A	U.M.B.B.
Encadreur	M ^{me}	Z.OUDINA		GCB

Année Universitaire 2020 – 2021

Remerciements

Nous remercions tout d'abord "Allah" tout puissant de nous avoir donné la force et le courage de faciliter et terminer notre modeste travail .

Merci à nos parents de nous encouragés et soutenus tout au long des années d'études .

Un grand merci à notre promotrices Mme **KHODJA Karima** et Mme **OUATIKI Saliha**, pour la confiance qu'elles nous accordées en acceptant d'encadrer ce travail, et pour le temps qu'elles ont consacré à nous enseignant les outils méthodologiques indispensables à la conduite de cette thèse, nous exprimons toute notre gratitude à Mme **KHODJA** pour sa patience , sa disponibilité, et surtout ses judicieux conseils .

Nous tenons également à remercier les membres de GCB qui sont orientés notre travail, et surtout notre encadreur Mme **OUIDINA Zineb** pour avoir la patience de répondre à nos innombrables questions et faciliter la compréhension de problématique de travail.

Nous tenons à remercier sincèrement les membres du jury qui me font le grand honneur d'évaluer ce travail.

Enfin un grand merci à toutes les personnes ayant participés de près comme le loin à la réalisation de ce mémoire .

Binôme :**BOUAZZOUNI et BACHI**

Dédicace

Au moment de mettre un point final à ce mémoire je veux remercier mes parents, à mon cher père et ma mère bien aimée .

Je dédie ce travail à ma sœur **Yasmina** et mon seul frère **Zin Eddine**.

Je tiens à remercier tous ceux qui m'ont encouragé et soutenu de la famille **Bachi** et **Ben mammar**. Surtout mes oncles **Ferhat** ,**Daradji** et leurs femmes , ma tante **Sabiha** et son mari, ma cousine le docteur **Farida**.

Je ne peux pas oublier avec gratitude tous les professeurs qui m'ont enseigné à ce jour.

Un dédicace spécial pour mes amies **Naima** et **Nadine**.

Enfin je garde les meilleures pour la fin, merci beaucoup et toute ma gratitude à ma belle et chère amie **Ikram Mechiet** et ma petite chère soeur **Amina** pour leurs soutien, leurs encouragements et leurs prières spéciales pour moi, et un merci tout spécial à mon binôme de ce travail **Karim** pour toute l'aide et les encouragements.

Je tiens aussi à remercier tous ceux qui ont de prés ou de loin, aide à réaliser ce travail que ce soit par des idées ou par les encouragement.

Louiza

Dédicace

Je dédie ce travail a mes très chères parents qui représentent pour moi une source de tendresse, de gentillesse et d'amour, pour tout leur dévouement et leur encouragement qui n'a jamais cessé, que dieu vous protège et vous donne longue vie dans le bonheur.

A mes très chères frères et sœurs ainsi qu'a tous mes neuves et nièces.

A mon beau-frère **Kaid** qui nous a aidés pour réaliser ce travail.

A mes amis en gage de l'amitié qui nous unit et des souvenir qui nous relit.

A mon binôme **Louiza** un merci spécial pour tout l'aide et les encouragements.

Abdelkrim

Sommaire

Introduction général	13
1 Présentation de l'entreprise GCB	15
1.1 Création	15
1.2 Ressources	16
1.2.1 Ressources humaines	16
1.2.2 Ressources matérielles	16
1.3 Infrastructures	16
1.4 Activités de GCB	19
1.5 Organisation	20
1.6 Les partenaires	21
1.7 Analyse SWOT	22
1.7.1 Les points forts	23
1.7.2 Les points faibles	23
1.7.3 Les menaces	24
1.7.4 Opportunités	24

2	Problématique	26
2.1	Politique actuelle de gestion du matériels	26
2.2	Positionnement du problème	28
2.3	Conclusion	31
3	Problèmes d'optimisation et technique de résolution	32
I	Problèmes d'optimisation	33
3.1	Définition d'un problème d'optimisation classique (mono-objectif) :	35
3.2	Classification des problèmes d'optimisation	35
3.2.1	Problème d'optimisation sans contraintes	36
3.2.2	Problème d'optimisation avec contraintes	36
3.2.3	Problème d'optimisation continue	36
3.2.4	Problème d'optimisation convexe	37
3.2.5	Problème d'optimisation stochastique et robuste	37
3.2.6	Problème d'optimisation linéaire	37
3.2.7	Problème d'optimisation non linéaire	37
3.2.8	Problème d'optimisation quadratique	38
3.2.9	Problème d'optimisation discrète (combinatoire)	38
3.2.10	Problème d'optimisation multi-objectifs	42
3.3	La complexité algorithmique	47
3.3.1	Classes des problèmes	48

II	Méthodes de résolution des problèmes d'optimisation	50
3.4	Introduction	51
3.5	Méthodes de résolutions exactes	53
3.6	Les méthodes approchées (Méta-heuristique)	55
3.6.1	Méta heuristique à base de solution unique	55
3.6.2	Méta heuristique à base de population de solutions	59
4	Modélisation	66
4.1	Introduction	66
4.2	Les hypothèses de GCB	69
4.2.1	Le temps de transfère d'engins	69
4.2.2	La disponibilité des engins et satisfaction des demandes	69
4.2.3	La distance entre deux projets (chantiers)	69
4.2.4	Les perturbations	70
4.2.5	Temps de montage et démontage	70
4.2.6	L'importance des projets	70
4.3	Le modèle mathématique	71
4.4	Les données fondamentaux	72
4.4.1	Les indices utilisés	72
4.4.2	Les variables de décision	73
4.4.3	Les contraintes	73
4.4.4	Les fonctions objectifs	75

4.4.5	Récapitulatif du modèle mathématique	76
4.5	Évaluation et critique du modèle	76
4.5.1	Nombre des variables et des contraintes	77
4.5.2	Caractéristiques du modèle mathématique	78
5	Méthode choisie et implémentation du résultats	79
5.1	Heuristique	80
5.1.1	Justification du choix	80
5.1.2	Le principe de l'heuristique « plan »	80
5.1.3	Principe d'application	82
5.2	Méta heuristique (algorithme génétique)	83
5.2.1	Justification de l'utilisation de NSGA II	83
5.2.2	Stratégie d'application de NSGA II	84
5.2.3	Adaptation de NSGA II à notre problème	88
5.2.4	Création de la première population	90
5.3	Implémentation et résultats	100
5.3.1	Introduction	100
5.3.2	Présentation de "python"	100
5.3.3	Choix de langage "python"	101
5.3.4	Présentation du programme « GPLAN »	102
5.3.5	Observations sur les résultats	106
	Conclusion	107

Bibliographie

110

Table des figures

1.1	Image représentant un conducteur saibom	17
1.2	Image représentant un bulldozer	18
1.3	Image représentant des camions d'entretiens	18
1.4	Image représentant un chargeur sur pneu « Caterpillar »	19
1.5	Organigramme de l'entreprise « GCB »	21
1.6	Tableau représentant la forme d'analyse SWOT.	23
2.1	Un organigramme qui présente la politique actuelle de la société	28
3.1	Image présente les relations entre les espaces	44
3.2	Exemple de front de paréto pour deux fonctions objectifs. [1]	46
3.3	Fonctionnement général d'un algorithme évolutionnaire.	61
3.4	L'organigramme d'un algorithme génétique [3]	63
3.5	Comportement des fourmis	65
5.1	Cas hypothétique de l'assignation de la Crowded distance.[51]	85
5.2	comment ce fait la représentation d'une solution	89

5.3	Menu « GPLAN »	103
5.4	Affichage des chantiers	104
5.5	liste d'engins	105
5.6	Planification d'affectation du matériels	106

List of Algorithms

1	Algorithme de la méthode ϵ contraintes [32]	55
2	Algorithme de Metropolis [50] :	56
3	Algorithme de recuit simulé [50] :	57
4	Algorithme de recherche Tabou [33] :	59
5	Algorithme Génétique	62
6	Algorithme Heuristique	81
7	Opérateur d'assignation de la Crowded distance [15]	87
8	NSGA-II [41]	88
9	Algorithme génér popul initiale	91
10	Algorithme Ranking	92
11	Algorithme RAND	95
12	Algorithme croisement	97
13	Algorithme mutation	99

Introduction général

Pour l'évolution de n'importe quelle société il faut que cette dernière prend des décisions et des initiatives importantes, l'une d'elle est l'acquisition des projets, en effet il faut avoir de plus en plus de projets au file des années pour pouvoir se développer et faire grandir la société, ainsi que son bénéfice, mais à un moment donné ce point de force va causer des problèmes, en effet pour un grand nombre de projets et une mauvaise gestion du matériels la société sera confronté à des problèmes, les plus importants d'entre eux est le dépassement du délai de réalisation de ces projets et des dépenses colossales, ceci entraîne une perte que ce soit du côté physique (perte d'argent, usage de machines) ou du côté morale (mauvaise réputation). Pour que la société fait face à tous cela il faut qu'elle se dote d'outils d'aide à la décision, intégrant des techniques et des méthodes de recherche opérationnelle pour résoudre ces problèmes.

La société GCB (société national de génie civil et bâtiment) s'occupe de la réalisation de projets de génie civil et détient actuellement à son compte un très grand nombre de réalisation dans divers domaines, comme toute autre société GCB est confrontée à un problème très important qui est l'optimisation de la planification en matière d'affectation du matériels.

Ce dernier est un classique en optimisation combinatoire .

Introduction général

En recherche opérationnelle, le problème d'affectation consiste à attribuer au mieux de matériels à des projets, des tâches à des agents,...,ect.

Dans le cadre de ce mémoire, nous avons adapté le problème d'affectation classique, auquel nous avons greffé les contraintes spécifiques de la société GCB en vue de l'optimisation de la planification en matière d'affectation du matériels.

Notre étude est structuré en cinq chapitres conformément à ce qui suit :

Chapitre 1 : Présentation générale de la société GCB (création, organisation, activités,...,ect.)

Chapitre 2 : Énoncé du problème étudié, objectifs assignés à l'étude et la description de l'affectation actuel.

Chapitre 3 : État de l'art de ce qui a été entrepris dans la littérature permettant de positionner les axes de notre sujet. Il traitera plus particulièrement des problèmes d'affectations et différents méthodes de résolution.

Chapitre 4 : Présentation de la modélisation mathématique et évaluation du modèle.

Chapitre 5 : Méthode de résolution utilisées et analyse des résultats obtenus. Traitent également l'environnement de programmation et implémentation du logiciel.

Enfin nous terminerons notre mémoire par une petite conclusion.

Présentation de l'entreprise GCB

1.1 Création

Société Nationale de Génie Civil et Bâtiment par abréviation GCB est une Société issue de restructuration de Sonatrach créée 1^{er} Août 1981 à partir de deux directions : Travaux et Constructions (DTC) et Travaux Pétroliers (DTP) de la SONATRACH. Son démarrage opérationnel n'a eu lieu qu'en janvier 1983.

GCB est érigée en société par actions le 21 juillet 1998 avec un capital social de 2.000.000.000 DA détenu entièrement par Sonatrach. Le 30 décembre 2007, la société a assisté à une augmentation de son capital à 7.630.000.000 DA.[35] Le capital de la société a dépassé 8160000000 DA après l'approbation de l'assemblée générale extraordinaire du 27 décembre 2018.

GCB est une entreprise au rayonnement large et grâce à ses 35 années d'expérience, elle a su rassembler des compétences, gagner la confiance de ses nombreux partenaires et satisfaire ses clients, elle accompagne effectivement la société mère Sonatrach dans l'élaboration de son plan de développement.

GCB tire sa force de la valorisation de ressources humaines, de la diversification réussie de son offre d'activités et de sa politique stricte de protection des personnes, des biens et de l'environnement.

1.2 Ressources

1.2.1 Ressources humaines

GCB emploie 15000 travailleurs, la plupart spécialisés dans le domaine des grands travaux d'aménagement et de construction, notamment au Sahara.

1.2.2 Ressources matérielles

GCB dispose d'un parc conséquent de plus 4465 équipements dont 3500 engins et équipements lourds et moyen de production qui sont constamment modernisés donnant à l'entreprise une grande autonomie d'actions.

1.3 Infrastructures

GCB dispose d'une infrastructure stable à des fins industrielles, administratives, sociales et professionnelles, comprenant :

- Engins de terrassement et de levage.
- Véhicules de transport lourds.
- Stations de production d'agrégats, de béton, d'asphalte,...,ect.
- Divers autres matériaux spécifiques qui lui permettent d'effectuer tous travaux.

Ils sont situés de manière significative dans :

- Alger (les régions industrialisées d'El Harrach et de Wadi Samar).

- Boumerdes (Boumerdes et Boudouaou).
- Arzew (zone industrielle d'Arzew).
- Hassi - Masoud - Rordnos - Nazla
- Hassi Raml - Islah - Adrar.
- In Amenas - Illizi - Tin Foye - Tabankort - Adrar.

Nous allons maintenant présenter quelques images des engins lourd que cette société possède :



FIGURE 1.1 – Image représentant un conducteur saibom



FIGURE 1.2 – Image représentant un bulldozer



FIGURE 1.3 – Image représentant des camions d'entretiens



FIGURE 1.4 – Image représentant un chargeur sur pneu « Caterpillar »

1.4 Activités de GCB

Ses activités se sont diversifiées ces dernières années et sont : Activités de Terrassement, Routes et Autoroute, Piste d'atterrissage et plate-forme pétrolières, Génie Civil (industriel, hydraulique, pipe-line, ferroviaire et bâtiments), Fabrication et montage de Charpente métallique, l'engineering.

GCB opère dans les différents corps de métier de génie-civil :

- L'Engineering et Procurment.
- Les Terrassements en grande masse.
- Les Plates-formes de forage et pistes d'accès.

- Le Génie-civil industriel des grandes installations pétrolières et gazières.
- Les Travaux de canalisation.
- Les réalisations des routes et autoroutes.
- Les réalisations de pistes d'atterrissage et aérodromes.
- Les ouvrages hydrauliques et transferts.
- Le Bâtiment et VRD (Voiries et Réseaux divers).
- Le Génie-civil ferroviaire.

1.5 Organisation

La Société est organisée en structures régionales polyvalentes dont chacune dispose des capacités matérielles et managériales pour répondre aux besoins des clients de GCB dans sa région.

Ces structures régionales ont l'appui des structures support en matière de matériel de transport, moyens généraux, techniques et financiers en fonction des besoins tout en jouant un rôle régulateur pour optimiser les moyens de la Société.

GCB dispose d'une direction d'engineering, d'architecture, topographie et d'une direction de réalisation d'ouvrages métalliques.

L'organigramme de l'entreprise GCB qui décrit ses différentes directions se présente dans la figure qui suit :

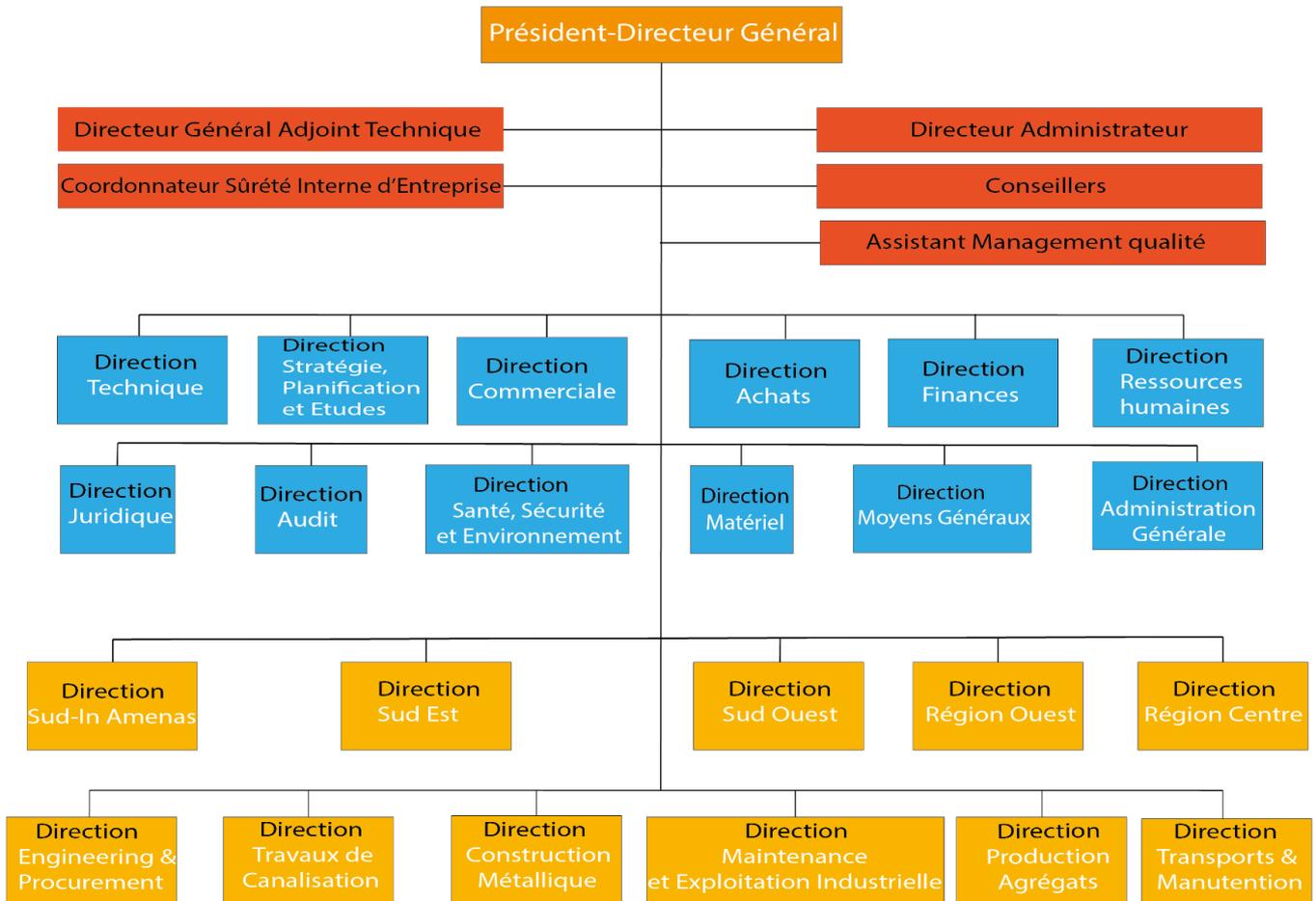


FIGURE 1.5 – Organigramme de l'entreprise « GCB »

1.6 Les partenaires

Le partenariat de qualité de GCB lui permet une meilleure adaptation à son environnement.

Par sa présence sur les principaux pôles économiques, GCB entend construire des relations étroites et permanentes avec ses clients, avec pour seul objectif de réaliser leurs ouvrages au meilleur prix, dans les délais requis et selon les exigences de la profession. Cela lui a valu de compter dans son portefeuille clients, de grandes Sociétés nationales et internationales telles :

- Sonatrach et ces filiales comme : NAFTAL.

- Les associés de sonatrach comme : SH/BP/STATOIL, SH/AAC, GROPEMENT TFT, GROUPEMENT TIM, GROUPEMENT BRS, GROUPEMENT REGGANE.
- L'administration comme : DTP Algérie, ANBT, ANESRIF, ONA.
- Les grandes firmes comme : KBR, BP, REPSOL(Espagne), SADEMI (Italie) , JGC (japon),...,ect .

1.7 Analyse SWOT

Le SWOT (Strengths - Weaknesses - Opportunities - Threats) est un outil très pratique lors de la phase de diagnostique stratégique, il inclut deux axes : axe interne et axe externe [24].

