

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

OPTIMISER LE COUT ET LE TEMPS D'UN PROJET DE RESSOURCES MULTI-TYPES EN
APPLIQUANT LA NSGA-II

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA
MAÎTRISE EN GESTION DE PROJET

PAR
AKBAR KHOSHAKHLAGHVARNOFADERANI

SEPTEMBRE 2022

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

SOMMAIRE

Avec l'expansion des projets de construction et l'augmentation des investissements dans ce secteur afin de réaliser du profit, ces dernières années, les chercheurs et experts, dans leurs recherches et études, examinent l'ordonnancement des projets dans le but d'optimiser le temps et le coût des projets avec un moindre risque. Par ailleurs, dans le cadre de la planification des projets, les activités comportent généralement deux limitations principales : 1- Priorité et ordre des travaux 2- Restrictions d'accès aux ressources. De nombreuses recherches ont été menées sur les problèmes d'ordonnancement des projets en tenant compte des contraintes de ressources (RCPSP). Toutefois, dans les projets du monde réel, nous discutons également des ressources multi-types. En fait, dans la réalité, lors de la planification de projet, nous rencontrons la MTRCPSP (planification de projet avec plusieurs types de ressources) en plus des RCPSP. En effet, pour réaliser une activité à partir d'une ressource spécifique, différents types peuvent être disponibles (par exemple, travailleur junior, travailleur senior, travailleur superviseur). L'objectif de ce mémoire est d'optimiser le temps et les coûts d'un projet en considérant plusieurs ressources. Il s'agit de créer un modèle mathématique à l'aide d'un algorithme génétique multiple (NSGA-II) et de l'implémenter dans un exemple de projet dans le logiciel MSP (*Microsoft Project*), car le logiciel MSP est l'un des logiciels les plus utiles pour préparer un calendrier de projet avec la possibilité de définir le projet et la structure de répartition du travail (WBS) et la capacité de définir, ainsi qu'allouer et niveler les ressources dans le projet. Pour ce faire, nous utilisons MSP (*Microsoft Project*) pour garantir la performance et la précision du modèle et de l'algorithme.

TABLE DES MATIERES

SOMMAIRE	II
TABLE DES MATIERES	III
LISTE DES TABLEAUX	VIII
LISTE DES ABRÉVIATIONS	IX
REMERCIEMENTS	X
CHAPITRE 1 INTRODUCTION	11
1.1 INTRODUCTION ET ÉNONCIATION DU PROBLÈME	11
1.2 NÉCESSITÉ ET IMPORTANCE DES RESSOURCES DE NIVELLEMENT DU PROJET	14
1.3 OBJECTIFS DE RECHERCHE:	16
1.4 QUESTIONS DE RECHERCHE :	17
1.5 STRUCTURE DE RECHERCHE :	18
CHAPITRE 2 : REVUE DE LA LITTERATURE	19
2.1 APERÇU DES PRINCIPAUX CONCEPTS D’UN PROJET :.....	19
2.2 PROBLEME D’OPTIMISATION TEMPS-COUT :	28
2.3 PROBLEME DE L’ORDONNANCEMENT DE L’ALLOCATION DES RESSOURCES :	37
2.4 ALGORITHMES ET METHODES POUR RESOUDRE LE PROBLEME D’ORDONNANCEMENT DE L’ALLOCATION DES RESSOURCES :.....	41
2.5 SOMMAIRE :	48
CHAPITRE 3 : ALGORITHME GENETIQUE ET NSGA-II	49
3.1 ALGORITHME GENETIQUE.....	49

3.2	CONCEPTS DE BASE DE L'ALGORITHME GENETIQUE :	51
3.3	LA SELECTION.....	53
3.4	LE CROISEMENT (RECOMBINAISON) :	54
3.5	LA MUTATION	55
3.6	GROUPES D'ELITE ET ELITISTES :	56
3.7	FACTEURS DE BASE AFFECTANT LES ALGORITHMES GENETIQUES :	56
3.8	LA FAÇON DONT LE CHROMOSOME EST AFFICHE :	56
3.9	POPULATION INITIALE :	57
3.10	STRATEGIE DE SELECTION DES PARENTS :	57
3.11	ALGORITHMES GENETIQUES EN OPTIMISATION MULTI-OBJECTIFS (NSGA-II)	63
CHAPITRE 4 : METHODOLOGIE DE RECHERCHE		77
4.1	UN MODELE MATHEMATIQUE	77
4.2	ALGORITHME GENETIQUE NSGA-II IMPLEMENTE.....	80
4.3	LIEN ENTRE MATLAB ET LES LOGICIELS DE PLANIFICATION (MS PROJECT/PRIMAVERA).....	89
4.4	RESUME.....	92
CHAPITRE 5 : RESULTAT		94
5.1	CONTEXTE	94
5.2	VALIDATION DE LA METHODE	98
5.3	VERIFICATION DE L'EXECUTION DU MODELE DANS MATLAB PAR MSP :	112
5.4	RESUME :	124

CHAPITRE 6 : CONCLUSION	126
6.1 SOMMAIRE :	126
6.2 CONTRIBUTIONS :	127
6.3 APPLICATIONS DE LA RECHERCHE :	129
6.4 LIMITES ET SUGGESTIONS POUR DES ETUDES FUTURES :	129
REFERENCES:	131
ANNEXES:	150

LISTE DES FIGURES

Figure 2.1 Groupe de processus de gestion de projet	21
Figure 2.2 Diagramme de la portée du projet	22
Figure 2.3 relations entre les activités.....	24
Figure 2.4 La différence entre les problèmes P, NP, NP-Complet et NP- Difficile	28
Figure 3.1 Classification du renseignement.....	49
Figure 3.2 Point de croisement	54
Figure 3.3 Mutation avant et après	55
Figure 3.4 Chromosome à croisement en un point.	61
Figure 3.5 Chromosome croisé en deux points.....	61
Figure 3.6 Sélection de la qualité et du coût de production comme problème d'optimisation multi-objectifs.....	65
Figure 3.7 Image d'optimisation de Pareto dans l'espace cible.....	66
Figure 3.8 : Le front de Pareto et le point idéal	67
Figure 3.9 Équilibre fort et faible dans un problème multi-objectifs.....	68
Figure 3.10 Classement des réponses dans différents algorithmes	69
Figure 3.11 Diagramme de flux de travail dans l'algorithme NSGA-II	72
Figure 3.12 Estimation de la compression des réponses.....	73
Figure 3.13 Processus de sélection et de recréation des membres de couche dans l'algorithme NSGA-II.....	76
Figure 4.1 Exemple de projet.....	81
Figure 4.2 Algorithme de génération de liste d'activités	82
Figure 4.3 Classification par numéro de catégorie	86
Figure 4.4 Concept de distance de congestion pour le point i.	88

Figure 4.5 Exemple de Front de Pareto obtenu par NSGA-II.....	90
Figure 4.6 Détails des points du front de Pareto	91
Figure 4.7 Tableau relatif à la priorité et au type de ressources requises pour les activités	91
Figure 4.8 Tableau des dates de début et de fin des activités	92
Figure 5.1 Exemple numérique de réseau (Zheng and Ng, 2005)	94
Figure 5.2 Front de Pareto de différentes nPop	104
Figure 5.3 Front de Pareto de différents taux de mutation.....	108
Figure 5.4 Front de Pareto de différents taux de croisement	111
Figure 5.5 Diagramme de Gantt du projet en supposant la sélection des ressources pour les activités avec le premier type de ressources avant de niveler les ressources.	113
Figure 5.6 Allocation des ressources du projet en supposant la sélection des ressources pour les activités avec le premier type de ressources avant de niveler les ressources.....	115
Figure 5.7 Diagramme de Gantt du projet avec la 7 ^e solution de croisement 0.4 du tableau 5.7.	117
Figure 5.8 Allocation des ressources du projet avec la 7 ^e solution de crossover 0.4.....	118
Figure 5.9 Diagramme de Gantt du projet avec exécution de la 5 ^e solution du croisement 0,6 ..	120
Figure 5.10 Allocation des ressources du projet avec l'exécution de la 5 ^e solution de croisement 0,6	121
Figure 5.11 Diagramme de Gantt du projet avec la 12 ^e solution en cours d'exécution du croisement 0,8	123
Figure 5.12 Allocation des ressources du projet avec la 12 ^e solution de croisement 0.8.....	124

LISTE DES TABLEAUX

Tableau 4.1 Le processus de production d'un exemple de liste d'activités	83
Tableau 5.1 : Exécution des activités (coût-durée) (Zheng and Ng, 2005).....	95
Tableau 5.2 : Exécution des activités (Ressources)	97
Tableau 5.3 : Analyse de sensibilité de la taille de la population	101
Tableau 5.4 Analyse du front de Pareto pour différentes tailles de pop.	104
Tableau 5.5 : Analyse de sensibilité au taux de mutation	105
Tableau 5.6 Analyse du front de Pareto pour différents taux de mutation.....	108
Tableau 5.7 : Analyse de sensibilité au taux de croisement.....	109
Tableau 5.8 Analyse du front de Pareto pour différents taux de croisement	112
Tableau 5.9 détail des activités du projet pour la 7e solution du croisement 0.4 du tableau 5.7 .	116
Tableau 5.10 détail des activités du projet pour la 5e solution du croisement 0.6 du tableau 5.7	119
Tableau 5.11 Détail des activités du projet pour la 12e solution du crossover 0.8	122

LISTE DES ABRÉVIATIONS

RCPSP - Resource-Constrained Project Scheduling Problem

NSGA-II - Non-Dominated Sorting Genetic Algorithm

MSP - Microsoft Project

WBS - Work Breakdown Structure

GA - Genetic Algorithm

MTRCPSP - Multi Type Resources Consider Project Scheduling Problem

NP Problems - Non-Polynomial Problems

TCO - Time and Cost Optimization

MAWA - Modified Adaptive-Weight Approach

LSM - Linear Scheduling Model

ACO - Ant Colony Optimization

NLP - Nonlinear Programming

CPM - Critical Path Method

TLBO – Teaching Learning Based Optimization

RLP - Resource Leveling Problem

CBC – Coin Branch & Cut

MFMC - Max Flow Min Cut

REMERCIEMENTS

Je suis très reconnaissant envers le professeur Alencar Bravo, car sans ses conseils, il aurait été très difficile de terminer ce mémoire. De plus, je lui suis extrêmement reconnaissant pour ses encouragements constants et les connaissances qu'il a partagées avec moi.

Je tiens à remercier l'équipe de la chaire, le professeur Christophe Bredille, Milena, Adrienne, Daniel Etsuo, Raphaëlle, ainsi que mes amis de l'UQTR Masoumeh, Niloofar et Farnaz qui ont grandement contribué à ce mémoire tout au long de son déroulement. Je tiens à remercier tous ceux qui ont participé à mon programme.

Mes parents ont tant sacrifié pour moi et ont fait partie intégrante de la réalisation de mes rêves. Je tiens à les remercier infiniment pour tout ce qu'ils ont fait pour moi.

Enfin, je souhaite remercier tout particulièrement ma conjointe, Mahya, qui m'a toujours accompagné sur ce chemin. Je lui dédie ce mémoire en remerciement pour son cœur plein d'amour et de sagesse qui m'a offert un environnement plein de santé, de sécurité, de paix et de confort.

CHAPITRE 1 INTRODUCTION

1.1 INTRODUCTION ET ÉNONCIATION DU PROBLÈME

L'un des principes les plus importants de la planification et de la gestion des projets de construction est la planification du projet avec une limitation contrainte des ressources. Une estimation précise du temps d'achèvement, du coût d'exécution et de la quantité de ressources consommées dans un projet s'avère très importante, en raison de l'incertitude dans la nature du contrôle du projet et de la planification du projet de construction qui se déroulent dans des environnements très variables. Ainsi, généralement, des retards dans l'achèvement d'un projet dans les délais augmentent les coûts d'exploitation. Par conséquent, le rôle de la prise de décision et de la gestion des ressources du projet est nettement visible.

L'un des problèmes rencontrés par les entrepreneurs et les maîtres d'ouvrage dans les projets de construction est l'absence de gestion des coûts et les mauvaises performances de la gestion de projet en matière de contrôle des coûts. En examinant les nombreux rapports qui ont été publiés sur les projets, on peut conclure que ce problème existe aussi bien dans les petits que dans les grands projets tels que la construction de centrales électriques ou de systèmes de transport, et les projets à coûts excédentaires. Raftery (Raftery 2003) a déclaré dans son étude que les projets de construction sont connus pour leur faiblesse en matière de système de gestion du temps et des coûts. Selon un rapport de la Banque mondiale sur la performance financière des projets qu'elle finance, la majorité des projets (1 120 sur 1 778) sont confrontés à des problèmes de gestion des coûts et leur budget dépasse le budget initial et alloué (World Bank, 2012). Kaming et al. (Kaming, Olomolaiye et al. 1997), après avoir étudié des projets de construction de grande hauteur en Indonésie, ont conclu que l'absence d'une gestion adéquate du temps et des coûts était fréquemment observée dans les projets, augmentant ainsi le temps et les coûts du projet.

Selon le PMBOK (PMBOK sixième édition 2017), l'une des exigences dans la section projet d'ordonnancement est de définir les ressources nécessaires à la réalisation

des activités, leur disponibilité, et de calculer le montant nécessaire à chacune d'entre elles pour réaliser les activités. Blazewicz (Błażewicz, Ecker et al. 2001) a divisé les ressources en deux catégories générales : 1) les ressources non consommables (renouvelables) et 2) les ressources consommables (non renouvelables). Les ressources non consommables sont des ressources disponibles utilisées d'une activité/projet à une autre activité/projet, telles que la main-d'œuvre et les machines. Les ressources sont des ressources qui ne peuvent pas être utilisées dans une autre activité/projet après avoir été utilisées dans une activité/projet, comme les matières premières ou le ciment et le sable.

Le problème d'ordonnement de projet sous contrainte de ressources (RCPS) consiste à altérer les horaires de début des activités d'un projet de manière à ce que l'utilisation des ressources varie le moins possible dans le temps tout en respectant les exigences de priorité. En raison de considérations telles que les périodes d'inactivité des ressources, la libération et la réembauche de la main-d'œuvre temporaire, le RCPS s'avère crucial, car l'utilisation quotidienne des ressources résultant de l'ordonnement des activités a une influence sur le coût d'un projet. Par conséquent, des projets plus nivelés aident à réduire ces dépenses.

Le RCPS implique de planifier les activités, de s'assurer qu'elles soient priorisées et de maximiser l'utilisation des ressources renouvelables. Partant du principe que le RCPS original ne couvre pas toutes les circonstances qui peuvent survenir dans la réalité, Hartmann et Briskorn (Hartmann et Briskorn 2010) décrivent des variantes du RCPS, telles que les objectifs, les ressources ou les activités. Le MTRCPS est une version améliorée du RCPS.

Les questions liées à la planification des projets tenant compte des contraintes de ressources (RCPS) sont parmi les questions les plus importantes qui ont été examinées par les chercheurs ces dernières années (Huang, Huang et al. 2015, Kaiafa et Chassiakos 2015, Tran, Cheng et al. 2016, Bhosale, Shastri et al. 2017, Christodoulou 2017, Giran, Temur et al. 2017, Kadri et Boctor 2018, Hartmann et Briskorn 2022, Lotfi, Yadegari et al. 2022) afin d'optimiser le temps du projet en prenant en considération des contraintes de

ressources (les contraintes de ressources signifient en fait l'accès aux ressources renouvelables telles que la main-d'œuvre et les machines).

En revanche, dans les projets réels, nous sommes confrontés à la problématique des ressources multi-types, et dans la recherche, l'ordonnement des projets doit être pensé en termes de ressources multi-types. De plus, dans les problématiques RCPSP, seuls l'ordonnement et l'optimisation du temps d'activité sont traités, alors que dans les projets de construction, l'un des enjeux les plus importants est la gestion des coûts et la réduction des coûts du projet. Par conséquent, il existe un besoin pour un modèle complémentaire afin d'optimiser le temps et le coût en considérant les ressources multi-types (MTRCPSP).

Dans ce mémoire, nous présentons des algorithmes pour résoudre le problème d'ordonnement de projet multi-types à ressources contraintes avec des décalages temporels minimum et maximum (MTRCPSP). Le but de MTRCPSP est de trouver une combinaison réalisable de types de ressources pour toutes les tâches, ainsi que de construire un calendrier réaliste basé sur les ressources et les restrictions de priorité, de sorte que le temps et les coûts du projet soient réduits au minimum. Les ressources ont des scénarios d'exécution multiples, mais finis (reflétant différentes manières de les exécuter) appelés types. Chaque scénario a éventuellement un impact différent sur la durée de l'activité, les coûts qui y sont associés et ses besoins en ressources. Une fois le type d'une ressource sélectionné pour une activité, l'activité doit être planifiée avec ce type pendant toute la durée correspondante et sans interruption.

Le problème d'ordonnement de projet à contraintes de ressources multiples (MTRCPSP) est défini par des activités de projet $i=1, \dots, n$ qui doivent être réalisées sans interruption et conformément à une hiérarchie de priorité.

Pour $(i,j) \in PR$, où PR est l'ensemble des restrictions de précédence, chaque tâche j ne doit être démarrée qu'après que son prédécesseur i a été terminé.

Chaque activité j est réalisée par un type $m \in M$ de ressources, qui établit le temps de traitement d_{jm} , la demande en ressource renouvelable $k \in R$.

À chaque temps « t », les ressources renouvelables sont renouvelées à leur pleine capacité « Lk ». « T » est la durée maximale de l'ensemble du projet.

Le problème de l'ordonnancement du projet est d'attribuer le temps de début ou « Si » aux activités du projet afin de trouver le moment de fin des activités le plus proche et enfin le moment de fin du projet. L'optimisation du temps et du coût du projet représente un problème NP- Difficile, car il n'y a pas d'algorithme spécifique pour résoudre ces problèmes et les chercheurs, dans leurs recherches et articles, cherchent à trouver des algorithmes heuristiques et méta-heuristiques pour résoudre ces problèmes (Lenstra et Rinnooy Kan 1978). Blazewicz et al (Blazewicz, Lenstra et al. 1983) ont prouvé dans un article de 1983 que le problème d'ordonnancement entre dans la catégorie des problèmes NP- Difficile en raison de contraintes de ressources. Ils ont prouvé qu'en général, l'ajout de contraintes de ressources à un problème d'ordonnancement de projet ajoute à sa complexité de calcul. Trouver une solution réaliste pour MTRCPSP constitue un problème NP- Difficile, selon Kolisch et Drexl (Kolisch & Drexl, 1997). Comme il s'agit d'un modèle étendu du modèle RCPSP, il n'existe pas de solution unique à ces problèmes. De plus, selon la revue de la littérature, d'autres auteurs ont déclaré que les problèmes d'optimisation du temps et des coûts ainsi que les problèmes RCPSP sont des problèmes NP- Difficile.

1.2 NÉCESSITÉ ET IMPORTANCE DES RESSOURCES DE NIVELLEMENT DU PROJET

En gestion de projet, la gestion de toute activité de projet est devenue un processus critique qui assure le succès de l'entreprise et la satisfaction du client. Les techniques de gestion traditionnelles contrôlent généralement les objectifs de projet à court terme et ne répondent pas aux critères de réussite de l'exécution du projet, alors que les techniques modernes de gestion de projet donnent aux entreprises la possibilité d'atteindre des objectifs à moyen et long terme de croissance continue et de compétitivité. Cependant, la gestion de projet n'a pas correctement pris en compte la gestion des ressources et l'a laissée

à la compétence des chefs de projet dans l'attribution et l'ordonnement des activités. Cette situation et la complexité des projets réels font qu'il s'avère difficile de définir la ligne de base du projet pour parvenir à une allocation optimale des ressources et l'ordonnement approprié des activités et pour estimer le budget approuvé. Dans l'industrie de la conception et du conseil en ingénierie, en raison de l'existence de différents départements impliqués dans le projet et du volume élevé de documents d'ingénierie, la gestion et l'affectation de superviseurs appropriés afin de créer le meilleur équilibre entre les coûts et les délais sont vitales. L'importance de cette question est tout à fait évidente dans les cas où le délai de livraison du contrat et de lourdes amendes pour livraison non ponctuelle dans chaque période ont été fixés.

La gestion des ressources est une composante interne de la gestion de projet. La gestion des ressources garantit que le projet est achevé dans les délais au même coût et avec la même qualité que ce qui a été prédéfini, ce qui se révèle essentiel même pour les entreprises orientées projet telles que les entrepreneurs. En fait, la question de l'ordonnement du projet est l'un des problèmes les plus importants auxquels sont confrontés ceux qui sont impliqués dans la planification, surtout lorsqu'ils sont tenus d'utiliser des ressources de consommation efficaces sans augmenter la durée du projet. Avec ces interprétations, parce que certaines ressources sont rares, les utiliser pour réaliser des activités conduira à des problèmes d'ordonnement du projet (Neumann, Schwindt, & Zimmermann, 2002). Dans la gestion des ressources, l'une des procédures les plus importantes peut être appelée nivellement des ressources, qui est responsable de la gestion de l'allocation des ressources du projet et résout les problèmes liés à l'allocation des ressources. Le chef de projet et les experts en planification au début du projet doivent planifier les ressources du projet en fonction du besoin et de l'accès à celles-ci. Cela évitera que le projet soit confronté à des problèmes d'allocation de ressources et celui-ci sera achevé à temps. Le nivellement des ressources constitue l'un des éléments principaux et clés de la gestion des ressources dans le projet. Les problèmes dans le projet surviennent lorsque les ressources ne sont pas correctement allouées (en raison d'une surallocation des

ressources) lors de la préparation du calendrier du projet et entraînent un risque financier dans le projet (Rieck, Zimmermann et al. 2012).

En général, il existe trois raisons principales expliquant le recours au nivellement des ressources d'un projet (Ponz-Tienda, Yepes et al. 2013) :

- 1- De nombreux changements dans le niveau des ressources telles que la main-d'œuvre, les problèmes et les coûts d'embauche et de licenciement, ainsi que le coût et le temps requis pour former de nouvelles personnes, les problèmes liés à l'efficacité à long terme des personnes et les causes dues à un mauvais dossier. On évite d'embaucher et d'employer des gens efficaces pour la collecte. De plus, le niveau des besoins en ressources étant régulier, le coût de location et de transport d'équipements lourds et coûteux est réduit.
- 2- En raison du niveau élevé de ressources, en raison de l'existence de périodes où ils ont besoin de beaucoup de ressources, l'efficacité de l'utilisation des ressources peut diminuer et les coûts peuvent augmenter.
- 3- Pour les ressources telles que les machines, l'existence d'un niveau fixe de besoin de ressources peut aider à expliquer les plans d'approvisionnement, de location, de transport et même d'entretien et de réparation des machines et équipements.

Par conséquent, lors de la préparation d'un projet d'ordonnancement, les ressources doivent être allouées correctement et en fonction de la quantité de ressources disponibles, et en cas de problèmes dus à une surallocation, nous devons niveler les ressources. Sinon, à l'avenir, nous serons confrontés au problème du manque de ressources et du non-respect du calendrier et augmenterons le temps et les coûts du projet.

1.3 OBJECTIFS DE RECHERCHE:

Dans les projets, le nivellement des ressources est très important et les chefs de projet ont toujours besoin d'un calendrier basé sur l'utilisation optimale des ressources nécessaires pour mener à bien leur projet. La plupart des recherches consacrées aux ressources ont été effectuées sur un seul type de ressource, alors que dans de nombreuses

organisations, les ressources sont multi-types. De plus, dans la plupart des modèles et des recherches, les deux paramètres de la ressource requise et la limitation de la source ne sont pas pris en compte dans le modèle, mais dans ce mémoire, les paramètres mentionnés sont inclus dans le modèle et le modèle est plus proche des projets du monde réel.

L'une des questions importantes dans les étapes de mise en œuvre de tout projet est la gestion correcte des facteurs impliqués et la nature de la gestion de projet nécessite plusieurs décisions telles que le nivellement des ressources du projet dans une variété de ressources ou la fourniture d'équipements et de main-d'œuvre nécessaires et appropriés pour le projet.

Nous avons l'intention d'étudier la modélisation du problème après avoir examiné le concept général de nivellement des ressources et indiqué les différentes définitions et solutions qui existent pour niveler les ressources dans les projets de construction, et les résultats du nivellement des ressources du projet pour améliorer la planification. Nous évaluerons également la productivité globale du projet. Nous introduirons ensuite un modèle mathématique dans le but de minimiser le temps et les coûts du projet, tout en tenant compte de l'utilisation limitée des ressources avec multi-types de ressources.

1.4 QUESTIONS DE RECHERCHE :

- A) Quelle est la nécessité du nivellement des ressources dans les projets ?
- B) L'utilisation de ce modèle permettra-t-elle d'optimiser le projet en termes de temps et de coûts ?
- C) Le nivellement des ressources du projet a-t-il réduit le risque de manquer de ressources au cours du projet ?

1.5 STRUCTURE DE RECHERCHE :

Le premier chapitre contient les généralités de la recherche et est conçu pour se familiariser avec la position du problème. Dans le chapitre 2, le contexte de la recherche et des études dans ce domaine est discuté. Le chapitre 3 comprend la définition de l'algorithme génétique et NSGA-II .

Le chapitre 4 présente le modèle mathématique et méthodologique et énonce les étapes de la méthode utilisée pour résoudre le problème. Dans le chapitre 5, les résultats des calculs effectués dans le chapitre 4 sont passés en revue et dans le chapitre 6, des solutions pour améliorer le problème ainsi que des suggestions pour de futures recherches sont présentées.

CHAPITRE 2 : REVUE DE LA LITTERATURE

2.1 APERÇU DES PRINCIPAUX CONCEPTS D'UN PROJET :

Dans cette section, nous avons l'intention d'expliquer les concepts liés à un projet dans différentes sections pour familiariser les lecteurs de ce mémoire. Il s'agit de concepts tels que la définition de projet, la planification et l'ordonnancement, la définition des ressources du projet, etc.

2.1.1 Projet :

Différentes définitions des projets ont été proposées jusqu'à présent, mais la meilleure définition est peut-être celle donnée dans le PMBOK (PMBOK sixième édition 2017) : un projet est un ensemble d'activités temporaires pour atteindre un objectif spécifique en temps opportun.

Un projet peut également être défini comme un ensemble d'activités commençant à un moment précis et basé sur des relations prérequis, et se terminant à un moment précis après la période d'exécution définie.

Les caractéristiques suivantes doivent être présentes dans cet ensemble d'activités :

- Chaque activité doit être priorisée, retardée ou exécutée en parallèle avec d'autres.
- Les paramètres d'une activité ne peuvent être définis que par une combinaison de trois facteurs : le coût, le temps et les ressources. Le paramètre temps doit être affecté à chaque activité.
- Il y a un début et une fin à l'ensemble des activités.
- Les fonctionnalités suivantes doivent être incluses dans tout projet :
 - Le projet est une œuvre nouvelle, originale et non répétitive.

- Le projet est fait pour produire un livrable (produit, service ou résultat) qui peut être livré une fois terminé.
- Les projets sont temporaires, ce qui signifie qu'ils ont un point de départ et un point final spécifiques, et que leur durée est limitée à un certain laps de temps.
- Un projet se termine lorsque ses objectifs sont atteints ou lorsqu'il est terminé.

2.1.2 Gestion de projet :

Selon le PMBOK (PMBOK sixième édition 2017), un chef de projet efficace applique les connaissances, compétences, outils et techniques nécessaires pour gérer le flux exécutif des activités. Afin d'atteindre le but ultime du projet, la gestion de projet consiste principalement à coordonner la mise en œuvre des activités pour la bonne utilisation des ressources et des installations. Diverses contraintes du projet doivent être équilibrées lors de la gestion du projet. Ces contraintes incluent : Risque, temps, coût, portée, qualité et ressources.

Selon le PMBOK (PMBOK sixième édition 2017), il existe 5 processus différents pour chaque projet. Ce sont des processus qui sont un ensemble d'activités pour atteindre un résultat spécifique. Ces processus sont exécutés sous la supervision de la direction du projet et par des exécutants du projet. Ces 5 processus comprennent : initial, planification, exécutif, contrôle et final. Le processus initial comprend l'élaboration de la charte du projet et l'identification des bénéficiaires du projet. Le processus de planification comprend l'élaboration du plan de gestion de projet, du plan de domaine de projet, de la création de la WBS, de la construction de la liste des activités, du calendrier du projet, du plan des coûts, du plan des risques et du plan des ressources du projet. Le processus de mise en œuvre comprend la gestion de la mise en œuvre des activités prévues dans la phase de planification, où la majeure partie du budget du projet est dépensée. En outre, dans cette phase du plan de travail, l'équipe de projet est implémentée. Dans la phase de contrôle, il comprend le suivi de l'avancement réel du projet selon le calendrier établi et la prise d'actions correctives en cas de retard dans le projet. Et enfin, le processus final, qui est le dernier processus du cycle de gestion de projet, est la clôture officielle du projet avec la

signature d'approbation ou de garantie par le client. La phase finale peut inclure des activités telles que : la livraison du projet, l'archivage du projet et les leçons apprises, la dissolution ou la libération officielle de l'équipe du projet. La figure 2.1 montre les groupes de processus de gestion de projet.

Figure 2.1 Groupe de processus de gestion de projet

Source (PMBOK sixième édition 2017)

Domaines de connaissance	Groupes de processus de management de projet				
	Groupe de processus d'initialisation	Groupe de processus de planification	Groupe de processus d'exécution	Groupe de processus de maîtrise	Groupe de processus de clôture
4. Gestion de l'intégration du projet	4.1 Élaborer la charte du projet	4.2 Élaborer le plan de management du projet	4.3 Diriger et gérer le travail du projet 4.4 Gérer les connaissances du projet	4.5 Maîtriser le projet 4.6 Maîtriser les changements	4.7 Clôre le projet ou la phase
5. Gestion du périmètre du projet		5.1 Planifier la gestion du périmètre et du contenu 5.2 Recueillir les exigences 5.3 Définir le périmètre 5.4 Créer le WBS		5.5 Valider le périmètre 5.6 Maîtriser le périmètre et le contenu	
6. Gestion de l'échéancier du projet		6.1 Planifier la gestion de l'échéancier 6.2 Définir les activités 6.3 Organiser les activités en séquence 6.4 Estimer la durée des activités 6.5 Élaborer l'échéancier		6.6 Maîtriser l'échéancier	
7. Gestion des coûts du projet		7.1 Planifier la gestion des coûts 7.2 Estimer les coûts 7.3 Déterminer le budget		7.4 Maîtriser les coûts	
8. Gestion de la qualité du projet		8.1 Planifier la gestion de la qualité	8.2 Gérer la qualité	8.3 Maîtriser la qualité	
9. Gestion des ressources du projet		9.1 Planifier la gestion des ressources 9.2 Estimer les ressources nécessaires aux activités	9.3 Obtenir les ressources 9.4 Développer l'équipe 9.5 Gérer l'équipe	9.6 Maîtriser les ressources	
10. Gestion des communications du projet		10.1 Planifier la gestion des communications	10.2 Gérer les communications	10.3 Maîtriser les communications	
11. Gestion des risques du projet		11.1 Planifier la gestion des risques 11.2 Identifier les risques 11.3 Mettre en œuvre l'analyse qualitative des risques 11.4 Mettre en œuvre l'analyse quantitative des risques 11.5 Planifier les réponses aux risques	11.6 Appliquer les réponses aux risques	11.7 Maîtriser les risques	
12. Gestion des approvisionnements du projet		12.1 Planifier la gestion des approvisionnements	12.2 Procéder aux approvisionnements	12.3 Maîtriser les approvisionnements	
13. Gestion des parties prenantes du projet	13.1 Identifier les parties prenantes	13.2 Planifier l'engagement des parties prenantes	13.3 Gérer l'engagement des parties prenantes	13.4 Maîtriser l'engagement des parties prenantes	

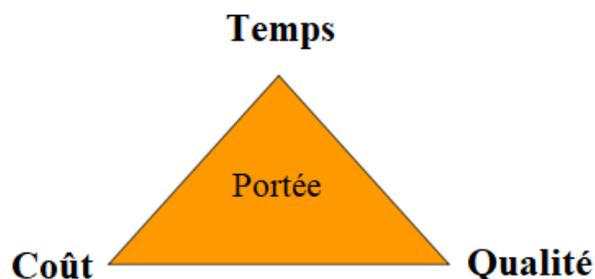
2.1.3 Planification et ordonnancement du projet :

Après avoir défini le projet et précisé sa portée, la phase de planification commence. En fait, la planification de projet fait référence à la préparation d'un ordonnancement exécutif en fonction des contraintes du projet. En fait, l'existence de ces contraintes est la principale raison de la planification dans le projet. Les principales et les plus importantes de ces contraintes, également appelées diagramme du champ d'application (figure 2.2), comprennent le coût, le temps et la qualité. Cette étape comprend des éléments importants tels que : créer un WBS, définir et identifier les activités du projet, déterminer la durée des activités, identifier les relations entre les activités, définir et allouer les ressources et identifier les contraintes de planification. Pour faciliter la planification du projet, le projet est divisé en parties gérables (WBS). WBS donne une vue d'ensemble du projet afin de diviser le projet en lots de travail. WBS est responsable de l'identification des activités spécifiques pour réaliser le projet. Une activité est le plus petit membre exécutif d'un projet. Pour qu'un projet soit mené à bien, il s'avère essentiel que toutes ses activités soient pleinement mises en œuvre.

Projet d'ordonnancement se définit comme la création d'une ligne de base pour le projet dans laquelle les activités sont hiérarchisées, et pour chaque activité, la date de début, la date de fin, la quantité de ressources nécessaires et le coût des travaux sont spécifiés (PMBOK sixième édition 2017).

Figure 2.2 Diagramme de la portée du projet

Source (PMBOK sixième édition 2017)



2.1.4 Types de relations entre deux activités :

Le projet comprend un certain nombre d'activités qui doivent être exécutées par les ressources du projet dans une période spécifique et dans une séquence spécifique. Il existe un ensemble de relations entre le début et la fin de ces activités (PMBOK sixième édition 2017).

Par exemple, vous ne pourrez pas exercer une activité de construction tant qu'un permis de fabrication n'aura pas été obtenu. Dans le jargon de la gestion de projet, l'activité « d'obtention d'une licence » est un prédécesseur de l'activité de « démarrage de la construction ». En fait, chaque activité peut être « prédécesseur » ou « successeur » d'une ou plusieurs autres activités du projet.

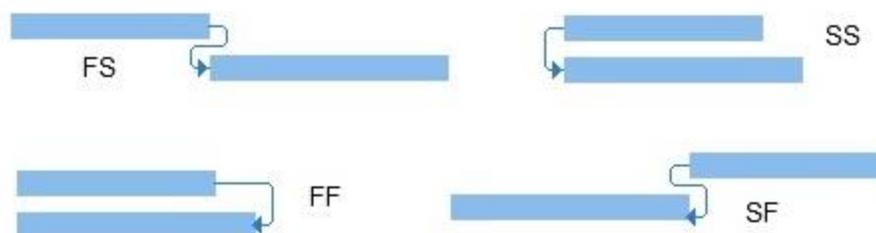
Les chefs de projet et l'équipe de planification doivent identifier toutes les relations et interdépendances entre les activités, y compris les relations « prédécesseur » et « successeur », lors de la préparation du calendrier du projet.

Le PMBOK (PMBOK sixième édition 2017) définit quatre types de relations entre les activités du projet :

- De la fin au début (FS) : cette relation signifie que l'activité « successeur » n'est pas autorisée à démarrer tant que l'activité « prédécesseur » n'est pas entièrement terminée. Cette relation est le type de relation le plus courant entre les activités d'un projet (PMBOK sixième édition 2017).
- Début au début (SS) : La relation suivante la plus couramment utilisée dans les projets est la relation SS, ce qui signifie que l'activité « successeur » ne peut pas commencer tant que l'activité « prédécesseur » n'a pas commencé. Dans cette relation, les activités n'ont pas besoin d'être terminées ou démarrées en même temps, mais la deuxième activité ne peut pas être démarrée tant que la première activité n'est pas démarrée. Par exemple, l'activité de coulage d'asphalte est une activité « prédécesseur » de « nivellement de la route ». L'activité successeur « Nivellement de la route » peut commencer en même temps que l'activité préalable « couler l'asphalte » (PMBOK sixième édition 2017).

- Fin à Fin (FF) : La troisième relation entre les activités du projet est FF, et cela signifie que lorsque nous pouvons dire que deux activités ou plus ont été achevées, ces activités seront achevées en même temps. Par exemple, l'activité « Poser les fils électriques » est un « prédécesseur » de l'activité « Inspecter les fils électriques » et elles doivent être réalisées en même temps (PMBOK sixième édition 2017).
- Du début à la fin (SF) : la dernière relation que nous rencontrons rarement dans les projets du monde réel est la relation SF, ce qui signifie que l'activité « successeur » se termine lorsque l'activité « prédécesseur » commence. En fait, la fin de l'activité « successeur » dépend du début de l'activité « prédécesseur ». Par exemple, si nous supposons que l'activité de « gardien de jour » est une activité prédécesseur de « gardien de nuit », l'activité « successeur » ne peut pas se terminer tant que l'activité de « prédécesseur » n'a pas commencé. En fait, le gardien de nuit ne prendra fin que lorsque le gardien de jour commencera à fonctionner (PMBOK sixième édition 2017).

Figure 2.3 relations entre les activités
Source (PMBOK sixième édition 2017)



2.1.5 Ressources :

Les ressources du projet comprennent tout ce qui est nécessaire à la mise en œuvre et à l'achèvement des activités du projet. Les ressources du projet sont essentiellement tout ce dont nous pouvons avoir besoin du début à la fin de celui-ci et au moment de la livraison

du projet, et si une ressource fait défaut, le projet sera confronté à des limites (PMBOK sixième édition 2017).

Les ressources du projet sont divisées en deux catégories générales : ressources renouvelables (non consommables) et non renouvelables (consommables) (PMBOK sixième édition 2017).

Les ressources renouvelables, qui comprennent la main-d'œuvre et les machines, sont calculées et allouées sur une base horaire et ont une utilisation limitée. Ces ressources peuvent être utilisées d'une activité à l'autre (de projet à projet) (PMBOK sixième édition 2017).

Les ressources non renouvelables telles que les matières premières, le ciment, le sable, etc., ont une unité de mesure (comme le KG) et sont consommées dans une activité et ne peuvent pas être utilisées dans une autre activité (Błażewicz, Ecker et al. 2001).

Dans la discussion sur la gestion des ressources et projets d'ordonnement, l'un des principaux sujets de l'équipe de gestion de projet est l'allocation des ressources, c'est-à-dire comment sont ordonnancées les activités et les ressources qui devraient compléter ces activités. La partie la plus importante de l'allocation des ressources est de savoir quel est le budget pour chaque ressource, puis de planifier les ressources en fonction du calendrier du projet (Błażewicz, Ecker et al. 2001).

En fait, une partie importante de la gestion de projet est l'allocation correcte des ressources pendant le calendrier du projet et doit être effectuée lors de la phase de planification du projet afin de réduire les coûts et d'améliorer la productivité du projet pour finalement conduire au succès du projet et susciter la satisfaction du client.

Lorsqu'une ressource se voit attribuer plus de travail qu'elle ne peut réellement en gérer dans un temps donné, on dit qu'elle est surallouée.

