



HAL
open science

Par-delà les Métaheuristiques ...

Laurent Deroussi

► **To cite this version:**

Laurent Deroussi. Par-delà les Métaheuristiques Informatique [cs]. Université Clermont Auvergne (UCA), 2023. tel-04933592

HAL Id: tel-04933592

<https://hal.science/tel-04933592v1>

Submitted on 6 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Habilitation à Diriger des Recherches

Par-delà les Métaheuristiques ...

Présentée par

Laurent Deroussi

Tome 1

Soutenue publiquement le 3 juillet 2023 à l'IUT de Clermont Auvergne (site de Montluçon) devant le jury :

Rapporteurs :	Carola Doerr	Directrice de recherche	CNRS, Sorbonne Université
	Jin-Kao Hao	Professeur	Université d'Angers
	Marc Sevaux	Professeur	Université Bretagne Sud
Examineurs	Mourad Baiou	Directeur de recherche	Université Clermont Auvergne
	Michel Gourgand	Professeur émérite	Université Clermont Auvergne
Référente	Sylvie Norre	Professeure	Université Clermont Auvergne

Laboratoire d'Informatique de Modélisation
et d'Optimisation des Systèmes
LIMOS CNRS UMR 6158
Axe Outils Décisionnels pour la Production
et les Services



REMERCIEMENTS

Déjà vingt ans que j'exerce au sein de l'IUT, d'abord de Montluçon, puis celui de l'Allier, puis celui de Clermont-Auvergne. Bien qu'il prenne de l'ampleur à chaque changement de nom, c'est bien le même établissement. J'y ai toujours trouvé sourire et bonne humeur. Que tous les personnels, collègues enseignants ou enseignants-chercheurs, techniciens ou administratifs que j'ai côtoyé en soient remerciés.

Un informaticien aussi peu informaticien que moi a toujours besoin d'informaticiens plus informaticien pour résoudre mes problèmes. Faire fonctionner des applications d'un autre âge, ouvrir des accès pour accéder aux ressources informatiques à distance des étudiants, équiper des salles de micros et caméras, gérer le matériel de l'entrepôt, etc. Un grand merci à Marc, Alexis, Joffray, Paul et Alexis pour leur disponibilité et leur réactivité sans faille.

Mes collègues du département Métiers de la Transition et de l'Efficacité Energétiques ont toujours fait preuve de beaucoup de compréhension à mon égard, en me laissant m'émanciper dans le département voisin. J'aurai largement le temps de les en remercier durant les trois prochaines années...

Ce sont donc les redoutables walkyries du département Management de la Logistique et du Transport qui m'ont offert le gîte. Comme tout bon processus symbiotique, je leur ai parfois offert le couvert en partageant les réussites, et aussi parfois les échecs culinaires de ma moitié. Je ne garderai que d'excellents souvenirs de partages et de convivialités.

Ce fut une grande chance de co-animer la Licence Professionnelle LOGIN pendant près de 15 ans. En même temps que le mariage pour tous, nos étudiants ont pu bénéficier d'un Papa systèmes d'information et d'un papa aide à la décision. David, merci de m'avoir accompagné dans la dernière partie de cette aventure, et pour tout ce que j'ai appris à tes côtés.

De Gardanne à Montluçon, le LIMOS est sans doute le plus grand laboratoire français. Malgré les distances, les réunions d'axe et de laboratoire nous permettent d'échanger régulièrement. Merci aux directeurs qui se sont succédé (Alain, Farouk et Mourad) et qui ont toujours soutenu notre petite équipe montluçonnaise. Merci également à les tous les collègues du labo pour les échanges toujours fructueux, qu'ils soient scientifiques ou pas.

"La recherche avance !". C'est par cette formule que j'exprime souvent mon agacement à faire quelques activités de peu d'importance. Heureusement, nos doctorants et post-doctorants sont là pour nous aider et faire en sorte que la recherche avance vraiment. Merci à Sylvie, Benjamin, Fabrice, Arnaud, Raïssa et Youssef pour les travaux passés, et à Arnaud, Youssouf et Hector pour les travaux présents et futurs.