1. **Axe interne** : Le diagnostique interne présente les forces et faiblesse d'une entreprise.
2. **Axe externe** : Le diagnostique externe présente les éléments qui ont un impact possible sur l'entreprise, il permet d'évaluer les opportunités et les menaces générée par son environnement .

L'analyse SWOT permet de synthétiser tous ces éléments dans un simple tableau qui prend la forme suivante :

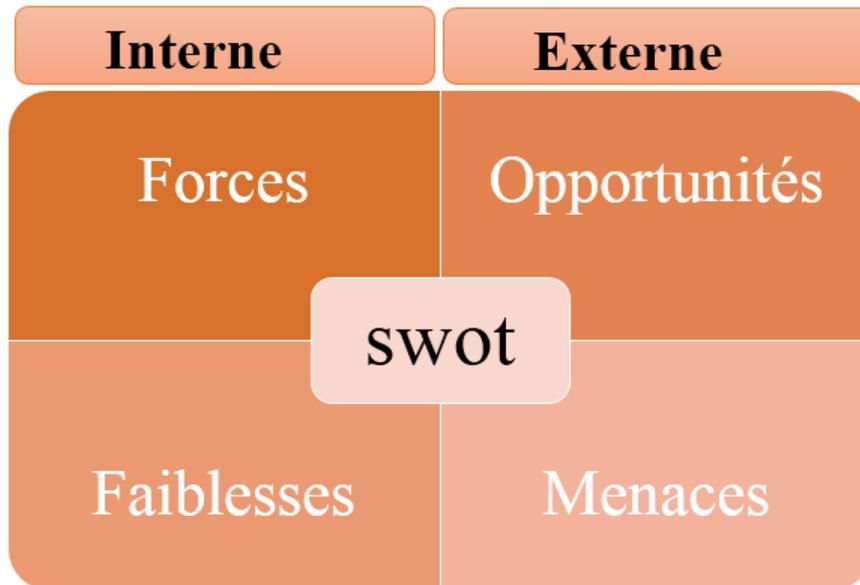


FIGURE 1.6 – Tableau représentant la forme d'analyse SWOT.

Nous pouvons résumer l'analyse SWOT de la GCB comme suite :

1.7.1 Les points forts

- GCB a plus de 40 ans d'expérience dans son domaine.
- une Flotte (Parc de Matériel) en très bon état.
- GCB a des activités qui couvrent une importante partie du territoire national (Implantation géographique importante) notamment dans le grand Sud de l'Algérie .
- Elle a une main d'ouvre qualifié .

1.7.2 Les points faibles

- Bien que l'entreprise possède un grand nombre de matériels de différents types , elle fait toujours appelle à la location pour couvrir les besoins de ses projets .
- GCB présente un nombre des pannes élevés et malheureusement la maintenance de la société

n'est pas suffisante pour les réparations.

1.7.3 Les menaces

- La concurrence .

1.7.4 Opportunités

- L'environnement politique favorise la production nationale.
- GCB évolue dans un environnement caractérisé par un marché clé en main, offrant des opportunités intéressantes.

GCB répond au nouvel environnement pour les cinq (05) ans à venir par :

- Le développement des capacités d'engineering et de procurement pour élargir son champ d'activité et se muer en une Entreprise EPC (Équipement de protection collective).
- La focalisation sur le génie civil comme métier de base et avec le développement de son expertise dans ce domaine par la consolidation des compétences actuelles.
- La diversification de son portefeuille clients dans de nouveaux marchés et secteurs dans ses métiers de base.
- La diversification de son portefeuille activités par l'intégration de nouveaux créneaux qui se traduit par ce qui suit :
 - Le développement de l'activité travaux de canalisation petit diamètre.
 - La création d'une structure production des agrégats.
 - Le développement de l'activité transport et manutention pour prendre en charge l'important volume de transport de matériels et marchandises de l'entreprise et élargir la prestation au groupe Sonatrach.

- La fabrication de la cabine Saharienne pour équiper ses bases de vie et satisfaire la demande de la société mère Sonatrach .

1

Problématique

2.1 Politique actuelle de gestion du matériels

Après nos recherches au sein de la société, nous avons appris qu'il y a plusieurs principaux décideurs dans la gestion du matériel, et sont placés sur deux niveaux différent, ce sont en effet les directions des différentes structures et la direction technique, qui jouent un rôle majeur dans la prise des décisions (soit valider les demandes du matériels, soit faire des demandes de locations).

Chaque structure possède un nombre de chantiers qu'ils lui sont attribués et c'est à la structure de les gérer.

Au début de la procédure de réquisition de matériel, c'est au chef de chantier de faire une demande à la direction de la structure à n'importe quelle période de l'année et il déterminera quels matériels le chantier en a besoin, les employés de cette dernière vont faire une recherche du matériels au niveaux des chantiers de la structure. Si la demande peut être prise en charge (c'est à dire le matériels en question est disponible) alors la direction va valider la demande et affecte les matériels au chantier sinon, si tous les matériels qui sont à la disposition de la

structure sont occupés alors la direction de la structure va faire une demande à la direction technique pour répondre à la demande du chef de chantier, cette dernière va procéder à la résolution de ce problème en parcourant trois étapes, la première étape consiste à voir s'il y a du matériel qui vient de département de maintenance, MTL (autrement dis les engins réparés), si la direction technique ne peut pas satisfaire la demande alors elle aura recours à la deuxième possibilité qui consiste à faire une recherche approfondis dans toutes les autres structures pour satisfaire la demande sinon une seule option reste qui peut couvrir cette demande c'est de faire appel à la location.

Nous citons que le matériel dans le tout début de la période se trouvent au niveau des chantiers et même après l'achèvement des projets le matériels restent au niveau de la direction des structures en attente pour être affectés à d'autres projets, et que le transfère du matériels se fait entre les chantiers et les structures (n'importe quelle structure peut transférer un matériel à n'importe quel chantier) sachant que le seul lien entre ces structures reste administratif c'est à dire le parc d'une structure ne va pas transférer du matériel à un autre parc d'une autre structure.

En ce moment la société n'utilise aucun logiciel pour affecter son matériels aux projets. Cependant elle a créé un logiciel propre à elle, son nom est "GMAINT", ce logiciel n'est pas conçu pour la gestion de matériels en thème d'affectation du matériel au chantier, mais il joue un rôle majeur dans le suivi de matériels en effet à l'aide des informations fournis sur le matériel (modèle, marque, code,...,etc), ils peuvent connaître l'endroit où se trouve le matériel, et quelle est son état dans la période présente et même les précédentes.

- Organigramme de la politique actuelle de la société

Un organigramme qui présente la politique actuelle de la société :

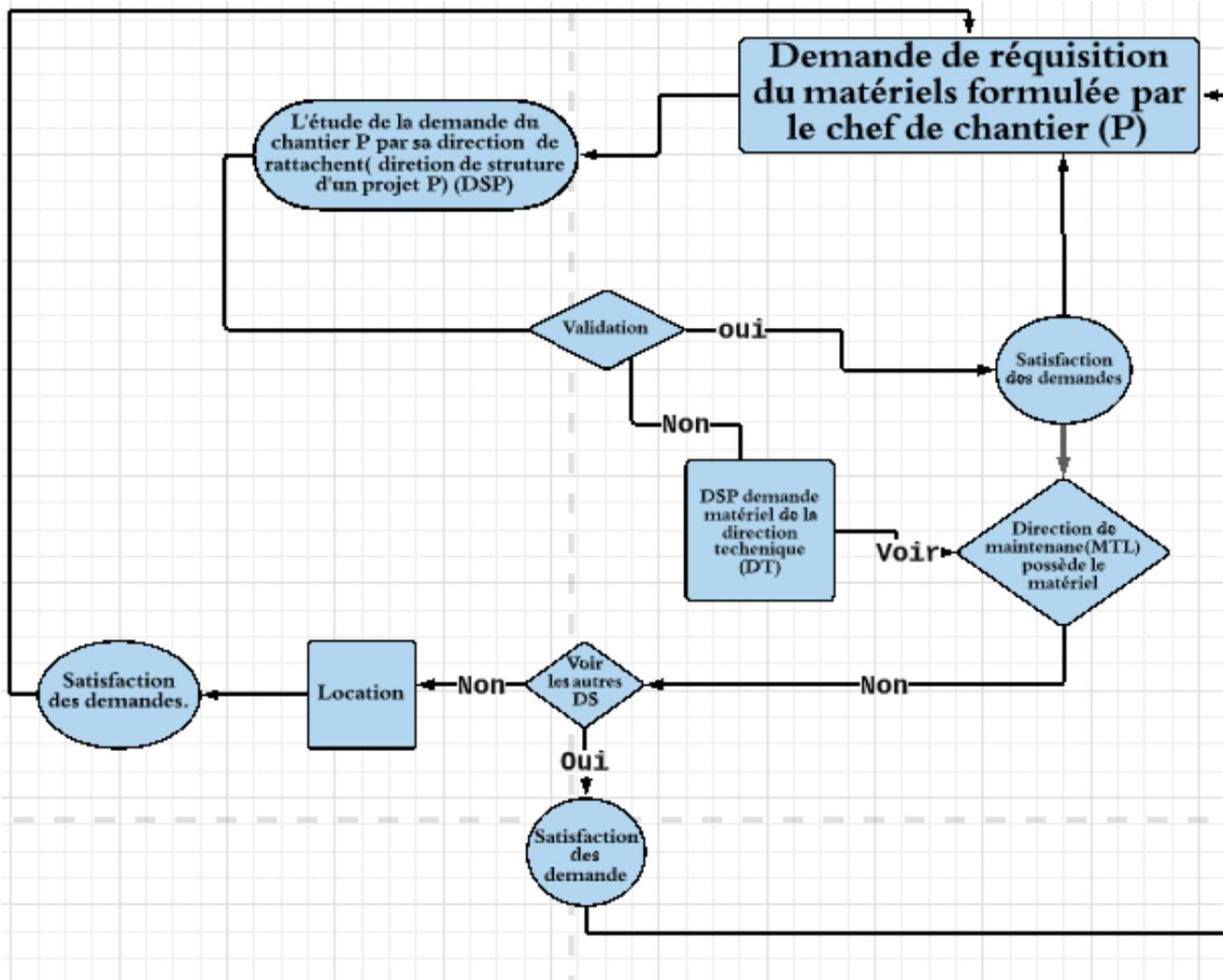


FIGURE 2.1 – Un organigramme qui présente la politique actuelle de la société

2.2 Positionnement du problème

Pour pouvoir suivre le développement de l’Algérie de façon général, l’entreprise GCB devrais faire face aux différents problèmes et elle sera poussée à les surmonter, pour cela elle se trouve obliger d’acquérir plus de projets et d’investir dans tous les domaines.

L'entreprise GCB est confrontée à deux problèmes qui influencent son développement et le bon déroulement de ses projets, ces problèmes sont le problème de dépassement de délais de réalisation des projets et problème de dépense (coût) excessive. Ces problèmes sont dus à une mauvaise gestion des matériels. Pour améliorer la planification et ainsi essayer de résoudre ces problèmes nous nous sommes intéressés au problème d'affectation des matériels arrêté par l'entreprise GCB .

Les motifs exposés pour justifier ce choix sont notamment le manque important dans la couverture simultanée de toutes les demandes des chantiers en équipement, ce qui mène à un manque à gagner considérable .

L'entreprise va affecter le matériel Selon le besoin de ses structures, elle va satisfaire les demandes de ces dernières en utilisant sa propre flotte d'engins s'il y aura d'engins libres, dans le cas contraire l'entreprise aura recours à la location, cette procédure ne prend en considération aucune contraintes.

Les problèmes qui se posent dans la politique actuelle de la société sont : le non suivi de la situation des matériels présents dans les projets et une mauvaise affectation des matériels dû à des mauvaises informations fournies par les différentes structures, en effet l'affectation est basée sur ces dernières ainsi si ces informations ne reflètent pas la réalité, ceci engendra des pertes, soit sur la durée de réalisation, soit sur les dépenses, à titre illustrative considérons l'exemple suivant : si un chantier aura besoin d'un matériel, le chef de chantier va faire une demande à la direction de la structure, et cette dernière va à son tour faire une recherche du

matériel en question dans les projets voisins, il arrive parfois qu'un matériel libre est déclaré comme actif pour des besoins dans le future, ainsi si la direction de la structure ne peut pas satisfaire cette demande alors elle va faire une demande à la direction technique qui va le satisfaire dans tous les cas défigure. Les mauvaises informations requissent et l'application de cette procédure pour satisfaire cette demande vont augmenter le temps de réalisation et les coûts des projets, ce résultat va l'encontre de notre objectif (minimiser la durée et le coût du réalisation des projets).

Dans ce problème chaque type d'engin est indépendant des autres types, dans ce cas si un matériel quelconque soit affecté a un chantier quelconque (suite a une demande de ce dernier), dès son arrivé au chantier il va commencer son travail sans se soucier des autre types, ce point va aider dans la modélisation du problème, ainsi le modèle mathématique va traiter et résoudre notre problème pour chaque type a part.

Pour résoudre ce problème nous allons essayer de faire un plan d'affectation du matériels en se basant sur la planification de chaque projet élaboré par la société et traiter chaque type séparément. Ce plan va exploiter au maximum les ressources matériels de la société ainsi nous allons essayer de minimiser l'appel à la location et minimisé le dépassement du délai de réalisation des projets et pourquoi pas les terminés avant le délai et de cette façon avoir plus de projets dans le future.

2.3 Conclusion

A partir de cela, nous pensons que les techniques de recherche opérationnelle vont nous aider à élaborer un modèle mathématique de ce problème et ainsi construire un logiciel informatique qui nous donnera un plan d'affectation idéal (optimal) en utilisant les demandes de chaque structure selon une période définie, dans le but de satisfaire les demandes dans les plus brefs délais ce qui entraîne la minimisation de la durée de réalisation des projets et évite de faire des demandes de location ce qui résulte la minimisation du coût des projets.

Chapitre 3

Problèmes d'optimisation et technique de résolution

Introduction

Ce chapitre est organisé en deux parties : dans la première partie nous présentons l'optimisation mathématique et plusieurs classes de problèmes d'optimisation, dans la deuxième partie nous présentons les différentes méthodes de résolution (exactes et approchés).

Première partie
Problèmes d'optimisation

Introduction

L'optimisation d'un point de vue globale est le fait d'améliorer ou de perfectionner soit une méthode soit un comportement soit une rentabilité d'une façon plus générale c'est faire en sorte d'avoir le meilleur résultat possible.

L'optimisation est une branche des mathématiques, principalement un outil d'aide à la décision (recherche opérationnelle) [28], son but est de modéliser, d'étudier et de résoudre les problèmes de minimisation (respectivement maximisation) d'une certaine quantité appelée objectif.

Elle joue un rôle très important dans plusieurs domaine scientifique comme les mathématiques appliquées en général (statistique), l'informatique (optimiser les codes des programmes informatique), l'économie (augmenter le bénéfice), dans la recherche opérationnelle (problème de transport, gestion de stock, ...), en analyse numérique (approximation, résolution de système linéaire, non linéaire, ...), en automatique (modélisation de système, filtrage), en ingénierie (dimensionnement de structures, conception optimale de système (réseaux, ordinateurs) [49].

Dans ce partie nous allons présenter les différentes notions de base de l'optimisation, nous aborderons aussi des rappels sur la complexité.

3.1 Définition d'un problème d'optimisation classique (mono-objectif) :

Un problème d'optimisation est un problème d'analyse fonctionnelle qui est noté par (P) [28], Ça forme général est de la forme [48] :

$$(P) \begin{cases} \text{"Opt"}F(x). \\ x \in X \subseteq \mathbb{R}^n \end{cases}$$

Vocabulaire :

- Opt $F(x)$ peut être minimiser ou maximiser $F(x)$.
- La fonction F appelée fonction objectif du problème (P) (ou fonction critère) .
- $X \subseteq \mathbb{R}^n$ est un domaine des solutions réalisables du problème (P) (Une solution est dite réalisable si elle vérifie l'ensemble des contraintes).
- Les contraintes peuvent être présentés comme des relations ou liaisons entre les variables .
- Le point x est appelé solution réalisable du problème (P) si et seulement si $x \in X$.

.Le point x^* est dite solution optimale de problème (P) si et seulement si $F(x^*) \leq F(x)$,

(resp, $F(x^*) \geq F(x)$) , $\forall x \in X$ pour un problème de minimisation (resp de maximisation).

Remarque : Le problème (P) est appelé problème d'optimisation mono-objectif car on a une seul fonction objectif à optimiser .

3.2 Classification des problèmes d'optimisation

Pour résoudre un problème d'optimisation, il est important de définir à quelle catégorie ce problème appartient.[48]

La nature d'un problème d'optimisation dépend des caractéristiques de la fonction objectif $F(x)$ et du domaine X .

3.2.1 Problème d'optimisation sans contraintes

Dans ce genre de problèmes l'ensemble des solutions réalisables $X=\mathbb{R}^n$, sa formule générale est de forme :

$$(P) \begin{cases} \text{"Opt"} F(x). \\ x \in \mathbb{R}^n. \end{cases}$$

3.2.2 Problème d'optimisation avec contraintes

Contrairement au cas avec contraintes l'ensemble des solutions réalisables est $X \subset \mathbb{R}^n$, ça formule générale est de forme :

$$(P_C) \begin{cases} \text{"Opt"} F(x). \\ x \in X. \end{cases}$$

3.2.3 Problème d'optimisation continue

L'optimisation continue étudie le cas où les variables sont de nature continue ($X \subseteq \mathbb{R}^n$), ça formule générale est de forme :

$$(P) \begin{cases} \text{"Opt"} F(x). \\ x \in X. \quad \text{avec } X \text{ domaine continue.} \end{cases}$$

3.2.4 Problème d'optimisation convexe

Étudie les problèmes du type : optimiser une fonction donnée dans un espace donné, elle traite le cas où la fonction objectif et les contraintes sont convexes .

3.2.5 Problème d'optimisation stochastique et robuste

Dans plusieurs cas réels nous optimisons un problème où des données sont connues en probabilité (variables aléatoires), ces problèmes sont appelés les problèmes d'optimisation stochastique [25], ou sont connus de façon imprécise (cadre de l'optimisation robuste).

Exemple [25] :

- Prévoir le futur (demande des clients, les pannes, défauts. . .).
- Réactions de concurrents (course de la bourse, du marché . . .).

Remarque : Dans le cas où les données sont connues, nous parlons alors de problèmes d'optimisation déterministe.

3.2.6 Problème d'optimisation linéaire

Le problème d'optimisation linéaire est une catégorie particulière, celle où la fonction objectif est une fonction linéaire et les contraintes sont linéaires .

3.2.7 Problème d'optimisation non linéaire

Cette classe est le cas contraire des problèmes d'optimisation linéaire, donc la fonction objectif est une fonction non linéaire et (ou) les contraintes sont non linéaires.

3.2.8 Problème d'optimisation quadratique

Ce genre de problème étudie le cas où la fonction objectif est quadratique et les contraintes sont linéaires.

Un problème quadratique s'écrit sous la forme de programme suivante :

$$(P) \begin{cases} \text{"Opt"} F(x) = \frac{1}{2}x^t Q x + c^t x. \\ x \in X. \end{cases}$$

Où $X = \{ x \in \mathbb{R}^n / Ax = b, x \geq 0 \}$.

avec Q une $n \times n$ matrice symétrique, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ et $A \in \mathbb{R}^{m \times n}$.

Exemple : optimisation de portefeuille financier .