L'objectif du nivellement des ressources est de résoudre les conflits de ressources et les surallocations afin de s'assurer que les projets sont réalisés avec les ressources disponibles.

2.1.6 Coûts du projet :

La catégorisation des coûts et des dépenses d'un projet est un outil qui aide le chef de projet et l'équipe de projet à comprendre le type de coûts qu'ils encourront pendant le cycle de vie d'un projet.

L'équipe de projet doit catégoriser et répertorier leurs coûts directement et indirectement lors de la budgétisation du projet et de la détermination des coûts du projet.

Fondamentalement, cette division des coûts directs et indirects est effectuée à des fins de budgétisation et de comptabilité, et ce n'est pas une tâche facile, car les coûts directs et indirects sont différents selon le type d'activité et de produit (Zandhuis et Wuttke 2019).

- Coûts directs du projet : Tous les coûts liés à l'utilisation de la main-d'œuvre directe du projet (comme les travailleurs), des matériaux, des machines et des équipements, et en général tous les coûts qui peuvent être spécifiquement comptabilisés dans l'exécution d'une activité, sont appelés coûts directs du projet. Ces coûts sont également appelés coûts des ressources. Pour réduire la durée du projet, les coûts directs du projet augmentent (Zandhuis et Wuttke 2019).
- Coûts indirects du projet : Tous les coûts qui ne peuvent être directement et explicitement liés à la mise en œuvre d'une activité sont considérés comme des coûts indirects du projet. Les éléments les plus importants de ces coûts comprennent l'équipe de projet (comme le chef de projet, les spécialistes de la planification et du contrôle de projet, etc.), les frais administratifs, les intérêts d'emprunt ainsi que les pénalités de retard pour l'achèvement du projet (Zandhuis et Wuttke 2019).

2.1.7 Problèmes NP et NP-difficiles :

Les problèmes NP non polynomiaux sont des problèmes difficiles qui ne peuvent être résolus par les méthodes traditionnelles. Par ailleurs, de nombreuses étapes doivent être franchies pour identifier des algorithmes rapides ou polynomiaux, et d'autre part,

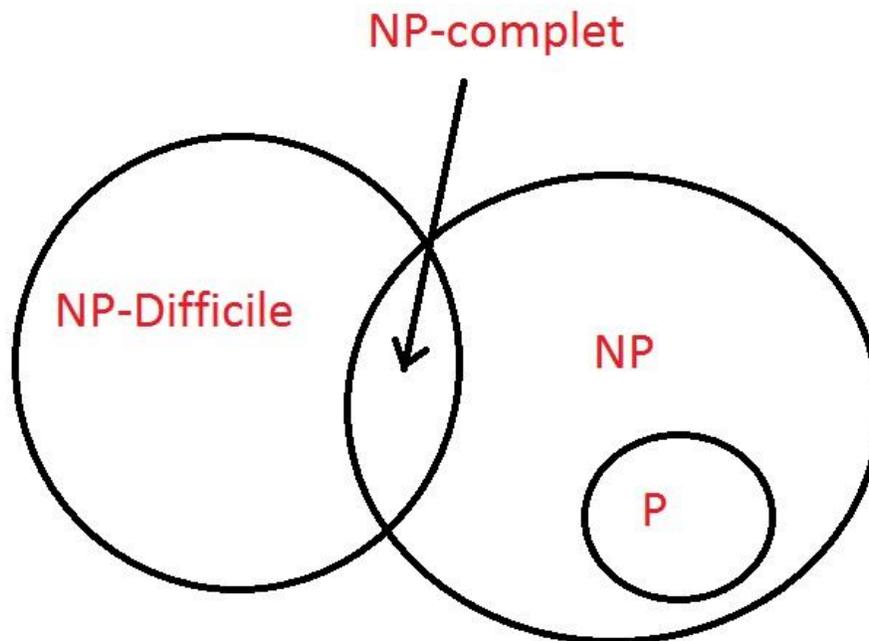
certains problèmes ne peuvent pas être résolus de manière algorithmique. Dans certains problèmes polynomiaux, il n'est pas possible de résoudre simultanément deux problèmes polynomiaux ou plus. Bien sûr, quand vous n'avez pas de réponse, il est très difficile de la vérifier, mais si vous avez la réponse, c'est beaucoup plus facile de vérifier la réponse. Ce sont les problèmes NP-complets (Plaisted 1984, Guo, Chang et al. 2005, Lodewijks 2019).

Le problème NP-complet concerne les problèmes de décision. NP représente un polynôme indéfini, ce qui signifie que la réponse peut être devinée par les algorithmes NP à la fois, puis examinée. Si nous avons une machine à deviner, alors nous pouvons trouver la réponse à un moment acceptable. Afin d'étudier les questions NP-complètes, nous devons considérer que la question elle-même est simple, car la réponse peut être oui ou non. En raison du résultat complexe des questions, une série de questions est appelée NP-difficile (Plaisted 1984, Guo, Chang et al. 2005, Lodewijks 2019).

Le mode NP-difficile est destiné aux problèmes d'optimisation. Ce groupe de problèmes n'est pas aussi limité que les questions np-complètes. Les questions NP face à ces questions peuvent facilement trouver un ou plusieurs algorithmes appropriés pour résoudre ce problème et leur seul travail est de rechercher toutes les réponses possibles. Ces problèmes comprennent plus d'un millier de problèmes qui s'appliquent dans la vie réelle, par exemple trouver le temps le plus court pour réaliser un projet ou ordonnancer un mariage. Mais du fait de la difficulté à résoudre de tels problèmes, on utilise des méthodes inexactes, c'est-à-dire des algorithmes approchés ou des algorithmes heuristiques et méta-heuristiques. Bien que ces méthodes ne résolvent pas exactement les problèmes NP-Difficile, elles sont plus avantageuses que les méthodes exactes, car elles produisent des réponses presque exactes pour nous dans les plus brefs délais (Plaisted 1984, Guo, Chang et al. 2005, Lodewijks 2019).

Figure 2.4 La différence entre les problèmes P, NP, NP-Complet et NP- Difficile

Source (Sparrow 2022)



2.2 PROBLEME D'OPTIMISATION TEMPS-COUT :

Les deux principaux facteurs à prendre en compte dans les projets, en particulier les projets de construction, sont l'optimisation du temps et du coût du projet. L'optimisation du temps et du coût du projet est quelque chose que les entrepreneurs et les clients essaient de faire. Dans le passé, les études ont été réalisées d'une telle manière que le temps de projet était supposé constant et l'optimisation a été faite sur l'autre objectif restant, à savoir le coût (Zhang et Thomas Ng 2012).

En 2004, Zheng et ses collègues (Zheng, Ng et al. 2004) ont proposé un modèle d'optimisation du temps et des coûts de projet (TCO) à objectifs multiples basé sur des algorithmes génétiques (GA). Dans cette étude, ils ont démontré le concept de leur modèle à travers une simple simulation manuelle. Dans ce modèle, ils ont intégré les poids

d'appariement dérivés et induit la pression de recherche à un point idéal. En fait, leur modèle était un modèle de poids adaptatif et modifié (MAWA), revendiquant le temps et le coût comme cible unique pour la simulation. Chassiakos et Sakellaropoulos (Chassiakos et Sakellaropoulos 2005) ont développé deux méthodes, linéaire (LP) et linéaire/entier (LP/IP), pour résoudre le problème d'échange de temps et de coût de projet. Le but de cette recherche est de sélectionner parmi les options de réalisation d'une activité, celles qui minimisent le coût du projet dans différents les dates limites (en fait, le but est d'optimiser et d'améliorer le cycle temps-coût du projet). Ils ont présenté deux méthodes détaillées et une méthode approximative. En ce qui concerne la première méthode, des substitutions coût-temps ont été effectuées pour chaque activité dans le but de minimiser le coût du projet, tandis que la seconde méthode examine la réduction de la durée du projet et sélectionne les activités qui raccourcissent le temps du projet.

En 1997, Li et Love (Li et Love 1997) ont déclaré que le problème d'optimisation coût-temps du projet constituait un problème NP difficile, car il n'existe pas de solution unique à ce problème. Ils ont fait valoir que si seule la question du raccourcissement du temps du projet était soulevée, une option pour résoudre le problème était de compresser le temps des activités du chemin critique, mais lorsqu'il s'agissait d'optimiser le temps et le coût du projet, il fallait utiliser des méthodes méta-heuristiques. Selon eux, les algorithmes génétiques (GA) représentent l'une des méthodes les plus appropriées pour résoudre les problèmes combinés temps-coût du projet. Dans leur étude, ils ont présenté des GA améliorés pour résoudre les problèmes de temps-coût du projet.

Le problème de l'optimisation temps-coût du projet est une tâche difficile et stimulante, car le temps et le coût des activités liées à un projet ne sont pas certains et ces problèmes doivent être pris en compte lors de l'optimisation du calendrier du projet (Ezeldin et Soliman 2009). Rogalska, Bozejko et leurs collègues (Rogalska, Bozejko et al. 2008) ont écrit un article sur la planification et l'ordonnancement de projets de construction en 2008 et ont développé le modèle d'ordonnancement linéaire (LSM), les diagrammes d'équilibre (LOB) et d'ordonnancement de réseau PERT. Ils ont considéré des méthodes multi-objectifs incluant le temps de projet, le coût et les ressources limitées dans des

méthodes définies et approximatives. Dans leur article, ils ont analysé la relation temps/coût du projet et optimisé leur modèle à l'aide de la méthode d'appariement temporel TCMIII et de l'algorithme évolutif combiné HEA.

Dans leur étude, Liu et Rahbar (Liu et Rahbar 2004) ont proposé une approche basée sur la théorie MFMC (*max flow-min cut*) pour optimiser le temps et le coût du projet. Dans cet article, ils ont utilisé la procédure opérationnelle pour appliquer un débit maximum et un cisaillement minimum. Dans cette méthode, le MFMC (*max flow-min cut*) résout d'abord le coût minimum du projet, et le cycle MFMC est répété jusqu'à ce que le temps du projet soit optimisé et compressé.

Le Construction Industry Institute (CII) définit l'ordonnancement comme une réduction des coûts quels qu'ils soient, et la compression du temps comme une réduction des coûts avec les coûts comme les deux principaux éléments d'ordonnancement. L'institut affirme que trouver un moyen peu coûteux de raccourcir les activités (réaliser des activités dans les plus brefs délais et au moindre coût) constitue une tâche très complexe dans la pratique, étant donné que dans le monde réel, les projets impliquent un grand nombre d'activités, dont chacune a des coûts différents pour compresser le temps d'exécution, et aussi les relations de dépendance entre les activités d'un projet sont très élevées et il est difficile de maintenir ces relations de travail (Liu 1998).

Dans leur étude de 2005, Zheng et al (Zheng, Ng et al. 2005) ont développé un modèle multi-objectifs basé sur un algorithme génétique pour optimiser le temps et le coût. Ils ont plus précisément essayé d'équilibrer la priorité de chaque objectif en intégrant un poids adaptatif dans leur modèle en fonction des performances de la génération précédente. Dans leur modèle, ils ont également utilisé la technique du classement de Pareto comme critère de sélection pour améliorer la diversité. Ils ont implémenté un petit projet (l'exemple de projet utilisé dans ce mémoire) dans le logiciel Microsoft Project pour s'assurer que leur modèle était correct (Zheng, Ng et al. 2005).

En 2008, Ng et Zhang (Ng et Zhang 2008) ont proposé un modèle d'optimisation des colonies de fourmis (ACS) pour résoudre le problème d'optimisation temps-coût

(TCO) qui optimise les deux cibles en même temps. Leur modèle tente de résoudre le problème du TCO en utilisant la technique MAWA en combinant les deux fonctions objectives de temps et de coût comme objectif général. Ils ont mis en œuvre un projet prototype qui a été principalement utilisé pour évaluer les performances des algorithmes génétiques dans la résolution des problèmes de temps et de coûts du projet afin d'évaluer leur modèle et leur algorithme.

En 2002, Que Bryan Christopher (Que 2002) a proposé une approche basée sur des algorithmes génétiques pour résoudre le problème du temps et du coût des projets, et pour rapprocher son modèle des projets du monde réel, il a essayé de prédire la date d'achèvement des activités à partir de paramètres comme les relations entre les activités, la latence, l'utilisation de contraintes, les améliorations et les ressources, et également essayé de tirer parti de la capacité de planifier en prédisant les dates d'achèvement en intégrant le système de gestion de projet avec l'algorithme génétique.

Al Haj et El-Sayegh ont déclaré dans leur article (Al Haj et El-Sayegh 2015) que, dans les projets de construction, l'objectif principal des chefs de projet est de terminer le projet dans les délais requis et avec le budget fixé. En fait, il y a un compromis entre le temps et les coûts du projet. Perdre le temps de fin du projet s'avère certainement coûteux pour le projet, et d'autre part, gagner du temps et faire le projet dans un délai court crée certainement des avantages pour le projet, mais ce gain de temps ne devrait pas entraîner de coûts pour le projet, donc l'optimisation du projet, du temps et des coûts des projets de construction se révèle critique. En fait, l'objectif principal de l'optimisation du temps et des coûts d'un projet est de déterminer la durée optimale d'un projet en fonction du coût minimum. Ils pensaient que pour ce faire, la durée totale du projet devait être raccourcie en raccourcissant le temps des activités critiques. L'inconvénient était que cela réduisait le flottement total des activités non critiques et perdait la flexibilité du calendrier. Par conséquent, dans leur étude, en utilisant la méthode la plus optimale, ils essaient de trouver une solution efficace tout en maintenant la flexibilité du projet d'ordonnancement (Al Haj et El-Sayegh 2015).

Zhang et Thomas (Zhang et Thomas Ng 2012) ont développé en 2012 un nouveau modèle du problème d'optimisation du temps et des coûts du projet (TCO) basé sur les techniques de l'algorithme de clonage optimal des fourmis (ACO) pour réduire le temps et le coût du projet et simultanément pour des projets de construction. Pour résoudre le problème du modèle multi-objectifs, dans cette étude, ils ont utilisé une approche de poids adaptatif modifié (MAWA) pour créer une fonction objectif générale et unifiée pour combiner les objectifs du modèle (réduire le temps et réduire le coût du projet) aux valeurs du poids cible. Après avoir implémenté le modèle, ils ont développé un modèle d'optimisation du temps et des coûts basés sur l'algorithme ACO utilisant le codage VBA dans le logiciel Microsoft Project en créant trois modules comprenant l'entrée des données, l'optimisation et la sortie.

En 2009, Ezeldin et Soliman (Ezeldin et Soliman 2009) ont proposé une technique combinée pour résoudre le problème de la réduction du temps de construction et du coût des projets de construction en intégrant des algorithmes génétiques à la programmation dynamique. Ceux-ci sont développés par un codage informatique en deux étapes qui comprend : 1) l'utilisation d'un système de codage dans l'algorithme génétique pour trouver les solutions les plus optimales et les plus proches disponibles 2) pour rechercher la proximité de proche de l'optimal tel qu'utilisé dans l'étape précédente trouvée par GA et sa convergence avec l'optimal global en utilisant la programmation dynamique. Ils ont testé la fiabilité de leur modèle à travers un petit projet de 14 activités et la mise en œuvre du modèle.

Kalhor, Khanzadi et al (Kalhor, Khanzadi et al. 2011) ont également eu recours à une approche de colonie de fourmis non dominantes en 2011 pour résoudre le problème d'optimisation du temps et des coûts du projet. Pour ce faire, ils ont développé un modèle qui recherche des solutions de temporisation en considérant le temps et les coûts comme deux objectifs. Ils ont fait valoir que pour que le modèle TCO soit plus proche des projets du monde réel, l'incertitude dans le temps et les coûts du projet devrait être considérée comme un principe. Dans ce modèle, ils ont utilisé l'approche α -cut pour déterminer le

niveau de risque accepté par le chef de projet, ainsi que les méthodes de domination gauche et droite pour trouver des solutions non dominantes.

Zheng et Thomas (Zheng et Ng 2005) ont créé un nouveau modèle pratique en 2005 pour améliorer leur modèle, qu'ils avaient développé en 2004, afin de combler les lacunes du modèle précédent. Ils ont tenté d'améliorer leur approche précédente en remplaçant le front de Pareto par l'introduction du concept de coût.

Hegazy et Wassef (Hegazy et Wassef 2001) ont également introduit un modèle pratique basé sur un algorithme génétique en 2001 pour résoudre le problème d'optimisation du temps et des coûts. Le but de leur modèle était d'optimiser le coût total de la construction, qui comprend les coûts directs et indirects du projet.

En 1993, Moselhi et El-Rayes (Moselhi et El-Rayes 1993) ont proposé un modèle de planification de projet visant à optimiser les coûts du projet. L'objectif principal de leur modèle est d'introduire le coût comme variable de décision clé dans le processus d'optimisation. Ils ont résolu leur modèle en deux étapes grâce à la programmation dynamique. Ces deux étapes consistent à : d'abord identifier les conditions locales minimales avec un processus en avant, puis assurer les conditions générales minimales avec un processus en avant. Dans cet article, ils ont utilisé un exemple de projet pour démontrer la validité de leur modèle.

L'optimisation du temps et des coûts du projet dans les projets de construction est une question d'optimisation combinée et figure parmi les aspects les plus importants de la planification (Xiong et Kuang 2008). Dans leur article, présenté en 2008 par Xiong et Kuang, ils ont utilisé l'ACO (algorithme de colonie de fourmis) pour optimiser le temps et les coûts du projet afin de résoudre le modèle TCO. En combinant l'approche MAWA (Approche de pondération adaptative modifiée), ils ont pu identifier les solutions optimales avec l'algorithme ACO et définir le front de Pareto.

Pour optimiser le temps et les coûts du projet, Klanšek et Pšunder (Klanšek et Pšunder 2010) ont mis en œuvre un modèle de planification non linéaire étendu (PNL) sous les relations prioritaires entre les activités. Ils ont déclaré que les modèles NLP existants

se concentrent davantage sur des solutions de coûts de projet optimales, tandis que dans leur modèle NLP, ils cherchent à optimiser le temps et les coûts du projet dans la gestion de projet. Leur méthode a été généralisée à la méthode de la nuit réduite dans le modèle d'optimisation NLP, et leurs résultats dans cet article sont le coût total minimum du projet et le point temps-coût du projet. Ils ont déclaré qu'avec ce modèle, l'entrepreneur du projet pouvait estimer l'impact de l'échéance du projet sur l'augmentation/la diminution des coûts du projet.

L'un des aspects les plus importants de la planification et du contrôle de projet dans les projets de construction est la relation (compromis) entre le coût et le temps du projet. En effet, ces deux éléments jouent un rôle primordial dans l'exécution et la réalisation des activités du projet, et l'optimisation de l'un devient l'affaiblissement de l'autre. Par conséquent, il s'avère très difficile d'optimiser et d'améliorer le coût et le temps du projet en même temps. Une façon de réduire les coûts du projet en utilisant la méthode du chemin critique (CPM) dans les petits projets consiste à utiliser des ressources bon marché dans des activités non critiques (ce qui, bien sûr, peut réduire la qualité des activités) (Feng, Liu et al. 1997). Feng, Liu et al. ont fourni un modèle pour rechercher et identifier des solutions optimales dans de grands projets basés sur des algorithmes génétiques et fournir un programme informatique pour une mise en œuvre plus efficace de l'algorithme (Feng, Liu et al. 1997).

L'une des entreprises les plus prospères au monde peut être introduite dans la construction de bâtiments, en particulier d'immeubles de grande hauteur, ce qui a entraîné de nombreux investissements dans ce secteur. Pour cette raison, l'une des principales tâches des chercheurs est de travailler sur des sujets tels que la manière de mener à bien des projets en toute sécurité et avec la plus grande rentabilité. L'un de ces enjeux est l'optimisation du temps et des coûts du projet.

Abdelkhalek, Refaie (Abdelkhalek, Refaie et al. 2020) et al. ont montré dans leur étude comment optimiser le temps et les coûts du projet en donnant des valeurs optimales de taux d'apprentissage à la main-d'œuvre qui réalise les activités du projet et en calculant correctement leur nombre pour réaliser les activités. Leur enquête a montré l'importance

de la recherche sur l'optimisation du temps et des coûts des projets dans les organisations, à laquelle les gestionnaires de projets doivent faire attention. Ils ont prévu 10 limitations afin d'améliorer le fonctionnement de leur modèle. Ces 10 limitations sont les suivantes :

- 1- Il est nécessaire pour chaque organisation de développer sa propre courbe d'apprentissage
- 2- Un changement dans les procédures, les travailleurs ou même dans la conception peut affecter un peu la courbe d'apprentissage
- 3- Certains facteurs peuvent modifier la courbe d'apprentissage, notamment les changements dans les procédures, la culture du lieu de travail et la disponibilité des ressources
- 4- Il n'est peut-être pas évident que les week-ends affectent la continuité du travail et donc pas le taux d'apprentissage des ouvriers
- 5- Les longs congés qui durent deux jours ou plus peuvent affecter le taux d'apprentissage des ouvriers alors que la continuité du travail devrait rester ininterrompue
- 6- Les situations météorologiques terribles pendant deux jours ou plus peuvent affecter négativement la continuité du travail
- 7- La répartition des taux d'apprentissage varie de 51 % (apprenant très rapide) à 100 % (aucun apprenant)
- 8- Selon le modèle proposé, la surface adulte moyenne pour les hommes est de 1,9 m², la surface adulte moyenne pour les femmes est de 1,6 m² et la surface adulte moyenne pour un travailleur est fixée à 2m²
- 9- Sauf si elles sont considérées comme des activités de base, les activités non répétitives sont exclues du modèle proposé
- 10- Le modèle proposé suppose que le même nombre d'étages est pertinent pour toutes les activités répétitives.

Pour accélérer et réduire l'ordonnancement du projet, il est nécessaire d'augmenter le coût total du projet. L'un des problèmes d'échange de temps et de coût du projet est d'identifier les coûts et les délais optimaux du projet pour augmenter les

bénéfices du projet (Toğan et Eirgash 2019). Dans leur article, Toğan et Eirgash ont proposé un modèle multi-objectifs basé sur le système TLBO (Optimisation basée sur l'enseignement et l'apprentissage) avec MAWA (Approche de pondération adaptative modifiée). Ils ont examiné quatre exemples de projets avec 7 à 63 activités (le projet à 7 activités est également utilisé dans ce mémoire) pour assurer la performance de leur modèle (TLBO-MAWA) (Toğan et Eirgash 2019).

Dans son projet, Shahriari (Shahriari 2016) fournit un modèle mathématique à deux objectifs pour optimiser et équilibrer la réduction du temps et des coûts du projet afin que les chefs de projet puissent prendre des décisions plus faciles concernant leur projet. L'une des hypothèses qu'il a émises pour rapprocher son modèle des projets du monde réel est la valeur temporelle de l'argent. Parce que l'un des facteurs qui ont un grand impact sur les décisions des managers en matière de compression et de raccourcissement du temps de projet est la valeur temporelle de l'argent, il a utilisé l'algorithme NSGA-II pour résoudre le problème d'optimisation du temps et des coûts dans son article.

En 2019, Podolski et Sroka (Podolski et Sroka 2019) ont mené une étude sur la rentabilité dans les projets de construction de tours à plusieurs logements. Ils ont essayé de fournir un modèle qui aide les gestionnaires de projet en matière de coûts dans la mise en œuvre de projets de construction tels que des bâtiments résidentiels, commerciaux, industriels. Ils ont défini la date limite du projet comme une contrainte dans leur modèle et ont utilisé l'algorithme de recuit pour résoudre ce modèle. Dans ce modèle, une seule fonction objective est définie, qui optimise le coût total du projet (y compris les coûts directs et indirects). Ils ont examiné l'efficacité de leur modèle, en utilisant un projet à 9 activités, et la mise en œuvre du modèle dans ce projet.

Au cours des vingt dernières années, l'un des sujets fréquemment étudiés et analysés est la question de l'optimisation du temps et des coûts des projets (Li, Xu et al. 2018).

En 2018, Li, Xu et al. (Li, Xu et al. 2018) ont déclaré que les modèles présentés dans les projets avec un nombre élevé d'activités (plus de 500 activités) ne disposaient pas

de performances de calcul précises et ont essayé de fournir un modèle qui comblerait cette lacune. Ils ont développé deux algorithmes heuristiques multi-objectifs pour optimiser le temps et les coûts. Leur premier algorithme est basé sur l'algorithme NSGA-II et un système de croisement conçu sur le chemin critique du projet, et le second algorithme a identifié des solutions efficaces en utilisant un algorithme heuristique descendant avec résolution répétée du modèle.

2.3 PROBLEME DE L'ORDONNANCEMENT DE L'ALLOCATION DES RESSOURCES :

Dans la gestion de projet basée sur le PMBOK (Zandhuis et Wuttke 2019), l'une des principales parties de la planification et de l'ordonnement de projet consiste à spécifier et à prévoir les dates des activités, à déterminer les risques, à gérer et à allouer les ressources du projet (en particulier les ressources limitées ou renouvelables). Comme nous le savons, dans la plupart des projets, l'équipe de gestion de projet (chefs de projet et spécialistes en planification et contrôle de projet) est confrontée au problème du manque de ressources. Dans ces cas précis, il y a un problème d'ordonnement du projet avec des ressources limitées (RCPS) (Kolisch et Hartmann 2006). Ashtiani, Leus et al. (Ashtiani, Leus et al. 2011) ont introduit une nouvelle classe de politiques de préprocesseur pour résoudre le problème SRCPS (Ordonnement basé sur les ressources et ordonnancement de projet). Dans cette méthode, lors de l'étape de prétraitement, des décisions sont prises concernant un certain nombre de décisions, tandis que les autres décisions restantes sont prises en observant les lignes de maillage qui sont basées sur les ressources pendant la mise en œuvre du projet. Ils ont utilisé une méthode de recherche en deux étapes pour générer ces politiques. La première étape de cette méthode consiste à trouver une liste de priorités appropriées. L'étape suivante consiste à créer une liste de choix consécutifs.

Le problème de nivellement des ressources (RLP) se produit lorsque nous avons une quantité limitée de ressources disponibles, une date d'échéance pour le projet et que

nous devons effectuer l'ordonnancement l'exécution de la tâche tout en minimisant les fluctuations de la quantité de ressources disponibles. Outre l'article de Burgess et Killebrew de 1962 et l'article de De Witte de 1964, les premières contributions dans ce domaine datent respectivement de 1962 et 1964.

Deux types de ressources sont considérés : les ressources renouvelables et les ressources non renouvelables. La présence de ressources doublement limitées et partiellement renouvelables est un autre type de ressource qui peut être considéré. Les machines, les opérateurs de machines et les équipements de transport sont des exemples de ressources renouvelables dont la disponibilité par période est limitée. En revanche, les ressources non renouvelables, comme l'argent du budget, sont accessibles en quantités finies pendant la durée de vie du projet (Słowinski 1981).

Selon la version traditionnelle du RCPSP, la planification consiste à attribuer des tâches ($j = 1, \dots, J$) par des liens de priorité zéro-décalage, du début à la fin, sans préemption. Les ressources sont renouvelables et limitées à tout moment. La durée et la consommation de ressources de chaque activité sont définies par la phase d'exécution (Blazewicz et al., 1983).

Blazewicz et al. (Blazewicz, Lenstra et al. 1983) ont déclaré que l'objectif du RCPSP est de réduire la durée du projet tout en respectant les priorités et les contraintes de ressources (Blazewicz, Lenstra et al. 1983). Il a été démontré que ce problème est NP-difficile.

Il est supposé dans le RCPSP standard qu'aucun temps de transfert n'est requis lorsque les ressources sont transférées entre différents emplacements pour exécuter différentes activités. En réalité, cependant, le transfert de ressources entre différents sites peut prendre beaucoup de temps. De l'équipement lourd, par exemple, peut s'avérer nécessaire sur de nombreux chantiers, et son transfert de l'un à l'autre peut prendre plusieurs heures.

Drexl et Gruenewald, en 1993 (Drexl et Gruenewald 1993), ont proposé des méthodes pour les formuler et les résoudre dans une classe générale afin de résoudre les

problèmes d'ordonnancement de projets sous contrainte de ressources (RCPSP) dans lesquels les temps d'exécution des projets sont séparés des fonctions de ressources renouvelables et non renouvelables.

Pour diverses raisons, comme la météo, la durée des activités est incertaine et dans le monde réel, dans les projets de construction, nous sommes confrontés au problème des contraintes de ressources. En principe, la planification du projet devrait inclure la gestion des ressources (Leu, Chen et al. 1999).

Pour le problème d'ordonnancement de projet avec contraintes de ressources (RCPSP), étant donné que l'objectif est de minimiser le temps d'activité, les activités du projet doivent être planifiées lors de l'ordonnancement en fonction des priorités et des relations entre les activités et les contraintes de ressources (Stork 2001). Stork a supposé que le temps d'exécution des activités dans son article est incertain et a suivi une certaine distribution possible dans son modèle. Il a cité des facteurs tels que les conditions météorologiques, l'accès aux ressources et les coordonnées géographiques du projet comme les principales raisons influentes de l'ordonnancement basé sur les ressources.

La problématique du RCPSP consiste à comprimer et à réduire la durée du projet en hiérarchisant les activités et en limitant les ressources renouvelables (machines et ressources humaines). Dans ce cas, il suppose des informations complètes sur la durée de l'activité (comme les conditions préalables) et sur la manière d'utiliser les ressources, et tente de déterminer un calendrier exécutable. Par exemple, préparer une liste du début et de la fin des activités. Les experts en planification considèrent généralement des délais d'exécution modérés et la quantité de ressources nécessaires et préparent le calendrier du projet en conséquence, ce qui peut entraîner des problèmes dans le projet, car au moment de la mise en œuvre du projet, nous sommes confrontés au problème du manque de ressources et de l'incertitude dans la réalisation des activités (Bruni, Beraldi et al. 2011).

Il existe un domaine d'étude particulier en recherche opérationnelle pour discuter de l'ordonnancement, dont le but principal est de déterminer l'ordre dans lequel les activités sont exécutées en allouant des ressources (généralement limitées) au projet. L'un

des problèmes principaux et les plus importants dans ce domaine est la question du RCPSP qui, en raison de son application élevée dans les projets, constitue encore l'un des principaux défis dans la recherche universitaire et commerciale, car comme nous le savons, c'est l'un des principaux problèmes et discussions dans la planification de même qu'une question d'ordonnement. Dans les projets de mise en œuvre d'activités qui sont le plus petit membre d'un projet, nous devons composer généralement avec deux limitations principales : 1- La priorité 2- Les ressources. La première limite indique que les activités ne peuvent pas commencer avant l'achèvement des activités prédécesseur, et la deuxième limite indique que l'allocation des ressources aux activités dans l'ordonnement des activités ne peut pas dépasser la disponibilité des ressources (Morillo-Torres, Moreno-Velásquez et al. 2015).

En 2015, Bruni, Beraldi et al. (Bruni, Beraldi et al. 2015) ont mené une étude de cas réelle sur la question du RCPSP dans la construction d'appartements pour étudiants à l'Université de Calabre (Italie) et ont examiné l'impact de cette question sur l'amélioration du processus de projet. Ils ont prouvé que le projet effectué par leur modèle était achevé plus rapidement qu'un autre projet similaire parce que dans ce calendrier, la gestion des ressources est faite correctement, les gestionnaires de projet gèrent facilement les ressources limitées de leur projet et les activités sur la base de la priorité ont été planifiées et mises en œuvre avec les ressources disponibles.

Chakraborty, Sarker et al. (Chakraborty, Sarker et al. 2017) ont modélisé et développé le problème RCPSP, en supposant que le calendrier des activités était incertain et que la quantité de ressources requises pour les activités était définie. Dans ce modèle, afin de déterminer la durée des activités, des variables aléatoires qui ont des fonctions de distribution de probabilité différentes sont utilisées. Pour résoudre ce problème, ils ont développé une approche basée sur le concept d'optimisation forte. Afin de créer une durée indéfinie, ils ont utilisé six heuristiques différentes comme contrainte définitive sur leur modèle. Le modèle conçu a été résolu à l'aide de CBC (Coin-Branch & coupe). Pour garantir la performance et l'efficacité du modèle, des exemples de projets ont été mis en œuvre par le modèle.

Uysal, İleşleyen et al. (Uysal, İşleyen et al. 2018) ont mené une étude sur la question du RCPSP en 2018 dans un but d'ordonnement des projets avec des ressources limitées. Le modèle qu'ils ont proposé a été examiné et utilisé pour modéliser les conditions de demande de ressources aléatoires et augmenter l'ordonnement des projets dans les projets complexes.

2.4 ALGORITHMES ET METHODES POUR RESOUDRE LE PROBLEME D'ORDONNANCEMENT DE L'ALLOCATION DES RESSOURCES :

Talbot (Talbot 1982) a proposé une extension de la formulation linéaire en nombres entiers du RCPSP pour décrire le MRCPSPP en utilisant la représentation conventionnelle de la discrétisation temporelle. Maniezzo et Mingozzi (Maniezzo et Mingozzi 1999) ont développé une nouvelle formulation mathématique pour le MRCPSPP, qui est une extension de la formulation du RCPSP. Zapata et al. (Zapata, Hodge et al. 2008) ont créé trois formulations distinctes de plus. Le temps est modélisé différemment dans chacun des modèles proposés.

La première formulation utilise la représentation traditionnelle en temps discret, la deuxième emploie une variable de décision de séquence pour éliminer le temps du modèle, et la troisième emploie une série d'événements, avec des variables de décision indiquant si les activités commencent ou se terminent à ces événements. Selon les données expérimentales des auteurs, toutes ces formulations sont capables de résoudre des problèmes de la même gamme de taille que le modèle MIP classique (Talbot 1982).

Talbot (Talbot 1982), Patterson et al. (Talbot 1981), Sprecher et al. (Sprecher, Hartmann et al. 1997) ont développé des algorithmes exacts de branchement et de coupure. Zhu et al (Zhu, Bard et al. 2006) ont présenté une approche de branchement et de coupure dans laquelle, pour accélérer la convergence, le processus conçu utilise des coupes spécifiques au problème générées à partir des conflits de ressources et des relations de précedence, ainsi que des schémas de resserrement des limites et des règles de branchement

spécialement conçues. Les instances de PSPLIB avec 20 activités et avec 30 activités ont été évaluées avec le processus de branche et de coupe et ont trouvé la meilleure solution. Par conséquent, il existe encore des exemples à 30 activités pour lesquels aucune solution idéale n'a été établie (Węglarz, Józefowska et al. 2011).

Les activités sont reliées par des relations de préséance sans décalage, du début à la fin. À chaque instant, les ressources nécessaires à la réalisation des opérations sont accessibles en quantités discrètes et restreintes. Chaque activité comporte une durée distincte et définie et consomme une certaine quantité de ressources tout au long de chaque phase d'exécution. En outre, la non-préemption est généralement considérée comme suggérant qu'une fois qu'une action commence, elle ne peut pas être arrêtée. La dureté NP a été déterminée pour ce problème (Blazewicz, Lenstra et al. 1983).

Boctor (Boctor 1993), Drexl et Gruenewald (1993), et Boctor (1993) ont proposé des procédures heuristiques pour résoudre le MRCPSP en se basant sur les principes de priorité. Une stratégie d'ordonnancement stochastique a été donnée par Drexl et Gruenewald (1993). Cette heuristique peut fournir des solutions viables ou infaisables, et ils ont recommandé qu'elle soit utilisée de nombreuses fois pour maintenir la meilleure option faisable. Malheureusement, malgré le fait que Boctor (Boctor 1993) ait utilisé l'heuristique dix mille fois, elle n'a répondu à aucun des 240 problèmes de test de Boctor.

Boctor (Boctor 1993) a résolu le MRCPSP à l'aide de nombreux algorithmes d'ordonnancement parallèles et sériels qui n'examinaient que les ressources renouvelables. Il a étudié les performances de vingt et une techniques de résolution, et toutes ont été capables de produire des solutions viables lorsque de telles solutions existaient. Malgré le fait qu'elles commencent par l'ordonnancement d'actions qui n'ont pas de prédécesseurs et qu'elles ne planifient aucune activité tant que les prédécesseurs n'ont pas été d'ordonnancement, toutes ces heuristiques sont des heuristiques d'ordonnancement prospectif.

Pour résoudre le MRCPSP, de nombreux algorithmes de recherche de voisinage ont été conçus. Enfin, une technique d'échantillonnage est utilisée pour créer un certain

nombre de solutions, la meilleure étant conservée. Saowiski et al. (1994), Boctor (1996b), Józefowska et al. (2001) suggèrent des techniques de recuit simulé, tout comme Bouleimen et Lecocq (2001). Thomas et Salhi (Thomas et Salhi 1998), Baar et al. (Baar, Brucker et al. 1999), et Nonobe et Ibaraki (Nonobe et Ibaraki 2002) qui sont parmi ceux qui ont suggéré des techniques de recherche tabou.

Mori et Tseng (Mori et Tseng 1997), Ozdamar (Ozdamar 1999), Hartmann (Hartmann 2002), Alcaraz et al. (2003), Lova et al. (2009), Van Peteghem et Vanhoucke (Van Peteghem et Vanhoucke 2014), et Zamani (2011) ont également créé des algorithmes génétiques pour le problème MRCPSP. D'autres méthodes de recherche locale, comme l'optimisation par essaims de particules, ont été publiées par Zhang et al. (Zhang, Tam et al. 2006) et Jarboui et al. (2008), tandis que Merkle et al. (2002) et Chiang et al. (Chiang, Huang et al. 2008) ont créé des méthodologies d'optimisation par colonies de fourmis. Enfin, Messelis et De Causmaecker (Messelis et De Causmaecker 2014) étudient le développement d'une méthode de sélection automatisée pour choisir l'heuristique à employer en fonction des propriétés du MRCPSP (problème de projet d'ordonnancement multimode à ressources limitées) à résoudre.

Dans de nombreux contextes pratiques, les actions doivent être menées dans de nombreux endroits (sites). Par conséquent, les ressources nécessaires doivent être transportées sur différents sites d'exécution. Par exemple, de l'équipement lourd dans des projets de construction peut être requis sur de nombreux chantiers, et son transfert d'un chantier à un autre peut nécessiter beaucoup de temps. Dans cette situation, la modélisation et la résolution de la question de l'ordonnancement du projet en tant que MRCPSP peuvent aboutir à des horaires qui ne tiennent pas compte de ses délais de transfert.

Pour résoudre le RCPSP jusqu'à l'optimalité, plusieurs méthodes de Branch and Bound ont été développées (Patterson, Talbot et al. 1990) (Carlier et Latapie 1991) (Demeulemeester et Herroelen 1996) (Brucker, Knust et al. 1998). Cependant, aucune de ces stratégies ne permet de traiter des situations réelles à grande échelle dans un délai raisonnable.

Leu, Chen et al. (Leu, Chen et al. 2001) ont proposé un algorithme GA pour résoudre le problème RCPSP basé sur des ressources floues. Ils ont exprimé plusieurs avantages pour leur modèle : 1- Dans ce modèle (basé sur l'algorithme GA), contrairement aux modèles précédents, il n'est pas nécessaire de suivre les règles heuristiques, ce qui rend le modèle plus flexible. 2- Ce modèle, contrairement aux modèles heuristiques, offre plusieurs solutions optimales qui permettent au décideur de choisir la solution appropriée parmi les solutions diverses et optimales créées par son projet.