La co-animation du Groupe de travail META m'a permis de travailler auprès de deux spécialistes des métaheuristiques. Un grand merci à Patrick Siarry et El-Ghazali Talbi pour m'avoir fait confiance. En plus de META, Patrick m'a accompagné tout au long de ma carrière. Il était déjà l'un des rapporteurs de ma thèse, m'a permis de rencontrer et de travailler avec Maurice Clerc, spécialiste de l'Optimisation par Essaim Particulaire, et donner l'opportunité d'écrire un ouvrage.

Un énorme merci à Carola, Jin-Kao et Marc qui ont spontanément accepté de rapporter mon HDR. J'ai essayé d'aborder les métaheuristiques avec un angle personnel et une pointe d'originalité. Votre regard sur ce travail était très important pour moi.

Je me suis lâchement appuyé sur une phrase souvent prononcée, mais je le concède un peu sortie de son contexte, comme fil conducteur de la soutenance. Merci Michel pour tout ce que vous m'avez appris et apporté.

Enfin, mon plus grand privilège a été de travailler pendant toutes ces années dans un environnement très bienveillant, aux côtés de deux personnes aux qualités humaines et scientifiques hors normes. Elles sont surtout devenues au fil du temps une seconde famille. Une grande sœur qui a toujours veillé sur moi, et une petite sœur ... qui a aussi veillé sur moi ! Sylvie et Nathalie, merci pour tout !

Bon ! le boulot c'est une chose, mais l'important est ailleurs ...

J'ai le bonheur et la chance d'avoir deux grands enfants qui font mon bonheur et ma fierté. Isciane, artiste comme sa mère et rêveuse comme son père ! Milo, ouvert et tolérant comme sa mère, scientifique comme son père ! ... L'avenir vous appartient !

Enfin, il est des évidences qui traversent une vie sans jamais vaciller, un soleil qui n'a de cesse de chasser les nuages.

Sans elle, rien ne serait possible !

Sans elle, rien n'est impossible !

Ce travail t'est entièrement dédié ...

SOMMAIRE TOME 1

TABLE DES MATIERES

Avant-propos.....	1
Curriculum Vitae.....	3
Sommaire	3
1. Synthèse de la carrière.....	4
2. Activités pédagogiques	6
3. Activités scientifiques	11
4. Responsabilités collectives.....	15
5. Liste des travaux et publications	17
Mémoire.....	23
Tables des matières.....	23
Introduction.....	25
1. Fondements des problèmes et algorithmes	29
2. métaheuristiques : Prémices et concepts.....	37
3. Métaheuristiques : avancées récentes et tendances actuelles.....	63
4. Exemples d'application des métaheuristiques	77
5. Conclusion générale et perspectives.....	101
6. Bibliographie	105
Table des figures	110

SOMMAIRE TOME 2

Sommaire	1
<i>New effective Neighborhoods for the permutation flow shop problem</i>	3
<i>A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles</i>	21
<i>An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem.....</i>	47
<i>A new split-based hybrid metaheuristic for the Reconfigurable Transfer Line Balancing Problem.....</i>	65

Avant-propos

Lorsque, au début des années 1980, je m'initiais au langage BASIC (Beginner's All-purpose Symbolic Instruction Code) sur mon Commodore VIC20 et 3 ko de mémoire vive, j'étais loin de penser que 20 ans plus tard je décrocherais un doctorat en informatique, et encore 20 ans après, je serai en train d'écrire ces lignes ...