3.2.9 Problème d'optimisation discrète (combinatoire)

C'est une classe d'optimisation particulièrement étudiée, dans ce genre de problème les variables sont de type entiers ($X \subseteq \mathbb{Z}^n$), dans ce cas la formule (POC) est dite problème en nombre entier.

$$(POC) \begin{cases} \text{"Opt"} F(x). \\ x \in X. \end{cases}$$

En particulier, si $x \in \{0,1\}$, on parlera de problème d'optimisation bivalente.

Remarque : Si l'ensemble X contient des variables discrètes en plus des variables continues, nous parlerons donc ici de problème d'optimisation mixte .

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle [48], en effet c'est un outil obligatoire qui englobe un large domaine d'étude [50]. C'est un problème de recherche en lequel nous exploitons un ensemble qui est dit ensemble des solutions

réalisables en espérant trouver un optimum, sinon essayer de trouver la plus proche solution à la solution optimale.

Ce problème dans la plupart des cas est définie comme une problématique de choix d'une meilleure alternative dans un très grande nombre fini d'alternatives (l'ensemble d'alternatives est l'ensemble des solutions réalisables) [48].

Nous pouvons représenter l'importance de ce problème en citant quelques points :

- Ce genre de problème à une très grande difficulté à résoudre (la majorité de ces problèmes sont de la classe NP-Difficile.

Remarque : nous aborderons la notion NP-Difficile d'une façon plus approfondie dans la partie 1.4 (La complexité algorithmique).

- Dans le cas pratique il y a un grand nombre de problèmes qui peuvent être représenté et formulé sous la forme d'un problème d'optimisation combinatoire.
- De nombreuses méthodes de résolution ont été créés ou développées dans beaucoup de domaines.

Parmi les problèmes les plus importants de l'optimisation combinatoire on trouve le problème de sac-à-dos, problème de voyage de commerce (PVC), problème de flot, problème de cheminement, problème d'affectation, problème d'ordonnancement, problème de tournée de véhicule (VRP),...,etc. [47]

- Problème d'affectation

Le problème d'affectation consiste à attribuer au mieux des tâches à des agents, de matériel à des projets,...,etc, chaque agent peut réaliser une unique tâche pour un coût ou rendement donné et chaque tâche doit être réalisée par un unique agent .

Les affectations (c'est à dire les couples agent-tâche) ont tout un coût (rendement) défini. Le but est de minimiser le coût (ou maximiser le rendement) total des affectation afin de réaliser toutes les tâches.

Ce problème consiste donc à réaliser une bijection entre les élément de deux ensembles I et J avec une certaine fonction objectif dépendant du choix du couple (i,j) (c'est à dire les affectations de i à j) à maximiser où à minimiser .

Ce problème se modélise par :

$$x_{ij} = \begin{cases} 1 & \text{Si le tâche } i \text{ sera affecté au l'agent } j. \\ 0 & \text{Sinon} \end{cases}$$

$$(P) \begin{cases} \sum_{i=1}^{|I|} x_{ij} = 1, & \forall j=1, \dots, |J|. \\ \sum_{j=1}^{|J|} x_{ij} = 1, & \forall i=1, \dots, |I|. \\ \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} c_{ij} \cdot x_{ij} = Z(\min)(ou Z(\max)) \\ x_{ij} \in \{0, 1\}, & \forall i=1, \dots, |I|, \forall j=1, \dots, |J| \end{cases}$$

- Affectation quadratique

Le problème d'affectation quadratique a été introduit par **Koopmans** et **Back mann** [52], il est présenté dans de nombreux domaines d'activités pratiques d'optimisation.

Cette affectation consiste à réaliser une bijection des éléments d'un ensemble (I) sur ceux d'un ensemble (J) de même cardinalité avec une certaine fonction objectif dépendant du choix des couples (i,j) (fonction à maximiser ou à minimiser) telle que l'élément i est affecté une seule fois à l'élément j pour une fonction F et l'élément j doit être affecté une seule fois à l'élément i [14].

La différence entre l'affectation quadratique et l'affectation simple c'est la fonction objectif, si la fonction est linéaire le problème est un problème d'affectation simple et si la fonction est quadratique alors c'est un problème d'affectation quadratique.

L'affectation quadratique est de classe NP complet, ce cas est plus difficile à résoudre que le cas simple.

Le problème d'affectation quadratique noté (QAP) s'écrit comme suit [14] :

$$(QAP) \left\{ \begin{array}{l} x_{ij} = 0 \text{ ou } 1, i \in I, j \in J, |I| = |J| \\ \sum_{i=1}^{|I|} x_{ij} = 1, \forall j \in J. \\ \sum_{j=1}^{|J|} x_{ij} = 1, \forall i \in I. \\ \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \sum_{k=1}^{|I|} \sum_{l=1}^{|J|} c_{ijkl} \cdot x_{ij} \cdot x_{kl} = Z(\min)(\text{ou } Z(\max)) \end{array} \right.$$

La résolution du problème d'affectation quadratique est importante car beaucoup de problèmes sont modélisés et représentés par ce dernier, un de ces problèmes par exemple est le problème de voyageur de commerce (TSP).

3.2.10 Problème d'optimisation multi-objectifs

Les premiers travaux menés sur les problèmes multi objectifs furent réalisés au 19 ième siècle sur des études en économie par **Edgeworth** [40] et généralisés par **Pareto** [40].

Dans ce genre de problèmes nous aurons plusieurs fonctions objectifs à maximiser et (ou) à minimiser (les fonctions objectifs sont souvent contradictoire), la formule générale de ces problèmes est :

$$(POM) \begin{cases} \text{"Opt"} & F(x) = (F_1(x), F_2(x), \dots, F_p(x)) , p \geq 2. \\ x \in X \subseteq \mathbb{R}^n. \end{cases}$$

avec $F_i(x)$, $i = \{1, \dots, p\}$, sont des fonctions à optimiser .

Remarque importante :

- La classification des problèmes d'optimisation multi-objectif est la même que celle de l'optimisation mono objectif (les mêmes caractéristiques).
- Dans le cas où nous avons une seule fonction objectif à optimiser, signifie qu'il faut trouver dans X une solution optimale (la meilleure solution qui existe) dans le cas contraire s'il y a plusieurs fonctions, il n'est généralement pas possible de trouver dans X une solution qui optimise simultanément les p fonctions (la notion de solution optimale n'est pas de sens). Alors dans ce cas nous allons parler de solution de meilleure compromis (autrement dit la meilleure solution possible pour toutes les fonctions à la fois).

Remarque : Dans ce qui suit nous n'envisagerons que le cas de maximisation, ceci n'est pas restrictif dans la mesure où la recherche de minimum d'une fonction F . Se ramène au problème de la maximisation de $(-F)$, $[\min F = -\max(-F)]$.

3.3.10.1 Concepts de bases**Définition 3.3.10.1.1**

On appelle espace de décision, l'espace de \mathbb{R}^n dans lequel se situe X .
 Pour espace des critères, on entend l'espace de \mathbb{R}^p dans lequel se situe
 $Z_x = \{ Z \in \mathbb{R}^p / Z_i(x) = F_i(x), \quad i = \{1, \dots, p\}, \quad x \in X \}$. [14]

La figure suivante représente la relation entre les espaces :

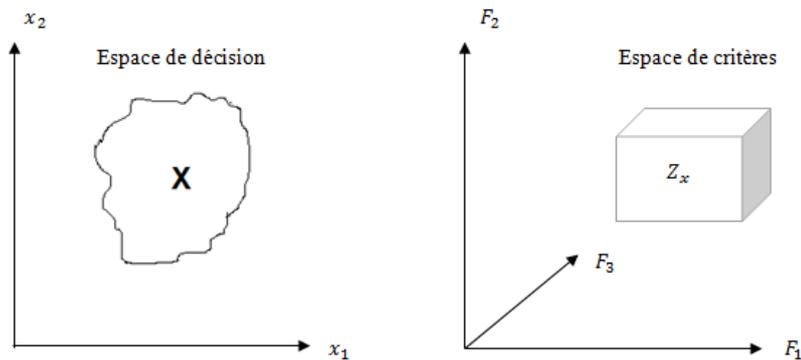


FIGURE 3.1 – Image présente les relations entre les espaces .

3.3.10.2 Notions de dominance et d'optimalité de parto

L'objectif est d'introduire la notion de dominance qui est basée sur l'idée de choisir entre deux vecteurs Z et Z' et trouvons qui est le meilleur où égal sur toutes les composantes, en étant strictement meilleure sur au moins une d'autre elles, tel que le vecteur $Z(x^*)=F(x^*)=(F_1(x^*), \dots, F_p(x^*))$

Dans cette section nous traitons les concepts fondamentaux d'efficacité et de dominance .

Définition 3.3.10.2.1

Dominance : Soit deux vecteurs $Z, Z' \in Z_X$, on dit que Z domine Z' si et seulement si $Z \geq Z'$ et $Z \neq Z'$.
 (ie $Z_i \geq Z'_i$, pour tout $i \in \{1, 2, \dots, p\}$ et $\exists i_0 \in \{1, 2, \dots, p\}$, tel que $Z_{i_0} > Z'_{i_0}$). [11] [18]
 Et écrire $Z \succeq Z'$.

Propriétés 3.3.10.2.1

La relation de dominance telle qu'elle est définie :

- N'est pas réflexive car une solution ne se domine pas elle même .
- N'est pas symétrique car nous ne trouve pas $Z_i \geq Z'_i$ et $Z'_i \geq Z_i$
- Est transitive car si $Z_i \geq Z'_i$ et $Z'_i \geq Z''_i$ alors $Z_i \geq Z''_i$. [4]

Définition 3.3.10.2.2

Dominance forte : Soit deux vecteurs $Z, Z' \in Z_X$, on dit que Z domine fortement Z' si et seulement si $Z > Z'$, pour tout $i \in \{1, 2, \dots, p\}$
Et écrire $Z \succ Z'$, alors Z est meilleure que Z' sur tous les critères.[18]

Définition 3.3.10.2.3

Efficacité : Une solution $x^* \in X$ est dite solution efficace ou paréto optimale (ou encore non dominée) s'il n'existe pas un $x \in X$ tel que $Z(x)$ domine $Z(x^*)$,
ie : $\nexists x \in X / Z(x) \succeq Z(x^*)$. [11][18]
La solution est dite paréto optimale s'elle n'est dominé par aucune autre solution réalisable .

Définition 3.3.10.2.4

Efficacité faible : Une solution $x^* \in X$ est une solution faiblement efficace s'il n'existe pas de $x \in X$ tel que $Z(x) \succ Z(x^*)$. [11]

Remarque : Le terme faiblement efficace est connu aussi sous le nom de Slater-optimale.

L'ensemble des solutions efficaces d'un problème (POM) sera noté par $\text{Eff}(P)$. [11] [18]

3.3.10.3 Front de paréto et surface de compromis

La solution d'un problème d'optimisation multi-objectif n'est pas un point unique mais un ensemble des points qui appelé l'ensemble des meilleurs compromis (surface de compromis ou front de paréto) [4].

Définition 3.3.10.3.1

Front de paréto : Le front de paréto qui on noté FP est l'ensemble des vecteurs de décision X qui ne sont pas dominés, elle définie comme suit : [55]
 $\text{FP} = \{ x' \in X \mid \nexists x \in X, x \prec x' \}$.

La figure suivante représente un exemple de front de paréto de deux fonctions :

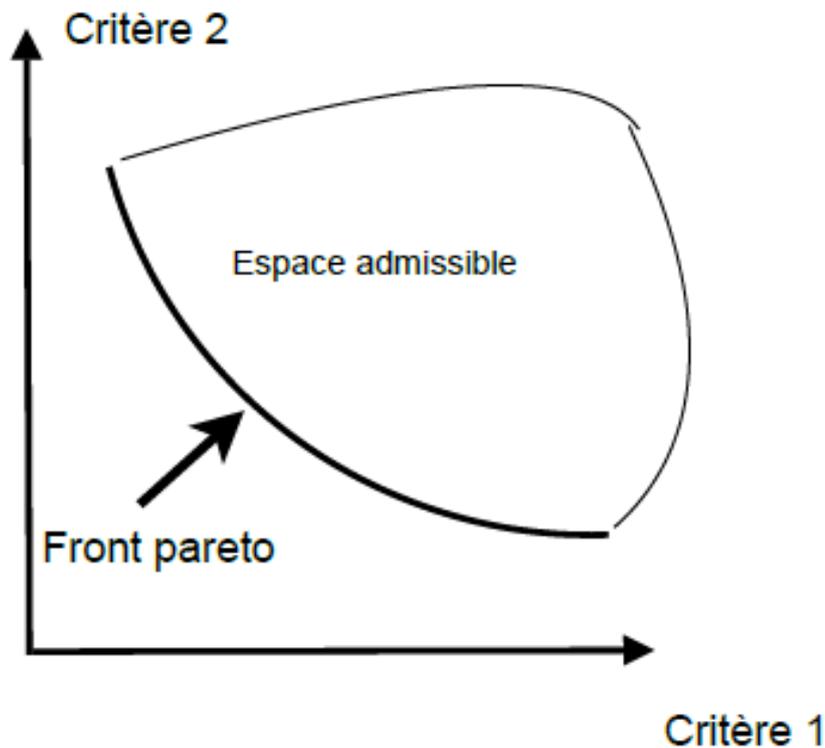


FIGURE 3.2 – Exemple de front de paréto pour deux fonctions objectifs. [1]

3.3.10.4 Points spéciaux

Définition 3.3.10.4.1

Point idéal : Le point idéal est le point obtenu en optimisant chaque fonction objectif individuellement, c'est le vecteur défini par : [11] [18]

$$Z^I = (\max_{x \in X} Z_1(x), \dots, \max_{x \in X} Z_p(x)) \in \mathbb{R}^p .$$

Généralement le point idéal n'est pas réalisable c'est-à-dire $Z^I \notin Z_X$.

Définition 3.3.10.4.2

Point anti-idéal : Le point anti-idéal c'est le vecteur défini par :

$$Z^A = (\min_{x \in X} Z_1(x), \dots, \min_{x \in X} Z_p(x)) \in \mathbb{R}^p .$$

le point anti-idéal en général n'est pas admissible c'est-à-dire $Z^A \notin Z_X$.

Définition 3.3.10.4.3

Point nadir : Le point nadir est le point de coordonnées

$$n_i = \min Z_{ij}, \quad i = \{1, \dots, p\}, \quad j = \{1, \dots, p\}$$

où $Z_{ij} = Z_i(\bar{x}^{(j)})$ avec $(\bar{x}^{(j)})$ est la solution optimale du critère Z_{ij} .

Les coordonnées de ce point correspondent aux pire valeurs obtenues par chaque fonction objectif, lorsque l'on restreint l'espace des solutions à la surface de compromis.[54]

La matrice $p \times p$ formée des éléments Z_{ij} est appelée matrice des gains.

$$Z_{ij} = \begin{pmatrix} Z_{11} & \cdots & Z_{1p} \\ \vdots & \ddots & \vdots \\ Z_{p1} & \cdots & Z_{pp} \end{pmatrix}$$

3.3 La complexité algorithmique

L'exécution d'un programme nécessite des ressources informatiques, par exemple un temps de calcul pour effectuer des instructions, un espace mémoire pour stocker les données. La performance d'un programme dépend de sa capacité à minimiser le temps d'exécution et l'espace mémoire utilisé [13].

Définition 3.4.1

L'algorithme : l'algorithme est une suite ordonnée des opérations finies pour résoudre un certain problème (P).[13]

Définition 3.4.2

Un algorithme non déterministe est un algorithme contenant une instruction "choix", qui opérant sur un ensemble fini, choisit un élément de cet ensemble sans spécifier comment ce choix est fait .[17]

Définition 3.4.3.

Un algorithme déterministe est un algorithme exécutant un ensemble d'instructions d'une manière précise dans un ordre précis.[17]

Définition 3.4.4

Un algorithme est dit polynômial si le nombre d'opérations élémentaires nécessaire pour résoudre un exemple de taille n est borné par un polynôme en n .Un algorithme sera efficace (rapide en terme de temps d'exécution) s'il est polynômial.[17]

Définition 3.4.5

La complexité : La complexité d'un problème est de pouvoir déterminer un algorithme et comparer à des autres afin de déterminer lequel est meilleure pour résoudre le problème donnée [53] .

3.3.1 Classes des problèmes

Les classes de complexité ont été pour les problèmes décisionnels c'est-à-dire des problèmes posant une question dont la réponse est oui ou non.

Un problème décisionnels peut appartenir à deux classes :

- La classe P (polynomial time) : est la classe des problèmes pouvant être résolus de manière exacte par un algorithme polynomial déterministe. C'est la classe des problèmes les plus faciles, (facile à résoudre).[8]
- La classe NP (Non déterministe polynomial time) : contient l'ensemble des problèmes dont on peut vérifier qu'une proposition donnée est bien une solution du problème avec un

algorithme de complexité polynômiale.

Les problèmes les plus difficiles de NP définissent la classe des problèmes NP-complet .

- Classe NP-complet : un problème est NP-complet si n'importe quel autre problème de NP peut être transformé en ce problème par une procédure polynômiale.

Nous distinguons également dans la classe NP : la classe NP-difficile ce sont les problèmes d'optimisation combinatoire dont le problème de décision associé est NP-complet.

Deuxième partie

Méthodes de résolution des problèmes d'optimisation

3.4 Introduction

Un très grand nombre de méthodes de résolution existent en recherche opérationnelle pour l'optimisation combinatoire, et comme beaucoup des problèmes d'optimisation combinatoire multi-objectif appartient à la classe NP-difficile alors ils ne possèdent pas toujours de solution optimale [53]. Dans ce cas, nous traiterons le problèmes d'optimisation combinatoire par deux grand types de résolutions : les méthodes de résolution exactes qui nous permettent d'obtenir une solution dont l'optimalité est garantie, mais dans certaines situations nous trouvons des solutions de bonne qualité sans garantir l'optimalité, ces méthodes de résolution sont appelées heuristiques et méta heuristiques (méthodes approchées).

Définition 3.5.1

Algorithme exact : Une méthode qui donne une garantie de trouver la solution optimale pour une instance de taille fini dans un temps limité et de prouver l'optimalité.[36]

Définition 3.5.2

Heuristique : Le mot heuristique est dérivé de la langue grec qui signifie le verbe trouver. Une heuristique est une méthode qui permet de trouver et d'obtenir efficacement des bonnes solutions approchées dans l'espace de solutions, dans un délai de temps raisonnable, pour un problème donné mais sans garantir l'optimalité [39].

L'objectif d'une heuristique est de trouver une solution la plus proche possible d'une solution optimale d'une méthode exacte.

Définition 3.5.3

Méta-heuristique : Le mot méta-heuristique est composé de deux mots grec méta et heuristique. Méta est un suffixe signifiant au delà c'est à dire plus haut niveau. [39]

Méta-heuristique est une stratégie qui guide et modifier d'autres heuristiques visant à résoudre des problèmes d'optimisations difficiles.[34]

Le but d'une méta-heuristique, est de réussir à trouver un optimum global.

Caractérisations de méta-heuristique [23],[41],[27]

- Les techniques qui constituent des algorithmes de type Méta-heuristique vont de la simple procédure de recherche locale à des processus d'apprentissage complexes.
- Les méta-heuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité.
- Les méta-heuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherches.
- Les concepts de base des méta-heuristiques peuvent être décrits de manière abstraite, sans faire appel à un problème spécifique.
- Les méta-heuristiques peuvent faire appel à des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur.
- Les méta-heuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite du processus de recherche.
- Généralité et application possible à une large classe de problèmes.
- Efficacité pour de nombreux problèmes.

- Possibilité de compromis entre qualité des solutions et temps de calcul.
- Nécessité d'adaptation de la méthode au problème traité.
- Nécessité de réglage des paramètres.
- Difficulté de prévoir la performance.