Étant donné qu'un projet implique un ensemble d'activités et de ressources (renouvelables et non renouvelables), l'un des problèmes du modèle RCPSP (ordonnancement du projet en tenant compte des contraintes de ressources) est de trouver le temps de travail le plus court avec les ressources possibles, de telle sorte que le projet soit achevé dans le temps le plus court possible (Brucker 2002). Li et Womer (Li et Womer 2015) ont développé des algorithmes ADP (programmation dynamique approximative) pour ces problèmes, qui ont utilisé CP (programmation par contraintes) pour améliorer la performance de cet algorithme.

La modélisation et la résolution du problème d'ordonnancement de projets sans tenir compte des délais de transfert peuvent donner lieu à des horaires infaisables. En utilisant les délais de transfert, Kruger et Scholl (Krüger et Scholl 2009) proposent des solutions aux problèmes d'ordonnancement de projets sous contrainte de ressources (RCPSPTT). Dans les deux cas, les auteurs ont formulé des modèles linéaires en nombres entiers, et une technique heuristique, utilisant des règles de priorité, a été utilisée pour résoudre les problèmes. Un algorithme génétique a été développé par Kruger (Krüger 2009) pour résoudre le problème. L'algorithme de recherche tabou basé sur le flux du RCPSP, développé par Poppenborg et Knust (Poppenborg et Knust 2016), tient compte des délais de transfert. Pour résoudre ce problème, ils proposent une technique génétique innovante et efficace.

Le problème de l'ordonnancement de projets sous contrainte de ressources avec temps de transfert (SRCPSPTT) a été résolu par Kruger et Scholl (Krüger et Scholl 2009) pour l'ordonnancement de projets uniques et multiples. Pour y arriver, auteurs forment

les deux problèmes sous forme de modèles linéaires en nombres entiers et élaborent des stratégies heuristiques basées sur des règles de priorité. La méthode génétique pour ce problème a été développée par Kruger (Krüger 2009) et Kadri et Boctor (Kadri et Boctor 2018). Un algorithme de solution par recherche tabou a également été développé par Poppenborg et Knust (Poppenborg et Knust 2016).

Les temps de configuration, qui représentent un sous-ensemble des temps de transfert, ont déjà été abordés dans le RCPSP, ce qui explique les temps de configuration comme le temps nécessaire pour préparer une ressource à utiliser dans une activité. Les catégories de configuration suivantes ont reçu la plus grande attention dans la littérature : temps de configuration indépendants de la séquence, temps de configuration dépendant de la séquence et temps de configuration dépendant de la planification.

Dans le deuxième scénario, les temps de configuration sont déterminés non seulement par l'activité réalisée et la ressource employée, mais également par l'ordre dans lequel ces activités sont gérées par cette ressource. Le temps de configuration dans des circonstances de configuration dépendant de la planification est déterminé par la séquence des opérations, la ressource nécessaire et d'autres aspects de la planification à exécuter. Mika et al. (Mika, Waligora et al. 2008), par exemple, ont étudié le défi de l'ordonnement du flux de travail sur une grille de ressources informatiques. Dans une telle circonstance, l'objectif est de définir le mode d'activité et le site d'exécution, ainsi qu'un temps de démarrage viable pour toutes les activités, afin que la durée du projet soit réduite au minimum (Mika, Waligora et al. 2008). Les contributions au RCPSP sont passées en revue dans de nombreuses publications. Korlisch (Kolisch 2000) présente un aperçu des modèles et des solutions pour le RCPSP déterministe ainsi que des contributions à la construction de systèmes de décision dans ce domaine. Herroelen et Leus (Herroelen et Leus 2005) se concentrent sur l'utilisation de l'incertitude pour l'ordonnement des projets. Plusieurs approches d'ordonnement sont discutées, notamment l'ordonnement réactif, l'ordonnement stochastique et l'ordonnement flou. Hartmann et Briskorn (Hartmann et Briskorn 2010) expliquent diverses extensions du RCPSP de base, où certaines conditions d'exécution des activités sont généralisées (par

exemple, en incluant des temps de préparation ou en permettant la préemption), la chronologie des activités est généralisée (en introduisant l'exécution parallèle, par exemple) et les contraintes de ressources sont généralisées (en prenant en compte les ressources partiellement renouvelables ou en utilisant des ressources dédiées). Quelques études plus anciennes ont été menées par Demeulemeester (Demeulemeester 1995), Elmaghraby (Elmaghraby 1995) et Herroelen et al. (Herroelen, De Reyck et al. 1998). Fournir une évaluation détaillée des temps de configuration et de leur catégorisation dans la planification du projet s'avère nécessaire.

NSGA-II a été appliqué à un projet de ressources afin d'optimiser la durée. Afin de réduire la durée du projet, la méthode sélectionne le type de ressource approprié pour chaque activité. En fonction de la durée, de la logique du réseau, de la capacité des ressources et des limites des relations entre les activités, l'approche comporte différents objectifs. L'utilisation de NSGA-II pour un projet d'optimisation des ressources est la première solution que nous connaissons (Mika, Waligora et al. 2008).

En 2013, Karam et Lazarova-Molnar (Karam et Lazarova-Molnar 2013) ont mené une étude sur les méthodes et les algorithmes d'ordonnancement de projet en tenant compte des contraintes de ressources du projet. Dans cette étude, l'application des algorithmes dans des projets échantillons a été étudiée. Les résultats de la recherche montrent que les algorithmes utilisés pour résoudre le problème RCPSP sont plus fiables que les autres algorithmes : PSO, SFLA et les algorithmes d'ensembles flous.

En 2015, Chakraborty, Sarker et al, (Chakraborty, Sarker et al. 2015), dans le but d'optimiser le temps du projet, ont présenté un modèle pour résoudre le problème RCPSP avec des ressources définies. Dans ce modèle, ils ont proposé et développé des variables et des contraintes d'une méthode B&C (Brancher et couper) pour les fonctions objectives, et ont résolu le modèle en utilisant la méthode CBC (Coin-Branche & Coupe). En outre, pour évaluer la performance et la fiabilité du modèle, il a été évalué avec des projets échantillons.

Fang, Kolisch et al., quant à eux, ont utilisé la méthode EDA (Estimation de l'algorithme de distribution) dans une étude de 2015 sur le RCPSP (projet d'ordonnancement avec ressource limitée). Ainsi, ils ont programmé l'EDA pour utiliser les données statistiques de la génération précédente d'élites afin de prédire la zone prometteuse dans l'espace de recherche. Ils ont utilisé la PBLX (stratégie de recherche locale basée sur la permutation) pour maximiser l'exploitabilité (Fang, Kolisch et al. 2015).

Kadri et Boctor (Kadri et Boctor 2018), dans leur étude de 2018 portant sur le RCPSP, ont conçu et développé un modèle basé sur un algorithme génétique (GA) utilisant un opérateur de croisement à deux points. Les résultats de leur recherche ont montré que l'algorithme GA livre une bonne performance pour résoudre les problèmes d'ordonnancement en tenant compte des contraintes de ressources. Dans cette recherche, ils ont développé une modélisation basée sur l'hypothèse que le droit de préséance et les relations de priorité ne sont pas autorisés. Ils ont également supposé la durée et la quantité de ressources nécessaires. Leur objectif dans cette recherche est d'obtenir le temps de début des activités afin de réduire la durée du projet.

ShaTao (Tao et Dong 2018) et al. Ont également suggéré que pour résoudre le problème d'ordonnancement de projet sous contrainte de ressources (RCPSP), il est préférable de considérer les structures de projet. Dans leur article, ils ont utilisé un réseau ET-OU pour présenter ces structures. Ils ont proposé un algorithme méta-heuristique basé sur le réseau pour résoudre le problème d'ordonnancement de projet sous contrainte de ressources et en utilisant l'algorithme NSGA-II.

Tirkolaei et al. (Tirkolaei, Goli et al. 2019), pour leur part, ont publié un article sur la question de l'ordonnancement de projets avec des ressources limitées pour minimiser le temps de projet. Ils ont d'abord validé le modèle en utilisant la méthode de contrainte Epsilon dans le logiciel GAMS, puis ont résolu le problème RCPSP à l'aide des algorithmes NSGA-II et du recuit simulé multi-objectifs.

Dans leurs recherches, Noemie Balouka et Izack Cohen (Balouka et Cohen 2021) ont utilisé l'approche de décomposition BENDERS avec des coupes spécialisées pour

minimiser le temps de projet sur des problèmes d'ordonnancement avec des ressources limitées. Leur objectif était de réduire la durée du projet en décidant de la manière d'allouer les ressources.

2.5 SOMMAIRE :

En examinant la littérature de recherche sur l'optimisation du temps et du coût des projets, nous concluons qu'il existe des lacunes dans les articles précédents pour créer un calendrier plus réaliste et efficace. Les plus importantes de ces lacunes sont les suivantes :

- 1- Les articles présentés sur l'optimisation du temps et du coût du projet ont été faits sans tenir compte des ressources limitées.
- 2- L'ordonnancement du projet a été réalisé en tenant compte des contraintes de ressources afin d'optimiser le temps et l'optimisation des coûts n'a pas été faite.
- 3- Lors de l'ordonnancement du projet, en considérant les contraintes de ressources, plusieurs types de ressources n'ont pas été considérés.

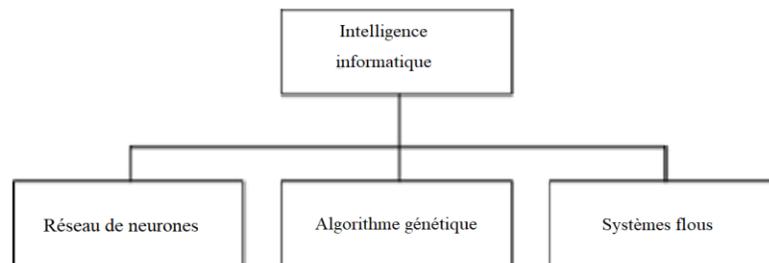
Pour répondre à ce qui précède, nous présentons dans ce mémoire un modèle mathématique pour optimiser le temps et le coût du projet en considérant les contraintes de ressources et dans le cas de plusieurs types de ressources et nous implémentons l'algorithme NSGA-II. À notre connaissance, aucune recherche n'a été faite à ce sujet jusqu'à présent.

CHAPITRE 3 : ALGORITHME GENETIQUE ET NSGA-II

3.1 ALGORITHME GENETIQUE

L'intelligence informatique ou le calcul logiciel comprend les algorithmes génétiques, les réseaux neuronaux et les systèmes flous. La théorie de la sélection naturelle de Darwin a inspiré le développement des algorithmes génétiques, qui sont des méthodes de recherche aléatoire. En appliquant le principe de survie correct, les algorithmes génétiques peuvent être utilisés pour approximer une population de solutions potentielles. Figure 3.1 Classification de l'intelligence.

Figure 3.1 Classification du renseignement



John Holland est l'inventeur des algorithmes génétiques. Avec la publication de l'article (Hayes-Roth 1975) intitulé « Adaptation des systèmes artificiels et naturels » en 1975, il a régulièrement répété deux points importants dans ses recherches : l'un est la capacité à afficher simplement des chaînes binaires pour coder des systèmes complexes, et l'autre est la capacité à simplifier une simple déformation pour modifier ces structures. Il a montré qu'avec un bon contrôle de la structure, des changements rapides dans les chaînes binaires se produisent avec certaines déformations, de sorte que la population des chaînes binaires évolue comme la population des organismes vivants.

Le livre de Holland (Holland 1992) contient des exemples dans lesquels des chaînes binaires sont utilisées pour résoudre des problèmes de codage. Des opérations

combinées sur ces chaînes binaires peuvent être utilisées pour la correction. Il définit les algorithmes génétiques comme suit :

« Les structures de contrôle peuvent être gérées de manière à faire évoluer les disciplines binaires en fonction du problème à résoudre, en apportant des réponses et des opérations. »

Un résultat important de 1975 est que même dans des espaces de recherche complexes avec certaines contraintes, les algorithmes génétiques ont tendance à générer des réponses qui sont universellement optimales ou tendent à être optimales.

Les principes et la base des algorithmes génétiques s'appuient sur l'évolution naturelle. Dans la discussion sur l'évolution, les changements de chaque organisme sont étudiés pour s'adapter efficacement à son environnement. La production de toute nouvelle « espèce » reflète l'arrangement des chromosomes de ses membres. Les opérations utilisées pour modifier l'arrangement des chromosomes sont utilisées lors de la production de nouveaux parents.

Holland (Holland 1992) a conçu des systèmes de classification, des systèmes inductifs à composantes génétiques. Avec l'invention de ces systèmes, Holland a développé un cadre dans lequel une population de règles codées sous forme de chaînes binaires amplifiées qui reçoivent alternativement de leur environnement évolue pour répondre de manière appropriée aux stimuli. Dans un système de classification, les règles représentent en fait une population changeante de disciplines.

Si ces systèmes sont bien conçus, il y aura des interactions entre les règles et l'algorithme génétique pour que le système évolue de manière à maximiser le renforcement positif de l'environnement. Dans les travaux antérieurs de Holland (Holland 1992), Wilson (Wilson, Ross et al. 2001) ou Booker (Booker 1993), des systèmes de classification ont été utilisés pour moduler l'acquisition des connaissances en fonction de la nature du mouvement dans le monde artificiel. Ces systèmes sont utilisés dans la plupart des problèmes d'apprentissage ; par exemple, dans le système proposé par David Goldberg (Goldberg 2002), il a montré qu'un système de classification peut contrôler le débit de gaz

naturel dans les lignes nationales, utiliser les fluctuations quotidiennes ou saisonnières de la demande pour trouver des fuites ou optimiser les profits.

Les faits présentés ci-dessus mettent en évidence les différences fondamentales entre les algorithmes génétiques et les méthodes conventionnelles de recherche et d'optimisation. Voici quelques-unes de ces différences :

- Au lieu de rechercher un point unique, les algorithmes génétiques recherchent des points parallèles.
- Les algorithmes génétiques ne nécessitent pas d'informations sur la dérivée et le rôle de la fonction et d'autres informations auxiliaires. Pour qu'une recherche soit efficace, elle doit être basée sur la fonction objective et le niveau de valeur souhaité.
- Les règles des algorithmes évolutionnaires suivent des probabilités au lieu de règles définies.
- L'utilisation des algorithmes génétiques est généralement plus explicite, car il n'y a pas de limite à la forme de la fonction objective en question.
- Les paramètres numériques constituent la base des algorithmes évolutionnaires, et non les paramètres eux-mêmes. Les algorithmes génétiques ont la capacité de fournir des réponses potentielles aux problèmes, et c'est à l'utilisateur de choisir la réponse finale ;

Par conséquent, ils sont très efficaces pour résoudre les problèmes d'optimisation multi-objectifs qui comportent un ensemble de réponses.

3.2 CONCEPTS DE BASE DE L'ALGORITHME GENETIQUE :

Voici quelques-uns des concepts et définitions qui sont communs à tous les algorithmes génétiques et qui doivent être mis à jour pour adapter l'algorithme à la question :

- Le Chromosome : Dans un GA simple, la réponse sélectionnée est affichée par des gènes séquentiels et est connue comme un chromosome. Les chromosomes sont des brins d'informations de longueur constante dans lesquels un certain nombre d'informations sont situées dans chaque locus (gène).
- Fit : Les chromosomes ont une fonction d'ajustement qui détermine leur capacité de survie et de reproduction. Chaque chromosome est évalué selon sa fonction dans cette fonction.
- La Population : Un ensemble de chromosomes uniformément sélectionnés est appelé une population.
- Génération : Chaque génération dans la société est connue comme une génération ; le nombre de personnes en son sein restera constant de génération en génération.
- Les principales fonctions des GA au fil des générations : Dans tout GA, pour passer de génération en génération, il faut modifier et améliorer les chromosomes de la génération en cours. Ce processus est réalisé par des fonctions génétiques telles que :
 - Opérateur de reproduction : Le transfert d'un chromosome d'une génération à l'autre en fonction de sa pertinence.
 - Opérateur d'intersection : Échange aléatoire ou défini de parties de chromosomes entre elles.
 - Opérateur de mutation : Modification aléatoire d'un chromosome en changeant la quantité de ses gènes.
 - Fonction inverse : Modification aléatoire d'un chromosome en changeant les valeurs de ses gènes.
- Enfant : Les chromosomes résultant de l'effet des actionneurs élevés sur les chromosomes de la génération actuelle sont appelés les enfants de la population de la génération suivante.

Ainsi, au lieu d'une seule réponse, GA modifie un ensemble de réponses simultanément et améliore la fonction objective ; dans cette méthode, la probabilité que GA se trouve dans un optimum localisé est fortement réduite.

3.3 LA SELECTION

La sélection est le processus par lequel les meilleurs chromosomes de chaque population sont sélectionnés pour la transplantation et la procréation. Le nombre de personnes au sein de chaque génération est déterminé par la valeur chromosomique. Par conséquent, la probabilité de sélectionner chaque chromosome au stade de la sélection est directement liée à l'importance de ce chromosome. La sélection à la roulette ou l'échantillonnage aléatoire est le schéma de sélection le plus simple et le plus largement utilisé.

Les bons chromosomes peuvent ne pas être sélectionnés, ce qui constitue la plus grande faiblesse de la méthode de la roulette. Une autre méthode s'appelle la méthode d'exécution.

Dans cette méthode, un certain nombre de chromosomes de la population sélectionnée sont sélectionnés au hasard avec une certaine quantité et le membre le plus précieux est sélectionné. Le facteur efficace dans la méthode de sélection des concours est le nombre de concours. À mesure que le niveau de compétition augmente, l'intensité de la sélection augmente et la probabilité de sélectionner des chromosomes plus précieux augmente. D'autre part, augmenter le nombre de compétitions réduit la diversité de la population parmi les parents sélectionnés. L'algorithme s'avère très efficace pour déterminer les valeurs optimales de la fonction objectif en déterminant le nombre de compétitions.

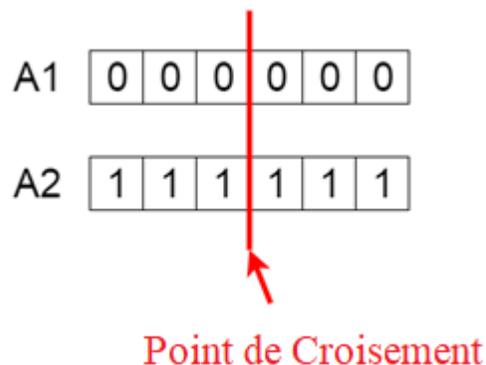
3.4 LE CROISEMENT (RECOMBINAISON) :

Le croisement est un opérateur génétique trouvé à la fois dans l'algorithme génétique et dans le calcul évolutif qui permet de combiner les informations génétiques de deux parents pour créer une progéniture. Le croisement est également connu sous le nom d'algorithme de recombinaison. Dans le domaine des mathématiques, c'est une façon de générer stochastiquement de nouvelles solutions à partir d'une population de solutions existantes. Il peut être comparé au croisement qui se produit lors de la reproduction en biologie. Il est également possible de générer de nouvelles solutions en clonant des solutions existantes, ce qui est analogue au processus de reproduction asexuée. Il arrive généralement que les solutions nouvellement générées soient mutées avant d'être incorporées dans la population.

Comme décrit précédemment, l'information génétique peut être stockée par une variété de structures de données dans le calcul évolutif, et grâce à différentes opérations de croisement, chaque représentation génétique peut être recombinaisonnée avec différentes représentations génétiques. Il existe plusieurs types courants de structures de données qui sont recombinaisonnées avec le croisement, comme les tableaux de bits, les vecteurs de nombres réels ou les arbres.

Figure 3.2 Point de croisement

Adapté de (Mallawaarachchi 2017)



3.5 LA MUTATION

Dans le cadre de l'évolution naturelle, la mutation est un processus aléatoire au cours duquel une partie d'un gène remplace un autre composant et crée une nouvelle structure génétique. Les mutations dans les algorithmes génétiques ont été considérées comme un opérateur aléatoire avec des performances très limitées, allant de 0,01 à 0,001. Les mutations se produisent toujours après le processus de croisement et changent aléatoirement les chromosomes du bébé. Cet opérateur évite une convergence prématurée de l'algorithme et garantit que le sous-espace recherché n'est jamais vide.

Une mutation est un facteur génétique utilisé pour maintenir la diversité génétique d'une génération de la population de chromosomes de l'algorithme génétique à la suivante. Celle-ci est semblable aux mutations biologiques.

L'opérateur de mutation peut provoquer des changements majeurs dans les chromosomes de la progéniture produits, entraînant la production de chromosomes ou de brins de progéniture pour avoir des gènes complètement différents du chromosome ou du brin parent.

En termes simples, l'opérateur de mutation est un processus aléatoire visant à perturber l'information génétique. Contrairement à l'opérateur de croisement, l'opérateur de mutation travaille au niveau du gène ; autrement dit, lorsque les gènes du brin ou du chromosome actuel sont copiés dans un nouveau brin ou chromosome, il est possible que chacun de ces gènes mute. Cette probabilité est généralement une très petite quantité appelée « probabilité de mutation ».

Figure 3.3 Mutation avant et après

Adapté de (Mallawaarachchi 2017)

Avant la mutation

A5

1	1	1	0	0	0
---	---	---	---	---	---

Après la mutation

A5

1	1	0	1	1	0
---	---	---	---	---	---

3.6 GROUPES D'ELITE ET ELITISTES :

Le groupe d'élite est la partie d'une population qui possède la valeur la plus élevée par rapport aux autres chromosomes. L'une des méthodes qui assure la survie des approximations sélectives dans un algorithme génétique est l'élitisme ; cela signifie que la population d'élite est préservée et transmise à la génération suivante. Cela préserve la survie des solutions optimales grâce aux propriétés aléatoires des algorithmes génétiques et aide beaucoup à sa convergence.

3.7 FACTEURS DE BASE AFFECTANT LES ALGORITHMES GENETIQUES :

Les sept facteurs clés qui affectent les performances des algorithmes génétiques sont les suivants :

- Comment les chromosomes sont affichés.
- Comment obtenir la population initiale.
- Stratégie de sélection des parents.
- Fonction de réglage.
- Opérateurs génétiques.
- Critères d'arrêt de l'algorithme.
- Critères d'évaluation des chromosomes.
- Nous examinons en détail chacun des éléments ci-dessus.

3.8 LA FAÇON DONT LE CHROMOSOME EST AFFICHE :

Pour chaque GA, une représentation chromosomique est requise afin que nous puissions expliquer les réponses à une question particulière. La façon dont les chromosomes sont affichés détermine la façon dont une question est formulée dans le format GA et quelle fonction de gène est utilisée. Comme mentionné ci-dessus, chaque chromosome formé à partir de gènes consécutifs peut être représenté comme binaire, correct ou une combinaison de caractères.

3.9 POPULATION INITIALE :

Pour exécuter chaque GA, nous devons commencer par l'ensemble initial de réponses. Les méthodes de production les plus courantes du groupe primaire sont :

- Génération aléatoire : dans cette méthode, en utilisant un algorithme de génération aléatoire de réponses possibles, la population initiale de réponses peut être obtenue.
- Nous devons ensuite utiliser les réponses obtenues pour résoudre le problème avec d'autres méthodes.

3.10 STRATEGIE DE SELECTION DES PARENTS :

Il est important de choisir les parents qui produiront la prochaine génération dans les GA. Afin de produire la prochaine génération, les meilleurs chromosomes et réponses seront sélectionnés. Les chromosomes de tous les individus ont une chance d'être sélectionnés pour la génération à venir. Étant donné que chaque chromosome est susceptible d'être sélectionné, les chromosomes importants seront plus susceptibles d'être sélectionnés. Par exemple, les parents peuvent être choisis par :

- Méthodes possibles :
 - Méthode de sélection de la roue de la roulette et ses généralisations.
 - Méthodes de mise à l'échelle
 - Méthodes de classement
- Méthodes improbables :
 - Sélection compétitive
 - La méthode de l'élitisme

Nous décrivons maintenant les méthodes ci-dessus :

3.10.1 Méthode de sélection de la roue de la roulette :

Dans cette méthode, on calcule d'abord la fonction d'ajustement de tous les chromosomes de la population actuelle (f_i s), puis on calcule les valeurs totales de la fonction d'ajustement des chromosomes, qui sont les suivantes :

$$F = \sum_{i=1}^{\text{Pop-Size}} f_i$$

Cela nous donne la probabilité suivante de choisir le chromosome « I » :

$$p_i = f_i / F$$

Dans la formule suivante, la probabilité cumulée des chromosomes, c'est-à-dire q_i , est calculée après que la probabilité de sélection de chaque chromosome a été calculée :

$$q_i = \sum_{r=1}^i p_r$$

Ensuite, un nombre aléatoire est généré « y » avec une distribution uniforme (1,0) dans la fonction $y \rightarrow (0,1)$; si $q_i < y \leq q_{i+1}$ chromosome $i + 1$ est alors sélectionné.

3.10.2 Méthode de classement géométrique standard :

Dans cette méthode, l'aptitude de la population est d'abord normalisée et les meilleurs et les pires chromosomes sont déterminés en fonction du degré d'aptitude ; chaque chromosome reçoit alors sa probabilité de sélection basée sur une certaine distribution, telle qu'une distribution géométrique ou triangulaire. Un exemple de distribution géométrique normalisée est :

$$\sum_k P(A_k) = \sum_{i=1}^P q' (1 - q)^{r-1}$$

Dans un sens :

$$q' = \frac{q}{1-(1-q)^p}$$

Probabilité de choisir la bonne personne : A_k

Nombre de personnes dans la population : k

Probabilité de choisir la meilleure personne : Q

Le rang de la personne dans le meilleur mode est 1 : R

3.10.3 Méthode de classement :

Dans cette méthode, les chromosomes les plus appropriés sont sélectionnés pour les opérations d'intersection et de mutation selon le processus d'échantillonnage aléatoire sans placement. Les recherches de Booker sur l'échantillonnage non aléatoire sans placement prouvent sa supériorité sur d'autres méthodes sélectionnées (telles que l'échantillonnage non aléatoire avec placement ou échantillonnage concurrentiel). Dans cette méthode, le nombre souhaité de chaque chaîne est égal à :

$$e_i = \left(\frac{f_i}{F}\right) \times K$$

« f_i » est égal à l'ajustement de la chaîne « i », « F » est l'ajustement moyen de toutes les chaînes de la population et « K » est la taille de la population. Chaque chaîne est attribuée à la génération suivante en fonction de la partie correcte de « e_i », et la partie fractionnaire de « e_i » indique la possibilité de transmettre une version supplémentaire de cette chaîne à la génération suivante. Par exemple, $e_i = 2,25$ signifie que la prochaine génération recevra deux copies de cette chaîne, et la probabilité de recevoir la troisième version est de 0,25.

3.10.4 Méthode de sélection compétitive

Le nombre de chromosomes « r » est sélectionné au hasard dans la population actuelle, puis le meilleur chromosome est sélectionné parmi les chromosomes « r » sélectionnés. Dans cette méthode, « r » est généralement considéré comme égal à 2 et est connu sous le nom de méthode compétitive binaire.

3.10.5 Méthode d'élitisme :

Dans cette méthode, la valeur de la fonction d'ajustement de chaque chromosome est d'abord calculée, puis les meilleurs chromosomes en matière de fonction d'ajustement sont sélectionnés.

3.10.6 Fonction de réglage :

Notre objectif dans l'optimisation de tout problème est d'optimiser la fonction d'ajustement. C'est le but de GA de trouver le gène dont la valeur de la fonction d'ajustement est la plus faible dans chaque population (dans les problèmes de minimisation).

3.10.7 Opérateurs génétiques

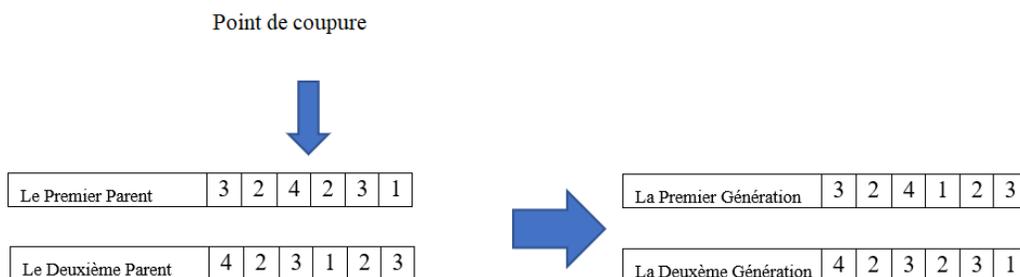
Les opérateurs génétiques sont utilisés pour produire des descendants. En supposant une chaîne de chromosomes, les opérateurs génétiques standard sont :

3.10.7.1 Intersection à un point

Dans l'opérateur de croisement à point unique, un point de coupure est sélectionné, et les parents 1 et les parents 2 sont coupés à partir de ce point pour l'intersection et divisés en deux parties, et les deuxièmes parties des parents sont inversées, et ainsi la génération suivante est produite.

Figure 3.4 Chromosome à croisement en un point.

Adapté de (Mallawaarachchi 2017)

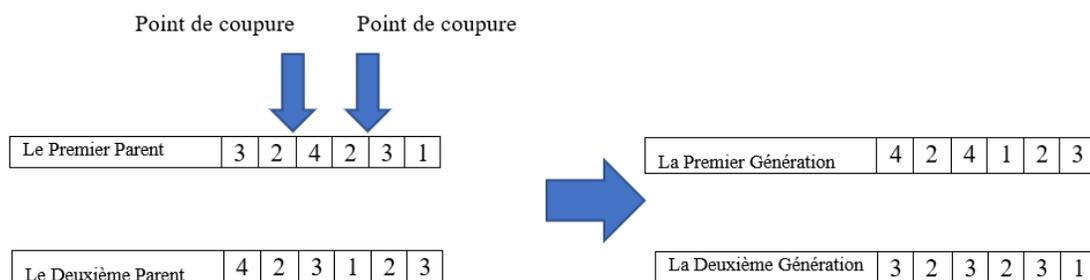


3.10.7.2 Intersection à deux points

Dans l'opérateur de croisement à deux points, deux points de coupure sont sélectionnés et les parents 1 et les parents 2 sont coupés à partir de ce point pour l'intersection et divisés en trois parties, puis les première et troisième parties des parents sont remplacées l'une par l'autre et donc la nouvelle génération est produite.

Figure 3.5 Chromosome croisé en deux points

Adapté de (Mallawaarachchi 2017)



3.10.7.3 *Opérateur de mutation alternatif*

Une fois que les opérateurs ont travaillé, une nouvelle progéniture est produite. Les enfants peu performants de la nouvelle génération devraient être remplacés par d'autres disciplines selon une stratégie alternative. La plus courante d'entre elles est une stratégie qui peut remplacer certaines des disciplines les plus faibles de la génération précédente dans la génération actuelle. Une stratégie cumulative implique que la chaîne la plus similaire de la génération précédente remplace la chaîne de la génération actuelle, tandis qu'une stratégie élitiste consiste à ajouter les chromosomes les plus similaires à la population actuelle, transmettant ainsi les chaînes les plus similaires à la nouvelle génération. Cette stratégie de substitution a été utilisée pour s'assurer que seuls les meilleurs chromosomes soient transférés de génération en génération.

3.10.8 **Critère d'arrêt de l'algorithme**

Pour que l'algorithme s'arrête après un certain temps et nous donne la réponse finale, nous devons considérer un critère d'arrêt pour celui-ci. Les critères d'arrêt qui sont régulièrement utilisés dans les GA sont :

3.10.8.1 *Nombre maximal de générations spécifié*

L'algorithme se termine lorsqu'il atteint le nombre maximum de générations.

3.10.8.2 *Convergence des populations*

En général, GA essaie de faire converger la population en une seule réponse. Si la population actuelle converge vers une seule réponse, l'algorithme se termine. La section suivante présente les performances de convergence en détail.

Stratégie de convergence.

Il ne fait aucun doute que lorsque la prochaine génération arrivera, il y aura une saturation des meilleurs chromosomes. Si un GA (*genetic algorithm*) ne maintient pas une grande diversité, il peut converger trop tôt avant de trouver la solution optimale ; sinon, il risque de ne pas trouver la solution optimale. Baker (Baker 1985) a observé

qu'une convergence rapide se produit lorsqu'un grand nombre de descendants d'une génération provient d'un petit ensemble de chromosomes ; en fait, lorsque de nombreux enfants naissent dans une population, mais laissent les autres sans enfant, le résultat est une perte rapide de diversité et de convergence, créant le problème de la dérive génétique. Il y a eu des suggestions pour empêcher une convergence rapide et améliorer la capacité des GA : l'une de ces suggestions était d'augmenter la population, mais ce n'est pas une option viable pour obtenir les réponses les plus complètes.

3.10.8.3 Variation de la population

Dans ce cas, si la somme des écarts des chromosomes de la population actuelle est inférieure à une certaine valeur, l'algorithme se termine.

3.10.8.4 Manque d'amélioration de la qualité des réponses obtenues

L'arrêt de l'algorithme pourrait être dû à un manque d'amélioration du meilleur type de réponse d'un certain nombre de générations.

3.11 ALGORITHMES GENETIQUES EN OPTIMISATION MULTI-OBJECTIFS (NSGA-II)

De nombreux problèmes d'optimisation en ingénierie sont multi-objectifs. Dans la plupart des cas, les concepteurs poursuivent des objectifs différents et parfois contradictoires ; l'exemple le plus clair de la poursuite d'objectifs multiples et contradictoires est la question de la qualité du produit et du coût de production ; dans ce cas précis, le designer veut minimiser le coût de production, mais aussi augmenter la qualité du produit ; d'autre part, il est évident que plus la qualité est élevée, plus le coût de production est élevé.

Le problème fondamental dans la résolution des problèmes d'optimisation multi-objectifs réside dans le fait qu'il y a rarement un seul point qui optimise tous les objectifs simultanément et au maximum. Au lieu de cela, il importe de trouver une satisfaction

équilibrée entre ces réponses et d'identifier un ensemble de réponses optimales. Pour ce faire, le concepteur sélectionne l'un des points optimaux les plus souhaitables. Par conséquent, différentes optimisations dans les problèmes d'optimisation multi-objectifs s'avèrent nécessaires.

Il existe de nombreuses méthodes innovantes pour résoudre des problèmes multi-objectifs, qui sont les méthodes classiques les plus importantes, telles que : les méthodes de pondération et les méthodes de planification idéale.

L'utilisation d'algorithmes génétiques pour résoudre des problèmes multi-objectifs s'est développée très rapidement au cours de la dernière décennie et a montré de très bons résultats. Les algorithmes génétiques conviennent à l'optimisation multi-objectifs pour diverses raisons, telles que :

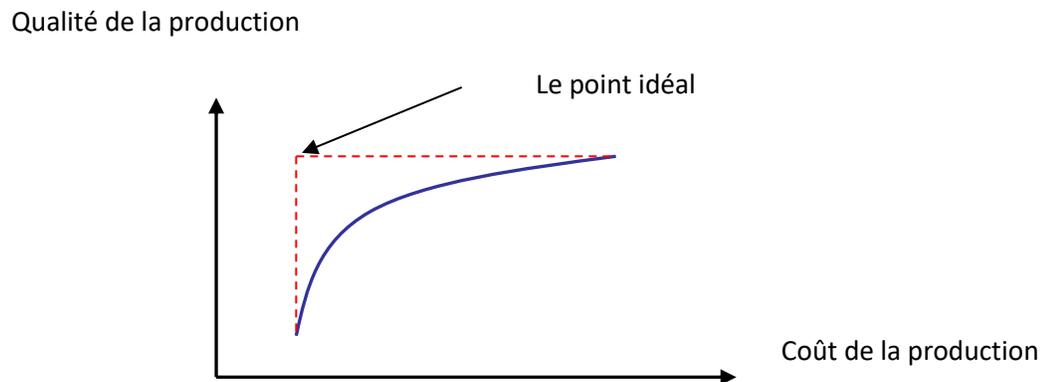
- Les algorithmes génétiques examinent simultanément un ensemble de réponses possibles et permettent au concepteur d'obtenir un ensemble de réponses optimales en une seule exécution de l'algorithme.
- Les algorithmes génétiques sont moins sensibles à la forme ou aux propriétés des points optimaux ; par exemple, la continuité ou la discontinuité, ainsi que l'exploitation minière cochléaire linéaire ou non linéaire n'ont aucun effet sur l'efficacité des méthodes évolutives.

Il s'agit des principaux problèmes des méthodes classiques de programmation mathématique : en utilisant les algorithmes génétiques dans l'optimisation multi-objectifs, on obtient suffisamment d'informations sur les solutions optimales possibles, puis le concepteur, en analysant ce large éventail de solutions, sélectionne le point optimal ; par conséquent, la condition de base dans la conception multi-objectifs, la résolution de problèmes et la décision du point optimal est souhaitable.

Les principaux objectifs de l'optimisation multi-objectifs sont les suivants :

- 1- Trouver l'ensemble optimal de réponses.
- 2- Sélectionner le point optimal dans la conception.

Figure 3.6 Sélection de la qualité et du coût de production comme problème d'optimisation multi-objectifs.



3.11.1 Problèmes multicritères

Les critères sont les règles et les normes qui guident le processus de prise de décision dans la sélection ou la formulation des caractéristiques. Les variables qui prennent une décision sur une question sont appelées la variable de décision.

Les objectifs sont exprimés sous forme de fonctions mathématiques et l'indicateur utilise une approche mathématique pour améliorer les fonctionnalités qui doivent être optimisées dans la prise de décision. À cet égard, il existe plus ou moins deux approches qui représentent respectivement la maximisation et l'extraction.