J'ai découvert le Problème du Voyageur de Commerce, lors d'un TER (Travail d'Encadrement et de Recherche) en Maîtrise d'Informatique, et programmé à l'occasion mes premières méthodes de recherche locale. J'ai poursuivi avec un stage au LIMOS en master, avec la programmation des premières métaheuristiques (algorithme tabou, recuit simulé, algorithme du kangourou, mais aussi des réseaux de neurones et les mouvements de Lin & Kernighan).

Je crois bien que depuis, le virus des métaheuristiques ne m'a jamais quitté !

Depuis, je n'ai jamais quitté le LIMOS, et j'ai toujours poursuivi ce travail de réflexion autour des métaheuristiques. Une question m'obsède, initiée par mon mentor en Recherche, Michel Gourgand, et qui répétait à l'envi que "*Les métaheuristiques, on ne sait pas très bien pourquoi ça marche, mais ça marche*".

Et pourquoi ça marche ? (et plutôt bien en plus !)

C'est un peu mon côté mouton noir, mais alors qu'autour de moi, d'aucuns s'intéressent aux problèmes, moi c'est toujours les méthodes qui ont constitué mon terrain de jeu favori.

20 ans de réflexions sur les métaheuristiques, sans avoir véritablement publié sur le sujet.

Alors, j'ai vu dans la rédaction de ce mémoire, un prétexte pour synthétiser tout ce que j'avais pu apprendre. Un prétexte pour écrire le document sur les métaheuristiques que j'aimerais lire. Il y avait pour moi une évidence : dans ce document, aucune métaheuristique ne devrait être explicitement décrite. Sur ce plan au moins, j'ai presque réussi !

Toujours ce côté mouton noir ! Ce mémoire n'est pas structuré classiquement, en faisant sur plusieurs chapitres une synthèse de mes activités de recherche, puis un dernier chapitre pour présenter un projet de recherche. Au lieu de cela, j'ai consacré les trois premiers chapitres pour rassembler les connaissances que j'avais pu accumuler sur les métaheuristiques. Le dernier chapitre reprend quelques-unes de mes activités de recherche, avec la loupe des chapitres précédents, et en se projetant sur les projets à venir.

J'ai pris beaucoup de plaisir à faire ce travail, à relire les articles que j'avais déjà lus lorsque j'étais en thèse, à redécouvrir (voire à découvrir lorsque j'étais passé à côté !) les premières notions sur les métaheuristiques, de revenir à la source de mes premières amours scientifiques.

"Celui qui ne sait pas d'où il vient ne peut savoir où il va"

Otto von Bismarck

Curriculum Vitae

SOMMAIRE

Sommaire	3
1. Synthèse de la carrière	4
1.1. Activités pédagogiques	4
1.2. Activités scientifiques	5
1.3. Responsabilités collectives	6
2. Activités pédagogiques	6
2.1. Présentation de l'activité d'enseignement	6
2.2. Présentation synthétique des enseignements.....	8
2.3. Formations suivies concernant vos activités pédagogiques	10
2.4. Responsabilités pédagogiques.....	11
3. Activités scientifiques	11
3.1. Présentation synthétique des thématiques de recherche	11
3.2. Encadrement Doctoral et scientifique.....	13
3.3. Diffusion et rayonnement	14
4. Responsabilités collectives	15
4.1. Présentation générale des responsabilités.....	15
4.2. Responsabilités administratives	16
4.3. Responsabilités et mandats locaux ou régionaux	17
4.4. Organisation de congrès	17
5. Liste des travaux et publications	17
5.1. Synthèse des publications	17
5.2. Ouvrage	18
5.3. Chapitres d'ouvrage.....	18
5.4. Articles dans une revue.....	18
5.5. Communication dans un congrès.....	19
5.6. Rapport	21

1. SYNTHÈSE DE LA CARRIÈRE

1.1. ETAT CIVIL

LAURENT DEROUSSI

Né le 1^{er} octobre 1970 à Riom-ès-Montagnes (15)

IUT Clermont Auvergne, site de Montluçon
Département Métiers de la Transition et de l'Efficacité Energétique
Avenue Aristide Briand
CS 82235
03101 Montluçon Cedex
France
Téléphone : 04-70-02-20-82
Courriel : laurent.deroussi@uca.fr

SITUATION PROFESSIONNELLE

Maître de Conférences hors classe au département Métiers de la Transition et de l'Efficacité Energétique de l'IUT Clermont Auvergne, site de Montluçon.