3.5 Méthodes de résolutions exactes

Les méthodes exactes (appelées aussi méthodes complètes) sont utilisées dans la recherche opérationnelle afin de trouver une solution ou des solutions réalisables pour les problèmes d'optimisation combinatoire .

Nous pouvons citer la méthode de séparation et évaluation, la méthode ϵ contraintes,...,etc.

1. La méthode de Séparation et évaluation

Cet algorithme est plus connu sous son appellation anglaise Branch and Bound, elle repose sur une méthode arborescente de recherche d'une solution optimale par séparations et évaluations, en représentant les états de solutions par un arbre d'état avec des noeuds et des feuilles.[50]

La procédure de séparation divise l'ensemble des solution réalisables en sous-ensembles (noeuds) .

La procédure d'évaluation évaluer les solutions d'un sous ensembles en majorant la valeur de la meilleure solution de ces sous-ensembles .

2. Méthode ϵ contraintes

La méthode ϵ contraintes est une technique qui considérée comme la plus connue pour résoudre les problèmes d'optimisation multi-objectifs , ce concept fut introduit par **Haines** en 1971 [56] et une discussions approfondie se trouve dans **Chankong** et **Haines** (1983) [56].

Considérons le problème suivant :

$$(POM) \begin{cases} \text{'Opt' } F(x) = (F_1(x), F_2(x), \dots, F_p(x)) , p \geq 2. \\ g_j(x) \leq b_j, \quad j \in \{1, \dots, m\}. \\ x \in X \subseteq \mathbb{R}^n. \end{cases}$$

(a) Le principe de la méthode

Le principe de cette méthode est de considérer uniquement une seul fonction objectif, pendant que les autres sont transformées en contraintes. Le problème (POM) est transformé alors au problème suivant : [32]

$$(P') \begin{cases} \text{'Opt' } F_{i'}(x) & i' \in \{1, \dots, p\}. \\ F_i(x) \geq \epsilon_i , & i \in \{1, \dots, p\} \quad i \neq i'. \\ g_j(x) \leq b_j, & j \in \{1, \dots, m\}. \\ x \in X \subseteq \mathbb{R}^n. \end{cases}$$

Algorithm 1 Algorithme de la méthode ϵ contraintes [32]

Début

- Définir le problème d'optimisation multiobjectif (POM).
 - Trouver les solutions efficaces de (POM) .
 - Transformé (POM) à (P') .
 - Soit X' , Y deux ensembles vides représentent l'ensemble de solutions efficaces et faiblement efficaces.
 - Tant que** : Le décideur n'est pas encore satisfait .
 - Fait** : Le décideur choisi un seul objectif à optimiser i' et les seuils d'acceptation des autres objectifs ϵ_i , $i \neq i'$.
 - Résoudre le problème (P').
 - Soit x^* la solution trouvée .
 - Si** x^* est unique alors :
 - $X' = X' \cup \{x^*\}$, et $Y = Y \cup \{x^*\}$.
- Fin algorithme.**
-

3.6 Les méthodes approchées (Méta-heuristique)

La classe de méta heuristique pour l'optimisation multiobjectif sont ceux qui emploient les méthodes approchées, ils ont été appliquées avec succès sur un grand nombre de problèmes académiques et réels : problème d'affectation quadratique, voyage de commerce,...,etc.

Ehrgott [38] et **Gandibleux** [38] classent ces méthodes essentiellement en deux, celles se basant sur une solution unique (les algorithmes d'exploration) et celle se basant sur une population de solutions (les algorithmes évolutionnaires).

3.6.1 Méta heuristique à base de solution unique

— Recuit Simulé

1. Définition

Le recuit simulé a été Introduit par **Kirkpatrick** en 1982 [31], et **Gerny** en 1985 [9]. L'idée de base est la suivante : à des paliers de températures décroissantes, l'algorithme utilise la procédure itérative de Metropolis, pour atteindre un état de quasi-équilibre thermodynamique.

Cette procédure permet de sortir des minima locaux avec une probabilité d'autant plus grande que la température est élevée.

Quand l'algorithme de recuit simulé atteint les très basses températures, les états les plus probables constituent en principe d'excellentes solutions au problème d'optimisation.[9]

Metropolis avait proposé en 1953 [37] un algorithme itératif qui permet d'atteindre l'état d'équilibre thermodynamique d'un système simulé à une température T. Son principe consiste à itérer les deux étapes suivantes [41] :

Algorithm 2 Algorithme de Metropolis [50] :

Début

- Évaluer la variation d'énergie associée à une transition élémentaire aléatoire de l'état courant i, d'énergie E_i , vers un nouvel état j, d'énergie E_j : $\Delta E_{ij} = E_j - E_i$
- Accepter la transition vers le nouvel état avec une probabilité P_{ij} .

ou

$$\begin{cases} P_{ij}(T) = 1 & \text{si } \Delta E_{ij} \leq 0 \\ P_{ij}(T) = \exp(-\Delta E/T) & \text{si } \Delta E_{ij} \geq 0 \end{cases}$$

Fin

Algorithm 3 Algorithme de recuit simulé [50] :

Début

- 1 - Définir la fonction objectif (F).
- 2 - Choix des mécanismes de perturbation d'une configuration ΔS .
- 3 - Tirer une configuration aléatoire S (ensemble des solutions).
- 4 - Calculer l'énergie associée à cette configuration E.
- 5 - Initialiser la température (T_0).
- 6 - **Tant que** Conditions d'arrêts pas satisfaites **faire** :
 - 6.1. **Tant que** l'équilibre thermodynamique pas atteint **faire** :
 - 6.1.1 .Tirer une nouvelle configuration S'.
 - 6.1.2. Appliquer la règle de Metropolis.
 - 6.1.3. **Si** $f(S') < f(S)$.
 - $f_{min} = f(S')$
 - $S_{opt} = S'$
 - Fin si**
 - Fin tant que**
 - 6.2. Décroître la température.
- Fin tant que.**
- 7 - Afficher la solution optimale.

Fin algorithme

2. Avantages de la méthode [36]

- Solution de bonne qualité.
- Méthode générale et facile à programmer.
- Souplesse d'emploi de nouvelles contraintes peuvent être facilement incorporées.

3. Inconvénients [41]

- Nombre important de paramètres.
- Temps de calcul excessif dans certaines applications.

— Recherche Tabou

1. Définition

Recherche tabou est une méthode de recherche local utilisée pour résoudre des problèmes complexes et de très grande taille, à été proposée à 1980 par **Fred Glover** [43] l'idée base de définir un voisinage V^* (comme dans la plupart des méthodes heuristique). La recherche tabou a plusieurs applications en programmation non linéaire .[43]

Partant de la solution courante x_n à l'étape n, nous calculons la meilleure solution x^* dans un sous voisinage V^* de $V(x_n)$, cette nouvelle solution courante x_{n+1} qu'elle soit meilleure au moins bonne que la précédente.[33]

2. Liste tabou

Une liste tabou présente une mémoire à court terme qui gardent les derniers solutions rencontrées, elle définit pour éviter le cycle de prévention [33].

Liste tabou déteint généralement une quantité d'information ou de mouvement interdits (solution interdits), la longueur de la liste est limitée (p) [33], elle est gérée comme une file avec discipline FIFO (First in , First out) [7] .

3. Principe de recherche tabou

Algorithm 4 Algorithme de recherche Tabou [33] :

Début

- **Initialisation** : $x_0 \in X$ solution initiale, $F^* := F(x_0)$, p = longueur de la liste tabou.
- **Étape n** : soit $x_n \in X$ la solution courante ;
 sélectionner, dans un sous-voisinage V^* de $V(x_n)$, la meilleure solution $x^* \in V(x_n)$ qui soit :
 non tabou ou tabou, mais satisfaisant un critère d'aspiration ;
 $x_{n+1} := x^*$;
si $F(x^*) < F$: $F := F(x^*)$
 mettre à jour la liste tabou ;
si la règle d'arrêt n'est pas satisfaite,
 passer à l'étape $n + 1$;
sinon, stop.

Fin

Remarque : La recherche tabou peut être très efficace, mais l'implémentation et l'ajustement des paramètres sont difficiles, ils sont très dépendants de la structure du problème [33].

Révolution de cette méthode par rapport aux autres permet de surmonter le problème des optimaux locaux par l'utilisation de la liste tabou (principe de mémoire). [33]

3.6.2 Méta heuristique à base de population de solutions

— Algorithme génétique

1. Définition

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes, nous les utilisons pour avoir une solution approchée (de bon qualité) lors de la résolution d'un problème d'optimisation qui a une taille importante (très grande)

ou lorsque ce dernier ne peut pas être résolu dans un temps raisonnable (quelques problèmes ont un temps de résolutions plus que la durée de vie de l'univers).

Ces algorithmes sont basés sur les mécanismes de la sélection naturelle et de la génétique, ils s'appuient sur un paradigme darwinien de l'évolution génétique d'une population.

Il combine une stratégie de "survie des plus forts" avec un échange d'information aléatoire mais structurée. Pour un problème pour lequel une solution est inconnue, un ensemble de solutions possibles est créé aléatoirement. [20],[45]

On appelle cet ensemble la population. Les caractéristiques (ou variables à déterminer) sont alors utilisées dans des séquences de gènes qui seront combinées avec d'autres gènes pour former des chromosomes et par après des individus. Chaque solution est associée à un individu, et cet individu est évalué et classifié selon sa ressemblance avec la meilleure, mais encore inconnue, solution au problème. Il peut être démontré qu'en utilisant un processus de sélection naturelle inspiré de **Darwin**, cette méthode convergera graduellement à une solution. [20]

2. Principes des algorithmes évolutionnaires [10]

En général, un algorithme évolutionnaire AE est caractérisé par trois faits :

1. A base de population (un ensemble de solutions candidates est maintenu).
2. Celui-ci subi un processus de sélection (les mieux adaptés survivent).
3. Héritage de patrimoine génétique.

4. Manipulé par des opérateurs génétiques, le plus souvent le croisement et la mutation.
5. Amélioration globale.
6. Fonction objectif .

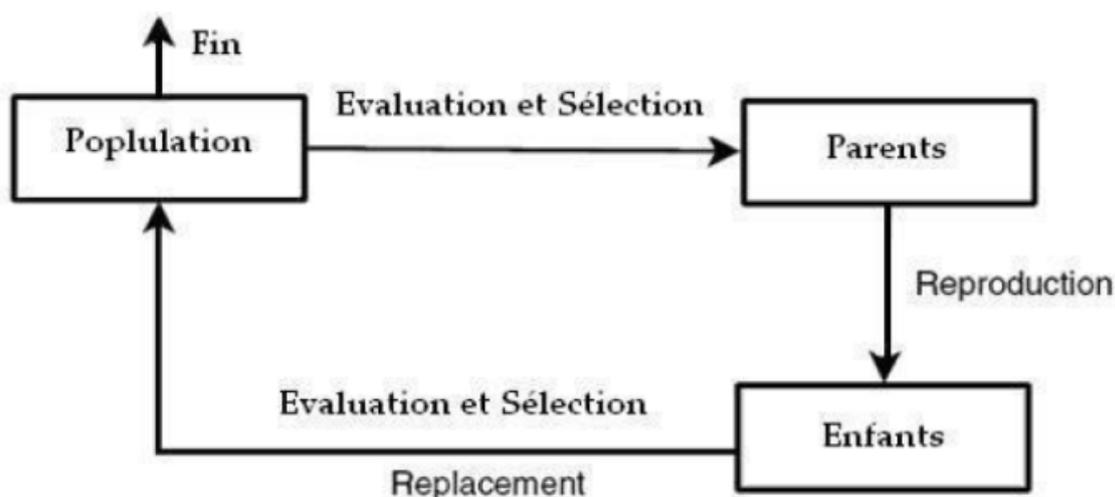


FIGURE 3.3 – Fonctionnement général d'un algorithme évolutionnaire.

3. La sélection naturelle de Darwin

La théorie de la sélection naturelle aide à comprendre l'action de l'environnement sur l'évolution des populations. Dans une population d'individus, un caractère présente des variations. Certains individus portent des variations qui leur permettent d'être mieux adapté à leur environnement (autrement dit avoir un avantage). Ils ont donc plus de chances de survivre et une meilleure probabilité de se reproduire. Leur descendance est donc plus nombreuse, et portera cette variation si elle est héréditaire. La conséquence logique est que cette variation héréditaire verra sa fréquence

augmenter dans la génération suivante. Et ainsi de suite,...[46]

4. Description de l'algorithme

Les algorithmes génétiques ont vu le jour grâce aux recherches de John Holland et de ses collègues et élèves de l'Université du Michigan en 1960, ces recherches portant sur l'évolution des espèces de la théorie développée par Charles Darwin.

Par analogie avec la théorie darwinienne, les solutions les plus adaptées tendent à demeurer plus longtemps dans l'espace de recherche et donc à se reproduire plus aisément pour former éventuellement de nouveaux individus plus performants.[46]

5. Les étapes d'un algorithme génétique [46]

Algorithm 5 Algorithme Génétique

Début

- **Data** : soit $X^{(n)} \subseteq X$ une population initiale générée aléatoirement.
- **Tant que** la règle d'arrêt n'est pas satisfaite faire :
- Soit $X^{(n)} \subseteq X$, la population courante ;
- Évaluer le degré d'adaptation de chaque individu ;
- Sélectionner dans $X^{(n)}$ un ensemble de paires de solutions de haute qualité.
- Appliquer à chacune des paires de solutions sélectionnées un opérateur de "croisement" qui produit une ou plusieurs solutions "enfants" ;
- Remplacer une partie de $X^{(n)}$, formée des solutions de basse qualité, par des Solutions "enfants" de haute qualité ;
- Appliquer un opérateur de mutation aux solutions ainsi obtenues ;
- Les solutions éventuellement mutées constituent la population $X^{(n+1)}$;

Fin.

La figure suivante représente le schéma d'un algorithme génétique :

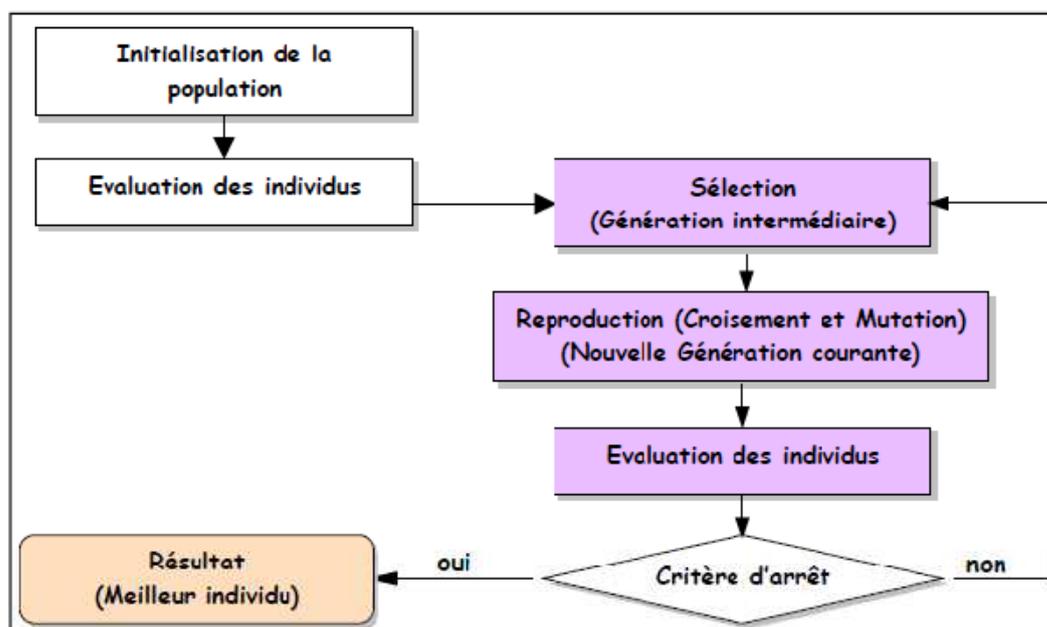


FIGURE 3.4 – L’organigramme d’un algorithme génétique [3]

— Algorithmes de colonies de fourmis [19]

Proposé à la fin des années 1980 par **Moyson** [19] et **Manderick** [19], puis par **Dorigo** [2] dans les années 90, pour la recherche de chemins optimaux dans un graphe.

L’idée originale est issue de l’observation du comportement collectif d’exploitation de la nourriture chez les fourmis. En effet, celles-ci, bien qu’ayant individuellement des capacités cognitives très limitées, sont capables collectivement de résoudre le problème de la découverte du plus court chemin entre une source de nourriture et leur nid.

Les scientifiques ont donc remarqué, après plusieurs expériences, que lorsque les fourmis ont le choix entre deux parcours de longueurs différentes, menant au même point de nourriture, elles ont automatiquement tendance à emprunter le plus court chemin.

Voici un modèle grossier qui explique ce comportement :

- Une fourmi « éclairceuse » parcourt plus ou moins aléatoirement l'environnement autour de la colonie .
- Si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones .
- Ces phéromones étant attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe, cette piste .
- En revenant au nid, ces mêmes fourmis vont renforcer la piste .
- Si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps, parcourue par plus de fourmis que la piste longue .
- La piste courte sera donc de plus en plus renforcée par les phéromones, et donc de plus en plus attractive .
- La piste longue, elle, finira par disparaître, les phéromones étant plus ou moins volatiles.
- A terme, l'ensemble des fourmis a donc « choisi » la piste la plus courte.

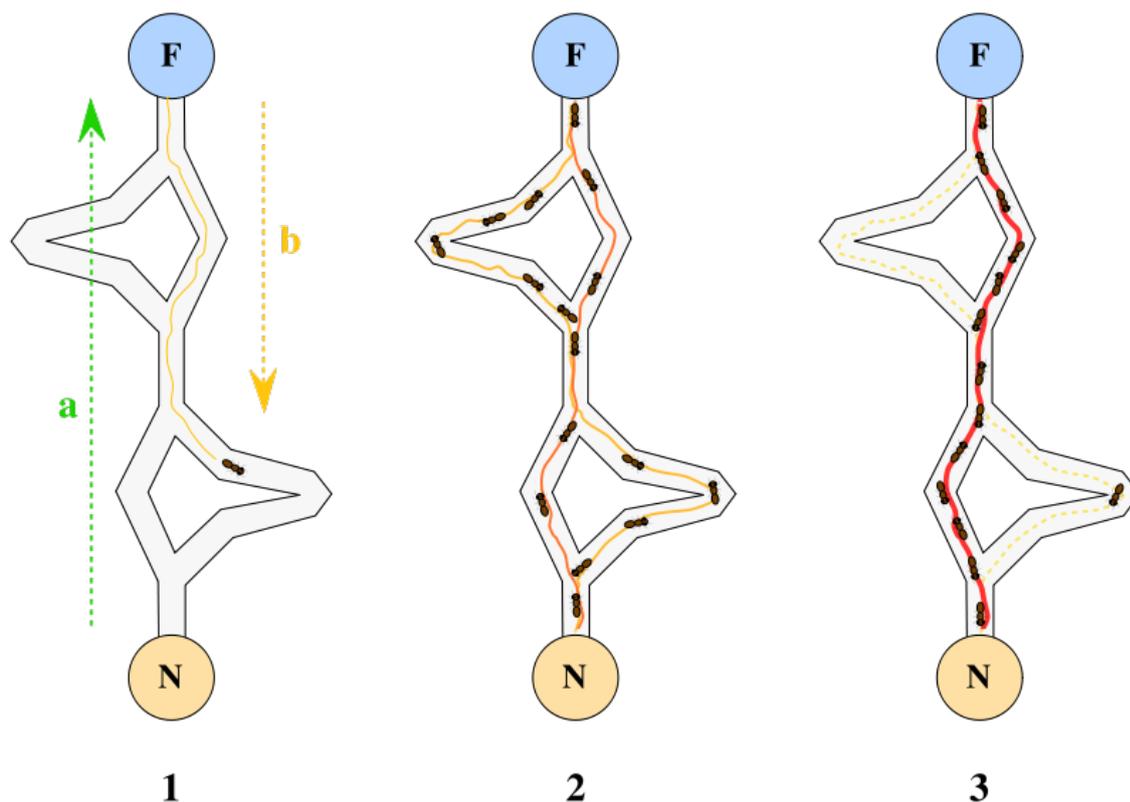


FIGURE 3.5 – Comportement des fourmis

Les algorithmes de colonies de fourmis sont beaucoup étudiés depuis quelques années, le problème du voyageur de commerce a fait l'objet de la première implémentation de l'algorithme « Ant System » (système de fourmis); le passage de la métaphore à l'algorithme est dans ce cas relativement facile à faire puisque le problème du voyageur de commerce est bien connu et étudié. Il est tout de même bon de mentionner l'existence d'une littérature importante sur toutes sortes d'autres problèmes solvables grâce aux variantes de cette approche, comme par exemple : coloriage de graphes, affectation de fréquence, affectation généralisée, sac à dos multi dimensionnel,...,etc.