3.11.2 Domination de Pareto

Pour les deux vecteurs de décision X_1 et X_2 , on dit que X_1 surmonte le vecteur X_2 , si deux conditions sont remplies : la première est que X_1 n'est pas plus mauvais que X_2 dans tous les buts, et la seconde est que X_1 est absolument meilleur que X_2 dans au moins un but. Le concept ci-dessus s'exprime mathématiquement de la manière suivante :

$$X_1 < X_2 \Leftrightarrow (\forall i \in \{1, 2, \dots, n\}: f_i(X_1) \leq f_i(X_2)) \wedge (\exists i \in \{1, 2, \dots, n\}: f_i(X_1) < f_i(X_2))$$

Ainsi, entre les deux vecteurs de décision X_1 et X_2 , les relations de dominance, de faible dominance et de non-dominance s'expriment comme suit :

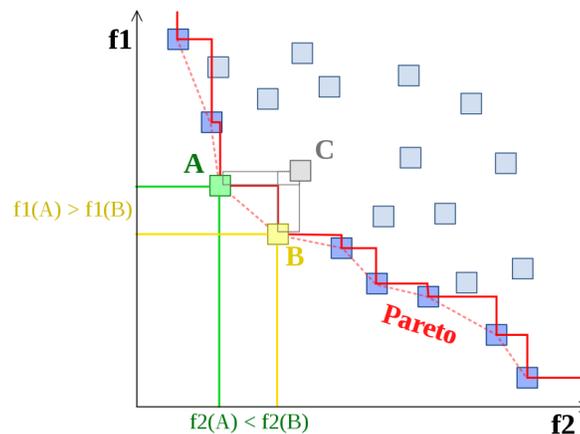
Dans le premier cas, X_1 surpasse X_2 ; dans le deuxième cas, X_1 surpasse faiblement X_2 et dans le troisième cas, X_1 ne surpasse pas X_2 .

$$X_1 < X_2 \quad \text{if} \quad F(X_1) < F(X_2)$$

$$X_1 \leq X_2 \quad \text{if} \quad F(X_1) \leq F(X_2)$$

$$X_1 \sim X_2 \quad \text{if} \quad (F(X_1) \not\leq F(X_2)) \wedge (F(X_2) \not\leq F(X_1))$$

Figure 3.7 Image d'optimisation de Pareto dans l'espace cible



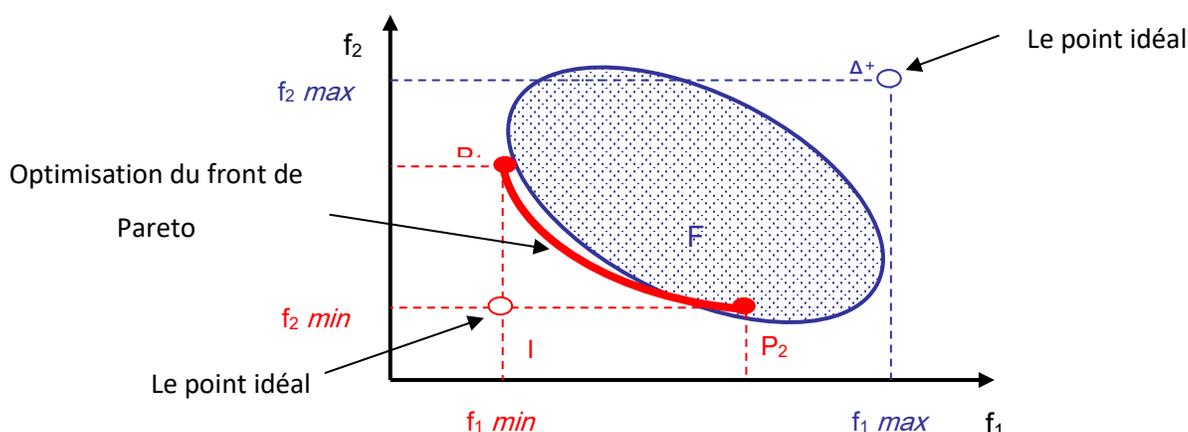
En utilisant la dominance de Pareto, un problème multi-objectif est défini comme ayant le critère d'optimalité. « Wolfrido Pareto » a introduit ce concept en 1896 et porte donc son nom.

Le vecteur de décision $X \in X_f$ par rapport à (mot manquant) $A \subseteq X_f$ est dominant si :

$$\exists a \in A: X < a$$

La différence fondamentale entre un problème à objectif unique et un problème multi-objectif est identifiée ici. Les problèmes multi-objectifs ne se limitent pas à une solution optimale, mais ils contiennent un ensemble de solutions optimales. Aucune des réponses n'est supérieure aux autres, sauf si la décision du décideur est définie.

Figure 3.8 : Le front de Pareto et le point idéal

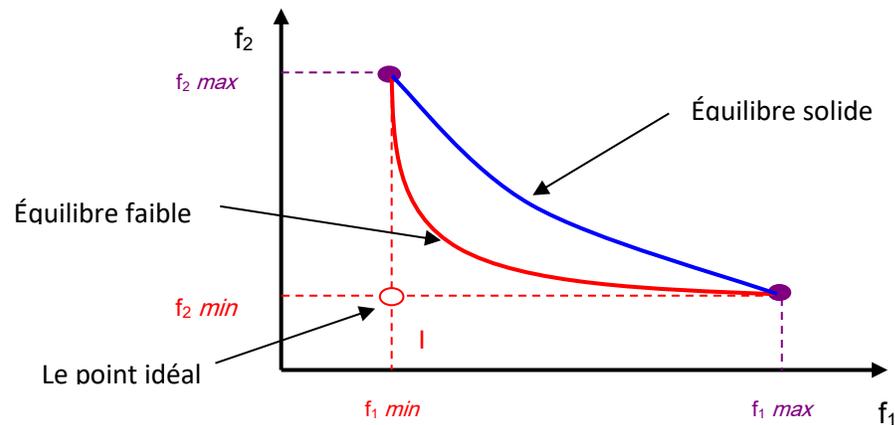


L'ensemble optimal de Pareto et les vecteurs objectifs correspondants sont appelés front optimal de Pareto si toutes les solutions optimales de Pareto existent dans un problème multi-objectifs.

Le point idéal (solution utopique) est le point auquel les valeurs de toutes les fonctions correspondent à l'objectif minimum. Évidemment, il n'existe généralement pas de point idéal dans l'espace cible. Le point anti idéal (point nadir) représente également le point où les valeurs de toutes les fonctions objectives sont maximales.

Le Front de Pareto nous indique comment les objectifs sont équilibrés. Sur la base de la forme du front de Pareto, cet équilibre indique à quel point les objectifs sont sensibles les uns aux autres.

Figure 3.9 Équilibre fort et faible dans un problème multi-objectifs.



La forme du front de Pareto est très proche du point idéal en équilibre faible ; ce n'est qu'à proximité de ce point que les valeurs des fonctions cibles augmentent rapidement. En équilibre fort, en raison de la forme du front de Pareto, il n'y a pas beaucoup de proximité avec le point idéal. Au lieu de cela, sur le front de Pareto, pour une diminution d'un objectif, il y a une augmentation d'un autre objectif.

3.11.3 Optimisation multi-objectifs par les algorithmes génétiques

L'idée d'utiliser les algorithmes génétiques dans une requête multi-objectifs a été proposée pour la première fois par Rosenberg en 1968 sous la forme de l'utilisation de propriétés multiples dans la recherche génétique. Cependant, la première idée a été mise en œuvre par Schaffer en 1984. Dans l'algorithme proposé par Schaefer, appelé VEGA, chaque ensemble de génération est divisé en plusieurs sous-ensembles de taille et de nombre d'objectifs du problème égaux. Dans chacun de ces sous-ensembles, une cible est indépendamment la base de la sélection des chromosomes. À l'étape suivante, les membres de ces sous-ensembles sont mélangés par des opérateurs de reproduction et d'hybridation pour former la génération suivante.

collection soient classés. Enfin, le rang de chaque membre constituera le critère de sélection et de production dans la prochaine étape de l'algorithme. Dans la méthode ci-dessus, la probabilité de reproduire toutes les réponses infructueuses est la même, mais cette probabilité s'avère plus élevée que les autres réponses infructueuses. Dans les algorithmes génétiques, les méthodes communautaires de Pareto utilisent l'idée d'optimisation de Pareto. L'optimisation de Pareto a été incorporée dans une variété d'algorithmes. Certains d'entre eux directement et certains d'entre eux considèrent le rang obtenu par les réponses ci-dessus comme la base de valeur, appelée « base de Pareto ». Certains des algorithmes les plus importants sont :

- NPGA (1991); MOGA (1993); NSGA (1994); SPEA (1998); ESAP (1999); SPEA-II (2001) et NSGA-II (2002) qui est la méthode discutée dans ce mémoire.

Les problèmes les plus importants des algorithmes évolutionnaires qui font la différence sont :

- Méthode d'évaluation des réponses : sélectionne les meilleures réponses et les reproduit, ce qui conduira les générations futures vers le front optimal de Pareto.
- Convergence de l'algorithme vers le front optimal de Pareto : Calcul de la vitesse de convergence de l'algorithme vers le front optimal de Pareto.
- Distribution des réponses le long du front de Pareto : l'objectif principal des réponses aux algorithmes multi-objectifs est de maintenir l'équilibre entre les cibles sur le front optimal de Pareto. Par conséquent, bien que les réponses doivent converger vers le front optimal, elles doivent être réparties de manière égale sur le front.

3.11.4 Algorithme génétique standard

Voici une description simple de l'algorithme génétique standard et de base :

Pop_{size} : Taille de la population

G_{max} : Nombre maximal de générations.

0.Début.

1. Déterminer la taille des paramètres suivants

2. Taille de la POP_{size} , G_{max} et le taux d'utilisation des opérateurs

3.production de la population initiale :

Générer la réponse au nombre Pop_{size} et la placer dans la population initiale.

4.Déterminer la population parentale :

Choisir la population parentale à partir de la population actuelle en utilisant l'une des méthodes de sélection des parents.

5.Fabriquer la nouvelle génération :

Génération par opérateurs génétiques dans les membres de la population parentale, en nombre Pop_{size} , et remplacement dans la population courante.

6. Si le critère d'arrêt est saturé, il faut l'arrêter et imprimer la meilleure réponse sur la sortie ; sinon, il faut passer à l'étape 3.

3.11.5 Algorithme génétique NSGA-II

L'algorithme NSGA-II est une version modifiée de l'algorithme NSGA. Cet algorithme couvre les faiblesses de la méthode NSGA et réduit la méthode habituelle de solutions défectueuses, de calcul et de temps pour superposer les réponses (Deb, Pratap et al. 2002).

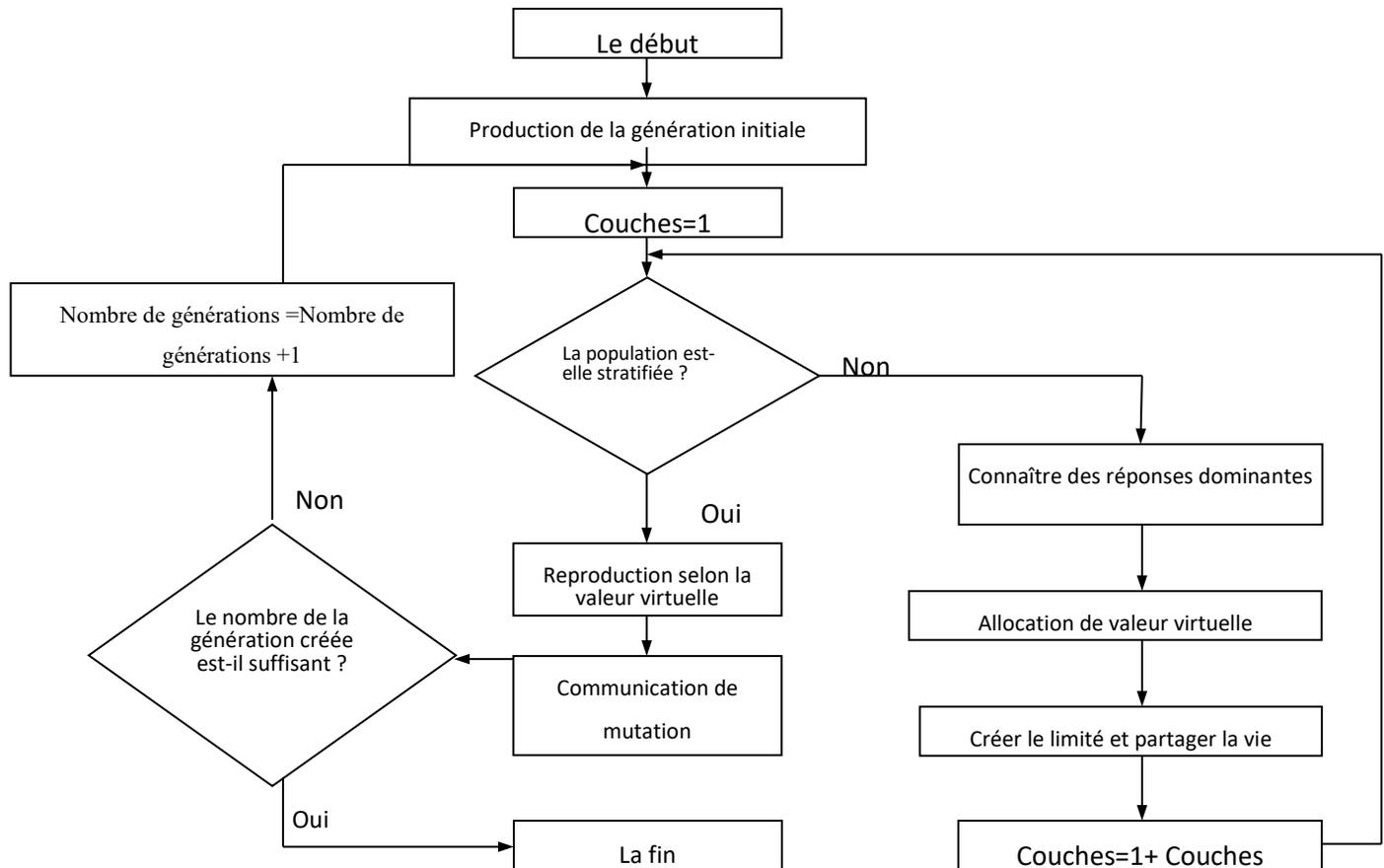
De plus, l'ensemble d'élite, qui contient les réponses infructueuses, a augmenté l'efficacité et la convergence de l'algorithme ; de plus, l'opérateur de comparaison de congestion maintient la distribution des réponses le long du front de Pareto sans que l'utilisateur ait besoin de spécifier des opérateurs.

3.11.5.1 Trier rapidement les réponses non résolues

Dans l'étape de classement rapide des réponses infructueuses, deux valeurs sont calculées pour chaque réponse dans l'ensemble actuel :

Figure 3.11 Diagramme de flux de travail dans l'algorithme NSGA-II

Adapté de (Kumar 2019)



1- Ensemble dominant : Les réponses qui ont vaincu le membre désiré (n_p).

2. Ensemble rejeté : Toutes les réponses qui ont été rejetées par le membre désiré (S_p).

Pour les réponses avec $n_p = 0$, le membre du front est invaincu. Une fois que le premier front invaincu est formé pour tous les membres, un nombre est déduit du nombre dominant de tous les membres de l'équipe vaincue. Les membres avec un numéro de victoire de zéro sont placés dans un ensemble séparé. Cet ensemble constituera la deuxième

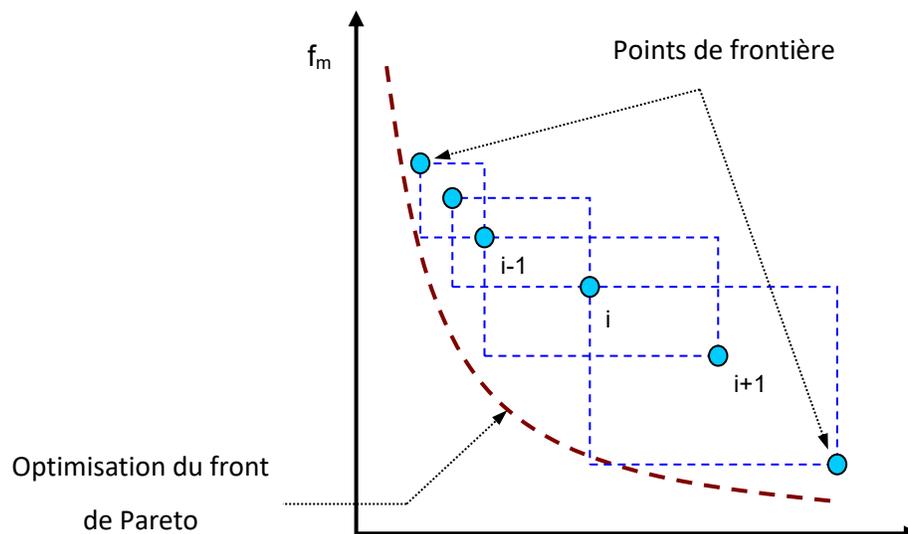
couche défavorable. L'opération est ensuite répétée pour les membres de la deuxième couche, et ce processus se poursuit jusqu'à ce que l'ensemble des réponses soit stratifié.

3.11.5.2 Opérateur de comparaison d'encombrement

Avant de décrire l'opérateur de comparaison de congestion, l'échelle d'estimation de densité est utilisée. Cette valeur sert à estimer la compression des solutions autour d'une solution donnée. À cet effet, l'indice de distance, i_{distance} , est pris en compte à toutes fins utiles.

La valeur de cet indice est égale à la distance entre les réponses dans lesquelles la réponse souhaitée est enfermée à l'écart maximum et minimum des objectifs correspondant de ce côté. Ensuite, l'échelle d'estimation de densité est obtenue à partir de la somme de chaque indice de distance ; par conséquent, les réponses situées dans des zones plus isolées ont un indice de distance plus élevé.

Figure 3.12 Estimation de la compression des réponses



3.11.5.3 *Partie principale de l'algorithme NSGA-II*

Initialement, N membres sont sélectionnés pour créer la communauté. Ensuite, leurs scores de dominance de Pareto sont comptabilisés. Chaque réponse se voit attribuer une compétence autorisée égale au niveau d'échec de cette réponse. Ainsi, les réponses avec une compétence moins autorisée sont préférables. Par le processus de sélection et de compétition binaire, les membres de la communauté produisent des enfants de taille N et une combinaison de la population de parents et d'enfants de taille 2N. Avec l'approche tri rapide des mauvaises réponses, toutes les mauvaises réponses sont identifiées dans la communauté, puis superposées.

À la génération suivante, la population est façonnée par les réponses de la première couche. Les membres restants de la population générée sont sélectionnés dans les couches suivantes si la taille de la première couche est inférieure à N. Les couches sont sélectionnées jusqu'à ce que la taille totale des couches dépasse N. Dans ce cas, les membres de la couche sont triés en ordre décroissant par l'opérateur <n jusqu'à ce que la taille de la population de la génération suivante après N n'augmente pas. Les meilleures réponses de cette couche sont sélectionnées et entrées dans la population de génération suivante, et une population P_{t+1} de taille N est reproduite. Cette population de parents, dans une compétition binaire, produit N enfants. Dans ce processus, l'opérateur <n est la principale raison de la sélection de membres plus précieux et amène les membres qui ont un certain inconfort et qui se trouvent dans une zone plus isolée à produire davantage d'enfants. Cela implique que l'algorithme se traduira également par une bonne répartition des réponses le long du front de Pareto puisqu'il aura convergé vers le front optimal.

3.11.5.4 *Mode d'application des restrictions*

Les algorithmes évolutionnaires appliquent des contraintes de manière très simple. Il existe deux types de contraintes : les équations et les inégalités.

$$H(X) = 0 \rightarrow |H(X)| - \varepsilon = 0$$

$$G(X) \leq 0 \rightarrow \max (G(X) - 0)$$

Avec le processus élevé, les contraintes deviennent des objectifs qui doivent être minimisés. Les contraintes sont appliquées par différents algorithmes de différentes manières, mais dans l'algorithme NSGA-II, les solutions sont comparées en surmontant les contraintes. Les contraintes d'un problème multi-objectifs définissent la zonabilité dans l'espace cible. Par conséquent, il y a un maximum de trois cas pour les deux réponses hypothétiques.

- 1- Les deux peuvent être répondues et une réponse avec une meilleure fonction objective est sélectionnée.
- 2- On peut répondre à l'un et non pas à l'autre. Dans ce cas, une réponse doit être sélectionnée.
- 3- Les deux sont sans réponse. Dans ce cas, une réponse avec moins d'écart que les contraintes est sélectionnée.

Pour les contraintes qui ont été converties selon l'équation ci-dessus, pour le troisième cas, il faut choisir une réponse qui a une valeur inférieure à la contrainte convertie.

Les deux derniers cas du processus de sélection décrit ci-dessus dans l'algorithme NSGA-II sont définis en définissant la dominance des contraintes comme suit :

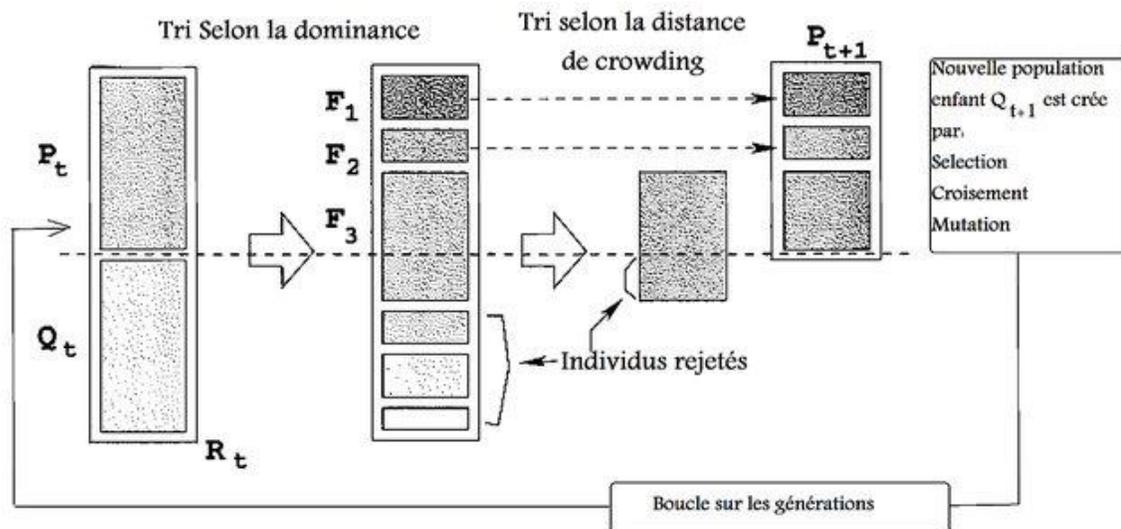
- La réponse « i » a un avantage limité sur la réponse « j » si les conditions suivantes sont remplies :
 - La réponse « i » est possible et la réponse « j » est impossible.
 - Les réponses « i » et « j » sont toutes deux impossibles, mais la réponse « i » s'écarte d'une borne inférieure.
 - Les deux réponses sont possibles, mais la réponse « i » surpasse la réponse « j » Pareto.
- Avec l'activité d'opérateur décrite ci-dessus, les réponses possibles sont mieux classées que les réponses impossibles, mais les réponses impossibles qui s'écartent moins de la contrainte sont sélectionnées. De cette façon, l'algorithme

va à l'espace possible dans la zone cible et les réponses possibles vont au front invincible de Pareto.

Figure 3.13 Processus de sélection et de recréation des membres de couche dans l'algorithme

NSGA-II

Source (Lee 2019)



Ce n'est pas aussi complexe en matière de calcul que la méthode NSGA-II. Nous avons pu obtenir des résultats précis en utilisant l'algorithme NSGA-II ; Cet algorithme a créé de bien meilleurs fronts par rapport aux méthodes précédentes. La méthode NSGA-II possède une fonction similaire à la méthode SPEA-II, mais est généralement plus simple.

CHAPITRE 4 : METHODOLOGIE DE RECHERCHE

4.1 UN MODELE MATHEMATIQUE

Étant donné l'ensemble des activités du projet à réaliser, leurs relations de priorité, la contrainte supérieure sur la quantité de ressources à affecter aux activités, et les types de ressources, notre problème est de déterminer la date de début et le type de ressources pour chaque activité, ainsi que la quantité de ressources à affecter au projet qui minimise le coût total d'exécution et le temps du projet tout en satisfaisant les relations de priorité et en n'utilisant pas plus de ressources que celles allouées au projet. Le produit de la quantité de ressources fournies à l'activité multipliée par son coût unitaire et la durée de l'activité est le coût des ressources.

Commençons par la nomenclature qui a été utilisée pour créer ce modèle avant de passer à la formulation mathématique recommandée.

L'indice	La description
i, j	L'indice de tâche ; $i, j = 1, \dots, N$
n	L'indice de la dernière activité
m	L'indice du type de ressource ; $m=1, \dots, M$ (M est le nombre de types de ressources) ; par exemple, nous avons 3 types de ressources : ouvrier junior, ouvrier et ouvrier senior.
t	Indice de la période de temps ; $t = 1, \dots, T$ (T est l'horizon de planification total/la limite supérieure du projet)
k	Ensemble de ressources ; $k= 1, \dots, R$ (R est le nombre de ressources)
Les paramètres	La description
C_{ikm}	Coût total de l'activité i par la ressource k dans le type m
d_{im}	Durée de l'activité i si elle est exécutée par des ressources de type m
PR_i	Ensemble des prédécesseurs immédiats de la tâche i

U_{kim}	Nombre d'unités de ressources k de type m nécessaires à l'exécution de la tâche i
L_k	Capacité de la ressource k
Les variables de décision	La description
X_{imt}, Y_{imt}	Variable binaire qui 1, si l'activité i commence avec des ressources de type m et au temps t ; 0 sinon
S_i	le temps de début de l'activité i
F_i	le temps de fin de l'activité i
X_{im}	Variable binaire qui 1, si l'activité i commence avec des ressources de type m ; 0 sinon

$$\text{Min } Z_1 = \sum_{k=1}^R \sum_{m=1}^{M_i} \sum_{i=1}^n (C_{kim} X_{im}) \quad 4.1$$

$$\text{Min } Z_2 = F_n \quad 4.2$$

s.t.

$$\sum_{m=1}^{M_j} X_{im} = 1, \quad i = 1, \dots, n, \quad 4.3$$

$$\sum_{m=1}^{M_j} \sum_{t=1}^T X_{imt} = F_i - S_i, \quad i = 1, \dots, n, \quad 4.4$$

$$S_i + \left(\sum_{m=1}^{M_j} d_{im} \cdot X_{im} \right) * pr_{(j)} \leq S_j, \quad \forall i \in pr_j \quad 4.5$$

$$F_j \geq S_j + \left(\sum_{m=1}^{M_j} d_{jm} \cdot X_{jm} \right), \quad j = 1, \dots, n, \quad 4.6$$

$$\sum_i^n \sum_m^{M_i} U_{kim} Y_{imt} \leq L_k \quad \forall k, t \quad 4.7$$

$$t + M * (1 - Y_{imt}) \geq S(i) \quad 4.8$$

$$t - M * (1 - Y_{imt}) \leq S(i) + d_{i,m} \quad 4.9$$

$$S_1 = 0 \quad 4.10$$

$$S_i \geq 0, \quad i = 2, \dots, n, \quad 4.11$$

$$X_{imt} \in \{0,1\}, i = 1, \dots, n, m = 1, \dots, M_i, t = 1, \dots, T \quad 4.12$$

$$y^+ \geq 0, \quad y^- \geq 0 \quad 4.13$$

Les équations (4.1) et (4.2), respectivement, sont les fonctions d'objectif du modèle pour diminuer le temps d'achèvement du projet et son coût. L'équation (4.1), en réalité, réduit le coût des ressources au niveau le plus bas possible. En réalité, le but de cette fonction objectif est de réduire l'utilisation des ressources. Le temps de réalisation du projet est calculé à l'aide de l'équation (4.2). L'équation 4.3 spécifie les ressources requises pour chaque activité. Les contraintes 4.4,4.8,4.9 garantissent que lorsque l'action est réalisée, aucune ressource n'est allouée. Les contraintes 4.5 et 4.6 assurent qu'aucune tâche ne peut commencer avant que ses prédécesseurs immédiats aient terminé leur travail. La contrainte 4.7 garantit que la quantité de ressources affectées ne dépasse pas la quantité qui peut être rendue accessible. Les restrictions 4.10 & 4.11 nécessitent des temps de démarrage non négatifs. Enfin, les restrictions 4.12 & 4.13 décrivent le type de variables de décision.

4.2 ALGORITHME GENETIQUE NSGA-II IMPLEMENTE

Un front de Pareto est ce à quoi ressemble un algorithme génétique multi-objectifs lorsque deux opérateurs essentiels sont ajoutés à l'algorithme génétique conventionnel à objectif unique, au lieu de trouver la meilleure réponse. Ces deux opérateurs comportent les caractéristiques suivantes :

- Basé sur un tri infructueux, cet opérateur donne aux membres d'une population une mesure de supériorité (rang).
- La même réponse peut être utilisée plusieurs fois au sein un seul opérateur.

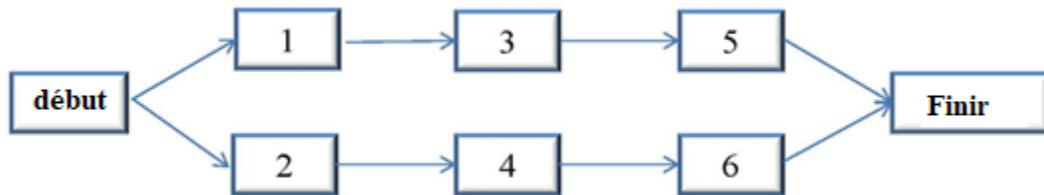
4.2.1 Conception de base

Avant que l'algorithme génétique ne soit mis en œuvre, nous effectuons un schéma de prétraitement avec le modèle (qui a été présenté par Sprecher et al. 1997). Nous cherchons à faire correspondre les données problématiques et à réduire l'espace. Après le schéma de prétraitement, l'algorithme multi-objectifs NSGA-II est mis en œuvre, dont les étapes sont les suivantes :

- La population initiale du test (P_t) comporte des chromosomes N (l'échantillon est sélectionné au hasard, donc aucun doublon n'est présent).
- Générer une population secondaire (Q_t) avec N chromosomes par promotion et mutation sur N chromosomes primaires (P_t).
- Rechercher les chromosomes extérieurs les plus proches de l'origine d'une population de $2N$ chromosomes ($P_t + Q_t$), (Recherche de points non post à l'aide d'un algorithme de tri non post).
- Sélectionner N membres non postaux ($P_t + 1$) dans l'ensemble des points non postaux avec $2N$ membres du front de Pareto ($P_t + Q_t$). En utilisant l'espace de coordonnées de nivellement du coût du temps de la ressource, le nombre de N chromosomes les plus proches de l'origine des coordonnées est sélectionné.

- Appliquer la sélection, l'accouplement et la mutation sur l'ancienne génération (P_t+1) et produire une nouvelle génération avec une taille de chromosome N (Q_t+1). En d'autres termes, les composants d'un algorithme génétique sont considérés comme le réseau d'activités d'un projet simple comportant 8 activités, comme le montre la figure (4.1).

Figure 4.1 Exemple de projet



4.2.2 Codage et représentation du modèle

En raison de la nature multi-types de notre scénario d'ordonnement de projet, l'algorithme génétique doit répondre à la fois aux problèmes d'ordonnement (spécification du temps de début) et de type (spécification du type de ressource). D'autre part, comme ces deux problèmes interagissent et s'influencent mutuellement, ils doivent être posés simultanément.

La matrice de paires $I = (\lambda, \mu)$ est utilisée pour représenter les membres du modèle. Dans la première ligne de cette matrice (λ), la représentation des activités du projet est spécifiée comme une liste d'activités et dans la deuxième ligne, l'affectation des types à la liste des ressources (μ) est spécifiée.

$$I = \begin{pmatrix} j_1 \dots j_J \\ \mu(j_1) \dots \mu(j_J) \end{pmatrix}$$

La longueur de ce chromosome à deux lignées est égale au nombre d'activités du projet (J). Dans ce qui suit, nous allons nous familiariser avec la représentation des chromosomes sous forme de liste d'activités :

- Liste des activités : la liste des activités en représentation chromosomique et les activités d'un réseau une rangée avec les prérequis suivants sont les suivantes :

$$\lambda = [j_1, j_2, \dots, j_g, \dots, j_J]$$

Dans la liste des activités, l'indice de position j_g est supérieur à l'indice de position de ses prérequis. En d'autres termes, l'activité j_g est listée après les prérequis.

- Algorithme de génération de la liste des activités : cet algorithme se compose de J étapes (c'est-à-dire qu'autant d'activités sont répétées). À chaque étape, une activité est sélectionnée dans l'ensemble des activités éligibles D_g (activités dont les prérequis ont été saisis dans la liste P_j à cette étape) et ajoutée à la liste d'activités PS_g .

Figure 4.2 Algorithme de génération de liste d'activités

$$\begin{aligned} j^* &:= j_h; \\ d_{j^*} &:= d_{j^*m}; \\ ES_{j^*} &= \max_{i \in P_{j^*}} \{F_h\}; \\ S_{j^*} &= \min \{t \mid ES_{j^*} \leq t, r_{j^*mk} \leq \tilde{R}_k(\tau), \tau = t, \dots, t + d_{j^*-1}, k = 1, \dots, K\}; \\ f_{j^*} &= S_{j^*} + d_{j^*}; \\ F_{h+1} &:= F_h \cup \{f_{j^*}\}; \\ PS_{h+1} &:= PS_h \cup \{j^*\}; \\ h &:= h + 1; \\ &END. \end{aligned}$$

Le tableau 4.1 montre la mise en œuvre étape par étape de l'algorithme dans le réseau d'activités du projet de la figure (4.2) :

Tableau 4.1 Le processus de production d'un exemple de liste d'activités

G	1	2	3	4	5	6
D _g	{1,2}	{1,4}	{1,6}	{3,6}	{3}	{5}1
J*	2	4	1	6	3	5

Les activités acceptées (j^*) à chaque étape sont sélectionnées dans l'ensemble D_g selon les règles de priorité. Dans la modélisation du problème, la loi de priorité RAN est utilisée pour sélectionner l'activité j^* et les activités de chaque étape sont sélectionnées aléatoirement parmi les activités éligibles, telles que :

$$\lambda = [1,3,5,2,4,6]$$

$$\lambda = [2,4,6,1,3,5]$$

$$\lambda = [1,2,3,5,4,6]$$

4.2.3 Créer une population primaire

Dans la population d'origine, chaque membre ou chromosome est assemblé en répétant les étapes suivantes, et ce, jusqu'à ce que le N chromosome ait été assemblé :

- Affectation du type de ressource : pour chaque ressource requise pour l'activité $j = 1, \dots, J$ au hasard, comme une fonction de Bernoulli avec 50 % de probabilité de 1 et 50 % de 2, nous affectons un type de l'ensemble des types exécutables ($\mu(j) \in M_j$) à cette ressource.
- Création d'une liste d'activités : selon le type de ressource attribué, nous établissons différents horaires en fonction des règles de priorité.

L'exemple suivant montre deux exemples de chromosomes liés au projet de la figure (4.2).

$$I^M = \begin{pmatrix} 2 & 4 & 1 & 6 & 3 & 5 \\ 2 & 2 & 1 & 1 & 1 & 1 \end{pmatrix} I^F \in \begin{pmatrix} 1 & 3 & 2 & 5 & 4 & 6 \\ 1 & 2 & 1 & 1 & 2 & 2 \end{pmatrix}$$

4.2.4 Opérateur Mix

En raison des caractéristiques particulières de la représentation du chromosome modèle, il n'est pas possible d'utiliser les opérateurs d'accouplement et de mutation standard dans l'algorithme génétique. Pour ce faire, nous décrivons l'étape de fusion suivante pour prendre en compte le concept de liste d'activités, la faisabilité des relations de prérequis et l'allocation de type. Pour l'accouplement, nous choisissons deux parents. Le chromosome mère $I^M = (\lambda^M, \mu^M)$ et le chromosome père $I^F = (\lambda^F, \mu^F)$.

$$I^M = \begin{pmatrix} j_1^M & \dots & j_J^M \\ \mu^M(j_1^M) & \dots & \mu^M(j_J^M) \end{pmatrix} \quad I^F = \begin{pmatrix} j_1^F & \dots & j_J^F \\ \mu^F(j_1^F) & \dots & \mu^F(j_J^F) \end{pmatrix}$$

Nous créons les quatre matrices PT1, PT2, PP1 et PP2 telles que le gène mère (positions) qui est dans la matrice PT1, le gène père qui est dans la matrice PT2, le type mère qui est dans la matrice PP1 et le type père qui est dans la Matrice PP2. Ainsi, deux membres, une fille $I^D = (\lambda^D, \mu^D)$ et l'autre fils $I^B = (\lambda^B, \mu^B)$, sont issus de l'union de deux parents. Nous créons d'abord une matrice constituée des types mère et père, soit la moitié des types de la mère (du début au milieu) et l'autre moitié du père (du bas à la bordure des types de la mère). Ensuite, nous considérons le chromosome I^D fille, qui est formé comme suit :

- Dans la liste des activités I^D , c'est-à-dire λ^D , les gènes (positions) $i = 1, \dots, p_{t1}$ sont transmis exactement de la mère. Cela signifie :

$$j_i^D := j_i^M$$

Dans la deuxième ligne du chromosome, nous mettons le type des ressources que nous avons créées auparavant. Le chromosome fils I^B est obtenu de la mère et du père de la même manière, sauf que la position $i = 1, \dots, q_1$ dans la liste des activités du chromosome fils λ^S est obtenue du père. De même, pour attribuer le type chromosomique masculin, on insère la matrice de type nouvellement créée.

Pour illustrer les définitions ci-dessus, considérons l'exemple suivant.

$$I^D = \begin{pmatrix} 2 & 4 & 1 & 6 & 3 & 5 \\ 2 & 2 & 1 & 1 & 2 & 2 \end{pmatrix} I^S = \begin{pmatrix} 1 & 3 & 2 & 5 & 4 & 6 \\ 2 & 2 & 1 & 1 & 2 & 2 \end{pmatrix}$$

4.2.5 Opérateur de mutation

En exécutant l'opérateur de mélange, une population mixte est créée, comprenant des parents et des enfants. La mutation modifie ensuite le contenu d'un gène (position) sur un chromosome choisi au hasard, ce qui augmente la diversité de la population et permet à l'algorithme de trouver les membres manquants. Les mutations empêchent les solutions optimales locales de devenir un problème pour l'algorithme. Ici, l'opérateur de mutation est appliqué séparément aux deux segments chromosomiques (liste d'activités et type de ressources allouées) avec une probabilité de $P_{\text{mut}} = 0,03$.

Dans le premier cas, pour la mutation dans la liste des activités λ , une activité (gène) est sélectionnée au hasard. Une position aléatoire est choisie parmi toutes les positions possibles pour représenter cette activité, entre la position la plus haute de ses prédécesseurs et la position la plus basse de ses successeurs. Pour clarifier le tout, considérons le chromosome fille. L'activité 4 est dans le deuxième gène (position) dans la liste. Selon le réseau d'activités de la figure (1.1), l'activité 2 ne constitue qu'un prédécesseur et l'activité 6 n'est qu'un successeur de cette activité. Ces activités sont répertoriées dans la liste des activités des chromosomes filles dans le premier et le sixième gènes (position). En conséquence, l'opérateur de mutation est tel que l'activité 4 peut sélectionner au hasard l'une des deuxième, troisième, quatrième et cinquième positions. Supposons que le cinquième gène (position) soit sélectionné au hasard. Le chromosome trouvé est illustré ci-dessous :

$$I^D = \begin{pmatrix} 2 & 6 & 1 & 4 & 3 & 5 \\ 2 & 1 & 1 & 2 & 2 & 3 \end{pmatrix}$$

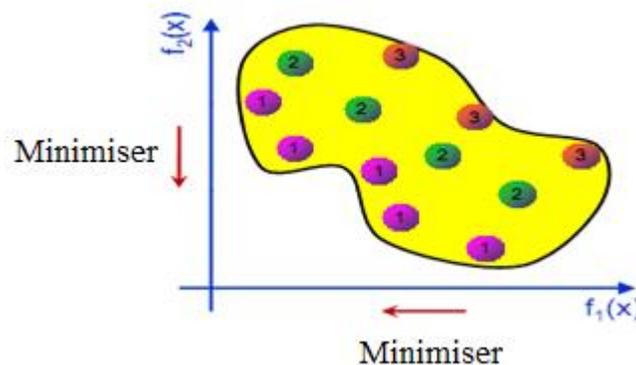
4.2.6 Trier les membres selon le concept de dominance

Dans l'algorithme NSGA-II, après la production de chromosomes descendants par fusion et mutation, la population parentale (P_t) est mélangée à la population descendante (Q_t) et une population mixte (R_t) est formée. Ensuite, pour trier les membres d'une population mixte sur la base du concept de dominance, les valeurs de la fonction objectif introduite doivent d'abord être effectuées pour chaque calcul. Le temps d'achèvement du projet, le coût total du projet et le nivellement des ressources sont les trois objectifs par lesquels l'éligibilité des membres est mesurée.

Les membres de la population mixte (R_t), qui résulte de la combinaison de la population des parents et des enfants, doivent être pondérés selon la compétence, ce qui a été expliqué en termes de deux fonctions objectives dans la section 3.4. Le tri et les différents fronts de Pareto sont déterminés selon la méthode de Goldberg.

Il est arrangé de telle sorte que les membres de la première catégorie forment un groupe qui n'a jamais été vaincu par les autres membres. Ce processus se poursuit de la même manière dans les autres catégories, de sorte que les membres de chaque catégorie reçoivent un classement basé sur le nombre de catégories auxquelles ils ont participé. Les membres de la deuxième catégorie ne sont battus par les membres de la première catégorie que sur cette base.

Figure 4.3 Classification par numéro de catégorie



Dans ce modèle, la méthode de tri rapide est utilisée pour trier les membres sur les fronts de Pareto. Pour chaque membre p de la population p , les étapes suivantes s'appliquent :

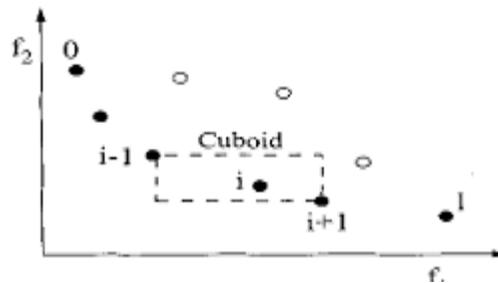
- ✓ $S(p) = [\emptyset]$: Stocke tous les membres dominés par le membre p .
- ✓ $n(p) = 0$: Cette variable stocke le nombre de membres qui dominent p .
- ✓ Pour chaque membre q de la population p :
 - Si p prévaut sur q ; q est ajouté à $s(p)$.
 - Si q prévaut sur p ; Une unité est ajoutée à $n(p)$. ($n(p)=n(p)+1$)
- ✓ Si $n(p) = 0$, cela signifie qu'aucun membre ne dépasse p . Donc, p appartient au premier front de Pareto et p est ajouté à $F(1)$ (F est l'ensemble de stockage des fronts de Pareto).
 - Les étapes ci-dessus s'appliquent à tous les membres de la population p .
 - La variable « i » est utilisée comme compteur de fronts de Pareto. « i » = 1
 - Le processus suivant continue jusqu'à ce que le front de Pareto ne soit pas vide. ($F(i) \neq [\emptyset]$)
- $Q = [\emptyset]$: Cet ensemble sert à agencer les membres des fronts de Pareto
- Pour chaque membre p sur le front $F(i)$:
 - Pour chaque membre q de l'ensemble $s(p)$. $s(p)$ est un ensemble qui stocke les membres par lesquels p est dominé.
 - 1- $n(q) = n(q) - 1$, le compteur diminue le nombre de membres dépassant Q d'une unité.
 - 2- $n(q) = 0$, alors aucun membre sur les fronts suivants ne surmonte q . Le membre q est ajouté à l'ensemble Q .
 - $I = i + 1$: Le compteur de fronts de Pareto augmente d'une unité.
 - Q est le prochain front de Pareto. $F_i = Q$

4.2.7 Détermination d'un paramètre de contrôle appelé distance de population

Ce paramètre est pris en compte pour chaque membre de chaque groupe et indique la proximité de l'échantillon avec les autres membres de cette population. Par conséquent,

tout point qui comporte une grande population couvre une plus grande zone, et le fait de l'omettre supprime la diversité de grands volumes de réponses. Les points qui ont une distance de population inférieure à l'ensemble de réponses de 3e ordre doivent être supprimés tant que la population initiale reste constante. De plus, les points de début et de fin liés à cet ensemble sont des points importants qui doivent être présents parmi les réponses et ne doivent pas être supprimés. La formule suivante peut également être définie pour chaque problème avec plusieurs fonctions objectifs.

Figure 4.4 Concept de distance de congestion pour le point i.



$$d_i^1 = \frac{|f_1^{i+1} - f_1^{i-1}|}{f_1^{\max} - f_1^{\min}}$$

$$d_i^2 = \frac{|f_2^{i+1} - f_2^{i-1}|}{f_2^{\max} - f_2^{\min}}$$

$$D = d_i^1 + d_i^2$$

Dans la formule ci-dessus, les valeurs de f_2^{\max} , f_2^{\min} , f_2^{i+1} , f_2^{i-1} , f_1^{\max} , f_1^{\min} , f_1^{i+1} , f_1^{i-1} sont spécifiées dans la figure et d_i^1 est relatif au point i et à l'ensemble de la fonction objectif, et d_i^2 est relatif au même point et à l'ensemble de la fonction objectif f_2 , et d est la somme de ces deux rapports, exprimant l'indice pour ce point appelé distance de congestion.

4.2.8 Sélection des membres de la prochaine génération

Après avoir déterminé les différents niveaux du Front de Pareto, les membres de la génération suivante (avec N membres) sont sélectionnés par les règles de l'algorithme NSGA-II parmi les membres de la population mixte (avec $2N$ membres). La transition commence par le meilleur front de F1 non vaincu. En ce qui concerne le dernier front autorisé, le nombre de membres existants peut dépasser le nombre de postes vacants dans la nouvelle génération. Dans ce modèle, les réponses des zones isolées sont privilégiées dans le processus de sélection. La base de sélection des membres de la population dans la méthode NSGA-II repose sur deux paramètres : le rang de dominance et la distance de surpeuplement. En utilisant l'opérateur de tournoi sélectionné pour sélectionner le meilleur membre du pool reproducteur, les deux membres sont sélectionnés au hasard dans l'ensemble p et q et comparés sur la base des éléments suivants :

- Comparer le rang de chacun de ces deux membres et sélectionner le membre ayant le rang le plus bas.
- Si les deux membres sont égaux, le critère de leur distance d'essaim est comparé.

4.3 LIEN ENTRE MATLAB ET LES LOGICIELS DE PLANIFICATION (MS PROJECT/PRIMAVERA)

Pour exécuter le modèle créé dans le logiciel Matlab, il faut entrer la liste des activités et leurs détails dans le fichier Excel, et après l'avoir appelé par le code donné initial (Annexe F du mémoire) dans le logiciel Matlab, soit le code NSGA-II (Annexe A du mémoire), nous exécutons le modèle. En fonction du nombre d'activités, des relations entre elles, de la liste des ressources et d'autres informations, après l'achèvement de l'algorithme NSGA-II, le diagramme Pareto Front est créé, (Figure 4.4 Exemple de diagramme de front de Pareto). Le sous-ensemble Workspace contient des informations sur l'algorithme et les réponses obtenues. L'option PF affiche les détails des points du graphique Pareto Front. Par exemple, nous sélectionnons le point Temps = 83 et Coût =

106 500. Il y a deux lignes d'informations sur ce point sur la page Position. La première ligne indique la priorité de l'activité, selon laquelle l'ordre des activités est (1 2 3 5 4 6 7), et la deuxième ligne indique le type de ressources requises pour les activités ; par exemple, pour l'activité numéro 1, il faut utiliser des ressources de type 3. La page Info indique le temps de début et de fin des activités. Pour assurer la performance du modèle et aussi pour préparer un planning dans les logiciels MS Project ou Primavera, il suffit de saisir les détails des activités au point sélectionné dans ces logiciels.

Figure 4.5 Exemple de Front de Pareto obtenu par NSGA-II

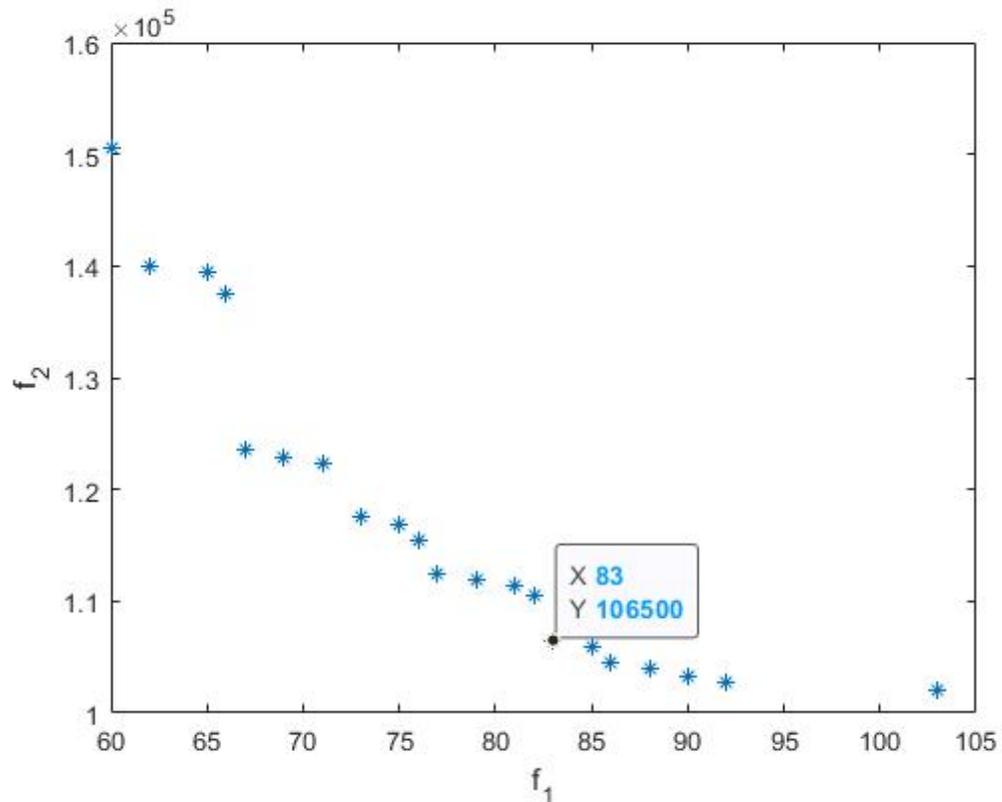


Figure 4.6 Détails des points du front de Pareto

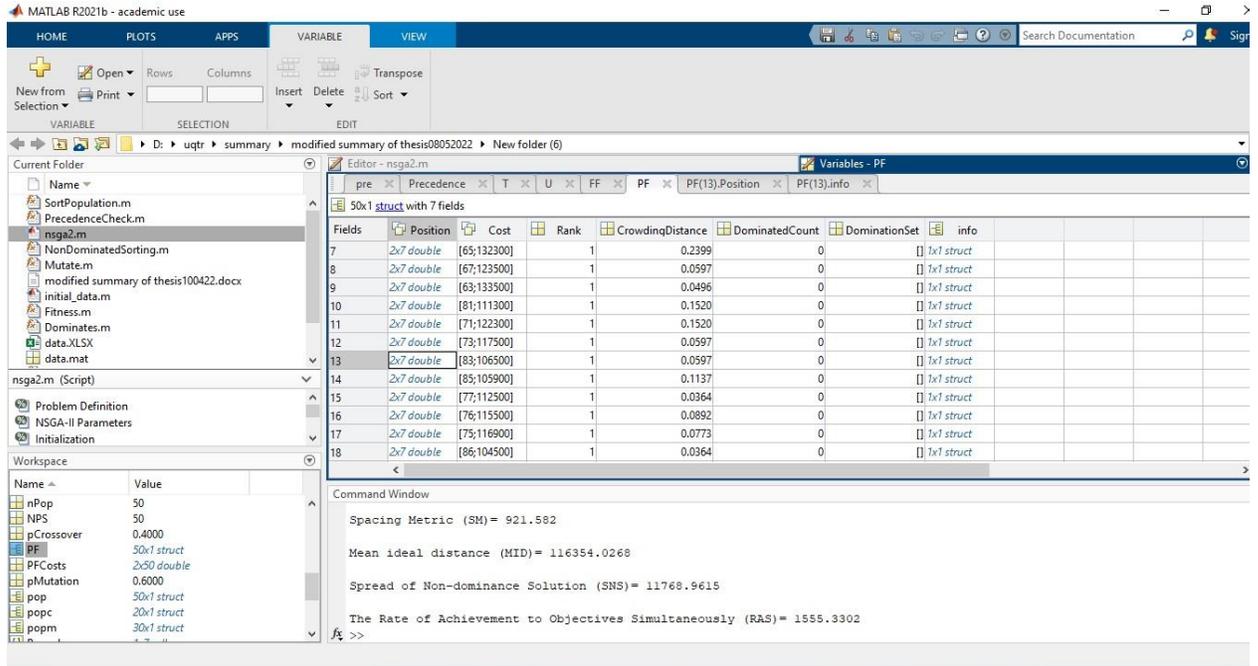


Figure 4.7 Tableau relatif à la priorité et au type de ressources requises pour les activités

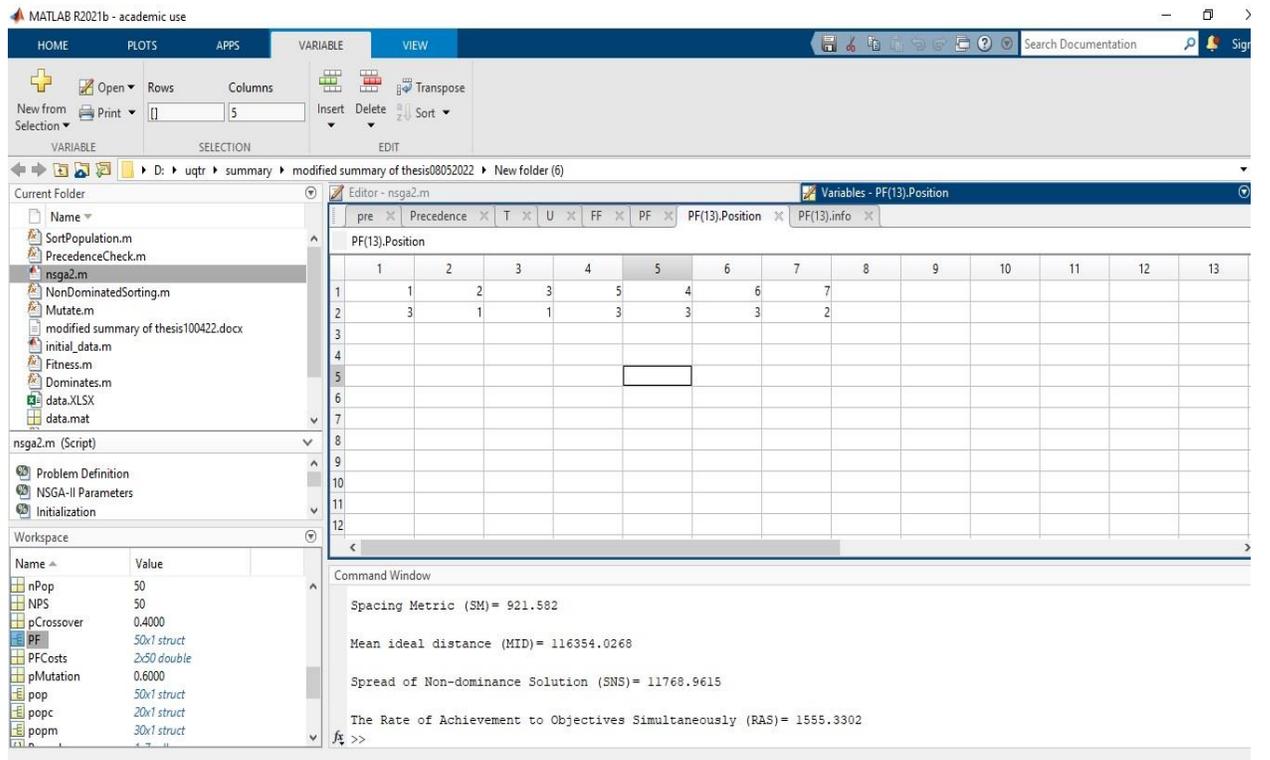
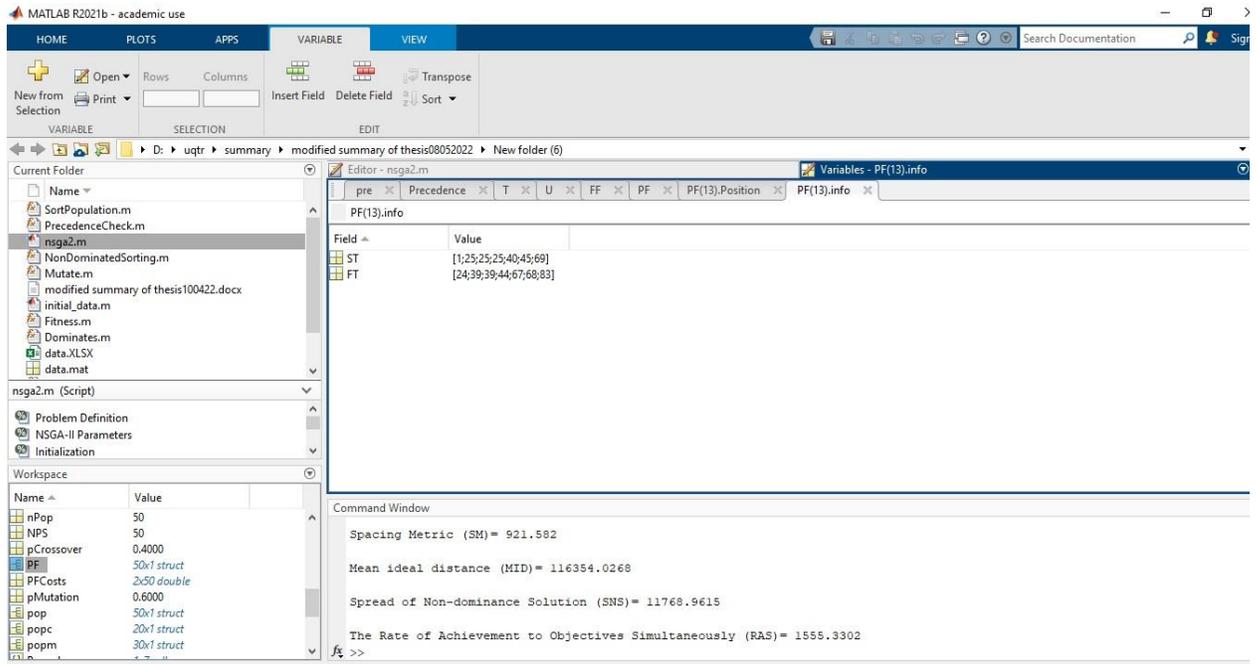


Figure 4.8 Tableau des dates de début et de fin des activités



4.4 RESUME

Ce chapitre présente les détails du modèle multi-objectifs, qui suivent les contraintes du problème (relations de prérequis...), ainsi que les objectifs qui (1) minimisent la durée du projet, (2) minimisent le coût total du projet suivi. De cette manière, différents points du bord de la pièce bidimensionnelle issus de l'exécution du modèle seront exposés, chacun appartenant à un réseau d'ordonnement différent. En sélectionnant chacune des réponses dominantes, nous aurons un calendrier particulier qui forcera l'exécution du projet selon les horaires et les types fixés pour chaque ressource. En raison des complexités inhérentes au type de problème à l'étude, des méthodes modifiées ont été utilisées pour générer la population initiale et pour coder et décoder le modèle innovant et le modèle proposé.

Le modèle proposé est différent des modèles précédents, c'est-à-dire qu'il considère à la fois des ressources multi-types et des ressources illimitées. Ce chapitre

présente l'algorithme génétique NSGA-II. L'algorithme génétique à objectif unique habituel a été amélioré avec deux opérateurs essentiels. Le nouvel algorithme multi-objectifs propose un ensemble de meilleures réponses, connu sous le nom de front de Pareto :

- Un opérateur qui attribue un critère de supériorité (rang) aux membres de la population en fonction de l'ordre dominant.
- Un opérateur qui garde une réponse variée parmi les réponses de même rang.

Étapes de l'algorithme génétique NSGA-II :

- Production de la population initiale
- Réalisation des opérations multi-objectifs (tri dominant, calcul de congestion, tri des réponses)
 - Mise en place la méthode de tri dominante
 - Calcul du paramètre de contrôle appelé distance de population (Crowding Distance)
 - Tri des réponses de haut en bas en fonction de la distance de foule et du front bas à haut
- Démarrage de l'anneau principal
- Production d'enfants par des méthodes de saut et de croisement
- Consolidation de la génération précédente et de la nouvelle génération
- Application des opérations multi-objectifs (tri dominant, calcul de congestion, tri des réponses)
- Choix du meilleur
- Réalisation des opérations polyvalentes (tri dominant, calcul d'encombrement, tri des réponses)
- Retour au début de la boucle principale jusqu'à ce que la condition d'arrêt soit établie

Dans le chapitre suivant, nous examinerons le modèle présenté dans ce chapitre en utilisant deux méthodes d'algorithme génétique.

CHAPITRE 5 : RESULTAT

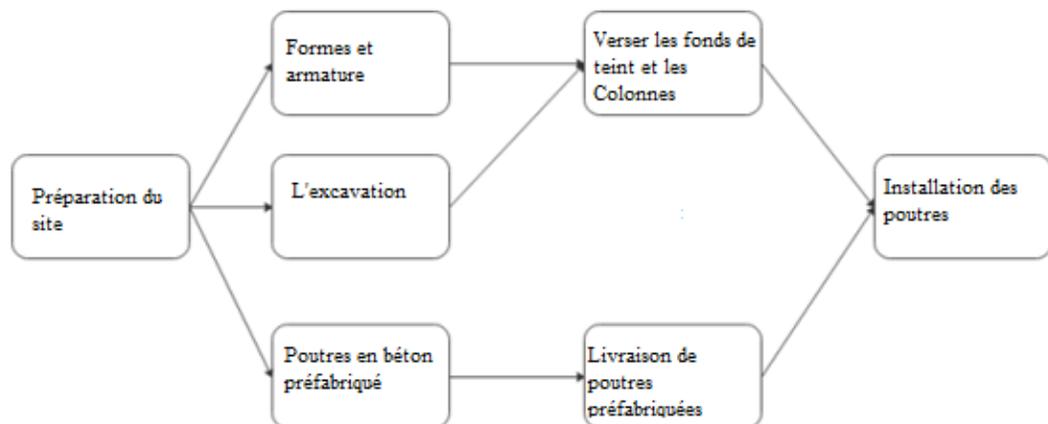
5.1 CONTEXTE

Dans le chapitre précédent, nous avons examiné la méthode méta-heuristique génétique, ou NSGA-II, pour résoudre le modèle mathématique proposé. Dans ce chapitre, nous résolvons d'abord le problème proposé en générant des modèles expérimentaux basés sur l'algorithme génétique proposé, et après avoir déterminé leurs fronts de Pareto par diverses méthodes, nous mesurons l'efficacité de l'algorithme NSGA-II et calculons les résultats de calcul. Les logiciels MATLAB 2021 et MSP2016 ont été utilisés pour concevoir les méthodes méta-heuristique utilisées.

L'application et les capacités de la méthode développée sont démontrées à l'aide d'un exemple numérique tiré de Zheng et Ng (2005). L'exemple numérique a été choisi parce qu'il couvre la majorité des variables utilisées dans cette étude et qu'il est largement utilisé dans les études d'ordonnancement de projets sous incertitude. Un exemple de réseau peut être trouvé dans la figure 5.1.

Figure 5.1 Exemple numérique de réseau

Adapté de (Zheng and Ng, 2005)



Selon le tableau 5.1, le projet se compose de sept activités, chacune d'entre elles ayant trois à cinq types de ressources à exécuter.

Tableau 5.1 : Exécution des activités (coût-durée) (Zheng and Ng, 2005)

Activité	Prédécesseur	Types de ressources	Durée	Coût
1. Préparation du site	-	1	14	23 000
		2	20	18 000
		3	24	12 000
2. Formes et armature	1	1	15	3000
		2	18	2400
		3	20	1800
		4	30	1200
		5	60	600
3. L'excavation	1	1	15	4500
		2	22	4000
		3	33	3200
4. Poutres en béton préfabriqué	1	1	12	45000
		2	16	35000
		3	20	30000
5. Verser les fonds de teint et les Colonnes	2,3	1	22	20000
		2	24	17500

Activité	Prédécesseur	Types de ressources	Durée	Coût
		3	28	15000
		4	30	10000
6. Livraison de poutres préfabriquées	4	1	14	40000
		2	18	32000
		3	24	18000
7. Installation des poutres	5,6	1	9	30000
		2	15	24000
		3	18	22000

La colonne Prédécesseur indique la relation entre les activités et les prédécesseurs, montrant en fait que chaque activité peut commencer après la fin de quelles activités. Par exemple, l'activité 'Préparation du site' n'a pas de prédécesseurs ou les prédécesseurs de l'activité 'Verser les fonds de teint et les Colonnes' sont les activités 'Formes et armature' et 'L'excavation'. En effet, l'activité 'Verser les fonds de teint et les Colonnes' ne peut débuter qu'une fois ces deux activités terminées. Les colonnes Types de ressources, Durée et Coût affichent le temps et le coût requis par chaque activité en fonction de chaque type de ressource. Par exemple, l'activité 'Préparation du site', si elle est effectuée avec le type de ressource 1, nécessite 14 jours et des coûts de 23 000,00 \$; si cette activité est effectuée avec le type de ressource 2, elle nécessite 20 jours et des coûts de 18 000, 00 \$; et si elle est effectuée avec le type de ressource 3, elle nécessite 24 jours et des coûts de 12 000,00 \$.

Afin de réaliser l'étude dans un environnement sous contrainte de ressources, certaines modifications ont été apportées aux paramètres du projet, car il existe peu de recherches sur l'ordonnancement sous contrainte de ressources en situation d'incertitude.

Voici ces modifications :

- 1- Sur la base de l'équation ci-dessous, le tableau 5.2 présente les ressources nécessaires pour chaque activité : Ressources nécessaires = Coût de l'activité / (durée de l'activité * coût des ressources/jour).

Tableau 5.2 : Exécution des activités (Ressources)

Activité	Types de ressources	Coût des ressources/jour	Coût	Ressources nécessaires
1. Préparation du site	1	400	23 000	4
	2	360	18 000	3
	3	320	12 000	2
2. Formes et armature	1	400	3000	1
	2	360	2400	1
	3	320	1800	1
	4	280	1200	1
	5	240	600	1
3. L'excavation	1	400	4500	1
	2	360	4000	1
	3	320	3200	1
4. Poutres en béton préfabriqué	1	400	45000	9
	2	360	35000	6
	3	320	30000	5
	1	400	20000	2

Activité	Types de ressources	Coût des ressources/jour	Coût	Ressources nécessaires
5. Verser les fonds de teint et les Colonnes	2	360	17500	2
	3	320	15000	2
	4	240	10000	1
6. Livraison de poutres préfabriquées	1	400	40000	7
	2	360	32000	5
	3	320	18000	3
7. Installation des poutres	1	400	30000	8
	2	360	24000	5
	3	320	22000	4

2- Ressources disponibles (ressource/jour) : 9 heures

Au total, il y a 7 activités dans le projet qui aboutissent à la formation d'un chromosome à 7 éléments. L'étude de cas est analysée afin de présenter des solutions Pareto-optimales pour une optimisation multi-objectifs du temps et du coût.

5.2 VALIDATION DE LA METHODE

Pour trouver le meilleur ensemble de solutions (front de Pareto), quatre analyses comprenant l'indice de préférence (PI), la distance idéale moyenne (MID), la propagation de la solution de non-dominance (SNS) et le taux de réalisation simultanée des objectifs (RAS) sont effectuées pour valider la taille de population, le taux de mutation et le taux de croisement proposés.

5.2.1 Analyser la sensibilité des algorithmes génétiques

Dans cette section, pour obtenir le meilleur front de Pareto dans l'algorithme NSGA-II pour résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types, nous devons comparer la taille de la population, le taux de croisement et le taux de mutation par 4 paramètres tels que PI, MID, SNS et RAS. Ensuite pour s'assurer de la performance de notre modèle, il faut exécuter quelques points de fronts de Pareto sur le logiciel MSP et vérifier le résultat obtenu sur MSP avec NSGA-II. Les objectifs de ce modèle sont de trouver une solution permettant d'optimiser la durée et le coût du projet sans surallocation des ressources.

Les paramètres de l'GA sont analysés en matière de sensibilité afin d'identifier ceux qui produisent des solutions de qualité. Dans les problèmes d'optimisation dont les paramètres de l'GA sont inconnus, le module de sensibilité dépendant du cas développé par les auteurs s'avère très utile. Dans notre étude de cas, les algorithmes peuvent converger de manière impulsive vers un optimum local étant donné la simplicité du problème.

En conséquence, les paramètres de l'algorithme génétique recommandés dans la littérature sont les suivants : taille de la population = 50, taux de crossover de 0,4 pour assurer une exploitation, une exploration et une convergence par étapes adéquates, taux de mutation de 0,9 pour réduire la convergence prématurée et introduire des matériaux génétiques utiles et la taille de la population. Pour chaque combinaison de probabilités de croisement (0,4 à 0,9), de probabilités de mutation (0,4 à 0,8) et de tailles de population (40, 50 et 70), des analyses de sensibilité sont effectuées. Pour tester la qualité des solutions, il a fallu modifier les paramètres de manière itérative.

5.2.1.1 Analyse de sensibilité de la taille de la population :

Pour analyser la sensibilité de nPop pour l'algorithme NSGA-II, nous utilisons les 4 paramètres suivants :

La distance idéale moyenne (MID) : La valeur de cet indice est égale à la distance des points de Pareto de l'algorithme étudié par rapport au point idéal. Dans ce

mémoire, les fonctions objectives du minimiseur sont considérées. Nous considérons que le point idéal est le minimum de chacune des fonctions objectives dans tous les algorithmes. L'indice MID peut être calculé par l'équation (5-1) :

$$MID = \frac{\sum_{i=1}^n C_i}{n}, C_i = \sqrt{(f_{1i})^2 + (f_{2i})^2} \quad (5-1)$$

La propagation de la solution de non-dominance (SNS) : La valeur de cet indice est égale à la distance de Pareto obtenue par chaque algorithme, et il est préférable que cet indice soit plus bas. L'indice SNS peut être calculé par l'équation (5,2) :

$$SNS = \sqrt{\frac{\sum_{i=1}^n (MID - C_i)^2}{n - 1}} \quad (5-2)$$

Le taux de réalisation simultanée des objectifs (RAS) : Cet indice est égal au degré d'équilibre dans l'optimisation des objectifs conflictuels. Il vaut mieux que cet indice soit plus bas. L'indice RAS peut être calculé par l'équation (5-3) :

$$RAS = \frac{\sum_{i=1}^n \left(\frac{f_{1i} - F_i}{F_i} \right) + \left(\frac{f_{2i} - F_i}{F_i} \right)}{n}, F_i = \min(f_{1i}, f_{2i}) \quad (5-3)$$

L'indice de préférence (PI) : Cet indice est égal au coût normalisé moyen et à la durée normalisée. Il vaut mieux que cet indice soit plus bas. L'indice PI peut être calculé par l'équation (5-4) :

$$PI = \frac{\text{Durée normalisée} + \text{coût normalisé}}{2} \quad (5-4)$$

Pour calculer ces deux valeurs, Coût Normalisé et Durée Normalisée, nous utilisons la formule de normalisation des données dans une méthode linéaire, dont la formule est égale à :

$$n_{ij} = \frac{x_{ij}}{\sum_1^m x_{ij}} \quad (5-5)$$

Tout d'abord, nous exécutons un exemple de projet avec une taille de population différente et supposons un taux de mutation fixe et un croisement comme recommandé dans la littérature (taux de croisement 0,4, taux de mutation 0,9). Le tableau 5.3 montre les résultats de l'analyse de sensibilité pour différentes tailles de population (40, 50, 70). Dans ce tableau, différents points des fronts de Pareto pour différents nPops peuvent être observés. En fait, ce tableau montre le temps de réalisation et le coût optimaux du projet pour différents nPops.

Tableau 5.3 : Analyse de sensibilité de la taille de la population

	Durée	Coût	Durée normalisée	Coût normalisé	PI
nPop=40	62	148 000	0,040 208	0,063 547	0,051 877
	103	102 000	0,066 796	0,043 796	0,055 296
	63	131 000	0,040 856	0,056 247	0,048 552
	90	103 300	0,058 366	0,044 354	0,051 36
	72	122 500	0,046 693	0,052 598	0,049 645
	67	123 500	0,043 45	0,053 027	0,048 239
	73	117 500	0,047 341	0,050 451	0,048 896
	83	106 500	0,053 826	0,045 728	0,049 777
	66	128 500	0,042 802	0,055 174	0,048 988
	77	112 500	0,049 935	0,048 304	0,049 12
	82	110 500	0,053 178	0,047 445	0,050 311
	65	130 400	0,042 153	0,055 99	0,049 071
	76	115 500	0,049 287	0,049 592	0,049 439

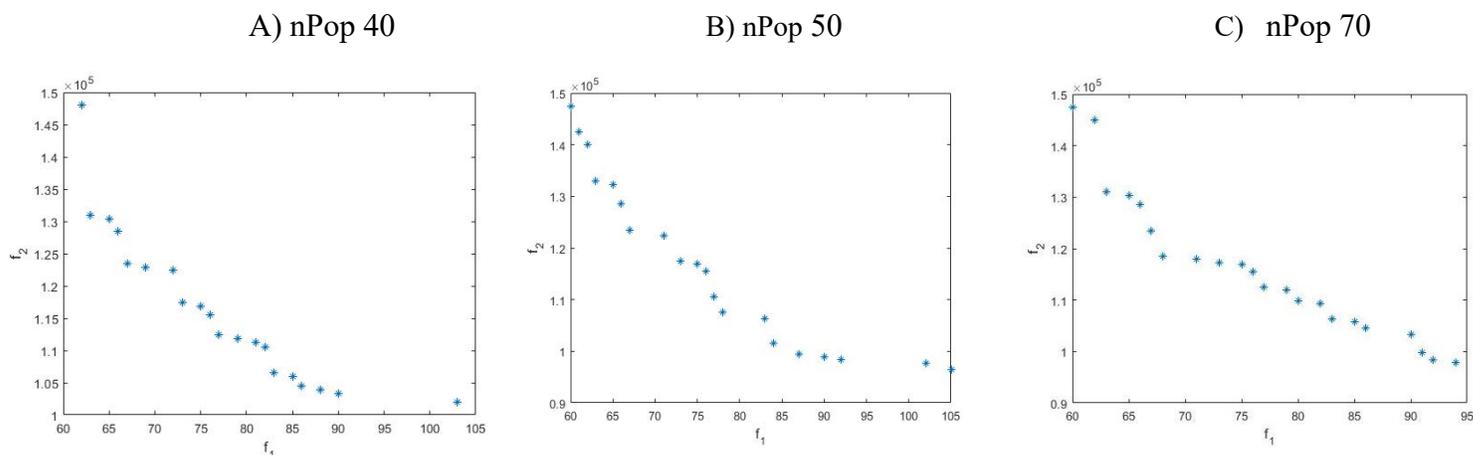
	Durée	Coût	Durée normalisée	Coût normalisé	PI
	69	122 900	0,044 747	0,052 769	0,048 758
	75	116 900	0,048 638	0,050 193	0,049 416
	85	105 900	0,055 123	0,045 47	0,050 297
	86	104 500	0,055 772	0,044 869	0,050 32
	81	111 300	0,052 529	0,047 789	0,050 159
	79	111 900	0,051 232	0,048 046	0,049 639
	88	103 900	0,057 069	0,044 611	0,050 84
	62	148 000	0,040 208	0,063 547	0,051 877
Recommandé nPop=50	94	97 800	0,055 786	0,038 332	0,047 059
	60	147 500	0,035 608	0,057 811	0,046 71
	62	145 000	0,036 795	0,056 832	0,046 813
	63	131 000	0,037 389	0,051 344	0,044 367
	68	118 500	0,040 356	0,046 445	0,043 401
	90	103 300	0,053 412	0,040 488	0,046 95
	86	104 500	0,051 039	0,040 958	0,045 998
	67	123 500	0,039 763	0,048 405	0,044 084
	66	128 500	0,039 169	0,050 365	0,044 767
	82	109 300	0,048 665	0,042 839	0,045 752
	83	106 300	0,049 258	0,041 663	0,045 461
	77	112 500	0,045 697	0,044 093	0,044 895
	71	117 900	0,042 136	0,046 21	0,044 173
	91	99 800	0,054 006	0,039 116	0,046 561
	79	111 900	0,046 884	0,043 858	0,045 371
	80	109 900	0,047 478	0,043 074	0,045 276
	65	130 400	0,038 576	0,051 109	0,044 842
	76	115 500	0,045 104	0,045 269	0,045 187
	92	98 300	0,054 599	0,038 528	0,046 564
	75	116 900	0,044 51	0,045 818	0,045 164
85	105 800	0,050 445	0,041 467	0,045 956	
73	117 300	0,043 323	0,045 975	0,044 649	

	Durée	Coût	Durée normalisée	Coût normalisé	PI
nPop=70	105	96 400	0,068 315	0,041 267	0,054 791
	60	147 500	0,039 037	0,063 142	0,051 09
	92	98 300	0,059 857	0,042 08	0,050 969
	71	122 300	0,046 194	0,052 354	0,049 274
	102	97 700	0,066 363	0,041 824	0,054 093
	62	140 000	0,040 338	0,059 932	0,050 135
	84	101 500	0,054 652	0,043 45	0,049 051
	78	107 500	0,050 748	0,046 019	0,048 384
	83	106 300	0,054 001	0,045 505	0,049 753
	67	123 500	0,043 591	0,052 868	0,048 23
	63	132 900	0,040 989	0,056 892	0,048 941
	65	132 300	0,042 29	0,056 635	0,049 463
	73	117 500	0,047 495	0,050 3	0,048 897
	66	128 500	0,042 941	0,055 009	0,048 975
	76	115 500	0,049 447	0,049 443	0,049 445
	61	142 500	0,039 688	0,061 002	0,050 345
	77	110 500	0,050 098	0,047 303	0,048 7
	87	99 500	0,056 604	0,042 594	0,049 599
	90	98 900	0,058 556	0,042 337	0,050 446
	75	116 900	0,048 796	0,050 043	0,049 42

Le but de l'analyse de sensibilité du nPop est d'obtenir la solution la plus optimale et donc de trouver le meilleur nPop pour l'algorithme NSGA-II afin de résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types. Comme nous pouvons le voir dans le tableau 5.3, parce que la huitième solution de nPop 50 a un PI 0,044084, une durée de projet de 67 jours et un coût de projet de 123 500,00 \$, elle a le taux de PI le plus bas parmi toutes les solutions disponibles dans différents Pops, donc si nous exécutons l'algorithme NSGA-II avec nPop 50, nous obtiendrons de meilleurs résultats que l'autre de nPop. Alors que la meilleure solution de nPop 40 et nPop 70 a une

durée de projet de 67 et un coût de projet de 123 500,00 \$, ils ont un PI plus élevé que nPop 50 (PI pour nPop 40 = 0,048 239 et pour nPop 70 = 0,048 23).