Chapitre 4

Modélisation

4.1 Introduction

La modélisation constitue une phase primordiale dans la résolution des problèmes rencontrés dans la pratique .

Tout simplement la modélisation mathématique d'un problème est une transcription simplifiée de la réalité, elle traduit en langage mathématique une situation (à l'aide d'équations, des suites, des fonctions, de configurations géométrique, des graphes, de lois de probabilités et d'outils statistiques,...).[22]

Son objectif est de traiter les problèmes réels de la vie ou des problèmes mathématique eux-mêmes ou des problèmes d'autre sciences, en convertissant n'importe quelle problème en un modèle mathématique .

Une fois la modélisation mathématique du problème est établi, l'étape qui suit consiste à trouver et à choisir les meilleures solutions à l'aide d'algorithmes qui correspondent à la nature du problème.[22]

Pour mettre au point une modélisation il faut suivre les étapes suivantes :

1. **Les hypothèses**

Proposer quelques hypothèses, en citant en l'occurrence une série de points essentiels (conditions nécessaires) à vérifier obligatoirement pour rendre possible l'utilisation du modèle élaboré.

2. **Le modèle mathématique [22]**

(a) **Phase 1 : La collections des données**

- **Détermination du but d'étude (résultat)**

Cette étape est très importante dans la création du modèle puisque si leur objectif n'est pas bien défini alors le modèle ne pourra pas résoudre le problème.

- **Réunir les données et les informations**

Faire une liste des données et indices disponibles ainsi que toutes les informations qui peuvent être extraites des données, et de plus les informations qu'il faut trouver ou calculer (les mesures, des coûts, des durées,...).

- **Identifier dans quel domaine le problème se situe ainsi que les équations utilisées**

Quelle sont les formules qui peuvent être utilisé dans notre modèle et qui peuvent résoudre le problème, bien sûr avant ce point il faut savoir bien définir le domaine de notre étude (certes cerner la problématique d'une bonne façon va aider à une définition plus exacte et de la des formules bien précises de ce dernier).

(b) **Phase 2 : Création du modèle mathématique**

- **Crée le modèle**

Après avoir terminé la première phase (la planification) dans cette étape, il faut créer le modèle, on utilise les données, les informations, les graphes (s'ils existent) et ainsi de suite, utiliser tous ce qu'ils a été collecté dans la phase 1, c'est crucial aussi de fixé les variables de décisions, les contraintes, et les fonctions objectif, bien sûr il faut s'assurer que la relation entre les données est conservée (le modèle reflète le problème).

• **Testé le modèle**

Dans cette partie après avoir créé un modèle et avant d'aller plus loin il faut le tester pour savoir si le modèle serra valider ou non, en appliquons les données il est possible de pouvoir conclure : Est-ce que ce modèle est bien formulé ? Est-ce qu'il a un sens ? Est-ce qu'il résout le problème confronté ? Cette étape est comme un correcteur dans le modèle.

• **L'amélioration du modèle**

C'est la dernière étape à faire dans la partie création du modèle mathématique, ils essayent de voire le modèle d'un point de vue critique, quels sont les points qu'il faut améliorer (bien sur si c'est possible) dans quels autres domaines ils peuvent utiliser ce modèle, si il peuvent changer les variables, contraintes, ou fonctions objectif pour améliorer les résultats .

3. Évaluation et critique [22]

A cette étape d'étude, ayant réussi à exprimer le problème par sa formulation en un programme mathématique, il faut effectuer une brève analyse concernant la taille du problème, en évaluant le nombre de variables et de contraintes mises en jeu, et en citeront aussi quelques caractéristiques importantes observées.

4.2 Les hypothèses de GCB

4.2.1 Le temps de transfère d'engins

Puisque dans notre étude nous essayons de terminer tous les projets dans un temps minimum et que la période de séjours des engins dans les chantiers est courte (semaine,mois) alors le temps de transfère du matériels est pris en compte.

4.2.2 La disponibilité des engins et satisfaction des demandes

Dans notre études les engins que la société possède peuvent ne pas être suffisant pour satisfaire les besoins de chaque chantier autrement dit le nombre d'engins est moins que le nombre de tâches, dans ce cas il faut faire appel à la location et puisque un des buts de notre études est de diminuer les dépenses alors la location sera prise en considération.

4.2.3 La distance entre deux projets (chantiers)

Nous allons prendre en considération la distance entre les chantiers, en effet si un chantier a besoin des engins il faut faire en sorte de lui affecter le matériels libre qui se trouve dans le plus proche des chantiers, cette démarche va nous aider à réaliser nos but de façon indirecte

(diminuer la distance de transport d'engins va résulter à minimiser le temps de réalisation des projets, ceci entraîne la minimisation du coût totale en diminuant les frais de transport).

4.2.4 Les perturbations

Nous supposons que dans notre études il y aura pas des éléments perturbateurs autrement dis nous allons supposer que la main d'oeuvre est bien formée et que le matériels est bien entretenue, ainsi que tous les conditions de travail sont selon les normes d'où l'absence d'éléments perturbateurs (grèves, panne d'engins,...,etc).

4.2.5 Temps de montage et démontage

La flotte du matériels de la société contient presque cinq milles engins diviser en trois classes (A, B, C) et chaque engin a son propre code, parmi ce matériels nous avons constaté que seulement cinquante engins (au plus) qui nécessitent un démontage et un montage, de plus la durée de séjour du matériel en question est bien plus grand que le temps de montage et démontage (autrement dit négligeable) et dans ce cas ce temps ne sera pas pris en considération (son influence n'a pas un grand impact) .

4.2.6 L'importance des projets

Pour l'entreprise les projets n'ont pas la même importance, donc notre étude prend en considération la priorité des projets (les projets critiques (c'est-à-dire les plus importants) passent toujours en premier), l'importance du projet ce base essentiellement sur les revenus de ce derniers (bien sur ils y a d'autres choses qui influences l'importance des projets mais elles sont négligeables par rapport au revenu alors elles ne sont pas prises en considération), dans ce cas nous allons attribuer à chaque projet une pondération de telle sorte que un projet d'une

grande importance aura une grande pondération .

4.3 Le modèle mathématique

Le travail consiste à :

- Avant le début de la période d'exercice, rassembler les données du problème (planning du matériel de chaque structure, liste des engins utilisables, affectation actuelle de chacun d'eux).
- Traiter les données pour construire un planning applicable sur toute la période future d'exercice.
- Affecter en chaque début de période les engins à leurs destinations respectives.

Donc, notre but lors de cette modélisation est de présenter de la manière la plus fidèle possible la problématique de GCB.

Le présent paragraphe décrit comment le problème a été structuré.

Les variables posées seront expliquées ainsi que les paramètres utilisés dans le but d'atteindre au mieux l'objectif cible.

Pour le faire, nous allons suivre les étapes suivantes :

- **Formulation mathématique du problème**
 - Identifications des indices, variables et données du problème.
 - Établissement des contraintes du modèle.
 - Établissement des objectifs de l'entreprise.
 - Modèle mathématique.
- **Évaluation du modèle**
 - nombre de variables.

- nombre de contraintes.

4.4 Les données fondamentaux

E : ensemble des engins d'un même type disponible sur l'ensemble de toutes les structures, $|E|= NE$.

K : ensemble des structures de l'entreprise , $|K|= NS$.

J : ensemble des chantiers (projets) total de l'entreprise concernés par la période de planification, $|J|= NJ$.

P : ensembles des mois de la période de planification à considérer $|P|= NP$.

4.4.1 Les indices utilisés

i : numéro d'engin dans une famille d'engins E , $i \in \{1, \dots, NE\}$.

j : numéro d'un chantier parmi l'ensemble J de chantiers concernés, $j \in \{0, \dots, NJ\}$.

k : numéro d'une structure parmi l'ensemble K des structures de l'entreprise , $k \in \{1, \dots, NS\}$.

l : numéro d'un mois de la période d'exercices P , $l \in \{0, \dots, NP\}$.

$l=0$ désigne le dernier mois de la période d'exercice précédente.

$j=0$ désigne le parc de la structure k .

ET_j : Pondération caractérisant l'état actuel du projet j , $ET_j \in \{1, 10\}$.

$$ET_j = \begin{cases} 10 & \text{Si le projet est stratégique .} \\ 1 & \text{sinon.} \end{cases}$$

D_l : Demande total des engins pour tous les chantiers de toutes les structures du mois l .

$Disp_l$: Disponibilité total des engins pour tous les chantiers de toutes les structures du mois l .

DS_{jl} : Paramètre utiliser pour pousser les engins présent au chantier j pendant le mois l a

rester dans le même chantier pendant le mois $l+1$.

$$DS_{jl} = \begin{cases} 1 & \text{Si la demande du mois } (l+1) \text{ dépasse la demande du mois } l \text{ pour le projet.} \\ 0 & \text{Sinon.} \end{cases}$$

DM_l : Paramètres utiliser pour accéder à la location dans le mois l .

$$DM_l = \begin{cases} 1 & \text{Si la demande des engins dans le mois } l \text{ dépasse la disponibilité, } (Disp_l < D_l). \\ 0 & \text{Sinon.} \end{cases}$$

EL_l : Ensemble des engins loués dans toutes les structures le mois l , $|EL_l| = NL_l$

$$NL_l = \begin{cases} D_l - Disp_l & \text{Si } Disp_l < D_l. \\ 0 & \text{Sinon.} \end{cases}$$

i' : Numéro d'engin dans la famille des engins loués, $i' \in \{1, \dots, NL_l\}$.

4.4.2 Les variables de décision

$$X_{ijkl} = \begin{cases} 1 & \text{Si l'engin } i \text{ sera affecté au projet } j \text{ de la structure } k \text{ durant le mois } l. \\ 0 & \text{Sinon.} \end{cases}$$

$$\forall i \in \{1, \dots, NE\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}, \forall l \in \{0, \dots, NP\}$$

X_{ijk0} = représente l'emplacement de l'engin i dans le dernier mois de la période précédente.

$$\forall i \in \{1, \dots, NE\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}.$$

$$Y_{i'jkl} = \begin{cases} 1 & \text{si l'engin } i' \text{ sera loué au projet } j \text{ de la structure } k \text{ durant le mois } l. \\ 0 & \text{sinon.} \end{cases}$$

$$\forall i' \in \{0, \dots, NL_l - DM_l\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}, \forall l \in \{0, \dots, NP\}$$

4.4.3 Les contraintes

1. Non libération conditionnelle du matériels

Dans le cas où un chantier j a besoin de l'engin i dans le mois $(l+1)$ alors le chantier j

garde l'engin i (sachant que l'engin i se trouve déjà au chantier j durant le mois l).

$$X_{ijkl+1} \geq DS_{jl} \cdot X_{ijkl}, \quad \forall i \in \{1, \dots, NE\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}, \forall l \in \{0, \dots, NP\}$$

2. La satisfaction des demandes

1- Contrainte de disponibilité :

Si $D_l \leq Disp_l$, (la demande des engins le mois l inférieure ou égale a la disponibilité des engins), nous aurons dans ce cas une satisfaction de la demandes :

$$\sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=1}^{NJ} X_{ijkl} = D_l, \quad \forall l \in \{0, \dots, NP\} \quad (1)$$

Sinon : si $D_l > Disp_l$:

$$\sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} < D_l, \quad \forall l \in \{0, \dots, NP\} \quad (2)$$

$$\text{De (1) et (2) : } \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} \leq D_l, \quad \forall l \in \{0, \dots, NP\} \quad (3)$$

dans ce cas nous ferons appel à la location

2- Contrainte de location :

Dans le cas où $D_l > Disp_l$ nous sommes donc contraints de louer .

$$\sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} + \sum_{i'=0}^{NL_l - DM_l} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} DM_l \cdot Y_{i'+1jkl} = D_l, \quad \forall l \in \{0, \dots, NP\}.$$

Remarque : Faisons remarque si $NL_l = 0 \implies$ nous aurons pas besoin de location ($DM_l = 0$).

Sinon $NL_l \geq 1 \implies i'+1 \geq 1$ ($i'+1$: le numéro d'engin loué (le premier engin loué a le numéro 1)).

3. Contraintes d'affectations

Ce type de contraintes est plutôt simple et appartient au problème classique d'affectations, tout engin ne peut pas être affecté qu'à seul projet à la fois dans un mois « l ».

Son expression est comme suit :

$$\sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} = 1, \quad \forall i \in \{1, \dots, NE\}, \quad \forall l \in \{0, \dots, NP\}$$

De même que l'engin i loué ne peut pas être affecté à deux endroits simultanément.

$$\sum_{k=1}^{NS} \sum_{j=0}^{NJ} Y_{i'+1jkl} = DM_l, \quad \forall i' \in \{0, \dots, NL_l - DM_l\}, \quad \forall l \in \{0, \dots, NP\}$$

4.4.4 Les fonctions objectifs

1. Minimiser le coût de transport et le coût de location

Notre objectif est de minimiser le coût total des projets. Nous allons prendre en considération la distance entre deux projets, si la distance est petite les dépenses diminueront, de plus le coût de locations aussi, cet objectif est :

$$\mathbf{Z}_1(\min) = \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{k'=1}^{NS} \sum_{j=0}^{NJ} \sum_{j'=0}^{NJ} \sum_{l=0}^{NP} CT \cdot L_{jj'kk'} \cdot X_{ijkl} \cdot X_{ij'k'l+1} + \sum_{i'=0}^{NL_l - DM_l} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} \sum_{l=0}^{NP} DM_l \cdot Y_{i'+1jkl} \cdot PC.$$

CT : Le coût de transport au kilomètre d'un engin de type i .

PC : Prix unitaire de location d'un type d'engin.

$L_{jj'kk'}$: Longueur (distance) en kilomètre, séparant deux projets et deux structures, avec $j, j' \in \{0, \dots, NJ\}$ et $k, k' \in \{1, \dots, NS\}$, nous avons ($L_{jj'kk'} = 0$ si $k=k'$ et $j=j'$).

2. L'importance des projets

Cette fonction objectif n'est donnée que pour favoriser les projets importants (critiques). Pour chaque projet nous allons associer une pondération.

Le but ici est de maximiser le nombre des projets importants, son expres-

sions est :

$$\mathbf{Z}_2(\max) = \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} \sum_{l=0}^{NP} ET_j \cdot X_{ijkl} + \sum_{i'=0}^{NL_l - DM_l} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} \sum_{l=0}^{NP} ET_j \cdot Y_{i'+1jkl}.$$

4.4.5 Récapitulatif du modèle mathématique

La formulation mathématique de notre problème est la suivante :

$$(P) \left\{ \begin{array}{l} \mathbf{Z}_1(\min) = \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{k'=1}^{NS} \sum_{j=0}^{NJ} \sum_{j'=0}^{NJ} \sum_{l=0}^{NP} CT.l_{jj'kk'} \cdot X_{ijkl} \cdot X_{ij'k'l+1} + \sum_{i'=0}^{NL_l - DM_l} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} \sum_{l=0}^{NP} DM_l \cdot Y_{i'+1jkl} \cdot PC. \\ \mathbf{Z}_2(\max) = \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} \sum_{l=0}^{NP} ET_j \cdot X_{ijkl} + \sum_{i'=0}^{NL_l - DM_l} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} \sum_{l=0}^{NP} ET_j \cdot Y_{i'+1jkl}. \\ X_{ijkl+1} \geq DS_{jl} \cdot X_{ijkl}, \quad \forall i \in \{1, \dots, NE\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}, \forall l \in \{0, \dots, NP\}. \\ \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} \leq D_l, \quad \forall l \in \{0, \dots, NP\} \\ \sum_{i=1}^{NE} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} + \sum_{i'=0}^{NL_l - DM_l} \sum_{k=1}^{NS} \sum_{j=0}^{NJ} DM_l \cdot Y_{i'+1jkl} = D_l, \quad \forall l \in \{0, \dots, NP\}. \\ \sum_{k=1}^{NS} \sum_{j=0}^{NJ} X_{ijkl} = 1, \quad \forall i \in \{1, \dots, NE\}, \quad \forall l \in \{0, \dots, NP\}. \\ \sum_{k=1}^{NS} \sum_{j=0}^{NJ} Y_{i'+1jkl} = DM_l, \quad \forall i' \in \{0, \dots, NL_l - DM_l\}, \quad \forall l \in \{0, \dots, NP\}. \\ X_{ijkl} \in \{0, 1\}, \quad \forall i \in \{1, \dots, NE\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}, \forall l \in \{0, \dots, NP\} \\ Y_{i'jkl} \in \{0, 1\}, \quad \forall i' \in \{0, \dots, NL_l - DM_l\}, \forall j \in \{0, \dots, NJ\}, \forall k \in \{1, \dots, NS\}, \forall l \in \{0, \dots, NP\} \end{array} \right.$$

4.5 Évaluation et critique du modèle

Après avoir créé notre modèle mathématique nous allons faire une petite analyse concernant la taille de notre problème, et évaluer le nombre de variables et le nombre de contraintes.

L'évaluation de ces derniers a pour but d'estimer la complexité et la taille du problème d'optimisation combinatoire multi-objectif en question.

4.5.1 Nombre des variables et des contraintes

1. Nombre de variables :

- Le nombre de variables de type x_{ijkl} est évalué comme suit : $NE \cdot NJ \cdot NP$
- Le nombre de variables de type y_{ijkl} est évalué comme suit : $NL \cdot NJ \cdot NP$
- Le nombre total de variables tous types confondus est : $NJ \cdot NP \cdot (NE + NL)$

2. Nombre de contraintes :

- Le nombre de contraintes de type (1) est évalué comme suit : $NE \cdot NJ \cdot NP$
- Le nombre de contraintes de type (2) est évalué comme suit : NP
- Le nombre de contraintes de type (3) est évalué comme suit : $NP \cdot (NE + NL)$.
- Le nombre total des contraintes est : $NP \cdot (NE \cdot (NJ) + NL + 1)$.

Étude d'un cas particulier :

Nous allons prendre un exemple particulier où l'entreprise possède environ 42 auto bétonnière en activité. Pour seulement 4 structures et chaque structure possède 10 projets pour une planification des 12 mois, pour cette étude nous aurons 25200 variables et 20676 contraintes.