Figure 5.2 Front de Pareto de différentes nPop



Afin d'analyser plus précisément les fronts de Pareto dans les populations de taille 40, 50 et 70, nous utilisons les trois paramètres MID, SNS et RAS, dont la formule de calcul est décrite précédemment. Le tableau 5.4 montre le résultat de ces paramètres.

Tableau 5.4 Analyse du front de Pareto pour différentes tailles de pop.

nPop	MID	SNS	RAS
40	116 450	13 696	1593
50	116 166	13 222	1571
70	118 254	16 496	1628

Selon le tableau 5.4, parce que nPop 50 a le taux le plus bas de MID (116 166), SNS (13 222) et RAS (1571) parmi tous les différents nPop, c'est le meilleur nPop, alors que nPop 40 a MID (116 450), SNS (13 696) et RAS (1593) et nPop 70 a avec MID (118 254), SNS (16 496) et RAS=1628. En conséquence, en analysant les tableaux 5.3 et 5.4, nous concluons que le nPop 50 est le meilleur nPop pour l'algorithme NSGA-II pour

résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types.

5.2.1.2 Analyse de sensibilité au taux de mutation :

À l'étape précédente, nous avons pu trouver le meilleur nPop pour l'algorithme NSGA-II; à présent, nous voulons obtenir le meilleur taux de mutation en considérant avoir nPop 50 et en supposant un croisement de 0,40. Pour ce faire, nous exécutons l'exemple de projet à l'aide de l'algorithme NSGA-II et du logiciel MATLAB. Le tableau 5-5 montre les points optimaux. Ce tableau montre les différents points des fronts de Pareto pour différents taux de mutation. En fait, ce tableau montre les points qui comportent le temps et les coûts les plus optimaux pour le projet.

Tableau 5.5 : Analyse de sensibilité au taux de mutation

	Durée	Coût	Durée normalisée	Coût normalisé	PI
Taux de mutation =0.40	129	97 800	0,080 374	0,041 978	0,061 176
	60	150 500	0,037 383	0,064 598	0,050 99
	103	102 000	0,064 174	0,043 781	0,053 978
	63	131 000	0,039 252	0,056 228	0,047 74
	62	145 000	0,038 629	0,062 237	0,050 433
	100	104 000	0,062 305	0,044 639	0,053 472
	89	104 800	0,055 452	0,044 982	0,050 217
	81	111 300	0,050 467	0,047 772	0,049 12
	83	106 500	0,051 713	0,045 712	0,048 713
	66	128 500	0,041 121	0,055 155	0,048 138
	67	124 800	0,041 745	0,053 567	0,047 656
	71	122 300	0,044 237	0,052 494	0,048 365
	73	118 800	0,045 483	0,050 992	0,048 237
	75	116 900	0,046 729	0,050 176	0,048 452
	77	113 800	0,047 975	0,048 845	0,048 41
	79	111 900	0,049 221	0,048 03	0,048 626

	Durée	Coût	Durée normalisée	Coût normalisé	PI
	69	122 900	0,042 991	0,052 751	0,047 871
	87	105 300	0,054 206	0,045 197	0,049 701
	85	105 900	0,052 96	0,045 455	0,049 207
	86	105 800	0,053 583	0,045 412	0,049 497
Taux de mutation =0.60	60	147 500	0,038 192	0,063 847	0,051 02
	103	102 000	0,065 563	0,044 152	0,054 858
	62	140 000	0,039 465	0,060 601	0,050 033
	63	131 000	0,040 102	0,056 705	0,048 403
	92	102 800	0,058 561	0,044 498	0,051 53
	67	123 500	0,042 648	0,045 458	0,044 053
	71	122 300	0,045 194	0,052 939	0,049 067
	66	128 500	0,042 011	0,055 623	0,048 817
	73	117 500	0,046 467	0,050 861	0,048 664
	76	115 500	0,048 377	0,049 996	0,049 186
	79	111 900	0,050 286	0,048 437	0,049 362
	77	112 500	0,049 013	0,048 697	0,048 855
	83	107 800	0,052 833	0,046 663	0,049 748
	82	110 500	0,052 196	0,047 831	0,050 014
	85	107 300	0,054 106	0,046 446	0,050 276
	88	103 900	0,056 015	0,044 974	0,050 495
	81	111 300	0,051 56	0,048 178	0,049 869
	90	103 300	0,057 288	0,044 715	0,051 002
	86	105 800	0,054 742	0,045 797	0,050 27
	87	105 300	0,055 379	0,045 58	0,050 48
Taux de mutation recommandé =0.90	103	102 000	0,063 975	0,041 068	0,052 521
	60	155 500	0,037 267	0,062 608	0,049 938
	61	142 500	0,037 888	0,057 374	0,047 631
	92	102 800	0,057 143	0,041 39	0,049 266
	63	131 000	0,039 13	0,052 744	0,045 937
	62	140 000	0,038 509	0,056 368	0,047 438

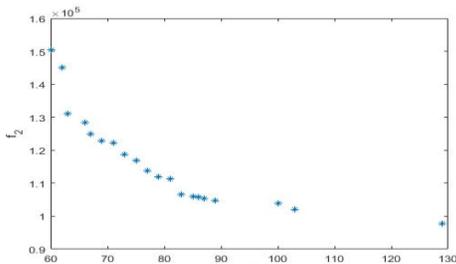
	Durée	Coût	Durée normalisée	Coût normalisé	PI
	67	123 500	0,041 615	0,049 724	0,045 67
	71	122 300	0,044 099	0,049 241	0,046 67
	73	117 500	0,045 342	0,047 308	0,046 325
	76	115 500	0,047 205	0,046 503	0,046 854
	66	128 500	0,040 994	0,051 737	0,046 366
	65	130 400	0,040 373	0,052 502	0,046 437
	77	112 500	0,047 826	0,045 295	0,046 561
	83	108 400	0,051 553	0,043 645	0,047 599
	85	105 900	0,052 795	0,042 638	0,047 717
	86	104 500	0,053 416	0,042 074	0,047 745
	82	110 500	0,050 932	0,044 49	0,047 711
	81	111 300	0,050 311	0,044 812	0,047 561
	88	103 900	0,054 658	0,041 833	0,048 246
	79	111 900	0,049 068	0,045 054	0,047 061
	90	103 300	0,055 901	0,041 591	0,048 746

Le but de l'analyse de sensibilité du taux de mutation est d'obtenir la meilleure solution parmi les solutions disponibles et finalement de trouver le meilleur taux de mutation pour l'algorithme NSGA-II pour résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types, en considérant que nPop 50 est le meilleur nPop. Tout d'abord, nous examinons les taux de PI liés aux différents taux de mutation. Le tableau 5-5 montre que la sixième solution obtenue au taux de mutation de 0,6 est le point où la durée optimale du projet est de 67 jours et le coût optimal du projet est de 123 500,00 \$ par rapport aux autres solutions disponibles, à la PI le plus faible.

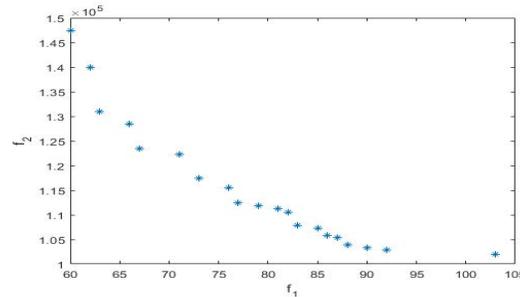
Pour ce qui est des taux de mutation de 0,4 et 0,90, les taux de PI sont respectivement de 0,047 656 et 0,045 67. Par conséquent, on peut conclure que si l'algorithme NSGA-II est mis en œuvre avec nPop 50 et Mutation Rate 0,60 et en supposant un Crossover 0,4, les meilleurs résultats peuvent être obtenus.

Figure 5.3 Front de Pareto de différents taux de mutation

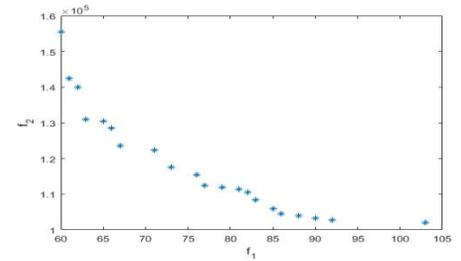
A) Taux de mutation 0.40



B) Taux de mutation 0.60



C) Taux de mutation 0.90



À cette étape, pour s'assurer que le taux de mutation 0,6 produit de meilleurs résultats que l'autre, nous examinons l'analyse de sensibilité du taux de mutation avec les 3 paramètres MID, SNS et RAS. Le tableau 5.6 montre les MID, SNS et RAS pour différents taux de mutation.

Tableau 5.6 Analyse du front de Pareto pour différents taux de mutation

Taux de mutation	MID	SNS	RAS
0.40	117 954	14 467	1576
0.60	115 166	12 517	1511
0.90	118 932	15 487	1619

Sur la base des informations du tableau 5.6, on peut constater que, comme l'analyse PI, le taux de mutation 0,6 a le MID (115 166), le SNS (12 517) et le RAS (1511) les plus bas par rapport au taux de mutation 0,4 qui a le MID (117 954), le SNS (14 467) et RAS (1576) et Mutation Rate 0.90 qui ont MID (118 932), SNS (15 487) et RAS (1619).

En conséquence, en analysant les tableaux 5.5 et 5.6, on peut conclure que si nous exécutons le problème d'optimisation du coût et du temps d'un projet de ressources multi-types par l'algorithme NSGA-II avec nPop 50 et taux de mutation 0,6 (en supposant un croisement de 0,40), de meilleurs résultats seront obtenus.

5.2.1.3 Analyse de sensibilité au taux de croisement :

Dans les deux étapes précédentes, nous avons calculé les meilleurs nPop et Mutation Rate pour l'algorithme NSGA-II afin de résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types (nPop 50 et Mutation Rate 0,60). À présent, nous voulons calculer le meilleur taux de croisement. À cette fin, nous examinons l'analyse de sensibilité au taux de croisement avec les 4 paramètres PI, MID, SNS et RAS. Le tableau 5.7 montre les points optimaux pour l'exemple de projet à différents taux de croisement (0,4, 0,60 et 0,80).

Tableau 5.7 : Analyse de sensibilité au taux de croisement

	Durée	Coût	Durée normalisée	Coût normalisé	PI
Croisement recommandé =0.40	60	150 500	0,048 701	0,080 905	0,064 803
	92	102 800	0,074 675	0,055 263	0,064 969
	63	132 900	0,051 136	0,071 444	0,061 29
	67	123 500	0,054 383	0,066 391	0,060 387
	81	111 300	0,065 747	0,059 832	0,062 79
	71	122 300	0,057 63	0,065 746	0,061 688
	65	130 400	0,052 76	0,070 1	0,061 43
	77	112 500	0,062 5	0,060 477	0,061 489
	83	106 500	0,067 37	0,057 252	0,062 311
	73	117 500	0,059 253	0,063 165	0,061 209
	88	103 900	0,071 429	0,055 854	0,063 641
	75	116 900	0,060 877	0,062 843	0,061 86
	76	115 500	0,061 688	0,062 09	0,061 889
	85	105 900	0,068 994	0,056 929	0,062 961
	86	104 500	0,069 805	0,056 177	0,062 991
	90	103 300	0,073 052	0,055 532	0,064 292
Croisement =0.60	103	102 000	0,063 463	0,041 521	0,052 492
	60	147 500	0,036 969	0,060 042	0,048 505

	Durée	Coût	Durée normalisée	Coût normalisé	PI
	92	102 800	0,056 685	0,041 846	0,049 266
	62	140 000	0,038 201	0,056 989	0,047 595
	67	123 500	0,041 282	0,050 273	0,045 777
	63	132 900	0,038 817	0,054 099	0,046 458
	61	142 500	0,037 585	0,058 007	0,047 796
	71	122 300	0,043 746	0,049 784	0,046 765
	73	117 500	0,044 978	0,047 83	0,046 404
	83	106 500	0,051 14	0,043 353	0,047 246
	75	116 900	0,046 211	0,047 586	0,046 898
	77	112 500	0,047 443	0,045 795	0,046 619
	82	110 500	0,050 524	0,044 981	0,047 752
	76	115 500	0,046 827	0,047 016	0,046 922
	85	105 900	0,052 372	0,043 108	0,047 74
	86	104 500	0,052 988	0,042 538	0,047 763
	81	111 300	0,049 908	0,045 307	0,047 607
	79	111 900	0,048 675	0,045 551	0,047 113
	88	103 900	0,054 221	0,042 294	0,048 257
	69	122 900	0,042 514	0,050 028	0,046 271
	90	103 300	0,055 453	0,042 05	0,048 751
Croisement =0.80	103	102 000	0,071 478	0,045 203	0,058 34
	60	147 500	0,041 638	0,065 367	0,053 502
	63	131 000	0,043 72	0,058 055	0,050 887
	92	102 800	0,063 845	0,045 557	0,054 701
	62	140 000	0,043 026	0,062 043	0,052 534
	66	128 500	0,045 802	0,056 947	0,051 374
	73	117 500	0,050 659	0,052 072	0,051 366
	83	106 500	0,057 599	0,047 197	0,052 398
	67	123 500	0,046 495	0,054 731	0,050 613
	71	122 300	0,049 271	0,054 199	0,051 735
	86	104 500	0,059 681	0,046 311	0,052 996

	Durée	Coût	Durée normalisée	Coût normalisé	PI
	76	115 500	0,052 741	0,051 185	0,051 963
	61	142 500	0,042 332	0,063 151	0,052 741
	77	112 500	0,053 435	0,049 856	0,051 646
	82	110 500	0,056 905	0,048 97	0,052 937
	90	103 300	0,062 457	0,045 779	0,054 118
	81	111 300	0,056 211	0,049 324	0,052 768
	69	122 900	0,047 883	0,054 465	0,051 174
	79	111 900	0,054 823	0,049 59	0,052 207

Étant donné que la 5e solution avec un taux de croisement de 0,60, avec un temps d'achèvement de projet de 67 jours et un coût de projet de 123 500,00 \$, a la durée normalisée et le coût normalisé moyens les plus bas de 0,045 777, la solution a le PI le plus bas parmi toutes les solutions disponibles. En conséquence, le taux de croisement 0,60 produit les meilleurs résultats par rapport aux autres. Pour leur part, les taux de PI les plus bas dans les deux autres croisements, 0,40 et 0,80, sont respectivement de 0,061 43 et 0,051 963.

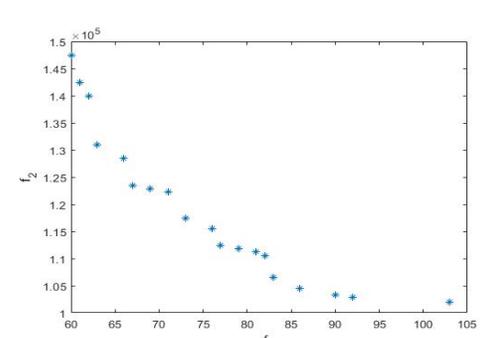
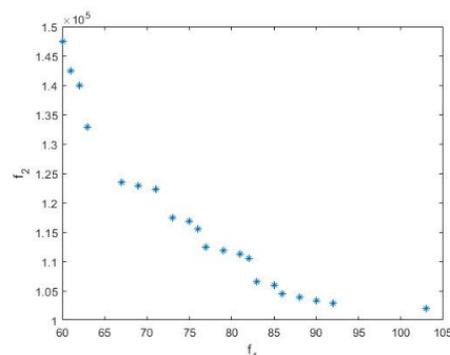
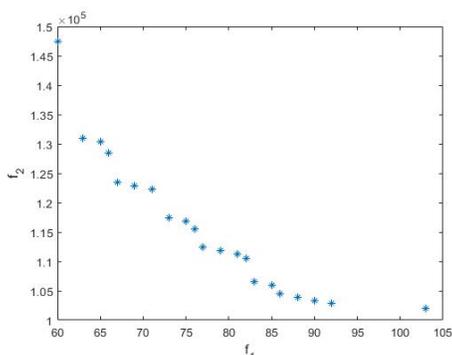
En analysant le taux PI dans le tableau 5.7, on peut conclure que si l'algorithme NSGA-II est mis en œuvre pour résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types avec un taux de croisement de 0,6, nous obtiendrons des résultats plus souhaitables.

Figure 5.4 Front de Pareto de différents taux de croisement

A) Croisement 0.40

B) Croisement 0.60

C) Croisement 0.80



Avec l'analyse PI, nous avons calculé le meilleur taux de croisement, mais pour assurer l'exactitude du résultat, nous vérifions l'analyse de sensibilité au taux de croisement cette fois avec trois paramètres MID, SNS et RAS. Le tableau 5.8 montre les valeurs de ces trois paramètres pour différents taux de croisement.

Tableau 5.8 Analyse du front de Pareto pour différents taux de croisement

Croisement	MID	SNS	RAS
0.40	117 952	13 529	1607
0.60	115 418	12 303	1522
0.80	119 092	14 425	1634

En examinant le tableau 5.8, nous pouvons constater que le taux de croisement de 0,60 présente les valeurs les plus faibles pour ces trois paramètres, à savoir MID 115418, SNS 12303 et RAS 1522, par rapport aux deux autres taux de croisement, donc le taux de croisement de 0,60 génère de meilleurs résultats pour notre algorithme NSGA-II. Pour sa part, le taux de croisement de 0,40 présente des valeurs de MID 117952, SNS 13529 et RAS 1607 et le taux de croisement de 0,80 présente des valeurs de MID 119092, SNS 14425 et RAS 1634.

En analysant différentes sensibilités, nous avons pu calculer les meilleurs nPop, taux de mutation et taux de croisement pour l'algorithme NSGA-II afin de résoudre le problème d'optimisation du coût et du temps d'un projet de ressources multi-types, les valeurs étant respectivement de 0,50, 0,60 et 0,60.

5.3 VERIFICATION DE L'EXECUTION DU MODELE DANS MATLAB PAR

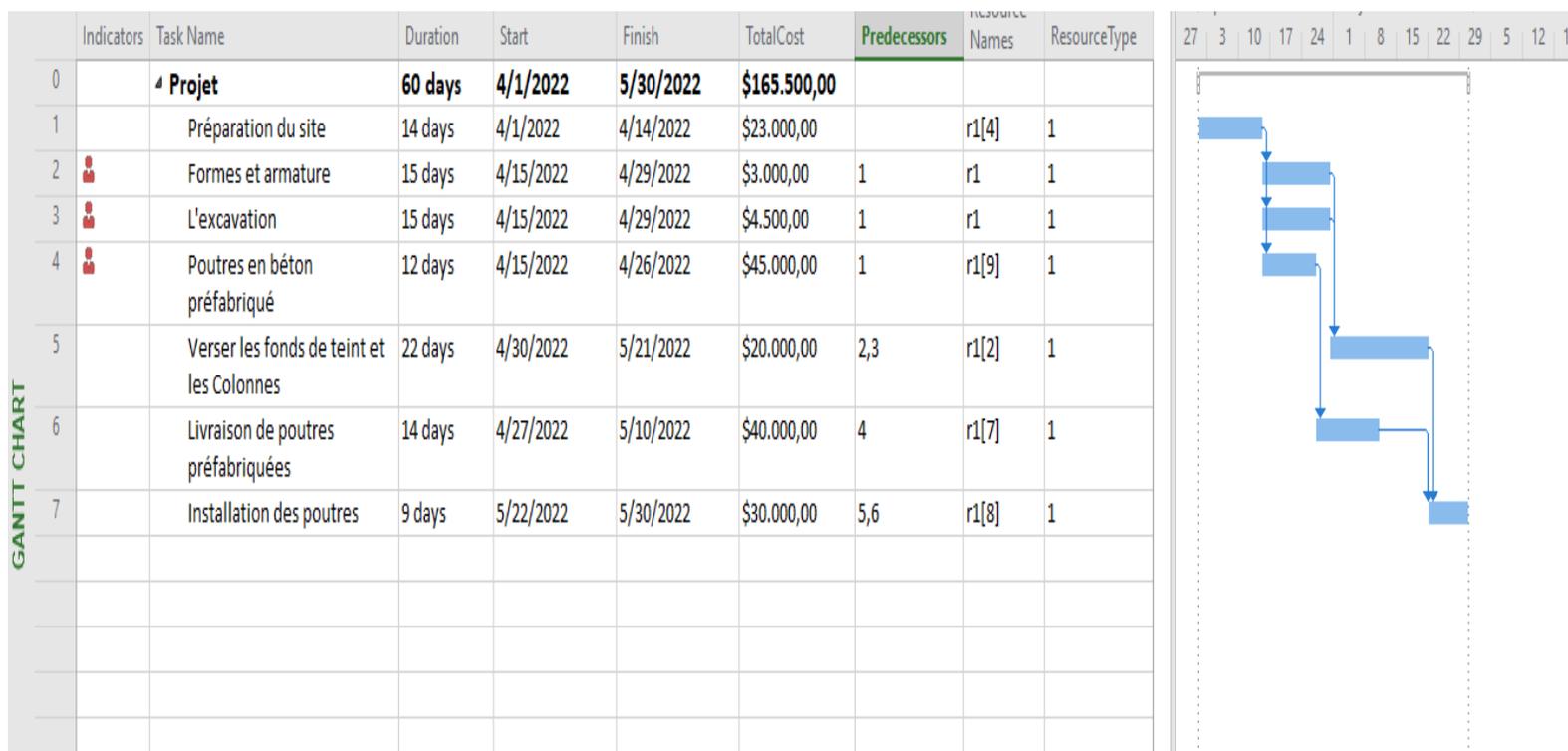
MSP :

Dans cette section, l'objectif est de s'assurer de la performance du modèle et des solutions et résultats obtenus. En fait, nous voulons nous assurer que les solutions obtenues à partir de l'algorithme NSGA-II respectent toutes les contraintes mentionnées dans le

modèle mathématique. Les limites les plus importantes du modèle qui doivent être observées sont, premièrement, le respect des relations entre les activités, deuxièmement, l'allocation des ressources aux activités en fonction du besoin des activités (pas plus et pas moins) et troisièmement, ne pas allouer aux activités des ressources supérieures aux ressources disponibles (pas de ressources surallouées).

Si ces contraintes ne sont pas respectées dans l'ordonnement du projet, celui-ci n'est pas fait correctement et n'est pas exécutable et doit être édité. Pour ce faire, nous utilisons le logiciel MSP, qui est l'un des logiciels les plus utiles dans la préparation de l'ordonnement de projet. Pour mieux comprendre les problèmes que pose l'ordonnement du projet en raison de la surallocation des ressources, nous ordonnons le projet type à l'aide du logiciel MSP en supposant que toutes les activités sont réalisées avec le type de ressource 1. La figure 5.5 présente l'ordonnement du projet.

Figure 5.5 Diagramme de Gantt du projet en supposant la sélection des ressources pour les activités avec le premier type de ressources avant de niveler les ressources.

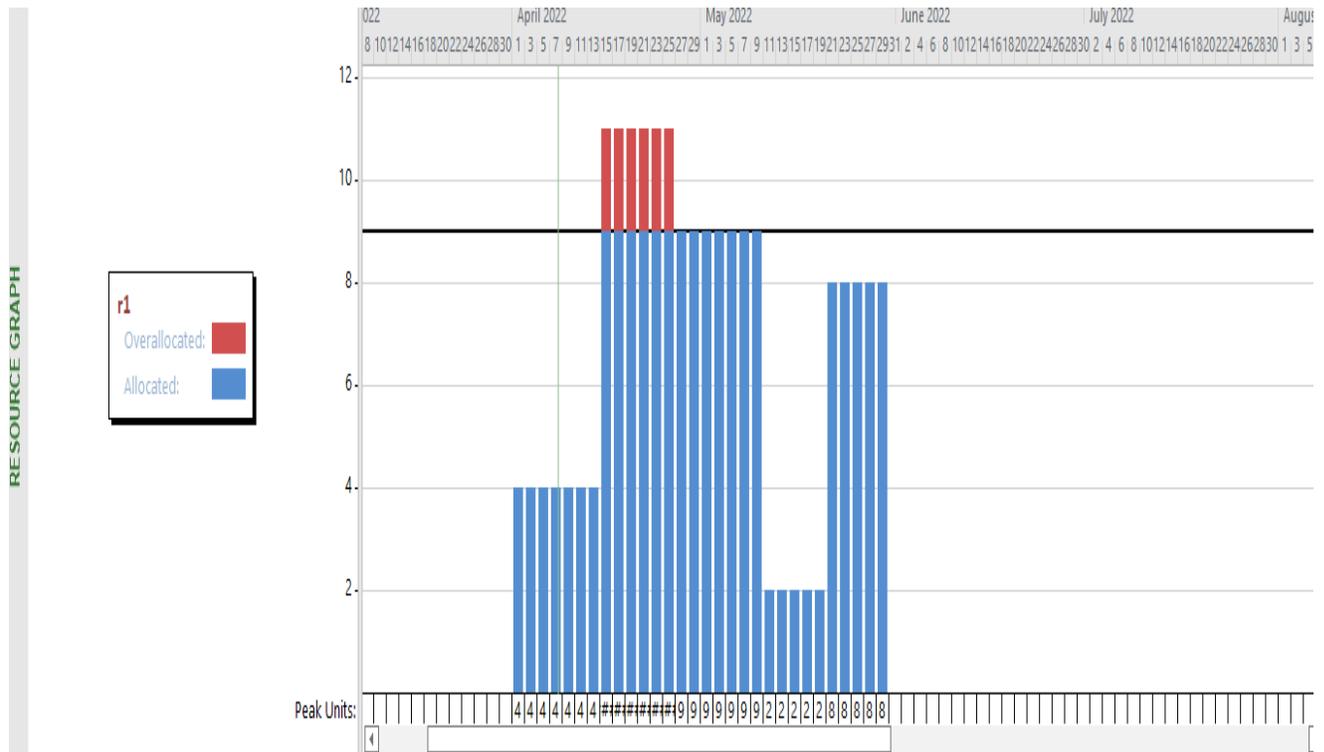


D'après la figure 5.5, le projet d'ordonnancement indique que le projet sera réalisé en 60 jours pour un coût de 165 500,00 \$. Nous voulons maintenant vérifier les contraintes du projet, car nous pouvons constater que la durée requise pour réaliser les activités dans l'ordonnancement (fichier MSP) est égale à la durée indiquée dans le tableau 5.1. Par exemple, l'activité 'Préparation du site' de l'ordonnancement nécessite 14 jours, ce qui est égal à l'information présentée dans le tableau 5.1.

Les relations entre les activités sont spécifiées dans la colonne Prédécesseur et indiquent que les relations entre les activités sont correctement suivies ; par exemple, l'activité 'Formes et armature' commence après la fin de l'activité 'Préparation du site'. La quantité de ressources requises pour chaque activité est également indiquée dans la colonne Noms des ressources. Par exemple, l'activité 'Préparation du site' nécessite 4 unités de ressources, ce qui correspond au tableau 5.1.

Étant donné que dans la colonne Indicateurs (la deuxième colonne de la figure 5.5), les activités 3 à 5 ont une ressource surallouée (le mannequin rouge dans cette colonne représente la ressource surallouée), l'ordonnancement s'avère problématique et non exécutable. Dans la figure 5.6 (histogramme d'allocation des ressources), les colonnes bleues représentent l'allocation des ressources et la colonne rouge représente la ressource surallouée. Les ressources du projet, en raison du problème de surallocation des ressources, ne peuvent pas terminer les activités 3-5 selon l'ordonnancement du projet et le projet ne sera pas terminé en 60 jours, donc ce calendrier n'est pas exécutable et les ressources doivent être nivelées.

Figure 5.6 Allocation des ressources du projet en supposant la sélection des ressources pour les activités avec le premier type de ressources avant de niveler les ressources.



Étant donné que ce projet dispose de ressources multi-types pour résoudre le problème de surallocation des ressources et obtenir le temps et le coût optimaux pour terminer le projet, nous ne pouvons pas utiliser le logiciel MSP pour niveler les ressources et devons utiliser des outils tels que l’algorithme NSGA-II.

Dans les étapes précédentes, nous avons trouvé des solutions pour résoudre le problème de l’optimisation du coût et du temps d’un projet de ressources multi-types en utilisant l’algorithme NSGA-II. Maintenant, nous devons nous assurer que les solutions obtenues par l’algorithme NSGA-II respectent toutes les contraintes du modèle mathématique. Pour ce faire, nous avons sélectionné les 3 solutions obtenues à partir du tableau 5.7 qui ont le PI le plus bas (une solution qui a le PI le plus bas de chaque croisement) et avons inscrit les informations d’activité obtenues dans le logiciel MSP et préparé l’ordonnancement du projet.

5.3.1 Projet d'ordonnancement basé sur les informations obtenues à partir de la 7^e solution de croisement 0,40 dans le tableau 5.7 :

Tout d'abord, nous choisissons la 7^e solution du croisement 0,4 avec une durée de 65 et un coût de 130 400,00 \$ qui a le PI le plus bas dans ce croisement pour nous assurer que toutes les solutions obtenues dans ce croisement respectent toutes les contraintes du modèle mathématique. Le tableau 5.9 présente les détails des activités de cette solution, notamment le type de ressources, la durée, la date de début des activités, la date de fin des activités et le coût des activités.

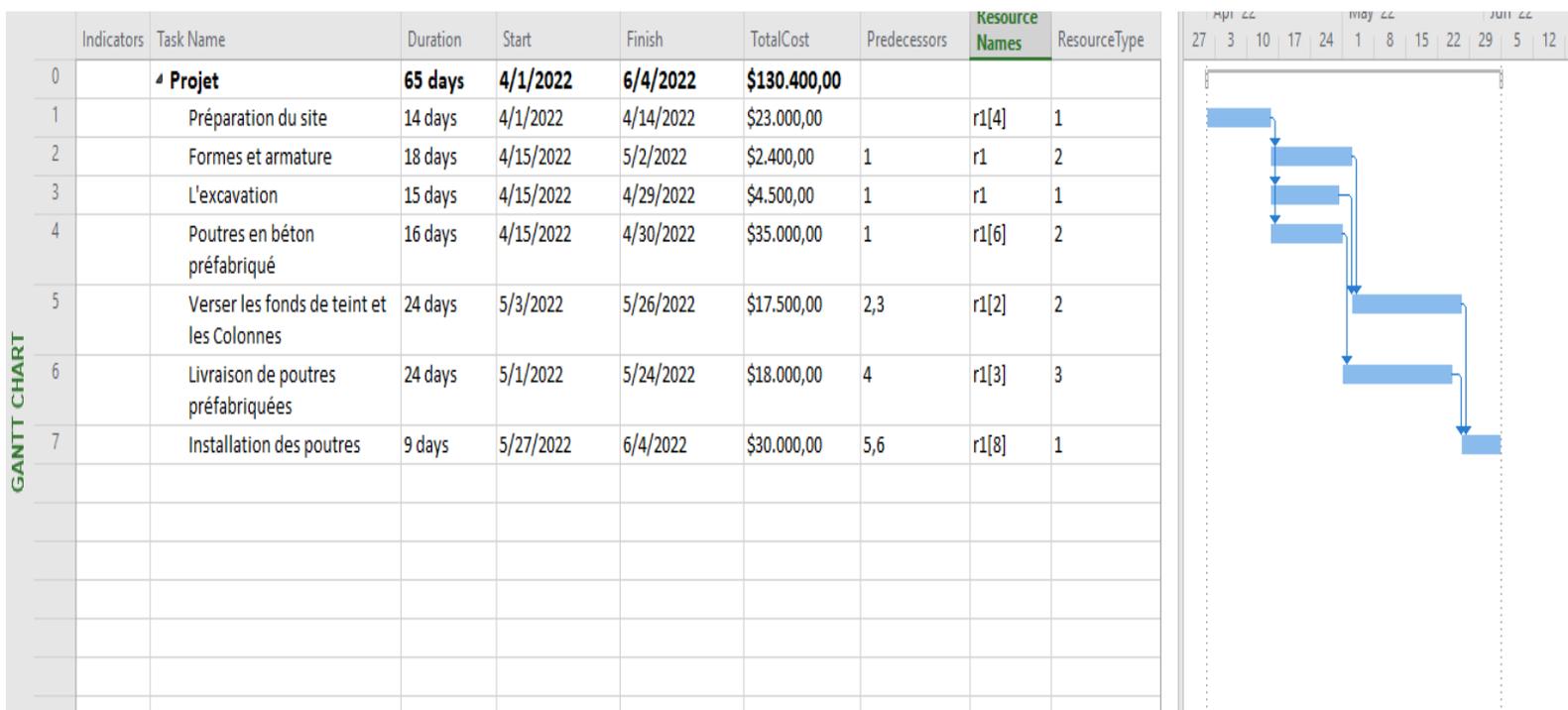
Tableau 5.9 détail des activités du projet pour la 7^e solution du croisement 0.4 du tableau 5.7

Identifiant d'activité	Nom de l'activité	Prec.	Type de ressource	Durée (jours)	début	Finir	Coût (\$)	Unité de ressource
1	Préparation du site	-	1	14	1	14	23 000	4
2	Formes et armature	1	2	18	15	32	2400	1
3	L'excavation	1	1	15	15	29	4500	1
4	Poutres en béton préfabriqué	1	2	16	15	30	35000	6
5	Verser les fonds de teint et les Colonnes	2,3	2	24	31	56	17500	2
6	Livraison de poutres préfabriquées	4	3	24	33	54	18000	3
7	Installation des poutres	5,6	1	9	57	65	30000	8
Durée totale du projet							65 jours	
Coût total du projet							\$ 130400.00	

Le tableau 5.9 présente les informations nécessaires à l'ordonnancement du projet. Par exemple, l'activité 'Préparation du site' avec le type de ressource 1, d'une durée de 14 jours et d'un coût de 23 000,00 \$, avec 4 unités de ressources par jour, commence le premier jour du projet et se termine le quatorzième jour.

Pour s'assurer que l'ordonnancement du projet dans cette solution respecte toutes les contraintes du modèle mathématique, nous utilisons le logiciel MSP pour ordonnancer le projet type sur la base des informations du tableau 5.9. La figure 5.7 montre le projet d'ordonnancement basé sur les informations du tableau 5.9.

Figure 5.7 Diagramme de Gantt du projet avec la 7^e solution de croisement 0.4 du tableau 5.7.



Selon le calendrier du projet (figure 5.7), le projet sera achevé en 65 jours à un coût de 130 400 \$, ce qui est conforme à la durée et au coût du projet basé sur la 7^e solution du croisement 0,4 présenté dans le tableau 5.9.

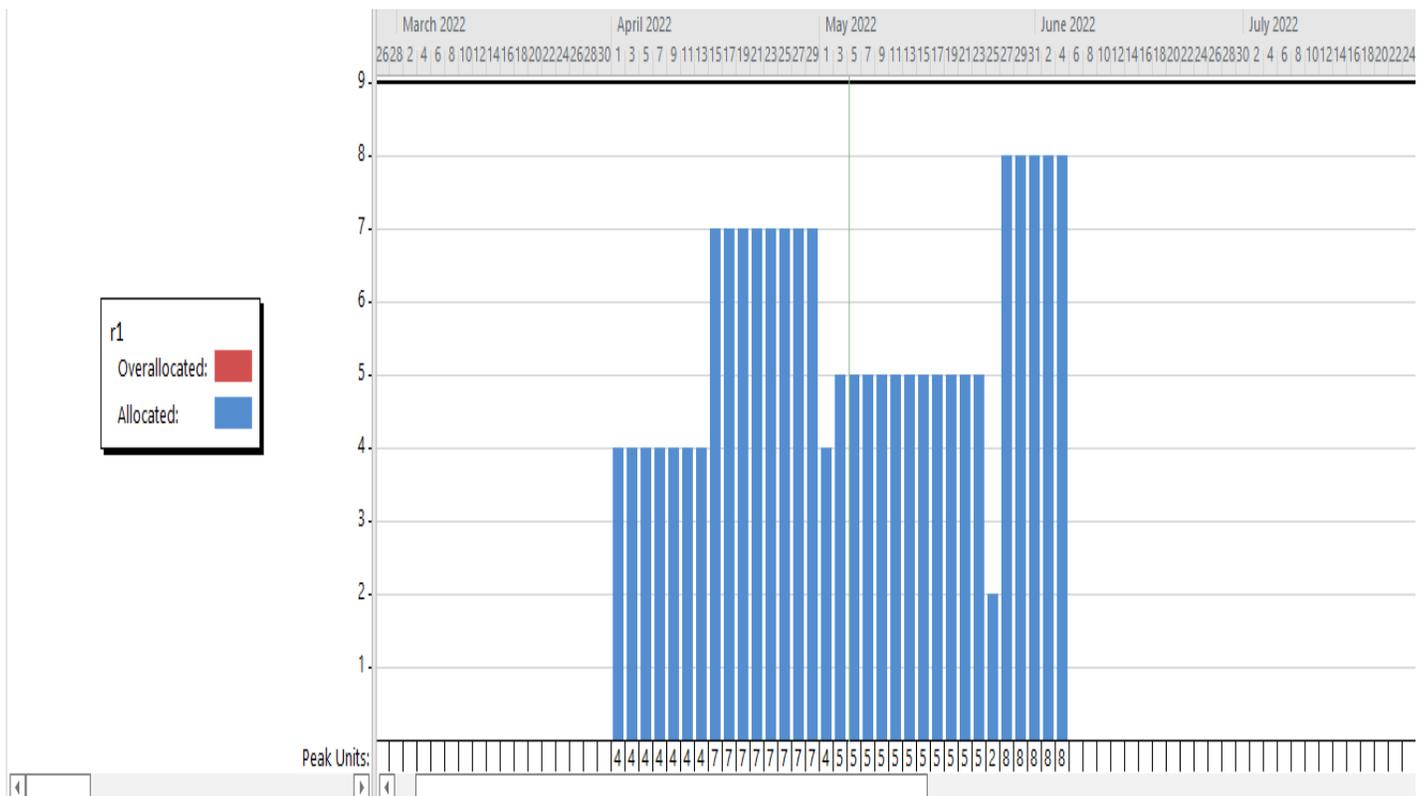
Pour que le calendrier du projet soit réalisable, il faut vérifier si toutes les exigences du modèle mathématique sont satisfaites ou non. Pour ce faire, il importe de comparer les informations sur les activités dans le calendrier du projet (Figure 5.7) avec le Tableau 5.9 comme le montre la figure 5.7.

La durée des activités dans la colonne Durée est égale à la durée des activités dans le tableau 5.9. Dans la colonne Prédécesseurs, les relations entre les activités sont

correctement suivies. Par exemple, l'activité 'Deliver Precast girder' commence une fois l'activité 'Precast Concrete girder' terminée. La quantité de ressources requises dans la colonne du nom de la ressource est égale à la quantité de ressources requises dans le tableau 5.9. Par exemple, 'Verser les fonds de teint et les Colonnes' nécessite 2 unités de ressources par jour, ce qui équivaut à la quantité de ressources requises dans le tableau 5.9.

Parce qu'aucune des activités de la colonne des indicateurs n'a de mannequin rouge et aussi, selon la figure 5.8 (histogramme d'allocation des ressources), parce que nous n'avons qu'une colonne bleue (colonne d'allocation des ressources) et aucune colonne rouge (indiquant des ressources sur-allouées), nous concluons que le calendrier créé à partir des informations obtenues de la septième solution de croisement 0,40 respecte toutes les contraintes du modèle mathématique et que le projet est applicable en 65 jours.

Figure 5.8 Allocation des ressources du projet avec la 7^e solution de crossover 0.4



5.3.2 Projet d'ordonnancement basé sur les informations obtenues à partir de la 5^e solution de croisement 0,60 dans le tableau 5.7 :

Dans les sections précédentes et en analysant divers paramètres tels que PI, MID, SNS et RAS, nous avons constaté que la meilleure solution disponible est la 5^e solution de croisement 0,60 dans le tableau 5.7 (une solution qui montre que le projet est terminé en 67 jours et coûte 12 3500,00 \$). Maintenant, nous voulons nous assurer que cette solution respecte toutes les contraintes du modèle mathématique. En fait, nous voulons vérifier si la meilleure solution créée par l'algorithme NSGA-II est exécutable ou non.

Pour assurer la performance de l'ordonnancement obtenu à partir de cette solution, il faut préparer l'ordonnancement du projet sur la base des informations sur les activités contenues dans cette solution, puis vérifier que l'ordonnancement respecte toutes les exigences du modèle mathématique ?

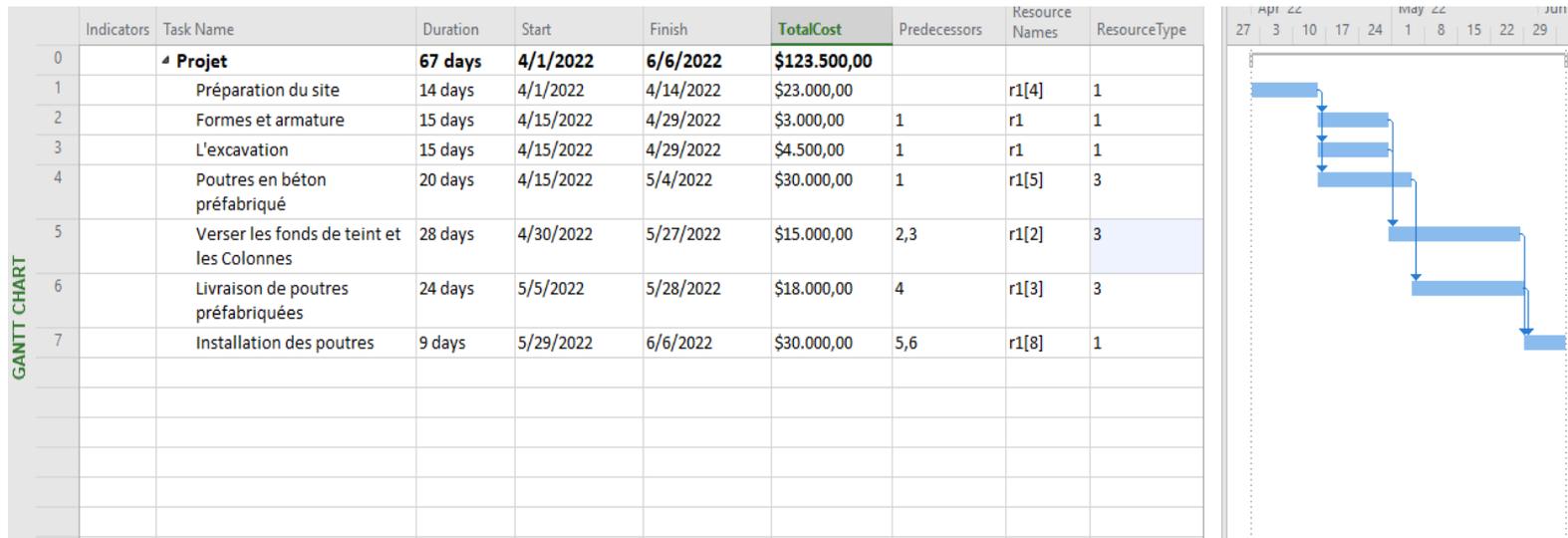
Tableau 5.10 détail des activités du projet pour la 5^e solution du croisement 0.6 du tableau 5.7

Identifiant d'activité	Nom de l'activité	Prec.	Type de ressource	Durée (jours)	début	Finir	Coût (\$)	Unité de ressource
1	Préparation du site	-	1	14	1	14	23 000	4
2	Formes et armature	1	1	15	15	29	3000	1
3	L'excavation	1	1	15	15	29	4500	1
4	Poutres en béton préfabriqué	1	3	20	15	34	30000	5
5	Verser les fonds de teint et les Colonnes	2,3	3	28	30	57	15000	2
6	Livraison de poutres préfabriquées	4	3	24	35	58	18000	3
7	Installation des poutres	5,6	1	9	59	67	30000	8
Durée totale du projet							67 Jours	
Coût total du projet							\$ 123500.00	

Les informations nécessaires à la préparation d'un projet d'ordonnement pour cette solution sont présentées dans le tableau 5.10. Selon les informations contenues dans le tableau ci-dessus, le projet sera réalisé en 67 jours et coûtera 12 3500,00 \$.

Pour préparer le projet d'ordonnement, nous disposons d'informations telles que les relations entre les activités, la durée de chaque activité et la quantité de ressources nécessaires. À présent, nous effectuons le projet d'ordonnement basé sur ces informations en utilisant le logiciel MSP.

Figure 5.9 Diagramme de Gantt du projet avec exécution de la 5^e solution du croisement 0,6



Maintenant, nous devons comparer le calendrier obtenu avec les informations du tableau 5.10. La première comparaison concerne la durée des activités, qui sont indiquées dans le logiciel MSP dans la colonne Durée. Cela montre que les activités ont la même durée que dans le tableau 5.10.

Ensuite, nous examinons les relations entre les activités. La colonne Prédécesseurs montre que les relations entre les activités dans le calendrier préparé sont conformes au tableau 5.10. ; par exemple, l'activité 'Installation des poutres' commence après 'Deliver Precast grider' et 'Pour Foundations et Piers' sont terminées.

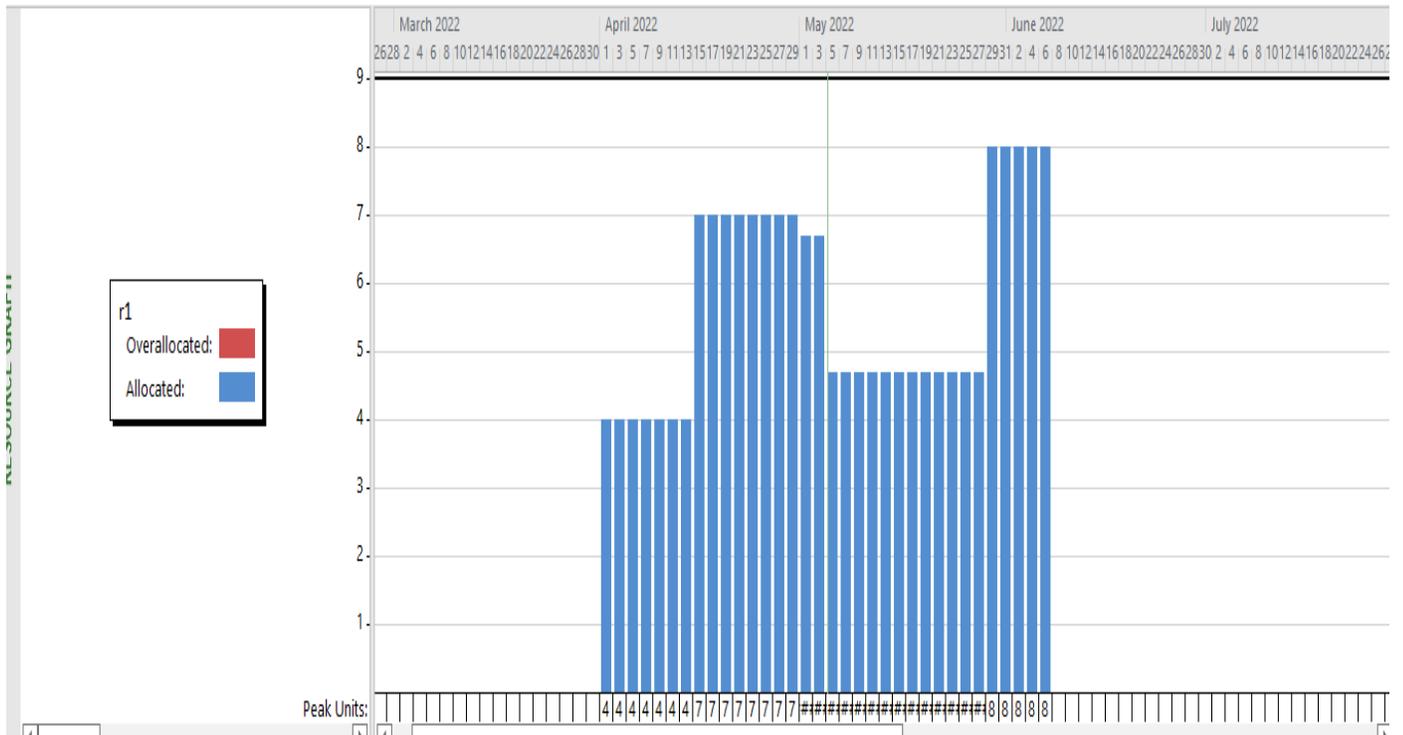
Par la suite, nous examinons si la quantité de ressources requises pour chaque activité est la même que dans le tableau 5.10. À partir de la colonne Nom de la ressource, nous pouvons conclure que les ressources requises dans le calendrier du projet sont les mêmes que dans le tableau 5.10.

Enfin, nous devons vérifier si les activités ont un problème de surallocation des ressources. Puisqu'aucune activité de la colonne des indicateurs n'a de mannequin rouge, et parce que la figure 5.10, qui montre l'historique d'allocation des ressources, ne comporte pas de colonne rouge, il n'y a pas de problème de surallocation des ressources.

Sur la base des comparaisons effectuées, nous pouvons conclure que le calendrier fourni dans cette solution ne présente pas de problème et que le projet peut être achevé en 67 jours sur la base des informations contenues dans cette solution.

Figure 5.10 Allocation des ressources du projet avec l'exécution de la 5^e solution de croisement

0,6



Dans les étapes précédentes, nous avons examiné deux des solutions obtenues à partir du tableau 5.7 (une solution du croisement 0,40 et une solution du croisement 0,60) pour vérifier les performances de l’algorithme NSGA-II.

Maintenant, nous voulons examiner une solution du croisement 0,8 qui comporte également le PI le plus bas et représente la meilleure solution de ce croisement, pour assurer la performance de toutes les solutions obtenues dans les trois croisements (0,40, 0,60 et 0,80). Pour ce faire, nous avons trouvé les informations d’activités pour la 12e solution dans Crossover 0.80, qui a un PI de 0,051 963, à partir de l’algorithme NSGA-II et les avons entrées dans le tableau 5.11.

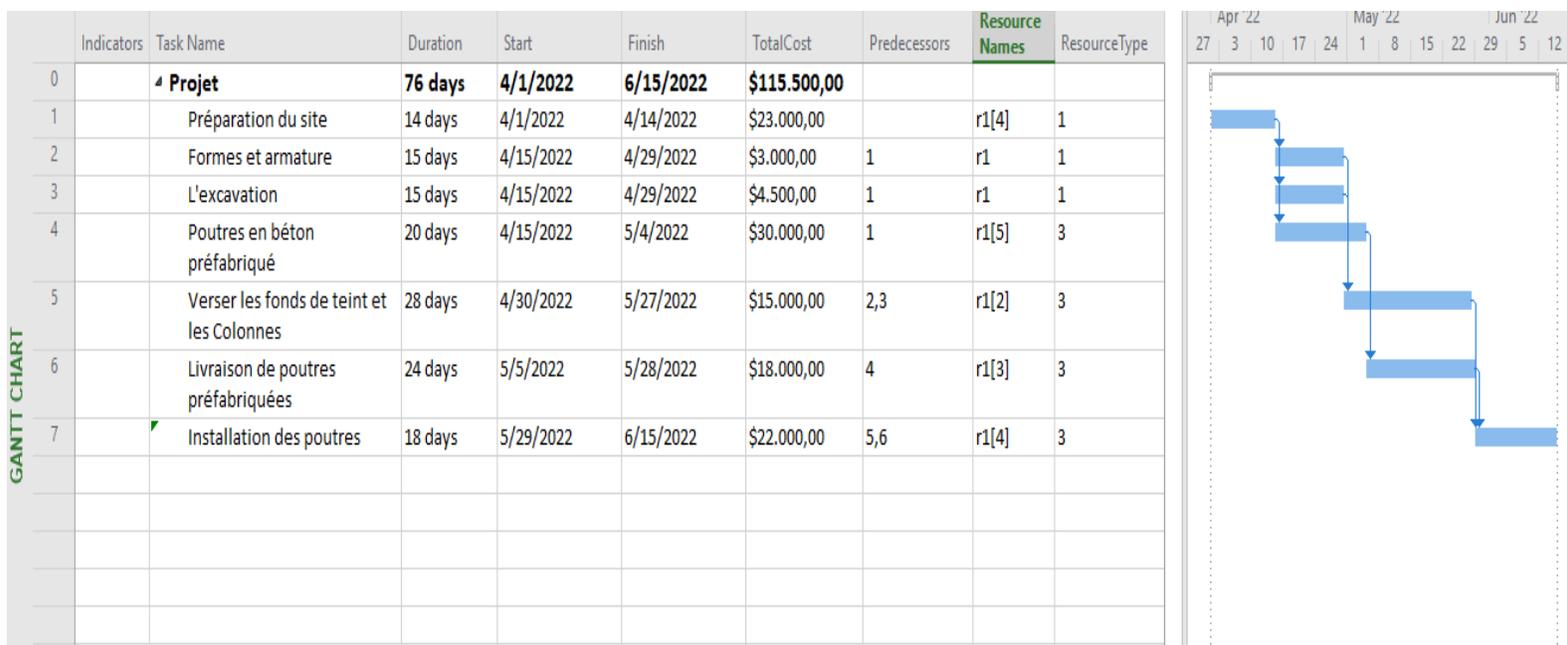
Tableau 5.11 Détail des activités du projet pour la 12^e solution du crossover 0.8

Identifiant d'activité	Nom de l'activité	Prec.	Type de ressource	Durée (jours)	début	Finir	Coût (\$)	Unité de ressource
1	Préparation du site	-	1	14	1	14	23 000	4
2	Formes et armature	1	1	15	15	29	3000	1
3	L'excavation	1	1	15	15	29	4500	1
4	Poutres en béton préfabriqué	1	3	20	15	34	30000	5
5	Verser les fonds de teint et les Colonnes	2,3	3	28	30	57	15000	2
6	Livraison de poutres préfabriquées	4	3	24	35	58	18000	3
7	Installation des poutres	5,6	3	18	39	76	22000	4
Durée totale du projet							76 Jours	
Coût total du projet							\$ 115500.00	

Le tableau 5.11 présente les informations nécessaires pour préparer un ordonnancement de projet basé sur la 12e solution de Crossover 0.80, telles que les relations entre les activités, la durée de chaque activité, les dates de début et de fin de chaque activité, le type de ressource pour chaque activité, et la quantité de ressources

nécessaires pour chaque activité. Nous pouvons maintenant préparer le calendrier du projet dans MSP en utilisant ces informations.

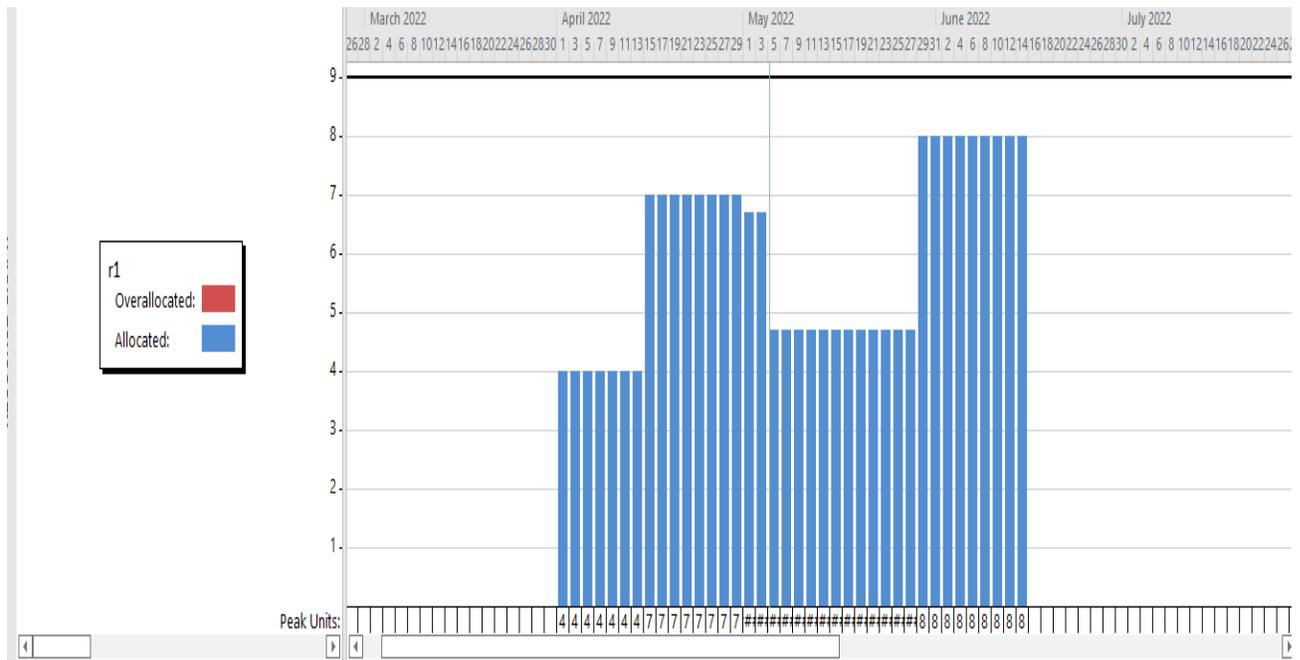
Figure 5.11 Diagramme de Gantt du projet avec la 12^e solution en cours d'exécution du croisement 0,8



Selon le calendrier préparé dans MSP (Figure 5.11), le projet sera achevé en 76 jours pour un coût de 115 500,00 \$, ce qui correspond à la durée totale et au coût total du tableau 5.11. Maintenant, pour s'assurer que le calendrier préparé respecte toutes les contraintes du modèle mathématique et est applicable, nous devons comparer les informations sur les activités dans le calendrier avec les informations sur les activités dans le tableau 5.11. Tout d'abord, nous voyons que la durée des activités dans le calendrier indiqué dans la colonne Durée est la même que celle du tableau 5.11. Deuxièmement, les relations entre les activités sont appliquées correctement, cela peut être vu dans la colonne des prédécesseurs du fichier MSP. Par exemple, les activités 'Formes et armature', « L'excavation » et « Poutres en béton préfabriqué » commencent après la fin de l'activité « Préparation du site ».

Nous pouvons également observer la quantité de ressources requises pour chaque activité dans la colonne du nom de la ressource, ce qui montre que les ressources requises pour chaque activité sont les mêmes dans le calendrier fourni dans le tableau 5.11. Enfin, parce qu’aucune activité de la colonne des indicateurs n’a de valeur fictive rouge et qu’il n’y a pas de colonne rouge dans l’histogramme d’allocation des ressources (figure 5.12) (la colonne rouge indique une ressource sur-allouée), nous concluons que le calendrier préparé a respecté toutes les contraintes du modèle mathématique et que le projet peut être achevé en 76 jours.

Figure 5.12 Allocation des ressources du projet avec la 12^e solution de croisement 0.8



5.4 RESUME :

En utilisant l’analyse des 4 paramètres PI, MID, SNS et RAS, nous avons pu obtenir les meilleurs nPop, taux de mutation et taux de croisement pour l’algorithme NSGA-II afin de résoudre notre modèle. Enfin, le meilleur ensemble de fronts de Pareto pour l’algorithme NSGA-II pour résoudre le modèle créé dans ce mémoire est nPop = 50,

taux de mutation = 0,6, et taux de croisement = 0,6. En outre, la meilleure solution pour le projet d'échantillon selon le tableau 5.7 est la 5e solution de Crossover 0,60, dans laquelle le projet est achevé en 67 jours et coûte 123 500,00 \$.

À l'étape suivante, pour s'assurer que les solutions obtenues à partir de l'algorithme NSGA-II ont respecté toutes les contraintes du modèle mathématique, dont les plus importantes sont les relations entre les activités, la quantité de ressources nécessaires pour chaque activité et ne pas avoir de surallocation Ressource, nous avons sélectionné trois des solutions obtenues à partir du tableau 5.7 qui ont le PI le plus bas (une solution de chaque croisement) et avons obtenu des informations sur l'activité pour ces solutions à partir de l'algorithme NSGA-II. En entrant les informations dans MSP, en préparant le calendrier du projet et en comparant le calendrier avec les contraintes du modèle mathématique, nous pouvons conclure que toutes les contraintes sont respectées par l'algorithme et que toutes les solutions sont exécutables par l'équipe du projet. Il suffit que le chef de projet choisisse la solution qu'il souhaite et ordonnance le projet en conséquence.

Les chefs de projet et les spécialistes de la planification et du contrôle des projets peuvent utiliser ce modèle pour préparer des projets d'ordonnancement dans le but d'optimiser le temps et le coût en tenant compte de la limitation des ressources et en utilisant des ressources multi-types. Pour ce faire, il suffit de sélectionner l'option la plus adaptée parmi les solutions obtenues sur le front de Pareto en fonction des impératifs de temps et de coût de leur projet, puis de saisir les informations sur les activités dans le logiciel MSP ou Primavera.

CHAPITRE 6 : CONCLUSION

6.1 SOMMAIRE :

Dans les projets de construction, la gestion du temps et des coûts s'avère très importante et la faiblesse de ce secteur a imposé des coûts élevés aux projets. Dans les systèmes de gestion et de planification, l'un des principes les plus importants et les plus fondamentaux des projets de construction qui doivent être abordés par les spécialistes de la gestion de projet est l'identification des activités de projet requises, la détermination de leur durée, l'identification et la définition des ressources (renouvelables et non renouvelables), l'étendue de leur disponibilité (en précisant la limitation des ressources) et en calculant le montant des ressources nécessaires pour mener à bien les activités.

Dans ce mémoire, l'objectif était d'optimiser la durée et le coût du projet en considérant la limitation de l'utilisation des ressources en mode multi-type de ressources. L'un des principaux problèmes des chefs de projet et des spécialistes de la planification et du contrôle de projet dans les projets à nombre élevé d'activités réside dans le nivellement des ressources du projet (non-surallocation des ressources du projet) étant donné que les logiciels de planification et de contrôle de projet tels que MSP et Primavera ne sont pas en mesure d'optimiser le temps et le coût ainsi que le niveau des ressources du projet en mode MTRCPSP. Par conséquent, nous avons essayé de résoudre ce problème en créant un modèle mathématique dans le troisième chapitre de ce mémoire tout en tenant compte des contraintes du projet telles que les relations entre les activités, l'utilisation limitée des ressources, etc., et ce, en implémentant le modèle avec le logiciel MATLAB et l'algorithme NSGA-II. Pour résoudre le modèle, l'algorithme génétique liste d'abord les activités et le type de ressource. Il se forme aléatoirement selon la loi de priorité (restriction des relations entre activités) (production de la population initiale).

Nous avons donc obtenu une liste d'activités et le type de ressources nécessaires pour les activités. À l'étape suivante, en effectuant des opérations multi-objectifs, en appliquant la méthode de tri défavorable, en calculant le paramètre de contrôle appelé distance de population et en triant les réponses de haut en bas en fonction de la distance d'encombrement et en fonction du front de haut en bas, nous avons formé les points du front de Pareto.

Dans le chapitre 5 de ce mémoire, afin d'obtenir la meilleure taille de population, le meilleur taux de mutation et le meilleur taux de croisement, nous avons comparé les résultats obtenus à partir de différentes solutions et avons finalement conclu que la meilleure solution est obtenue avec une taille de population de 50, le taux de mutation étant de 0,6 et Crossover de 0,6 (nous pouvons obtenir de meilleurs résultats avec ces paramètres).

Enfin, nous avons implémenté le projet type avec trois solutions obtenues à partir du tableau 5.7 dans le logiciel MSP afin de nous assurer de l'efficacité du modèle et des solutions obtenues. Nous avons ainsi observé : 1) Le nivellement des ressources du projet et l'ordonnancement du projet avec un temps et un coût optimal 2) Des résultats identiques obtenus par le logiciel MSP et l'algorithme NSGA-II, ce qui montre la bonne performance du modèle. En outre, le modèle peut être mis en œuvre dans des projets du monde réel.

6.2 CONTRIBUTIONS :

Créer un calendrier précis en tenant compte des contraintes, y compris les relations entre les activités, ainsi que limiter l'utilisation des ressources peut réduire le temps et les coûts des projets. Le modèle mathématique conçu dans ce mémoire a été implémenté à l'aide de l'algorithme métaheuristique NSGA-II et ses performances ont été étudiées et prouvées. Ce modèle peut aider les chefs de projet et les professionnels de la planification et du contrôle de projet à préparer un plan d'ordonnancement avec différents objectifs tels que l'optimisation du temps du projet, l'optimisation du coût du projet, l'allocation des ressources et le nivellement des ressources. En fait, en implémentant le modèle avec

l'algorithme NSGA-II, nous créons un front de Pareto des points de temps et de coût optimaux. À chaque point de ce schéma, les détails des activités tels que la date de début, la date de fin, la priorité des travaux, la quantité de ressources nécessaires, le type de ressources requises nous indiquent le coût de chaque activité. Il suffit de sélectionner l'un des points obtenus et de saisir les informations souhaitées dans des logiciels d'ordonnancement tels que MSP et Primavera.

Ce modèle présente des avantages par rapport aux modèles utilisés dans les recherches précédentes, qui sont également mentionnées dans la littérature de ce mémoire, soit des recherches sur l'optimisation du temps et des coûts sans tenir compte de la limitation des ressources, soit uniquement l'optimisation du temps du projet en tenant compte des contraintes de ressources (le coût du projet d'optimisation n'a pas été fait). À l'inverse dans ce mémoire, l'optimisation du temps et du coût du projet a été réalisée en considérant les ressources limitées et aussi en considérant multi types de ressources.

Ces avantages utilisés dans ce mémoire rapprochent notre modèle des projets du monde réel et montrent son applicabilité.

En mettant en œuvre ce modèle dans les projets, il est possible pour les spécialistes de la planification et du contrôle de projet d'obtenir les coûts et le temps les plus optimaux pour les projets (points de Pareto) dans les plus brefs délais.

En général, les contributions apportées par ce mémoire aux projets sont les suivantes :

Premièrement, le développement d'un modèle mathématique d'optimisation multi-objectifs de la durée et du coût du projet en tenant compte des contraintes de ressources en mode MTRCPSP. Deuxièmement, la création des fronts de Pareto pour la durée et le coût du projet qui permet aux gestionnaires de projet et aux spécialistes de la planification et du contrôle de projet de choisir les solutions optimales pour leur projet. Troisièmement, créer une méthode informatique permettant de résoudre le modèle en y consacrant le moins de temps possible.

6.3 APPLICATIONS DE LA RECHERCHE :

Les résultats de ce mémoire peuvent être utiles aux chefs de projet, en particulier dans les projets de construction, en raison de son applicabilité. Les sociétés de conseil, les employeurs et les entrepreneurs peuvent également bénéficier des résultats de la recherche. Les résultats de ce mémoire permettent aux chefs de projet et aux spécialistes de la planification et du contrôle de projet d'ordonner leurs projets avec une grande précision et dans les plus brefs délais avec des objectifs tels que l'optimisation du temps, l'optimisation des coûts, l'allocation des ressources et le nivellement des ressources.

6.4 LIMITES ET SUGGESTIONS POUR DES ETUDES FUTURES :

Dans ce mémoire, en utilisant le modèle mathématique et l'algorithme NSGA-II, nous avons optimisé le temps et le coût du projet en considérant la limitation des ressources et les ressources multi-types et nous avons obtenu des résultats intéressants. Il y a des limitations qui, si elles sont prises en compte, peuvent faire en sorte que le modèle mathématique soit plus proche des projets du monde réel :

- Tout d'abord, malgré le fait que l'objectif de ce mémoire était d'optimiser le temps et le coût du projet, dans les projets du monde réel, nous devons généralement composer avec des contraintes de temps et de coût, ce qui signifie que chaque projet a un temps et un coût acceptables, soit minimum soit maximum, et ne peut pas être inférieur ou supérieur à ceux-ci.
- Deuxièmement, le modèle développé dans ce mémoire ne peut être utilisé que pour optimiser le temps et le coût d'un projet, alors que les organisations peuvent utiliser leurs ressources dans plusieurs projets simultanément. En développant un modèle pour optimiser le temps et le coût de plusieurs projets, étant donné les ressources limitées et les ressources multi-types, le modèle peut être plus proche des projets du monde réel.
- Un troisième point qui peut aider à améliorer le modèle est le suivant : la qualité d'une activité réalisée par différents types de ressources peut varier. Par exemple,

si une activité est réalisée par un ouvrier soudeur, la qualité du travail peut être de 70 sur 100, mais si elle est réalisée par un ouvrier soudeur senior, elle peut être de 90 sur 100. Ce mémoire part du principe que les activités sont réalisées par différents types de ressources de même qualité. Dans des études futures, en ajoutant ce sujet au modèle présenté dans ce mémoire, le modèle pourrait être plus proche des projets du monde réel.

REFERENCES:

- Alcaraz, J., Maroto, C., & Ruiz, R. (2003). Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*, 54(6), 614-626.
- Baar, T., Brucker, P., & Knust, S. (1999). Tabu search algorithms and lower bounds for the resource-constrained project scheduling problem. In *Meta-heuristics* (pp. 1-18): Springer.
- Blazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete applied mathematics*, 5(1), 11-24.
- Boctor, F. F. (1993). Heuristics for scheduling projects with resource restrictions and several resource-duration modes. *The international journal of production research*, 31(11), 2547-2558.
- Boctor, F. F. (1996). A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. *European Journal of Operational Research*, 90(2), 349-361.
- Brooks, G. H. (1969). An algorithm for finding optimal or near optimal solutions to the production scheduling problem. *The Journal of Industrial Engineering*, 16(1), 34-40.
- Brucker, P., Knust, S., Schoo, A., & Thiele, O. (1998). A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 107(2), 272-288.
- Carlier, J., & Latapie, B. (1991). Une méthode arborescente pour résoudre les problèmes cumulatifs. *RAIRO-Operations Research*, 25(3), 311-340.
- Chiang, C.-W., Huang, Y.-Q., & Wang, W.-Y. (2008). Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling. *Journal of Intelligent & Fuzzy Systems*, 19(4-5), 345-358.
- Demeulemeester, E. (1995). Minimizing resource availability costs in time-limited project networks. *Management Science*, 41(10), 1590-1598.
- Demeulemeester, E. L., & Herroelen, W. S. (1996). An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem. *European Journal of Operational Research*, 90(2), 334-348.
- Elmaghraby, S. E. (1995). Activity nets: A guided tour through some recent developments. *European Journal of Operational Research*, 82(3), 383-408.
- Hartmann, S. (2001). Project scheduling with multiple modes: A genetic algorithm. *Annals of Operations Research*, 102(1), 111-135.

- Hartmann, S. (2002). A self-adapting genetic algorithm for project scheduling under resource constraints. *Naval Research Logistics (NRL)*, 49(5), 433-448.
- Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1), 1-14.
- Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research*, 25(4), 279-302.
- Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2), 289-306.
- Kadri, R. L., & Boctor, F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, 265(2), 454-462.
- Kelley, J. E. (1963). The critical-path method: resource planning and scheduling. *Industrial scheduling*.
- Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90(2), 320-333.
- Kolisch, R. (2000). Integrated scheduling, assembly area-and part-assignment for large-scale, make-to-order assemblies. *International Journal of Production Economics*, 64(1-3), 127-141.
- Kolisch, R., & Drexl, A. (1997). Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE transactions*, 29(11), 987-999.
- Krüger, D. (2009). *Multi-project scheduling with transfers*. Ph. D. thesis, University of Jena, Germany,
- Krüger, D., & Scholl, A. (2009). A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. *European Journal of Operational Research*, 197(2), 492-508.
- Maniezzo, V., & Mingozzi, A. (1999). The project scheduling problem with irregular starting time costs. *Operations Research Letters*, 25(4), 175-182.
- Messelis, T., & De Causmaecker, P. (2014). An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 233(3), 511-528.
- Mika, M., Waligora, G., & Węglarz, J. (2008). Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research*, 187(3), 1238-1250.

- Mingozi, A., Maniezzo, V., Ricciardelli, S., & Bianco, L. (1998). An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44(5), 714-729.
- Mori, M., & Tseng, C. C. (1997). A genetic algorithm for multi-mode resource constrained project scheduling problem. *European Journal of Operational Research*, 100(1), 134-141.
- Nonobe, K., & Ibaraki, T. (2002). Formulation and tabu search algorithm for the resource constrained project scheduling problem. In *Essays and surveys in metaheuristics* (pp. 557-588): Springer.
- Ozdamar, L. (1999). A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 29(1), 44-59.
- Patterson, J. H., Talbot, F. B., Slowinski, R., & Weglarz, J. (1990). Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. *European Journal of Operational Research*, 49(1), 68-79.
- Poppenborg, J., & Knust, S. (2016). A flow-based tabu search algorithm for the RCPSp with transfer times. *OR spectrum*, 38(2), 305-334.
- Pritsker, A. A. B., Waiters, L. J., & Wolfe, P. M. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1), 93-108.
- Slowinski, R. (1981). Multiobjective network scheduling with efficient use of renewable and nonrenewable resources. *European Journal of Operational Research*, 7(3), 265-273.
- Sprecher, A., Hartmann, S., & Drexl, A. (1997). An exact algorithm for project scheduling with multiple modes. *Operations-Research-Spektrum*, 19(3), 195-203.
- Talbot, F. B. (1981). An Algorithm for a general class of precedence and resource constrained scheduling problems.
- Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science*, 28(10), 1197-1210.
- Thomas, P. R., & Salhi, S. (1998). A tabu search approach for the resource constrained project scheduling problem. *Journal of Heuristics*, 4(2), 123-139.
- Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, 235(1), 62-72.

- Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes—A survey. *European Journal of Operational Research*, 208(3), 177-205.
- Zamani, R. (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *European Journal of Operational Research*, 229(2), 552-559.
- Zapata, J. C., Hodge, B. M., & Reklaitis, G. V. (2008). The multimode resource constrained multiproject scheduling problem: Alternative formulations. *AIChE Journal*, 54(8), 2101-2119.
- Zhang, H., Tam, C., & Li, H. (2006). Multimode project scheduling based on particle swarm optimization. *Computer-Aided Civil and Infrastructure Engineering*, 21(2), 93-103.
- Zhu, G., Bard, J. F., & Yu, G. (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, 18(3), 377-390.
- Błażewicz, J. (1986). Scheduling under resource constraints: Deterministic models, JC Baltzer.
- Guide, A. (2001). Project management body of knowledge (pmbok® guide). Project Management Institute.
- Kaming, P. F., P. O. Olomolaiye, G. D. Holt and F. C. Harris (1997). "Factors influencing construction time and cost overruns on high-rise projects in Indonesia." Construction Management & Economics 15(1): 83-94.
- Morris, P. W. and G. H. Hough (1987). "The anatomy of major projects: A study of the reality of project management."
- Raftery, J. (2003). Risk analysis in project management, Routledge.
- Abdelkhalek, H. A., H. S. Refaie and R. F. Aziz (2020). "Optimization of time and cost through learning curve analysis." Ain Shams Engineering Journal 11(4): 1069-1082.
- Al Haj, R. A. and S. M. El-Sayegh (2015). "Time–cost optimization model considering float-consumption impact." Journal of Construction Engineering and Management 141(5): 04015001.
- Ashtiani, B., R. Leus and M.-B. Aryanezhad (2011). "New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing." Journal of Scheduling 14(2): 157-171.
- Brucker, P. (2002). "Scheduling and constraint propagation." Discrete Applied Mathematics 123(1-3): 227-256.

- Bruni, M. E., P. Beraldi and F. Guerriero (2015). The stochastic resource-constrained project scheduling problem. Handbook on Project Management and Scheduling Vol. 2, Springer: 811-835.
- Bruni, M. E., P. Beraldi, F. Guerriero and E. Pinto (2011). "A heuristic approach for resource constrained project scheduling with uncertain activity durations." Computers & Operations Research **38**(9): 1305-1318.
- Chakraborty, R. K., R. Sarker and D. Essam (2015). Resource constrained project scheduling: A branch and cut approach. proceedings of the 45th international conference on computers and industrial engineering Metz, France.
- Chakraborty, R. K., R. A. Sarker and D. L. Essam (2017). "Resource constrained project scheduling with uncertain activity durations." Computers & Industrial Engineering **112**: 537-550.
- Chassiakos, A. P. and S. P. Sakellariopoulos (2005). "Time-cost optimization of construction projects with generalized activity constraints." Journal of Construction Engineering and Management **131**(10): 1115-1124.
- Drexl, A. and J. Gruenewald (1993). "Nonpreemptive multi-mode resource-constrained project scheduling." IIE transactions **25**(5): 74-81.
- Ezeldin, A. S. and A. Soliman (2009). "Hybrid time-cost optimization of nonserial repetitive construction projects." Journal of Construction Engineering and Management **135**(1): 42-55.
- Fang, C., R. Kolisch, L. Wang and C. Mu (2015). "An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem." Flexible Services and Manufacturing Journal **27**(4): 585-605.
- Feng, C.-W., L. Liu and S. A. Burns (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." Journal of computing in civil engineering **11**(3): 184-189.
- Hegazy, T. and N. Wassef (2001). "Cost optimization in projects with repetitive nonserial activities." Journal of Construction Engineering and Management **127**(3): 183-191.
- Kadri, R. L. and F. F. Boctor (2018). "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case." European Journal of Operational Research **265**(2): 454-462.
- Kalhor, E., M. Khanzadi, E. Eshtehardian and A. Afshar (2011). "Stochastic time-cost optimization using non-dominated archiving ant colony approach." Automation in Construction **20**(8): 1193-1203.
- Karam, A. and S. Lazarova-Molnar (2013). Recent trends in solving the deterministic resource constrained project scheduling problem. 2013 9th International Conference on Innovations in Information Technology (IIT), IEEE.