Remarque : On note que dans cet exemple nous avons posé que le nombre d'engin loué est nul ($NL=0$), et malgré ça la taille du modèle est relativement grande pour un exemple de 4 structures de 10 projets chacun sur une année d'étude, ceci reflète la complexité du problème étudié.

4.5.2 Caractéristiques du modèle mathématique

Après avoir créé notre modèle nous avons constaté quelques points importants, qui sont les suivants :

- Des variables de décisions bivalentes .
- Toutes les contraintes sont de nature linéaires.
- Une fonction objectif linéaire et une autre non linéaire.
- Une taille très importante du problème.

Nous avons pu construire un modèle mathématique qui reflète notre problématique sous la forme d'un programme d'optimisation combinatoire non linéaire multi objectif.

La taille du problème est très grande, dans ce cas nous allons sans doute résoudre notre problème à l'aide des méthodes approchées (heuristique et méta heuristique) .

Chapitre 5

Méthode choisie et implémentation du résultats

Le présent chapitre qui traite la résolution du problème s'articule autour de trois parties :

- Une première partie relative à l'utilisation d'une heuristique dans le but d'avoir une solution de bonne qualité.
- Une deuxième partie relative à l'utilisation d'une méta-heuristique dans le but d'essayer d'améliorer la solution donnée par l'heuristique définie dans la partie 1 ci-dessus.
- Enfin une dernière partie consacrée à l'implémentation du logiciel.

Partie 1 : Heuristique

5.1 Heuristique

La première étape de résolution qu'on va appliquer est l'utilisation d'une heuristique glou-
tonne, cette méthode sera conçue spécifiquement pour le traitement de notre problème.

5.1.1 Justification du choix

Dans ce genre de problème qui possède une taille très importante (un grand nombre de
données et variables) et de plus avoir deux fonctions dont l'une est non linéaire, il est difficile
d'utiliser une méthode de résolution exacte, pour le résoudre en un temps raisonnable, donc le
recours à une heuristique est une nécessité. Pour résoudre notre problème et pour avoir la pla-
nification désirée, nous procédons dans notre méthode progressivement en affectons les engins
mois par mois, dans ce cas à la fin de chaque mois nous allons réactualiser le positionnement
de nos engins ainsi que la disponibilité de ces derniers et les demandes des différents chantiers
pour le prochain mois, afin de faire une affectation qui correspond à nos objectifs.

5.1.2 Le principe de l'heuristique « plan »

Dans le tout début nous possédons les informations suivantes :

- La liste de tous les engins et leurs différents types que possède la société.
- Les demandes de matériels des différents chantiers pendant toute la période d'exercice.
- L'emplacement de tous les engins et chantiers et structures à travers le pays ainsi que la distance qui les sépare.

Et bien entendu nos buts sont la satisfaction maximum des demandes (tout en ce basons sur

la priorité des projets critique) l'utilisation maximum de notre flotte ainsi que la minimisation du temps et les dépenses de réalisation des projets.

En ce basons sur ça, nous allons construire l'heuristique qui traite un seule type a la fois, son algorithme est le suivant :

Algorithm 6 Algorithme Heuristique

Début

1. Fixer un type d'engins ; aller à (2) ;
2. Faire la liste des engins de ce type ;
3. Au début de la période :
 - 3.1) Faire le récapitulatif général des demandes en ce type d'engins par tous les chantiers pour le mois à venir ; aller en (3.2) ;
 - 3.2) **Si** les demandes sont nulles, alors affecter tous les engins de ce type vers les parcs des structures et passer au mois suivant ; aller en (3.1) ;
Sinon, aller a (3.3) ;
 - 3.3) Localiser l'engin durant le mois précédent ; aller en (3.4) ;
 - 3.4) Récapituler les demandes en ce type d'engins pour le prochain mois dans chacun des chantiers et classer ces derniers suivant l'ordre croissant du rapport :(distance sur pondération), aller en (3.5) ;
 - 3.5) Affecter cet engin au chantier demandeur le plus proche pour le mois à venir ; aller en (3.6) ;
 - 3.6) **Si** toutes les demandes sont satisfaites, aller en (3.7) ;
Sinon, **Si** nombre d'engins de la société non utilisés est non nul, aller en (3.3) ;
Sinon aller en (3.8) ;
 - 3.7) **Si** le nombre d'engins de la société non affecté est non nul, alors affecter ces derniers aux parcs des structures, aller en (3.11) ;
 - 3.8) Récapituler les demandes restantes dans chacun des chantiers et classer ces derniers selon leurs importance ;
 - 3.9) Loué un engin au chantier qui est le plus important ;
 - 3.10) **Si** toutes les demandes sont satisfaites, aller en (3.11) ;
Sinon aller à (3.9) ;
 - 3.11) **Si** tout les mois sont parcouru aller à (1) ;
Sinon, aller à (3.1) ;

Fin

5.1.3 Principe d'application

1. Faire le tableau récapitulatif de toutes les demandes de tous les chantiers pour chaque mois de la période et pour chaque type d'engins.
2. Pour chaque mois et chaque type d'engins, comparer les demandes de tous les chantiers, on aura les cas suivants :
 - Les demandes sont nulles, il n'y aura pas d'affectation du type d'engins considéré (tous les engins sont alors envoyés vers les parcs des structure pour le mois à venir).
 - Les demandes dépassent le nombre d'engins de ce type que possède l'entreprise (GCB), on fait appel à l'heuristique « plan », et on détermine le nombre de locations externes nécessaires.
 - Le patrimoine de GCB en ces engins est supérieur à toutes les demandes, ici aussi l'heuristique « plan » intervient et le nombre de locations externes est nul.

Partie 2 : Méta-heuristique

5.2 Méta heuristique (algorithme génétique)

Dans cette deuxième partie nous allons présenter un algorithme génétique que nous avons choisis pour résoudre notre problème, il s'agit de la méthode NSGA II (Nondominated Sorting Genetic Algorithm II), ce dernier s'avère très pratique et efficace pour résoudre les problèmes d'optimisation multi-objectif a variables bivalentes, à condition le bien régler pour qu'il soit applicable sur notre problème (méthode, codage, opérateur de sélection. . .).

5.2.1 Justification de l'utilisation de NSGA II

[44],[5]

Les algorithmes génétiques sont de plus en plus utilisés pour résoudre des problèmes complexes, et ça a cause de leurs capacité non seulement à explorer diverses régions du domaine de réalisabilité mais aussi améliorer les solutions non seulement de façons d'intensifications mais aussi de diversification ainsi nous évitons le risque de rester dans un optimum local.

Une autre raison qui nous aide à choisir et utiliser cette méthode est la structure des solutions de notre problème. Comme on a abordé dans le début de cette partie nos variables sont de type binaire et le codage du chromosome peut être structuré et présenté de façon simple.

Dans les problèmes multi-objectifs comme celui auquel nous sommes confrontés, une étape difficile des algorithmes génétiques est celle du calcul et de la comparaison des performances des vecteurs critères, pour les solutions d'une même population afin de les classer du meilleur au

pire. L'avantage de l'algorithme NSGA II est qu'il offre une technique intéressante de classement des solutions, selon la notion de dominance, mais aussi selon le concept de la distance de Crowding que nous allons soigneusement définir ci-après.

5.2.2 Stratégie d'application de NSGA II

[44],[15],[51]

En 2000, Deb et al ont proposé un algorithme évolutionnaire multi-objectifs élitiste qu'ils ont appelé NSGA II pour indiquer les origines de la nouvelle approche. la seule chose que NSGA II a hérité de son prédécesseur NSGA est le principe de base du calcul de la performance (notions de classement des solutions dans des fronts successifs de solutions non dominées, soit le Ranking, puis classement par performances décroissantes dans chaque front).

La présence d'élites augmente les chances de créer de meilleurs enfants [16]. Ceci implique qu'un algorithme utilisant l'Élitisme convergera plus rapidement. Dans le cas des problèmes d'optimisation mono-critère les élites se repèrent facilement. Ils ont la meilleure valeur par rapport à la fonction d'objectif. Pour les problèmes d'optimisation multicritères cette affirmation n'est plus valide, car il y a présence de plusieurs fonctions objectifs. Afin de gérer les élites, l'algorithme NSGA-II de Deb [15] utilise le rang de l'ensemble de Pareto ou se trouve la solution. Donc les élites seront identifiées comme étant les solutions faisant partie de l'ensemble de Pareto F_1 (l'ensemble qui contient les éléments non dominé de notre population autrement dit les élites de notre population). Pour gérer la diversité au niveau des ensembles de Pareto, l'algorithme utilise une implantation de l'opérateur de sélection bien particulière, soit le crowded

tournement (algorithme 7)

Définition 5.2.2.1

[40] Soit une population P classer en plusieurs sous-ensembles F_1, F_2, \dots , deux solutions $z_m^{(i)} \in F_m$ et $z_n^{(j)} \in F_n$, $F_m, F_n \subset P$, la solution z_m remportera le tournoi si :

1- $m < n$.

OU

2- $m=n$ et $d_1 > d_2$.

Dans la procédure d'assignation de la Crowded distance, notée par $D : P \rightarrow R^+$, la variable d_i évalue la densité des solutions présentes autour d'une solution i . L'assignation de cette valeur aura comme effet de diminuer les chances de survie d'une solution i présente dans une région où plusieurs autres solutions y sont concentrées. Concrètement, comme le montre la figure (5.1), le fait de trier le vecteur d'indice $I^{(j)}$ et d'assigner une distance très grande aux premières et dernières solutions, pour chaque fonction d'objectif (voir algorithme 7), permet de donner priorité aux solutions extrêmes d'un ensemble de Pareto. Pour les solutions intermédiaires, toujours grâce au vecteur de tri, la distance est donnée par le demi-périmètre du cuboïde entourant ces solutions.

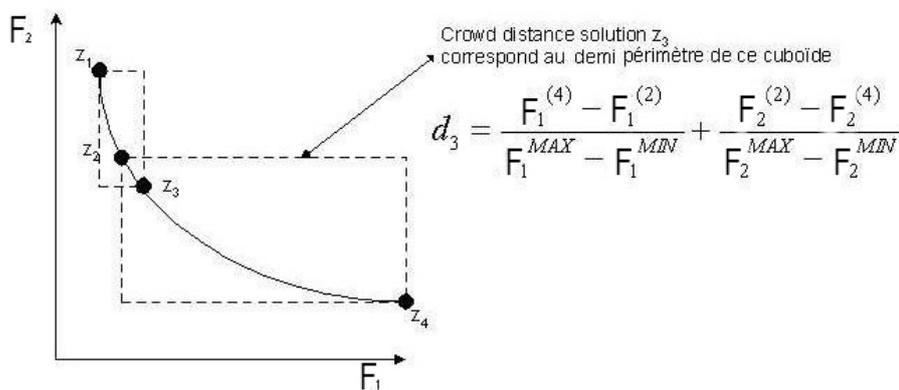


FIGURE 5.1 – Cas hypothétique de l'assignation de la Crowded distance.[51]

Dans l'exemple montré à la figure (5.1), la solution z_2 est celle qui aura plus de chance d'être rejetée étant donné que le périmètre de son cuboïde est le plus petit. Il est aussi important de noter que la densité est mesurée dans l'espace d'objectif et non de décision et que la méthode inclue directement une normalisation, ce qui est indispensable lors des calculs de distances. Finalement l'algorithme NSGA-II (algorithme 8) utilise deux populations. La population \mathbf{Q} , constituée de l'ensemble des individus qui ont été créés par l'application de l'opérateur de sélection 1 (définition 17) et une deuxième population \mathbf{P} de taille N . À chaque itération les deux populations, de taille N , seront combinés dans une même population R et triées de façon à obtenir les ensembles de Pareto. Par la suite, une nouvelle population P_{t+1} de taille N sera constituée des meilleurs ensembles de Pareto de la population R . Pour y arriver, les solutions des ensembles de Pareto $F_k R_t$ seront incluses dans la population jusqu'à ce que la taille devienne supérieure ou égale à N .

- Si, suite à l'ajout du dernier ensemble de Pareto possible, la taille de la population est supérieure à N , alors le dernier ensemble sera trié selon la crowded distance et les solutions ayant les plus petites distances seront éliminées jusqu'à ce que la taille de P_{t+1} soit égale à N . L'opérateur de sélection, croisement et mutation seront alors appliqués sur P_{t+1} pour créer la nouvelle population Q_{t+1} .

Algorithm 7 Opérateur d'assignation de la Crowded distance [15]1. **Données :**

I : Vecteur d'indice.

F_i : Sous-ensemble de Pareto.

d_j : Crowded distance de la solution i .

f_k : La $k^{\text{ème}}$ fonction objectif.

f_m^{max} : Valeur maximale de la $m^{\text{ème}}$ fonction objectif.

f_m^{min} : Valeur minimale de la $m^{\text{ème}}$ fonction objectif.

Début :

Initialiser $l = ||F_i||$, $i \in \{1, 2, \dots, n\}$.

pour $z^{(j)} \in F_i$ **faire :**

Initialiser $d_j = 0$.

Fin pour

pour chaque fonction objectif f_k **faire :**

Trouver le vecteur d'indice $I^{(k)} = \text{Trier}(f_k, <)$

(i.e : trier les éléments de l'ensemble F_i pour la fonction f_k objectif façon décroissante).

Fin pour

pour chaque fonction objectif f_k **faire :**

Assigner $d_1^m = d_l^m = \infty$.

pour chaque $z^{(i)}$ $i=2, \dots, (l-1)$ **faire :**

Calculer

$$d_{I_j}^m = d_{I_j}^m + \frac{f_{I_{j+1}}^m - f_{I_j}^m}{f_m^{\text{max}} - f_m^{\text{min}}}$$

Fin pour**Fin pour**

fin

Algorithm 8 NSGA-II [41]1. **Données :**

D : Opérateur d'assignation de Crowded distance.

F_k : Sous-ensemble de Pareto.

M : Opérateur de mutation.

P : Population d'individus.

Q : Population des enfants.

R : Population enfants - individus.

S : Opérateur de sélection.

T : Opérateur de tri par ensemble de Pareto.

X : Opérateur de croisement.

$z_{(i)}^k$: Solution i faisant partie de l'ensemble de Pareto k.

Début :

Créer une population initiale P_0 et Q_0 .

pour chaque itération t **faire** :

$$R_t = P_t \cup Q_t$$

Appliquer $R' = T(R)$.

Assigner $P_{t+1} = \emptyset$.

tant que $\|P_{t+1}\| < N$ **faire** :

k = k + 1 et $P_{t+1} = P_{t+1} \cup F_k$.

Calculer $D(z_k^{(i)}) \forall i \in F_k$

Trier($F_k < d$) (i.e : trier les éléments des ensembles F_k $k \in \{1, \dots, n\}$, de façon décroissante

en se basant sur la Crowded distance (d).

i = 1

répéter

$$F_k = F_k / z_k^{(i)}$$

i = i + 1.

jusqu'à ce que $\|P_{t+1}\| = N$.

Appliquer

$Q_{t+1} = M(X(S(P_{t+1})))$ (i.e : la procédure de l'algorithme NSGA-II : étape 1 la sélection (S), étape 2 le croisement (X), étape 3 la mutation (M).)

Fin pour

fin

5.2.3 Adaptation de NSGA II à notre problème

En arrivant à ce stade, nous appliquons la méthode NSGA II pour résoudre notre problème afin de trouver l'affectation de chaque engin durant chaque mois de notre période d'étude (un an dans notre cas), l'utilisation de cette méthode nécessite une bonne fixation des paramètres

de nos algorithmes.

5.2.3.1 Représentation d'une solution

Dans le tout début nous allons parler de la représentation de la solution, dans ce cas nous parlons du codage de cette dernière, en effet il faut faire en sorte de bien représenter notre solution de façon simple et explicite, ainsi elle ne va pas coûter beaucoup du côté d'espace mémoire. Dans notre cas notre solution étant composée d'éléments X_{ijkl} binaires, il suffira de stocker ses éléments dans un vecteur de bits de longueur $(NE).(NJ).(NP)$, disposés comme l'indique la figure suivante :

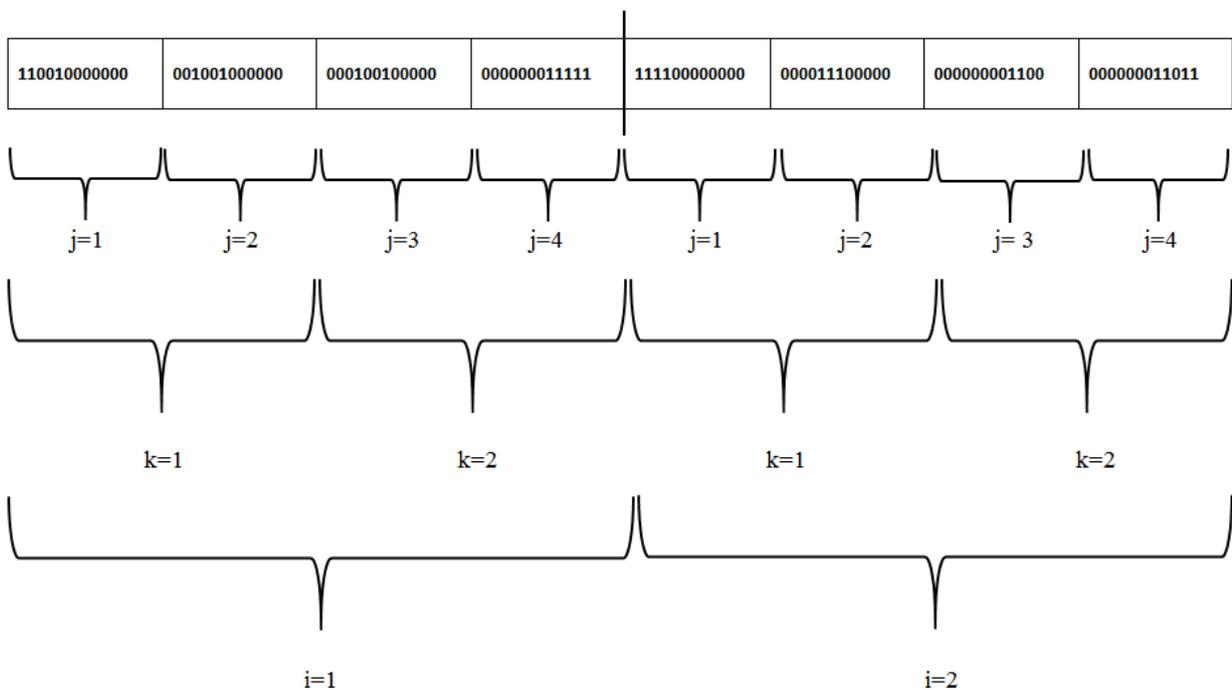


FIGURE 5.2 – comment ce fait la représentation d'une solution

Remarque : Dans ce schéma on a pris $NE=2$ engins, $NJ=4$ qui sont distribués à $NS=2$ structures dans la période d'exercice $NP=12$ mois .