- Klanšek, U. and M. Pšunder (2010). "Cost optimization of time schedules for project management." Economic research-Ekonomska istraživanja **23**(4): 22-36.
- Kolisch, R. and S. Hartmann (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update." European journal of operational research **174**(1): 23-37.
- Leu, S.-S., A.-T. Chen and C.-H. Yang (1999). "Fuzzy optimal model for resource-constrained construction scheduling." Journal of Computing in Civil Engineering **13**(3): 207-216.
- Li, H. and P. Love (1997). "Using improved genetic algorithms to facilitate time-cost optimization." Journal of Construction Engineering and management **123**(3): 233-237.
- Li, H. and N. K. Womer (2015). "Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming." European Journal of Operational Research **246**(1): 20-33.
- Li, H., Z. Xu and W. Wei (2018). "Bi-objective scheduling optimization for discrete time/cost trade-off in projects." Sustainability **10**(8): 2802.
- Liu, J. (1998). "“Realization of time-cost optimization method on computer.” Optimization Capital Construction." Journal of construction engineering and management
- 133.**
- Liu, J. and F. Rahbar (2004). "Project time-cost trade-off optimization by maximal flow theory." Journal of construction engineering and management **130**(4): 607-609.
- Morillo-Torres, D., L. F. Moreno-Velásquez and F. J. Díaz-Serna (2015). "A branch and bound hybrid algorithm with four deterministic heuristics for the resource constrained project scheduling problem (RCPSp)." Dyna **82**(190): 198-207.
- Moselhi, O. and K. El-Rayes (1993). "Scheduling of repetitive projects with cost optimization." Journal of Construction Engineering and Management **119**(4): 681-697.
- Ng, S. T. and Y. Zhang (2008). "Optimizing construction time and cost using ant colony optimization approach." Journal of construction engineering and management **134**(9): 721-728.
- Podolski, M. and B. Sroka (2019). "Cost optimization of multiunit construction projects using linear programming and metaheuristic-based simulated annealing algorithm." Journal of civil engineering and management **25**(8): 848-857.

- Que, B. C. (2002). "Incorporating practicability into genetic algorithm-based time-cost optimization." Journal of construction engineering and management **128**(2): 139-143.
- Rogalska, M., W. Bożejko and Z. Hejducki (2008). "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling." Automation in Construction **18**(1): 24-31.
- Shahriari, M. (2016). "Multi-objective optimization of discrete time–cost tradeoff problem in project networks using non-dominated sorting genetic algorithm." Journal of Industrial Engineering International **12**(2): 159-169.
- Stork, F. (2001). "Stochastic resource-constrained project scheduling."
- Toğan, V. and M. A. Eirgash (2019). "Time-cost trade-off optimization of construction projects using teaching learning based optimization." KSCE Journal of Civil Engineering **23**(1): 10-20.
- Uysal, F., S. K. İşleyen and C. Çetinkaya (2018). "Resource constrained project scheduling with stochastic resources." Journal of applied research on industrial engineering **5**(1): 39-49.
- Xiong, Y. and Y. Kuang (2008). "Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off." Journal of construction engineering and management **134**(2): 153-156.
- Zhang, Y. and S. Thomas Ng (2012). "An ant colony system based decision support system for construction time-cost optimization." Journal of Civil Engineering and Management **18**(4): 580-589.
- Zheng, D. X. and S. T. Ng (2005). "Stochastic time–cost optimization model incorporating fuzzy sets theory and nonreplaceable front." Journal of Construction Engineering and Management **131**(2): 176-186.
- Zheng, D. X., S. T. Ng and M. M. Kumaraswamy (2004). "Applying a genetic algorithm-based multiobjective approach for time-cost optimization." Journal of Construction Engineering and management **130**(2): 168-176.
- Zheng, D. X., S. T. Ng and M. M. Kumaraswamy (2005). "Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization." Journal of Construction Engineering and Management **131**(1): 81-91.
- World Bank, 2012. Construction Sector Transparency Program Goes Global. [Online]
- Abdelkhalek, H. A., H. S. Refaie and R. F. Aziz (2020). "Optimization of time and cost through learning curve analysis." Ain Shams Engineering Journal **11**(4): 1069-1082.

- Al Haj, R. A. and S. M. El-Sayegh (2015). "Time–cost optimization model considering float-consumption impact." Journal of Construction Engineering and Management **141**(5): 04015001.
- Ashtiani, B., R. Leus and M.-B. Aryanezhad (2011). "New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing." Journal of Scheduling **14**(2): 157-171.
- Baar, T., P. Brucker and S. Knust (1999). Tabu search algorithms and lower bounds for the resource-constrained project scheduling problem. Meta-heuristics, Springer: 1-18.
- Bhosale, V., S. Shastri and M. Khandare (2017). "A review of genetic algorithm used for optimizing scheduling of resource constraint construction projects." International Research Journal of Engineering and Technology **4**(5): 2869-2872.
- Błażewicz, J. (1986). Scheduling under resource constraints: Deterministic models, JC Baltzer.
- Blażewicz, J., J. K. Lenstra and A. R. Kan (1983). "Scheduling subject to resource constraints: classification and complexity." Discrete applied mathematics **5**(1): 11-24.
- Boctor, F. F. (1993). "Heuristics for scheduling projects with resource restrictions and several resource-duration modes." The international journal of production research **31**(11): 2547-2558.
- Booker, L. B. (1993). "Recombination distributions for genetic algorithms." Foundations of Genetic Algorithms **2**: 29-44.
- Brucker, P. (2002). "Scheduling and constraint propagation." Discrete Applied Mathematics **123**(1-3): 227-256.
- Brucker, P., S. Knust, A. Schoo and O. Thiele (1998). "A branch and bound algorithm for the resource-constrained project scheduling problem." European journal of operational research **107**(2): 272-288.
- Bruni, M. E., P. Beraldi and F. Guerriero (2015). The stochastic resource-constrained project scheduling problem. Handbook on Project Management and Scheduling Vol. 2, Springer: 811-835.
- Bruni, M. E., P. Beraldi, F. Guerriero and E. Pinto (2011). "A heuristic approach for resource constrained project scheduling with uncertain activity durations." Computers & Operations Research **38**(9): 1305-1318.
- Carlier, J. and B. Latapie (1991). "Une méthode arborescente pour résoudre les problèmes cumulatifs." RAIRO-Operations Research **25**(3): 311-340.

- Chakraborty, R. K., R. Sarker and D. Essam (2015). Resource constrained project scheduling: A branch and cut approach. proceedings of the 45th international conference on computers and industrial engineering Metz, France.
- Chakraborty, R. K., R. A. Sarker and D. L. Essam (2017). "Resource constrained project scheduling with uncertain activity durations." Computers & Industrial Engineering **112**: 537-550.
- Chassiakos, A. P. and S. P. Sakellariopoulos (2005). "Time-cost optimization of construction projects with generalized activity constraints." Journal of Construction Engineering and Management **131**(10): 1115-1124.
- Chiang, C.-W., Y.-Q. Huang and W.-Y. Wang (2008). "Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling." Journal of Intelligent & Fuzzy Systems **19**(4-5): 345-358.
- Christodoulou, S. E. (2017). "Entropy-based heuristic for resource-constrained project scheduling." Journal of Computing in Civil Engineering **31**(3): 04016068.
- Demeulemeester, E. (1995). "Minimizing resource availability costs in time-limited project networks." Management Science **41**(10): 1590-1598.
- Demeulemeester, E. L. and W. S. Herroelen (1996). "An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem." European Journal of Operational Research **90**(2): 334-348.
- Drexl, A. and J. Gruenewald (1993). "Nonpreemptive multi-mode resource-constrained project scheduling." IIE transactions **25**(5): 74-81.
- Elmaghraby, S. E. (1995). "Activity nets: A guided tour through some recent developments." European journal of operational research **82**(3): 383-408.
- Ezeldin, A. S. and A. Soliman (2009). "Hybrid time-cost optimization of nonserial repetitive construction projects." Journal of Construction Engineering and Management **135**(1): 42-55.
- Fang, C., R. Kolisch, L. Wang and C. Mu (2015). "An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem." Flexible Services and Manufacturing Journal **27**(4): 585-605.
- Feng, C.-W., L. Liu and S. A. Burns (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." Journal of computing in civil engineering **11**(3): 184-189.
- Giran, O., R. Temur and G. Bekdaş (2017). "Resource constrained project scheduling by harmony search algorithm." KSCCE Journal of Civil Engineering **21**(2): 479-487.

- Goldberg, D. E. (2002). The design of innovation: Lessons from and for competent genetic algorithms, Springer.
- Guide, A. (2001). Project management body of knowledge (pmbok® guide). Project Management Institute.
- Hartmann, S. (2002). "A self-adapting genetic algorithm for project scheduling under resource constraints." Naval Research Logistics (NRL) **49**(5): 433-448.
- Hartmann, S. and D. Briskorn (2010). "A survey of variants and extensions of the resource-constrained project scheduling problem." European Journal of operational research **207**(1): 1-14.
- Hartmann, S. and D. Briskorn (2022). "An updated survey of variants and extensions of the resource-constrained project scheduling problem." European Journal of Operational Research **297**(1): 1-14.
- Hayes-Roth, F. (1975). "Review of" Adaptation in Natural and Artificial Systems by John H. Holland", The U. of Michigan Press, 1975." ACM SIGART Bulletin(53): 15-15.
- Hegazy, T. and N. Wassef (2001). "Cost optimization in projects with repetitive nonserial activities." Journal of Construction Engineering and Management **127**(3): 183-191.
- Herroelen, W., B. De Reyck and E. Demeulemeester (1998). "Resource-constrained project scheduling: A survey of recent developments." Computers & Operations Research **25**(4): 279-302.
- Herroelen, W. and R. Leus (2005). "Project scheduling under uncertainty: Survey and research potentials." European journal of operational research **165**(2): 289-306.
- Holland, J. H. (1992). "Genetic algorithms." Scientific american **267**(1): 66-73.
- Huang, H.-H., C.-H. Huang and W. Pei (2015). "Solving Multi-Resource Constrained Project Scheduling Problem using Ant Colony Optimization." Journal of Engineering, Project & Production Management **5**(1).
- Kadri, R. L. and F. F. Boctor (2018). "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case." European Journal of Operational Research **265**(2): 454-462.
- Kaiafa, S. and A. P. Chassiakos (2015). "A genetic algorithm for optimal resource-driven project scheduling." Procedia Engineering **123**: 260-267.
- Kalhor, E., M. Khanzadi, E. Eshtehardian and A. Afshar (2011). "Stochastic time-cost optimization using non-dominated archiving ant colony approach." Automation in Construction **20**(8): 1193-1203.
- Karam, A. and S. Lazarova-Molnar (2013). Recent trends in solving the deterministic resource constrained project scheduling problem. 2013 9th International Conference on Innovations in Information Technology (IIT), IEEE.

- Klanšek, U. and M. Pšunder (2010). "Cost optimization of time schedules for project management." Economic research-Ekonomska istraživanja **23**(4): 22-36.
- Kolisch, R. (2000). "Integrated scheduling, assembly area-and part-assignment for large-scale, make-to-order assemblies." International Journal of Production Economics **64**(1-3): 127-141.
- Kolisch, R. and S. Hartmann (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update." European journal of operational research **174**(1): 23-37.
- Krüger, D. (2009). Multi-project scheduling with transfers, Ph. D. thesis, University of Jena, Germany.
- Krüger, D. and A. Scholl (2009). "A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times." European Journal of Operational Research **197**(2): 492-508.
- Lenstra, J. K. and A. Rinnooy Kan (1978). "Complexity of scheduling under precedence constraints." Operations Research **26**(1): 22-35.
- Leu, S.-S., A.-T. Chen and C.-H. Yang (1999). "Fuzzy optimal model for resource-constrained construction scheduling." Journal of Computing in Civil Engineering **13**(3): 207-216.
- Li, H. and P. Love (1997). "Using improved genetic algorithms to facilitate time-cost optimization." Journal of Construction Engineering and management **123**(3): 233-237.
- Li, H. and N. K. Womer (2015). "Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming." European Journal of Operational Research **246**(1): 20-33.
- Li, H., Z. Xu and W. Wei (2018). "Bi-objective scheduling optimization for discrete time/cost trade-off in projects." Sustainability **10**(8): 2802.
- Liu, J. (1998). "'Realization of time-cost optimization method on computer.'" Optimization Capital Construction. Journal of construction engineering and management **133**.
- Liu, J. and F. Rahbar (2004). "Project time-cost trade-off optimization by maximal flow theory." Journal of construction engineering and management **130**(4): 607-609.
- Lotfi, R., Z. Yadegari, S. H. Hosseini, A. H. Khameneh, E. B. Tirkolaei and G.-W. Weber (2022). "A robust time-cost-quality-energy-environment trade-off with resource-constrained in project management: A case study for a bridge construction project." Journal of industrial & management optimization **18**(1): 375.

- Maniezzo, V. and A. Mingozzi (1999). "The project scheduling problem with irregular starting time costs." Operations Research Letters **25**(4): 175-182.
- Messelis, T. and P. De Causmaecker (2014). "An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem." European Journal of Operational Research **233**(3): 511-528.
- Mika, M., G. Waligora and J. Węglarz (2008). "Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times." European Journal of Operational Research **187**(3): 1238-1250.
- Mingozzi, A., V. Maniezzo, S. Ricciardelli and L. Bianco (1998). "An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation." Management science **44**(5): 714-729.
- Mori, M. and C. C. Tseng (1997). "A genetic algorithm for multi-mode resource constrained project scheduling problem." European Journal of Operational Research **100**(1): 134-141.
- Morillo-Torres, D., L. F. Moreno-Velásquez and F. J. Díaz-Serna (2015). "A branch and bound hybrid algorithm with four deterministic heuristics for the resource constrained project scheduling problem (RCPSp)." Dyna **82**(190): 198-207.
- Moselhi, O. and K. El-Rayes (1993). "Scheduling of repetitive projects with cost optimization." Journal of Construction Engineering and Management **119**(4): 681-697.
- Ng, S. T. and Y. Zhang (2008). "Optimizing construction time and cost using ant colony optimization approach." Journal of construction engineering and management **134**(9): 721-728.
- Nonobe, K. and T. Ibaraki (2002). Formulation and tabu search algorithm for the resource constrained project scheduling problem. Essays and surveys in metaheuristics, Springer: 557-588.
- Ozdamar, L. (1999). "A genetic algorithm approach to a general category project scheduling problem." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **29**(1): 44-59.
- Patterson, J. H., F. B. Talbot, R. Slowinski and J. Węglarz (1990). "Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems." European Journal of Operational Research **49**(1): 68-79.
- Plaisted, D. A. (1984). "New NP-hard and NP-complete polynomial and integer divisibility problems." Theoretical Computer Science **31**(1-2): 125-138.
- Podolski, M. and B. Sroka (2019). "Cost optimization of multiunit construction projects using linear programming and metaheuristic-based simulated annealing algorithm." Journal of civil engineering and management **25**(8): 848-857.

- Poppenborg, J. and S. Knust (2016). "A flow-based tabu search algorithm for the RCPSP with transfer times." OR spectrum **38**(2): 305-334.
- Pritsker, A. A. B., L. J. Walters and P. M. Wolfe (1969). "Multiproject scheduling with limited resources: A zero-one programming approach." Management science **16**(1): 93-108.
- Que, B. C. (2002). "Incorporating practicability into genetic algorithm-based time-cost optimization." Journal of construction engineering and management **128**(2): 139-143.
- Raftery, J. (2003). Risk analysis in project management, Routledge.
- Rogalska, M., W. Bożejko and Z. Hejducki (2008). "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling." Automation in Construction **18**(1): 24-31.
- Shahriari, M. (2016). "Multi-objective optimization of discrete time–cost tradeoff problem in project networks using non-dominated sorting genetic algorithm." Journal of Industrial Engineering International **12**(2): 159-169.
- Słowinski, R. (1981). "Multiobjective network scheduling with efficient use of renewable and nonrenewable resources." European Journal of Operational Research **7**(3): 265-273.
- Sprecher, A., S. Hartmann and A. Drexl (1997). "An exact algorithm for project scheduling with multiple modes." Operations-Research-Spektrum **19**(3): 195-203.
- Stork, F. (2001). "Stochastic resource-constrained project scheduling."
- Talbot, F. B. (1981). "An Algorithm for a general class of precedence and resource constrained scheduling problems."
- Talbot, F. B. (1982). "Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case." Management science **28**(10): 1197-1210.
- Thomas, P. R. and S. Salhi (1998). "A tabu search approach for the resource constrained project scheduling problem." Journal of Heuristics **4**(2): 123-139.
- Toğan, V. and M. A. Eirgash (2019). "Time-cost trade-off optimization of construction projects using teaching learning based optimization." KSCE Journal of Civil Engineering **23**(1): 10-20.
- Tran, D.-H., M.-Y. Cheng and A.-D. Pham (2016). "Using fuzzy clustering chaotic-based differential evolution to solve multiple resources leveling in the multiple projects scheduling problem." Alexandria Engineering Journal **55**(2): 1541-1552.
- Uysal, F., S. K. İşleyen and C. Çetinkaya (2018). "Resource constrained project scheduling with stochastic resources." Journal of applied research on industrial engineering **5**(1): 39-49.

- Van Peteghem, V. and M. Vanhoucke (2014). "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances." European Journal of Operational Research **235**(1): 62-72.
- Węglarz, J., J. Józefowska, M. Mika and G. Waligóra (2011). "Project scheduling with finite or infinite number of activity processing modes—A survey." European Journal of operational research **208**(3): 177-205.
- Wilson, P. R., J. N. Ross and A. D. Brown (2001). "Optimizing the Jiles-Atherton model of hysteresis by a genetic algorithm." IEEE Transactions on Magnetics **37**(2): 989-993.
- Xiong, Y. and Y. Kuang (2008). "Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off." Journal of construction engineering and management **134**(2): 153-156.
- Zandhuis, A. and T. Wuttke (2019). A pocket companion to PMI's PMBOK® Guide sixth Edition: Based on PMBOK® Guide Sixth Edition, Van Haren.
- Zapata, J. C., B. M. Hodge and G. V. Reklaitis (2008). "The multimode resource constrained multiproject scheduling problem: Alternative formulations." AIChE Journal **54**(8): 2101-2119.
- Zhang, H., C. Tam and H. Li (2006). "Multimode project scheduling based on particle swarm optimization." Computer-Aided Civil and Infrastructure Engineering **21**(2): 93-103.
- Zhang, Y. and S. Thomas Ng (2012). "An ant colony system based decision support system for construction time-cost optimization." Journal of Civil Engineering and Management **18**(4): 580-589.
- Zheng, D. X. and S. T. Ng (2005). "Stochastic time–cost optimization model incorporating fuzzy sets theory and nonreplaceable front." Journal of Construction Engineering and Management **131**(2): 176-186.
- Zheng, D. X., S. T. Ng and M. M. Kumaraswamy (2004). "Applying a genetic algorithm-based multiobjective approach for time-cost optimization." Journal of Construction Engineering and management **130**(2): 168-176.
- Zheng, D. X., S. T. Ng and M. M. Kumaraswamy (2005). "Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization." Journal of Construction Engineering and Management **131**(1): 81-91.
- Zhu, G., J. F. Bard and G. Yu (2006). "A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem." INFORMS Journal on Computing **18**(3): 377-390.

- Rieck, J., J. Zimmermann and T. Gather (2012). "Mixed-integer linear programming for resource leveling problems." European Journal of Operational Research **221**(1): 27-37.
- Ponz-Tienda, J. L., V. Yepes, E. Pellicer and J. Moreno-Flores (2013). "The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm." Automation in Construction **29**: 161-172.
- Abdelkhalik, H. A., H. S. Refaie and R. F. Aziz (2020). "Optimization of time and cost through learning curve analysis." Ain Shams Engineering Journal **11**(4): 1069-1082.
- Al Haj, R. A. and S. M. El-Sayegh (2015). "Time-cost optimization model considering float-consumption impact." Journal of Construction Engineering and Management **141**(5): 04015001.
- Ashtiani, B., R. Leus and M.-B. Aryanezhad (2011). "New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing." Journal of Scheduling **14**(2): 157-171.
- Brucker, P. (2002). "Scheduling and constraint propagation." Discrete Applied Mathematics **123**(1-3): 227-256.
- Bruni, M. E., P. Beraldi and F. Guerriero (2015). The stochastic resource-constrained project scheduling problem. Handbook on Project Management and Scheduling Vol. 2, Springer: 811-835.
- Bruni, M. E., P. Beraldi, F. Guerriero and E. Pinto (2011). "A heuristic approach for resource constrained project scheduling with uncertain activity durations." Computers & Operations Research **38**(9): 1305-1318.
- Chakraborty, R. K., R. Sarker and D. Essam (2015). Resource constrained project scheduling: A branch and cut approach. proceedings of the 45th international conference on computers and industrial engineering Metz, France.
- Chakraborty, R. K., R. A. Sarker and D. L. Essam (2017). "Resource constrained project scheduling with uncertain activity durations." Computers & Industrial Engineering **112**: 537-550.
- Chassiakos, A. P. and S. P. Sakellariopoulos (2005). "Time-cost optimization of construction projects with generalized activity constraints." Journal of Construction Engineering and Management **131**(10): 1115-1124.
- Drexel, A. and J. Gruenewald (1993). "Nonpreemptive multi-mode resource-constrained project scheduling." IIE transactions **25**(5): 74-81.
- Ezeldin, A. S. and A. Soliman (2009). "Hybrid time-cost optimization of nonserial repetitive construction projects." Journal of Construction Engineering and Management **135**(1): 42-55.

- Fang, C., R. Kolisch, L. Wang and C. Mu (2015). "An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem." Flexible Services and Manufacturing Journal **27**(4): 585-605.
- Feng, C.-W., L. Liu and S. A. Burns (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." Journal of computing in civil engineering **11**(3): 184-189.
- Hegazy, T. and N. Wassef (2001). "Cost optimization in projects with repetitive nonserial activities." Journal of Construction Engineering and Management **127**(3): 183-191.
- Kadri, R. L. and F. F. Boctor (2018). "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case." European Journal of Operational Research **265**(2): 454-462.
- Kalhor, E., M. Khanzadi, E. Eshtehardian and A. Afshar (2011). "Stochastic time-cost optimization using non-dominated archiving ant colony approach." Automation in Construction **20**(8): 1193-1203.
- Karam, A. and S. Lazarova-Molnar (2013). Recent trends in solving the deterministic resource constrained project scheduling problem. 2013 9th International Conference on Innovations in Information Technology (IIT), IEEE.
- Klanšek, U. and M. Pšunder (2010). "Cost optimization of time schedules for project management." Economic research-Ekonomska istraživanja **23**(4): 22-36.
- Kolisch, R. and S. Hartmann (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update." European journal of operational research **174**(1): 23-37.
- Leu, S.-S., A.-T. Chen and C.-H. Yang (1999). "Fuzzy optimal model for resource-constrained construction scheduling." Journal of Computing in Civil Engineering **13**(3): 207-216.
- Li, H. and P. Love (1997). "Using improved genetic algorithms to facilitate time-cost optimization." Journal of Construction Engineering and management **123**(3): 233-237.
- Li, H. and N. K. Womer (2015). "Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming." European Journal of Operational Research **246**(1): 20-33.
- Li, H., Z. Xu and W. Wei (2018). "Bi-objective scheduling optimization for discrete time/cost trade-off in projects." Sustainability **10**(8): 2802.
- Liu, J. (1998). "“Realization of time-cost optimization method on computer.” Optimization Capital Construction." Journal of construction engineering and management

133.

- Liu, J. and F. Rahbar (2004). "Project time-cost trade-off optimization by maximal flow theory." Journal of construction engineering and management **130**(4): 607-609.
- Morillo-Torres, D., L. F. Moreno-Velásquez and F. J. Díaz-Serna (2015). "A branch and bound hybrid algorithm with four deterministic heuristics for the resource constrained project scheduling problem (RCPSp)." Dyna **82**(190): 198-207.
- Moselhi, O. and K. El-Rayes (1993). "Scheduling of repetitive projects with cost optimization." Journal of Construction Engineering and Management **119**(4): 681-697.
- Ng, S. T. and Y. Zhang (2008). "Optimizing construction time and cost using ant colony optimization approach." Journal of construction engineering and management **134**(9): 721-728.
- Podolski, M. and B. Sroka (2019). "Cost optimization of multiunit construction projects using linear programming and metaheuristic-based simulated annealing algorithm." Journal of civil engineering and management **25**(8): 848-857.
- Que, B. C. (2002). "Incorporating practicability into genetic algorithm-based time-cost optimization." Journal of construction engineering and management **128**(2): 139-143.
- Rogalska, M., W. Bożejko and Z. Hejducki (2008). "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling." Automation in Construction **18**(1): 24-31.
- Shahriari, M. (2016). "Multi-objective optimization of discrete time–cost tradeoff problem in project networks using non-dominated sorting genetic algorithm." Journal of Industrial Engineering International **12**(2): 159-169.
- Stork, F. (2001). "Stochastic resource-constrained project scheduling."
- Toğan, V. and M. A. Eirgash (2019). "Time-cost trade-off optimization of construction projects using teaching learning based optimization." KSCE Journal of Civil Engineering **23**(1): 10-20.
- Uysal, F., S. K. İşleyen and C. Çetinkaya (2018). "Resource constrained project scheduling with stochastic resources." Journal of applied research on industrial engineering **5**(1): 39-49.
- Xiong, Y. and Y. Kuang (2008). "Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off." Journal of construction engineering and management **134**(2): 153-156.

- Zhang, Y. and S. Thomas Ng (2012). "An ant colony system based decision support system for construction time-cost optimization." Journal of Civil Engineering and Management **18**(4): 580-589.
- Zheng, D. X. and S. T. Ng (2005). "Stochastic time–cost optimization model incorporating fuzzy sets theory and nonreplaceable front." Journal of Construction Engineering and Management **131**(2): 176-186.
- Zheng, D. X., S. T. Ng and M. M. Kumaraswamy (2004). "Applying a genetic algorithm-based multiobjective approach for time-cost optimization." Journal of Construction Engineering and management **130**(2): 168-176.
- Zheng, D. X., S. T. Ng and M. M. Kumaraswamy (2005). "Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization." Journal of Construction Engineering and Management **131**(1): 81-91.
- Tao, S. and Z. S. Dong (2018). "Multi-mode resource-constrained project scheduling problem with alternative project structures." Computers & Industrial Engineering **125**: 333-347.
- Balouka, N. and I. Cohen (2021). "A robust optimization approach for the multi-mode resource-constrained project scheduling problem." European Journal of Operational Research **291**(2): 457-470.
- Tirkolaee, E. B., A. Goli, M. Hematian, A. K. Sangaijah and T. Han (2019). "Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms." Computing **101**(6): 547-570.
- ard problem determined from its characteristic for DNA-based computing." BioSystems **80**(1): 71-82.
- Lodewijks, B. (2019). "Mapping NP-hard and NP-complete optimisation problems to Quadratic Unconstrained Binary Optimisation problems." arXiv preprint arXiv:1911.08043.
- Plaisted, D. A. (1984). "New NP-hard and NP-complete polynomial and integer divisibility problems." Theoretical Computer Science **31**(1-2): 125-138.
- Błażewicz, J., K. H. Ecker, E. Pesch, G. Schmidt and J. Weglarz (2001). Scheduling computer and manufacturing processes, springer science & Business media.
- Deb, K., A. Pratap, S. Agarwal and T. Meyarivan (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation **6**(2): 182-197.

- Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. Proceedings of an International Conference on Genetic Algorithms and their applications, Hillsdale, New Jersey.
- Kumar, H. (2019). "Hybrid NSGA-II based decision-making in fuzzy multi-objective reliability optimization problem." Retrieved 2022, from <https://link.springer.com/article/10.1007/s42452-019-1512-2>.
- Lee, C. M. (2019). "Solve Multi-Objective Problem using NSGA-II and DEAP in Python." 2022, from <https://medium.com/@rossleecooloh/optimization-algorithm-nsga-ii-and-python-package-deap-fca0be6b2ffc>.
- Mallawaarachchi, V. (2017). "Introduction to Genetic Algorithms — Including Example Code." Retrieved 2022-06-15, from <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>.
- Sparrow, J. (2022). "NP-Complétude | Série 1 (Introduction)." Retrieved 2022-07-01, from <https://fr.acervolima.com/np-completude-serie-1-introduction/>.

ANNEXES:

ANNEXE A:
CODE DE LA NSGA-II

```
clc;
clear;
close all;

%% Problem Definition
data=load('data.mat');
Precedence=data.Precedence;
state=data.state;
%% NSGA-II Parameters
MaxIt=100;
nPop=50;

pCrossover=0.6;
nCrossover=round(pCrossover*nPop/2)*2;

pMutation=0.6;
nMutation=round(pMutation*nPop);

%% Initialization
tic;
empty_individual.Position=zeros(2,data.N);
empty_individual.Cost=[];
empty_individual.Rank=[];
empty_individual.CrowdingDistance=[];
empty_individual.DominatedCount=[];
empty_individual.DominationSet=[];
empty_individual.info=[];

pop=repmat(empty_individual,nPop,1);

for i=1:nPop
    x=randperm(data.N);
    y=randi([1 state],1,data.N);
    x=PrecedenceCheck(x,Precedence);
```

```

    pop(i).Position=cat(1,x,y);
    [pop(i) pop(i).Cost]=Fitness(pop(i),data);
end

% Non-dominated Sorting
[pop F]=NonDominatedSorting(pop);

% Calculate Crowding Distances
pop=CalcCrowdingDistance(pop,F);

%% NSGA-II Loop
for it=1:MaxIt
    % Crossover
    popc= repmat(empty_individual,nCrossover/2,2);
    for k=1:nCrossover/2

        i1=BinaryTournamentSelection(pop);
        i2=BinaryTournamentSelection(pop);

        [popc(k,1) popc(k,2)]=Crossover(pop(i1),pop(i2));

        [popc(k,1) popc(k,1).Cost]=Fitness(popc(k,1),data);
        [popc(k,2) popc(k,2).Cost]=Fitness(popc(k,2),data);

    end
    popc=popc(:);
    % Mutation
    popm= repmat(empty_individual,nMutation,1);
    for k=1:nMutation

        i=BinaryTournamentSelection(pop);

        popm(k)=Mutate(pop(i));
    end
end

```

```
[popm(k) popm(k).Cost]=Fitness(popm(k),data);
end

% Merge Pops
pop=[pop
      popc
      popm];

% Non-dominated Sorting
[pop F]=NonDominatedSorting(pop);

% Calculate Crowding Distances
pop=CalcCrowdingDistance(pop,F);

% Sort Population
pop=SortPopulation(pop);

% Delete Extra Individuals
pop=pop(1:nPop);

% Non-dominated Sorting
[pop F]=NonDominatedSorting(pop);

% Calculate Crowding Distances
pop=CalcCrowdingDistance(pop,F);

% Plot F1
PF=pop(F{1});
PFCosts=[PF.Cost];
figure(1);
plot(PFCosts(1,:),PFCosts(2,:), '*');
xlabel('f_1');
ylabel('f_2');
```

```

    % Show Iteration Information
    disp(['Iteraion ' num2str(it) ': Number of F1 Members = '
num2str(numel(PF))]);

end

RunTime=toc;

%% Results
m=mean(PFCosts,2);
m1=PFCosts(1,:);
m2=PFCosts(2,:);
NPS=numel(PF);
SM=sqrt( max(PFCosts(1,:))-min(PFCosts(1,:))+max(PFCosts(2,:))-
min(PFCosts(2,:)));
SS=0;
FFF=0;
CI=0;
SNS=0;
for i=1:NPS
    CI=CI+sqrt(PFCosts(1,i)^2+PFCosts(2,i)^2);
    FF(i)=min(PFCosts(1,i),PFCosts(2,i));
    FFF=FFF+(PFCosts(1,i)-FF(i))/FF(i)+(PFCosts(2,i)-FF(i))/FF(i);
end
RAS=FFF/NPS;
MID=CI/NPS;

for i=1:NPS
    CI=sqrt(PFCosts(1,i)^2+PFCosts(2,i)^2);
    SNS=SNS+(MID-CI)^2;
end
SNS=(sqrt(SNS))/sqrt((NPS-1));

d=zeros(1,NPS);
distance=zeros(NPS,NPS);

```

```

    for i=1:NPS
        for ii=1:NPS
            distance(i,ii)=abs(PFCosts(1,i)-PFCosts(1,ii))+abs(PFCosts(2,i)-
PFCosts(2,ii));
        end
    end
    for i=1:NPS
        distance(i,i)=inf ;
    end
d=min(distance);
SM=sqrt(var(d));

disp(' ');
disp(['Total Run Time: ' num2str(RunTime) ' second(s)']);
disp(' ');
disp(['Number of Pareto Solution (NPS)= ' num2str(NPS) ]);
disp(' ');
disp(['Spacing Metric (SM)= ' num2str(SM) ]);
disp(' ');
disp(['Mean ideal distance (MID)= ' num2str(MID) ]);
disp(' ');
disp(['Spread of Non-dominance Solution (SNS)= ' num2str(SNS) ]);
disp(' ');
disp(['The Rate of Achievement to Objectives Simultaneously (RAS)= '
num2str(RAS) ]);

```

ANNEXE B:
TRIER LES POPULATIONS

```
function pop=SortPopulation(pop)

    % Sort according to Crowding Distance
    CD=[pop.CrowdingDistance];
    [CD CDSO]=sort(CD,'descend');
    pop=pop(CDSO);

    % Sort according to Rank
    R=[pop.Rank];
    [R RSO]=sort(R,'ascend');
    pop=pop(RSO);

end
```

ANNEXE C:
CONTRÔLE DE PRIORITÉ

```
function newx=PrecedenceCheck(x,Precedence)

newx=[];

while true

    for i=x

        if all(ismember(Precedence{i},newx))
            break;
        end
    end

    [newx]=[newx i];
    [j]=find(x==i);
    x(j)=[];

    if isempty(x)
        break
    end

end

end
```

ANNEXE D:
TRIAGE NON DOMINÉ

```
function [pop F]=NonDominatedSorting(pop)

    nPop=numel(pop);

    F{1}=[];

    for i=1:nPop
        pop(i).DominatedCount=0;
        pop(i).DominationSet=[];

        for j=[1:i-1 i+1:nPop]

            if Dominates(pop(i),pop(j))
                pop(i).DominationSet=[pop(i).DominationSet j];

            elseif Dominates(pop(j),pop(i))
                pop(i).DominatedCount=pop(i).DominatedCount+1;

            end

        end

        end

        if pop(i).DominatedCount==0
            F{1}=[F{1} i];
        end

    end

    k=1;

    while true

        Q=[];
```

```
for i=F{k}
    for j=pop(i).DominationSet
        pop(j).DominatedCount=pop(j).DominatedCount-1;

        if pop(j).DominatedCount==0
            Q=[Q j];
        end
    end
end

if isempty(Q)
    break;
end

F{k+1}=Q;
k=k+1;

end

for k=1:numel(F)
    for i=F{k}
        pop(i).Rank=k;
    end
end

end
```

ANNEXE E:

MUTATE

```
function y=Mutate(x)
xtemp=x;

    P=x.Position;

    [a b]=size(P);

    e=randi([1 3],1,b)

    P(2,:)=e;
    xtemp.Position=P;
    y=xtemp;
end
```

ANNEXE F:
DONNÉE INITIALE

```
clc
clear

pre=xlsread('data.xlsx','PRE');
N=size(pre,1);
for c=1:N
q=find(pre(:,c)==1);
Precedence{c}=q;
end
maxR=xlsread('data.xlsx','MAX_R');
num_R=size(maxR,2);
T=xlsread('data.xlsx','T');
state=size(T,2);
U=zeros(N,state,num_R);
U(:, :, 1) = xlsread('data.xlsx','U1');

CA=zeros(N,state,num_R);
CA(:, :, 1) = xlsread('data.xlsx','CA1')

save data
```

ANNEXE G:
APTITUDE

```

function [sol Z]=Fitness(sol,data)

Precedence=data.Precedence;
T=(data.T)-1;
U=data.U;
CA=data.CA;
maxR=data.maxR;
N=data.N;
num_R=size(maxR,2);

x=sol.Position;

ST=zeros(N,1);
FT=zeros(N,1);
ZZ=0;
sumT=3*sum(max(T,[],2));

RemainedR= repmat(maxR,sumT,1);

m=0;
for i=x(1,:)
    m=m+1;
    state(i)=x(2,m);
    if isempty(Precedence{i})
        t=0;
    else
        t=max(FT(Precedence{i}));
    end
    for r=1:num_R
        Ri(:,r)=repmat(U(i,state(i),r),sumT,1);

```

```

qq=RemainedR(:,r)-Ri(:,r);

[a1,~]=find(qq<0);
q=ones(1,sumT);
q(a1)=-1;
q(1:t)=-1;

p=find(q>0);
np=length(p);

for g=1:np
    j=p(g);

    if all(q(j:j+T(i,state(i))))>=0)==1
        break
    end
end
jj(r)=j;
end
ST(i)=max(jj(r));

FT(i)=ST(i)+T(i,state(i));
for r=1:num_R
    ZZ=ZZ+CA(i,state(i),r);
end
periodi=ST(i):FT(i);

for r=1:num_R
    RemainedR(periodi,r)=RemainedR(periodi,r)-
    repmat(U(i,state(i),r),T(i,state(i))+1,1);
end

```

```
end
```

```
Z1=max(FT);  
Z2=ZZ;
```

```
sol.info.ST=ST;  
sol.info.FT=FT;  
Z=[Z1  
  Z2];
```

```
end
```

ANNEXE H:

DOMINE

```
function flag=Dominates(x,y)

    if isfield(x, 'Cost')
        x=x.Cost;
    end

    if isfield(y, 'Cost')
        y=y.Cost;
    end

    flag=all(x<=y) && any(x<y);

end
```

ANNEXE I:
CROISEMENT

```
function [y1 y2]=Crossover(x1,x2)

    x1temp=x1;
    x2temp=x2;

    P1=x1.Position;
    P2=x2.Position;
    [a b]=size(P1);
    e=randperm(b-2);
    cros=e(1)+1;
    pt1=P1(1,:);
    pt2=P2(1,:);
    pp1=P1(2,:);
    pp2=P2(2,:);
    PPtemp=cat(2,pp1(1:cros),pp2(cros+1:b));
    P1=cat(1,pt1,PPtemp);
    P2=cat(1,pt2,PPtemp);

    x1temp.Position=P1;
    x2temp.Position=P2;

    y1=x1temp;
    y2=x2temp;

end
```

ANNEXE J:**CALCULER LA DISTANCE D'ENCOMBREMENT**

```
function pop=CalcCrowdingDistance(pop,F)

    for k=1:numel(F)

        C=[pop(F{k}).Cost];

        nObj=size(C,1);

        nFk=numel(F{k});

        D=zeros(nFk,nObj);

        for j=1:nObj

            Cj=C(j,:);

            [SortedCj SortOrder]=sort(Cj);

            D(SortOrder(1),j)=inf;
            D(SortOrder(end),j)=inf;

            Cjmin=SortedCj(1);
            Cjmax=SortedCj(end);

            for i=2:nFk-1
                D(SortOrder(i),j)=(SortedCj(i+1)-SortedCj(i-1))/(Cjmax-Cjmin);
            end

        end

        for i=1:nFk
            pop(F{k}(i)).CrowdingDistance=sum(D(i,:));
        end
    end
end
```

end

end

ANNEXE K:
SÉLECTION DE TOURNOIS BINAIRES

```
function i=BinaryTournamentSelection(pop)

    ii=randperm(numel(pop));

    i1=ii(1);
    i2=ii(2);

    if pop(i1).Rank<pop(i2).Rank
        i=i1;

    elseif pop(i2).Rank<pop(i1).Rank
        i=i2;

    else
        if pop(i1).CrowdingDistance>pop(i2).CrowdingDistance
            i=i1;
        else
            i=i2;
        end
    end
end

end
```