5.2.4 Création de la première population

L'étape initiale consiste en la création pseudo aléatoire d'individus réalisables pour notre problème, ainsi on forme la génération du départ et nous pouvons utiliser la solution qu'on aura dans l'étape première (heuristique). Soit la taille de la population à une génération quelconque et notons $M[i,j,k,l]$ les éléments d'un individu :

Algorithm 9 Algorithme génér popul initiale**Début**

```

— Pour i=1 à NE faire :
  Localiser l'engin au cours du mois l = 0 et remplir M [i,j,k,0];
Fait ;
  Créer le dataset(popul);
  Ajouté le premier individu dans le dataset (la solution heuristique)
  NL=0;
  t=2;
Tant que t ≤ N faire :
  Écrire (popul,t);
  Pour l =1 à NP faire :
  Pour k =1 à NS faire :
  Pour j=1 à NJ faire :
  Pour i=1 à NE faire :
  Tant que (M [i,j,k,l-1] =1) et ( demande (j,k,l) non satisfaite)
    et (DS[j,k,l] = 1) faire :
    M [i,j,k,0] = random 0;1;
  Fait
  Fait
  Fait
  Fait
  Pour k = 1 à NS faire :
  Pour j =1 à NJ faire :
  Tant que (demande (j,k,l) non satisfaite) et (les engins ne sont pas tous utilisés) faire :
  Choisir aléatoirement un engin i non affecté encore;
  M [i,j,k,l] =1;
  Affecter l'engin i au chantier j de la structure k
  Fait
  Fait
  Fait
  Pour k = 1 à NS faire :
  Pour j =1 à NJ faire :
  Si (demande (j,k,l) non satisfaite) et (tous les engins sont utiliser) alors Loué un
  engin i' pour le chantier j de la structure k;
  NL=NL+1;
  Fin si
  Fait
  Fait
  Pour l =0 à NP faire
  Pour k =1 à NS faire
  Pour j=1 à NJ faire
  Pour i=1 à NE faire
  Écrire (popul, M [i,j,k,l] );
  Fait
  Fait
  Fait
  Fait
  Calculer le critère 1;
  Écrire (popul, critère 1);
  Calculer le critère 2;
  Écrire (popul, critère 2);
Fait
Fin

```

5.2.4.1 Classement par fronts

Après avoir généré la population nous allons classer les solutions en des sous ensembles successifs de fronts de solutions non dominées, cette démarche est réalisée en utilisant les critères calculer lors de la création de notre population. Dans cette étape nous allons donner à chaque solution un rang de façon à ce que les solutions qui appartient au même ensemble possèdent le même rang, et que l'ensemble qui possède le rang 1 contient les solutions non dominées de la population. Cette méthode est généralement appelé « Ranking » de nos solutions.

La procédure de Ranking est la suivante :

Algorithm 10 Algorithme Ranking

Début

1. Soit S l'ensemble des solutions (individus) de la population ;
 $K=1$;
Tant que $S \neq \emptyset$ **faire** :
 $S' = \emptyset$;
Pour tout $s_i \in S$ **faire** :
Si s_i une solution non dominée dans S **alors**
 $\text{Rang}(s_i) = K$;
 $S' = S' \cup \{s_i\}$;
Fin si
Fin pour
 $S = S/S'$;
 $K = K + 1$;
Fin tant que

Fin.

Après avoir terminé cette étape nous pouvons à première vue classer les solutions de notre population d'après leurs qualité de meilleurs aux pires, ceci en ce basant sur le faite qu'une solution est de bon qualité si son rang est petit.

5.2.4.2 Distance de crowding

Après avoir ordonné nos solutions par fronts de dominances, la prochaine étape consiste à comparer entre les solutions qui ont le même rang.

C'est dans cette étape que la distance de crowding entre en scène, cette dernière joue un rôle très important dans la méthode NSGA II, en effet ce point est ce qui différencie la méthode NSGA II de son prédécesseur NSGA dont l'algorithme se base sur le calcul de fonctions de performance (fitness) des solutions pour pouvoir les classer.

La distance de crowding est une mesure de densité de surpeuplement pour chaque solution de notre population. Elle est donnée la somme des densités respectivement calculées pour chacun des critères (deux critères dans notre cas), cette méthode de calcul est expliquée plus haut dans la partie (Stratégie d'application de NSGA II).

Après la réalisation de cette procédure nous aurons une classification pour notre population Par le « ranking » de façon croissante et par « la distance de crowding » de façon décroissante, cette classification sera éventuellement sauvegardée dans un dataset (popul2).

5.2.4.3 Sélection pour le remplacement

Le nouveau dataset (popul2) va être créé de la façon suivante :

Premièrement les solutions des petits rangs seront favorisées et précèdent les autres solutions dont le rang est plus grand, et deuxièmement si deux solutions appartiennent au même ensemble (autrement dit possède le même rang) dans ce cas la priorité à la solution qui a plus grande

densité de surpeuplement (distance de crowding), ainsi cette dernière a plus de chance pour apparaître dans le dataset (popul2).

A cette étape nous allons éliminer $N/2$ individu qui représente les solutions les moins bons, dans ce cas nous allons garder l'autre moitié de population, éventuellement cette moitié sera présenté dans la prochaine génération, et ces derniers vont être utilisés pour se reproduire en utilisant « le croisement », c'est de cette façon qu'on va compléter notre nouvelle population et ainsi avoir N individus, cette méthode vas aider à faire une recherche plus avancée et plus diversifié ainsi évité la possibilité de ce bloquer dans une partie du domaine de réalisation.

5.2.4.4 Sélection pour la reproduction

Après avoir supprimé les plus mauvaises solutions nous allons tirer aléatoirement des couples des individus restants et ainsi complété notre population, vu que nous avons supprimé $N/2$ individus alors il faut $N/2$ enfants, et dans ce cas nous aurons besoin de $N/2$ parents aléatoirement tiré. Ce processus est présenté comme suit :

Algorithm 11 Algorithme RAND**Début**

1. $i = \text{random}\{1 \dots N/2\}$;
 $j = \text{random}\{1 \dots N/2\} \setminus \{i\}$;
Si $i < j$ alors $f=j$;
Sinon $f=i$;
Fin si ;
 $i = \text{random}\{1 \dots N/2\}$;
 $j = \text{random}\{1 \dots N/2\} \setminus \{i\}$;
Si $i < j$ alors $h=j$;
Sinon $h=i$;
Fin si ;

On ressort avec f et h les indices des deux parents, et on extrait alors les deux chromosomes correspondants du dataset « Popul 2 » vers deux structures « HOM[i,j,k,l] » et « FEM[i,j,k,l] » ;

Fin.

Fin.

5.2.4.5 Description de l'opérateur de croisement

Cette étape est la plus cruciale dans l'application des algorithmes génétiques (évolutionnaires), à ce niveau il faut fixer un opérateur de croisement qui va jouer un rôle important dans la reproduction des parents élu dans la partie précédente afin d'avoir les individus enfants qui complète notre population, et aussi aide à exploiter diverses régions dans notre domaine.

Pour définir un opérateur de croisement dans notre cas c'est un peu difficile et compliqué, en effet si cet opérateur ne serait pas bien élaboré alors nous pourrions avoir des individus non réalisables (soit affecté du matériel à des chantiers qui n'auront pas besoin), ceci peut orienter notre algorithme à une fausse piste ainsi nous n'aurons pas une bonne solution à la fin et même dans certains cas avoir une situation bien pire (celle d'avoir une solution finale non réalisable).

Malgré le fait que dans certains cas nous pouvons appliquer des méthodes qui pénalisent les solutions non réalisables qui peuvent être produites, mais malheureusement dans le cas où nos solutions sont codées en binaire et dans ce cas les solutions ont beaucoup de similitude entre elles, ainsi toute application qui touche une solution va systématiquement modifier d'autres solutions (individus). Nous allons présenter la procédure que nous allons utiliser dans l'algorithme de croisement ci-dessous :

Algorithm 12 Algorithme croisement**Début**

-
- 1 - Soient $HOM[i,j,k,l]$ et $FEM[i,j,k,l]$ les deux individus parents sélectionnés lors des choix pour la reproduction ;
- 2 - Soit P la probabilité de croiser deux individus, préalablement fixés ;
- 3 - Générer aléatoirement un nombre dans $[0;1]$, $x=random[0;1]$;
- Si** x inférieur ou égale P **alors** :
- Déterminer les mois sur lesquels on peut croiser, comme suit :
- $S = \emptyset$;
- Pour** l à NP **faire** :
- Croisable :=vrai ;
- Pour** $k=1$ à NS **faire** :
- Pour** $j=1$ à NJ **faire** :
- Si** $DS [j;k]=1$ **alors**
- Croisable :=faux ;
- Fin si**
- Fait** ;
- Fait** ;
- Si** croisable = vrai **alors**
- $S = S \cup \{k\}$
- Fin si**
- Fait** ;
- Si** $S \neq \emptyset$ **alors** :
- $n := random \{S\}$.
- Pour** $i = 1$ à NE **faire** :
- Pour** $j = 1$ à NJ **faire** :
- Pour** $k=1$ à NS **faire** :
- Pour** $l=n$ à NP **faire** :
- $C=HOM[i,j,k,l]$;
- $HOM[i,j,k,l]=FEM[i,j,k,l]$;
- $FEM[i,j,k,l]=C$;
- Fait**
- Fait**
- Fait**
- Fait**
- Fin si**
- Fin si**
- 4 - Calculer les valeurs des critères 1 et 2 pour $HOM[i,j,k,l]$;
- 5 - Ouvrir le dataset (popul 2) ;
- 6 - A la suite du dataset, enregistrer l'indice de $HOM[i,j,k,l]$;
- 7 - Enregistrer $HOM[i,j,k,l]$;
- 8 - Enregistrer la valeur du critère 1 puis celle du critère 2 de $HOM[i,j,k,l]$;
- 9 - Calculer les valeurs des critères 1 et 2 pour $FEM[i,j,k,l]$;
- $t := t+1$;
- 10 - A la suite du dataset, enregistrer l'indice t de $FEM[i,j,k,l]$;
- 11 - Enregistrer $FEM[i,j,k,l]$;
- 12 - Enregistrer la valeur du critère 1 puis celle du critère 2 de $FEM[i,j,k,l]$;
- Fin.**
-

Avec cette méthode malgré le fait que le nombre de croisement possible est devenu plus petit mais ça nous évite la possibilité de tomber dans les cas d'avoir des individus non réalisables, dans ce cas notre croisement est validé.

5.2.4.6 Description de l'opérateur de mutation

Après le croisement des $N/2$ parents sélectionnés dans la partie précédente, une petite proportion des enfants obtenus « nouveaux individus » vont subir une mutation, cette action va faire en sorte de varier de plus en plus les solutions obtenues (individus) tout en gardant notre algorithme génétique opérationnelle.

La mutation de nos individus se résume en prendre un élément de ce dernier qui égale à 1 et le changer à 0 ou l'inverse ainsi changer l'affectation du matériel (récupéré un engin d'un chantier et l'affecté à un autre chantier qui en a besoin) .

Dans notre cas il faut que les demandes des mois concernés par la mutation ne soient pas saturées car dans le cas où les demandes sont saturées alors même si nous allons prendre un engin nous n'aurons pas d'autre choix que de le remettre à sa place initiale et ainsi y'aura pas de mutation.

Toute cette procédure est représentée comme suit :

Algorithm 13 Algorithme mutation

Début

1. Soit $HOM[i,j,k,l]$ un individu « enfant » nouvellement créé ;
2. Déterminer l'ensemble d'indices $S = (j,k,l)$ avec $DS[j,k,l]$ et la demande du mois l du projet j de la structure k est non nul ($demande[j,k,l] \neq 0$) ;
3. Choisir aléatoirement (j^*,k^*,l^*) dans S ;
4. Choisir aléatoirement $i^* \in \{1, \dots, NE\}$ tel que $HOM[i^*,j^*,k^*,l^*]=1$;
5. Vérifier s'il existe $j' \neq j^*$ tel que la demande $[j^*,k^*,l^*]$ ne soit pas satisfaite ;
 Si oui, alors
 - 1- Affecter i^* au chantier j' au cours du mois l^* , ($HOM[i^*,j',k^*,l^*]=1$) ;
 - 2- Annuler l'ancienne affectation de i^* , ($HOM[i^*,j^*,k^*,l^*]=0$) ;

Fin.

Cette mutation est validée et assure la réalisabilité du nouvel individu « enfant ».

Partie 3 : Implémentation et résultats

5.3 Implémentation et résultats

5.3.1 Introduction

Dans ce partie nous présentons la dernière étape de notre travail, sachant que nous utiliserons le langage de programmation « python » qui permet d'appliquer plus facilement la méthode de résolution proposée à notre modélisation mathématique .

5.3.2 Présentation de "python"

Python est un langage de programmation puissant et facile à apprendre, a été crée en 1989 par **Guido Van Rossum** [29], la première version publiée de ce langage a été publiée en 1991. [12]

Les principales caractéristiques de python sont les suivantes : [30] [12]

- "Open-source" : l'utilisation de python est gratuite et ses fichiers sources sont disponibles et modifiables.
- Simple et très lisible.
- Importante quantité de bibliothèques disponibles : pour le calcul scientifique, les statistiques, les bases de données, la visualisation, . . .,etc.
- Il est multiplateforme, c'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry, Pi jusqu'aux supercalculateurs.

5.3.2.1 Domaines d'application [42]

Python est utilisé dans de nombreux domaines d'application comme :

- L'apprentissage de la programmation objet.
- Les scripts d'administration système ou d'analyse de fichiers textuels.
- Les développement liés à l'internet et en particulier au Web : scripts CGI, navigateurs Web, moteurs de recherche, agents intelligents, objets distribués,...,etc.
- L'accès aux bases de données (relationnelles).
- La réalisation d'interfaces graphiques utilisateurs.
- Le calcul scientifique, numériques et l'imagerie.

5.3.3 Choix de langage "python"

Le choix d'utiliser le langage python est justifié par le faites qui satisfait les critères suivants :

Python est le meilleure pour le Big Data grâce à ses différentes packages et bibliothèque de science des données qui nous aident dans notre travail de telle sort :

- **Pandas** : est une bibliothèque d'analyse et de modélisation de données.
- **Numpy** : est une d'une bibliothèque open source en langage Python, elle est très utile pour effectuer des calculs logiques et mathématiques sur des tableaux et des matrices. [21].
- **PyQt 5** : est un module qui permet de lier le langage python à la bibliothèque, il permet de créer des interfaces graphiques.
- **Scipy** : est une collection de package pour les mathématiques, les algorithmes numériques, optimisation, sciences et l'ingénierie [26] .

5.3.4 Présentation du programme « GPLAN »

Ce programme « GPLAN » est destiné à la gestion du matériels et plus précisément la planification en matière d'affectation du matériel aux différents chantiers de la société (GCB).

Lors du lancement du programme, une fenêtre s'ouvre et affiche un menu, proposant ainsi différentes fonctionnalités à l'utilisateur, dont :

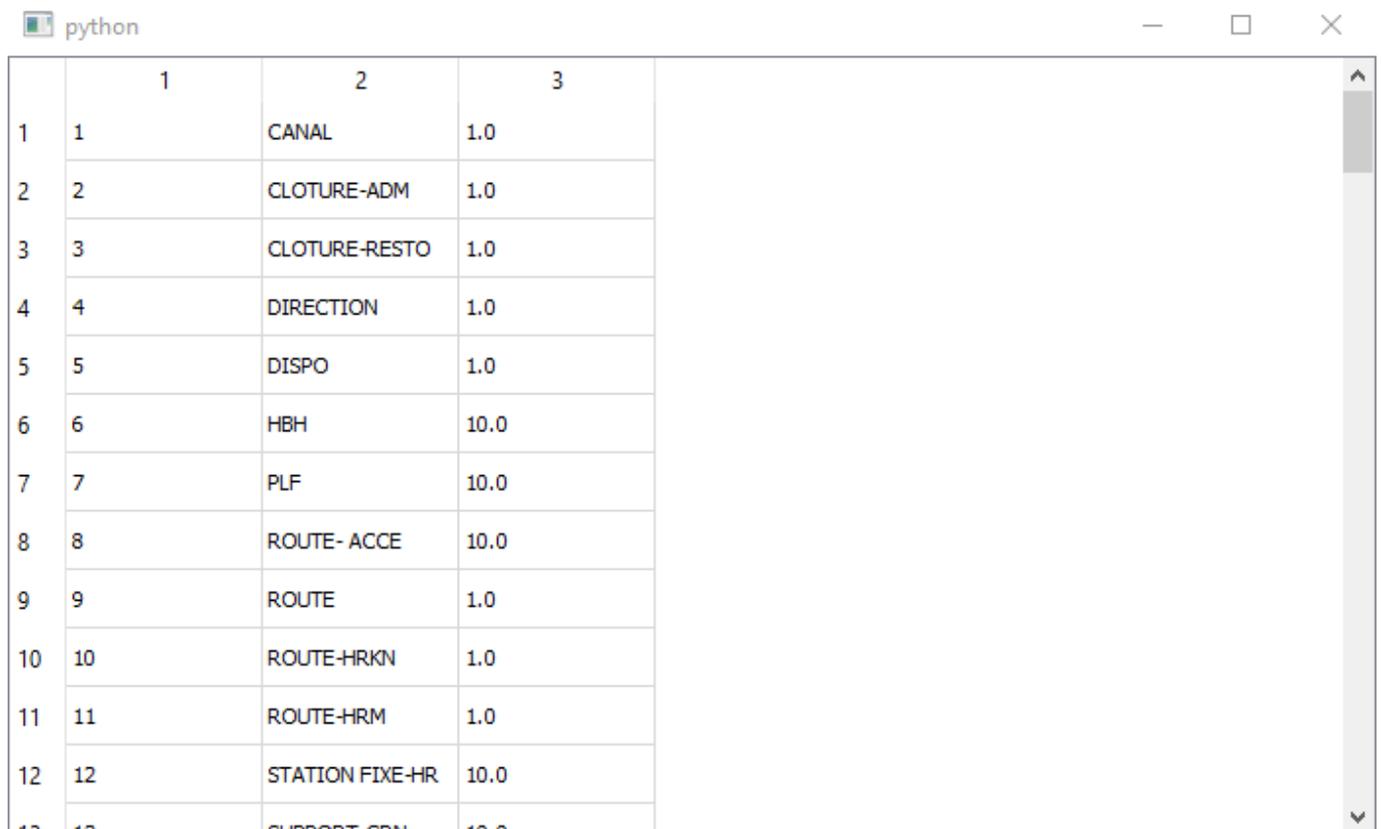
L'accès a une base de données contenant diverse information : Type d'engins, le nombre d'engins que la société possède de chaque type, les chantiers de la société GCB, la pondération de chaque type, et la planification des engins pour toute l'année.

La fenêtre principale est composée d'un menu à trois boutons donnant accès aux différentes fenêtres secondaires et d'un bouton pour quitter l'application.



FIGURE 5.3 – Menu « GPLAN »

On clicant sur le bouton « chantiers », nous aurons accès à un tableau qui contient trois genre d'informations, la première colonne contient le numéro du chantier, la deuxième contient le nom du chantier, et la dernière contient la pondération (importance du chantier), ces derniers sont représentés dans l'image ci-dessous

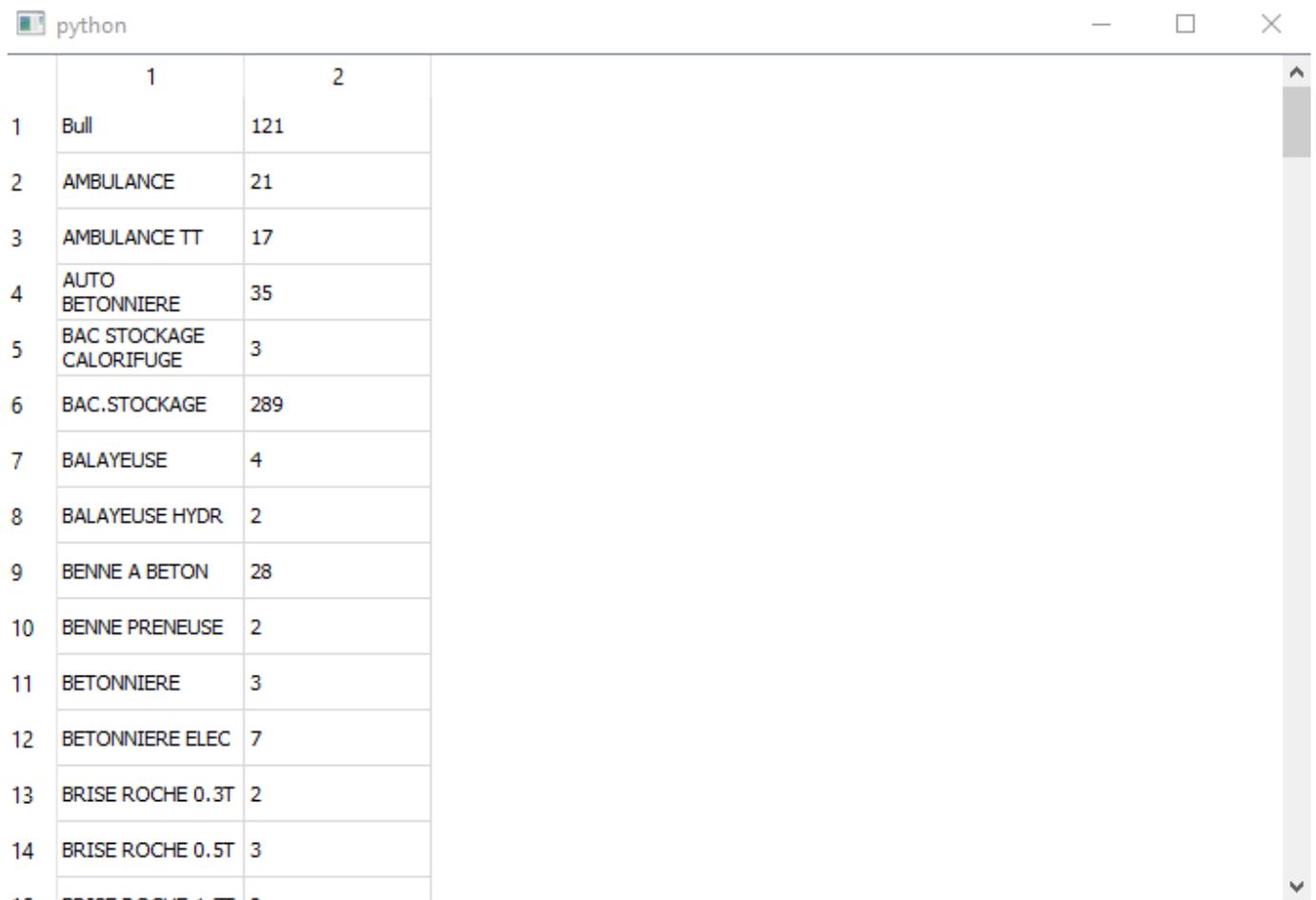


The image shows a screenshot of a Python application window titled 'python'. The window displays a table with three columns labeled '1', '2', and '3'. The table contains 12 rows of data, with the last row partially cut off. The data is as follows:

	1	2	3
1	1	CANAL	1.0
2	2	CLOTURE-ADM	1.0
3	3	CLOTURE-RESTO	1.0
4	4	DIRECTION	1.0
5	5	DISPO	1.0
6	6	HBH	10.0
7	7	PLF	10.0
8	8	ROUTE- ACCE	10.0
9	9	ROUTE	1.0
10	10	ROUTE-HRKN	1.0
11	11	ROUTE-HRM	1.0
12	12	STATION FIXE-HR	10.0
13	13	SUPPORT-ADM	10.0

FIGURE 5.4 – Affichage des chantiers

Un clique sur le bouton « nos engins » une deuxième fenêtre apparaît, cette dernière contenant un deuxième tableau qui contient deux colonnes, la première colonne contient les noms des engins (type d'engins), et la deuxième contient le nombre d'engins du même type mise en disposition de la société, cette fenêtre est représentée dans l'image ci-dessous.



The image shows a screenshot of a Python application window. The window title is 'python'. It displays a table with two columns labeled '1' and '2'. The table contains 15 rows of equipment data. The first column lists equipment names, and the second column lists numerical values. The data is as follows:

	1	2
1	Bull	121
2	AMBULANCE	21
3	AMBULANCE TT	17
4	AUTO BETONNIERE	35
5	BAC STOCKAGE CALORIFUGE	3
6	BAC.STOCKAGE	289
7	BALAYEUSE	4
8	BALAYEUSE HYDR	2
9	BENNE A BETON	28
10	BENNE PRENEUSE	2
11	BETONNIERE	3
12	BETONNIERE ELEC	7
13	BRISE ROCHE 0.3T	2
14	BRISE ROCHE 0.5T	3
15	BRISE ROCHE 1.5T	0

FIGURE 5.5 – liste d’engins

Enfin le dernier bouton « Planification » qui va afficher notre planification d’affectation du matériels sur toute la longueur de l’année, nous notons que chaque mois est représentée par deux colonnes, la première contient le nom du projet dans le quel l’engin est affecté pendant le mois, la deuxième contient le nom de structure qui gère le projet en questions.

Cette planification est représentée dans l’image ci-dessous.

Bull		2	3	4	5	6	
1	Bull	CINA-HMD	DSE	BASSIN.HBK	DSE	BV-BORMA	DSE
2	Bull	station fixe -HASSI R'MEL	DRO	DEV-GAZ-BRUT	DTC	DEV-GAZ-BRUT	DTC
3	Bull	CINA-HMD	DSE	HGA	DSE	HGA	DSE
4	Bull	station fixe -HASSI R'MEL	DPA	stations fixe - AOUGROUT	DPA	stations fixe - AOUGROUT	DPA
5	Bull	BASSIN.HBK	DSE	BV-BORMA	DSE	BV-BORMA	DSE
6	Bull	BBK	DSE	BBK	DSE	BBK	DSE
7	Bull	BBK	DSE	BBK	DSE	BBK	DSE
8	Bull	DEV-GAZ-BRUT	DTC	DEV-GAZ-BRUT	DTC	DEV-GAZ-BRUT	DTC
9	Bull	BASE DE VIE - H.ILATOU	DRO	PROJET-RRN	DTC	PROJET-RRN	DTC
10	Bull	Station fixe - AOUGROUT	DPA	stations fixe -HEH	DPA	stations fixe -HEH	DPA
11	Bull	ROUTE- P.ATTIRISSGE-...	DRO	COMPLEX-HMD	DSE	COMPLEX-HMD	DSE
12	Bull	Voie Ferrée Sidi Ladjel	DRO	BASE DE VIE - H.ILATOU	DRO	BASE DE VIE - H.ILATOU	DRO

FIGURE 5.6 – Planification d'affectation du matériels

Toute fenêtre peut être fermée on clicant sur la croix situe en haut à droite, et on quitte l'application à travers le bouton « **Quitter** » de la fenêtre principale.

5.3.5 Observations sur les résultats

Pour résoudre le problème proposé, nous avons choisi l'utilisation de deux étapes, à savoir une heuristique gloutonne et un méta heuristique basé sur l'évolution. Ceci nous a permis d'avoir dans la première étape un résultat de bonne qualité et dans la deuxième étape l'amélioration de nos résultats.

Malgré le fait que les algorithmes évolutionnaires pouvaient exploiter diverses parties du domaine de réalisabilité, nous avons remarqué que pour une partie des types d'engins la solution obtenue par l'heuristique gloutonne reste la meilleure, la raison de ce résultat vient partiellement du fait que la plupart de ces engins ont un nombre qui n'est pas grand, autrement dit les combinaisons des affectations réalisées sont très limitées et très peu variées et dans ce genre de cas l'application de l'algorithme génétique peu s'avérer inutile, de plus même les demandes d'engins exprimées par les chantiers peuvent aussi influencer l'application de l'algorithme génétique, en effet dans le cas où nous aurons peu de demandes alors les solutions sont générées seront limitées et ainsi avoir le même résultat que dans le premier cas.

Malgré le fait que les tests réalisés sont peu explicites, et nous ne pouvons pas dire quelle méthode est préférable pour ce genre de problème, et même si la méthode méta heuristique (algorithme génétique) ne peut pas toujours donner des résultats, la heuristique gloutonne arrive à donner une bonne solution (de bonne qualité) et nous pouvons être sûr de cela même sans beaucoup de tests numériques, car tout simplement cette heuristique est conçue spécialement pour ce problème et dans le but de donner une solution bonne et rapide, de plus cette dernière est acceptée par l'utilisateur.

Pour essayer d'avoir de plus en plus des solutions de bonnes qualités, il est préférable de faire beaucoup d'application du algorithme génétique tout en diversifiant ces paramètres (nombre de générations, taille de population,...), ainsi nous pouvons savoir si cette méthode peut atteindre des solutions Pareto-optimales.

Conclusion générale

Au cours de notre travail au sein de la société GCB, le département de planification nous a confié d'aborder la problématique de la planification en matière d'affectations du matériels aux différents chantiers pour une période d'exercice d'un an.

Les compétences acquises lors de étude sont premièrement l'apprentissage de diverses méthodes mathématiques pour résoudre les différents problèmes d'optimisation multi-objective de façon générale, et les problèmes d'affectation à plusieurs fonctions objectives (bi-objectives dans notre cas) en particulier, nous avons aussi acquis beaucoup de connaissances que se soit dans l'optimisation de façon générale ou dans l'aide à la décision, deuxièmement le développement de nos compétences en terme de langage de programmation et construction de logiciel et programme, et mieux ce familiariser avec le langage python qui est un environnement de programmation orienté objet (ce langage connaît un grand succès auprès des programmeurs).

Ce mémoire commence par la présentation de l'entreprise GCB ainsi que son domaine d'action et diverse autres informations la concernant, nous avons ensuite clairement défini notre problème, on posant bien les clauses à satisfaire dès le départ pour envisager une approche

scientifique de résolution. Dans la suite, nous avons présenté les différents problèmes d'optimisation et juste après nous avons entamé la présentation des méthodes de résolution les plus envisageables qui se divise en deux parties, exacte et approchée (heuristiques et méta heuristiques).

Ensuite la partie qui est après consiste à proposer c'est la modélisation mathématique du problème et nous avons évalué la taille des données et du nombre de contraintes y intervenant, et après présente la méthode qu'on va utiliser pour résoudre notre problème, dans cette partie nous avons opté pour deux étapes différentes parmi la catégorie des méthodes approchées pour la recherche de solution(s) acceptables. La première étape est rigoureusement construite, spécifiquement pour la recherche d'un planning des affectations relativement bon, en un temps d'exécution acceptable d'une séquence d'instructions bien structurée. L'heuristique gloutonne que nous avons pu concevoir est susceptible de donner d'assez bons résultats au problème posé sans pour autant nécessiter un traitement long des informations de départ, c'est ce qui lui confère à priori une efficacité à ne pas négliger.

La seconde étape est une méthode par exploration distribuée de l'ensemble des solutions réalisables, fondées sur un algorithme génétique, ce dernier est un outil méta heuristique puissant pour la recherche «pseudo aléatoire», mais qui nécessite un espace mémoire important pour le stockage d'informations générationnelles intéressantes. Plus on voudrait converger vers les meilleures solutions possibles, plus les paramètres de l'application devraient être augmentés (taille des populations et nombre d'itérations surtout), mais celle-ci mettrait un temps d'exécution croissant en fonction de cette modification, cette étape sera utilisé pour améliorer le

résultat de la premier étape.

Dans la dernière partie nous avons construis un programme qui résout notre problème, c'est claire que sans beaucoup de testes pratiques nous ne pouvons pas avoir la certitude que notre méthode soit le meilleur pour ce genre de problème, et pour savoir si c'est le cas et pour rendre ce point une vérité, il est évidant qu'il faut prendre en considération des cotés importants comme l'amélioration du programme du coté technique, la collection de plus des données de bons qualités, le développement et l'amélioration de la politique utilisé de la part de la société. Enfin nous constatons qu'il faut exploiter d'autres options et d'autre domaines pour mieux résoudre les différents problèmes de la société et ainsi l'aidé à se développe et grandis.

Bibliographie

- [1] : A. Berro. Optimisation multi-objectif et strategie d evolution en environnement dynamique. PhD thesis, 2001
- [2] : A. Coloni, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, Proceedings of ECAL'91, European Conference on Artificial Life, Elsevier Publishing, Amsterdam, 1991.
- [3] : A. Dipanda, cour "algorithmes génétiques", Université de Bourgogne, 2013.
- [4] : BENLAHRACHE, Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences, 2007.
- [5] : Berro A. «Optimisation multi objectif et stratégies d'évolution en environnement dynamique». Thèse de Doctorat - Université des sciences sociales Toulouse 1, 2001.
- [6] : Cherif Sadfi, 2002, Problèmes d'ordonnancement avec Minimisation des Encours. Thèse Ph.D, Institut National Polytechnique de Grenoble.
- [7] : C.H. PAPANITRIOU, K. STEIGLITZ, Combinatorial optimization – algorithms and complexity. Prentice Hall, 1982 ;
- [8] : Christophe Rapine et denis Trystram, 2002, Théorie de la Complexité, Notes de cours, ENSGI – INP Grenoble.
- [9] : Cerny, V., Thermodynamical approach to the traveling salesman problem : an efficient

simulation algorithm. J. of Optimization th. and applications. 1985

[10] : Clarisse Dhaenens, El-Ghazali Talbi, Optimisation multicritère : Approche Par métaheuristiques, Université de Lille 1, France, 2004.

[11] : Collette. Y ; Siarry P., Multiobjective Optimization, Springer, 2003.

[12] : Cour de python université de paris -UFR science de vivant.2021.

[13] : Cour informatique Appliquée iTi 3200, Conception et analysed ALGORITHMES. Bouma Mohamed

[14] : C.Roucairol "Un nouvel algorithme pour le problème d'affectation quadratique".RAIRO Recherche opérationnelle tome 13 ,1979.

[15] : Deb, K., Agrawal, S., Pratab, A., and Meyarivan. «A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization : NSGA II», 2000.

[16] : Deb, K., (2001) Multi-Objective Optimization using Evolutionary Algorithms, Chichester : John Wiley Sons

[17] : Drici.W " Cour Optimisation Combinatoire : La Complexité Algorithmique", université UMBB, 2019/2020.

[18] : Ehrgott. M Multicriteria Optimization, Springer, 2005.

[19] : F. Moyson, B. Manderick, « The collective behaviour of Ants : an Example of SelfOrganization in Massive Parallelism », Spring Symposium on Parallel Models of Intelligence, Stanford, Californie, 1988.

[20] : Gabriel Cormier, "Algorithmes Génétiques", Université de Moncton, 2008.

[21] : <https://datascientest.com/numpy>

[22] : [https://fr.wikihow.com/créer un modèle mathématique](https://fr.wikihow.com/créer_un_modèle_mathématique) .

- [23] : <https://www.gerad.ca/~alainh/Metaheuristiques.pdf>
- [24] : <https://www.manager-go.com/strategie-entreprise/swot.html>
- [25] : <https://www.math.u-bordeaux.fr/~pmeurdes/os1.pdf>
- [26] : <https://www.scipy.org/about.html>.
- [27] : <http://www.info.univ-angers.fr/pub/hao/papers/RIA.pdf>
- [28] : [http://www.lsis.org/master/ancien-site/documents/Support cours-113 introduction de l'optimisation.pdf](http://www.lsis.org/master/ancien-site/documents/Support%20cours-113%20introduction%20de%20l%27optimisation.pdf)
- [29] : Internet Programming with Python (IPwP), de Aaron Watters, Guido van Rossum et James C. Ahlstrom, MT Books, 1996
- [30] : Introduction à la programmation en langage python université paris-sud-MPI 2015-2016
- [31] : (Kirkpatrick, et al., 1983) Kirkpatrick, S., Gelatt, C. and Vecchi, M. P., Optimization by simulated annealing. Science. 1983, Vol. 220, 4598, pp. 671-680.
- [32] : Madani Bezoui, Cour "Optimisation multi-objectif", UMBB Boumerdes, Algérie, 2020.
- [33] : Marc Pirlot, Métaheuristiques pour l'optimisation combinatoire : un aperçu général, Chapitre 1, dans "Optimisation approchée en recherche opérationnelle", sous la direction de Jacques Teghem et Marc Pirlot, Hermes Science.
- [34] : Mémoire de magistère "APPLICATION D'UNE APPROCHE BIO-INSPIREE AU PROBLEME D'ORDONNANCEMENT DES INSTRUCTIONS" ,par GUERNA Abderrahim, M'sila 2012.
- [35] : Mémoire de master Optimisation de la planification des projets de l'entreprise nationale de Génie Civil et Bâtiment par Cherifi Cakib, Ould-hammouda Safa, U.S.T.H.B ,2009
- [36] : Mémoire de master informatique université Kemis mliana Algorithme de Branch and

Bound Appliqué au problème de sac à dos fait par L.imen et R.amina

[37] : Metropolis, N., Rosenbluth, M. and Teller, A., Teller, E., Equation of state calculation by fast computing machines. J. of Chemical Physics. 1953

[38] : M. Ehrgott et X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum, 2000, Vol. 22, pp.425- 460

[39] : Optimisation Combinatoire, M1 GADM, Dr. N. KHERICI, UBMA, 2020/2021

[40] : Pareto, V. (1897). The new theories of economics. Journal of Political Economy

[41] : Projet fin d'étude , Mémoire master "optimisation d'un algorithme irréversible de compression par l'utilisation des méta-heuristique" par ADAD Fatima , univesrsité Oum boughi 2012

[42] : Présentation du langage python .pdf

[43] : Recherche Tabou J. Ayas et M.A. Viau 16 novembre 2004.

[44] : Roudenko O. «Application des Algorithmes Evolutionnaires aux Problèmes d'optimisation Multi-Objectif avec Contraintes ». Thèse de Doctorat - École Polytechnique de paris,2004

[45] : R. Duvigneau, Introduction aux méthodes d'optimisation sans gradient pour l'optimisation et le contrôle en mécanique des fluides, Inria Sophia-Antipolis, 2006.

[46] : Selvaraj Ramkumar, Algorithmes Génétiques, Faculté des sciences-Université libre de Bruxelles, 26 Avril 2007.

[47] : Thème de magistère "Résolution des problèmes multiobjectif à Base colonies de Fourmi" par SAHA ADEL, university de Batna.

[48] : Thèse de doctorant de université Paris saclay : Modélisation et optimisation bi-objectif et multi-période avec anticipation d'une place de maarché de prospects-internet :Adequation

offre,demande par manal MAAMAR,7 Décembre 2015

[49] : Thèse de doctorat par ZIDANI Hafid "Représentation de solution en optimisation continue, multi-objectif et applications" 2013.

[50] : Thèse de doctorat Conception de méta-heuristiques d'optimisation pour la segmentation d'images. Application aux images biomédicales par Amir Nakib, université paris France 2007.

[51] : Thèse de doctorat "optimisation multi-objectif des problèmes combinatoire : application à la génération des horaires d'examens finaux" l'école de technologie supérieur université du QUÉBEC par Pascal Côté, Novembre 2004

[52] : T.C Koopmans . M.J Bekman Assigment problèmes and the location of economie activities.

[53] : Thèse d'ingénieur par CATHERINE Mancel "Modélisation et résolution de problème d'optimisation combinatoire issus d'application spatiale,juin 2004.

[54] : Thème de Magister en informatique "Optimisation Multi-objectifs par Programmation Génétique, par SAADI LEILA, juillet 2007

[55] : Thèse d'ingénieur " Optimisation multi-objectif discrète par progation de contraintes", par TRISTRAM Grabener et Alain Berro, 07/07/2008

[56] : Vira Chankong and Yacov Y Haimes. Optimization-based methods for multiobjective decision-making-an overview. Large Scale Systems In Information And Decision Technologies