Membre de l'équipe de recherche du LIMOS (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes), axe Outils Décisionnels pour la Production et les Services (ODPS), UMR 6158.

Co-responsable de la Licence Professionnelle Logistique et Systèmes d'Information, parcours Informatique et aide à la décision (LOGIN).

1.2. ACTIVITÉS PÉDAGOGIQUES

Les deux postes que j'ai occupés depuis mon arrivée sont rattachés au département Génie Thermique et Énergie¹ (GTE). Cependant, j'ai toujours travaillé dans plusieurs départements de l'IUT et mes activités de recherche autour de l'optimisation dans les systèmes logistiques m'ont amené à m'investir dans le département Gestion Logistique et Transport² (GLT). Mes enseignements portent principalement sur les **mathématiques**, la **bureautique**, la **recherche opérationnelle** et les **systèmes d'information**.

En tant que PRCE (de 2003 à 2007), j'ai enseigné les **mathématiques** dans les trois départements industriels du site de Montluçon (GTE, GEII et GMP), la **programmation** (GEII) et la **bureautique** (Word/Excel) au niveau du DUT (première et deuxième années), les **systèmes d'information**, l'**échange de données informatisé** (EDI) et l'**optimisation** en Licence Professionnelle Logistique et Systèmes d'Information (LOGIN)

Depuis 2007, en tant que MCF, j'ai recentré mon enseignement sur mon département de rattachement **GTE** et sur le département **GLT**, avec un service d'enseignement partagé entre ces deux départements. Mes enseignements en GTE portent sur les **mathématiques** et la **bureautique avancée** (solveur linéaire d'Excel et programmation VBA) (DUT 2^{ème} année). Dans le département GLT, je suis intervenu en DUT GLT 2^{ème} année sur un cours d'optimisation

¹ Aujourd'hui MTEE pour Métiers de la Transition et de l'Efficacité énergétique.

² Aujourd'hui MLT pour Management de la Logistique et du Transport.

logistique. L'essentiel de mes interventions sont au niveau de la Licence Professionnelle Logistique et Systèmes d'Information (LOGIN) où j'ai enseigné les cours **d'optimisation, d'ordonnement, de planification, de localisation, de simulation, de WMS (Warehouse Management System) et de conception de tableaux de bord.**

En dehors de l'IUT, je suis intervenu sur le site de Montluçon dans la formation d'ingénieur en alternance Génie des Systèmes de production (GSP) au niveau des trois années en **mathématiques** (séries de Fourier et Algèbre Linéaire en 1ère année, Probabilités-statistiques en 2ème année, théorie des graphes en 3ème année).

Je suis également intervenu en master d'Informatique en Recherche Opérationnelle, et j'interviens au niveau de l'Université Clermont Auvergne dans l'animation d'un module de l'école doctorale Sciences pour l'Ingénieur et du master Génie Industriel. Dans ces trois formations pour une **introduction aux métaheuristiques.**

Je m'implique dans **l'innovation pédagogique**, avec la mise en œuvre de nouvelles pratiques pédagogiques (suppression des cours magistraux en mathématiques, utilisation des outils et plateformes numériques, l'apprentissage des étudiants par les étudiants, animation de jeux-sérieux, exploitation pédagogique de l'entrepôt pédagogique, ...). Je suis co-auteur de deux articles pédagogiques ("une pédagogie innovante pour la conception de tableaux de bord logistiques" et "un entrepôt pédagogique grandeur nature pour les étudiants de Montluçon") qui ont été présentés lors d'un track pédagogique de la logistique et du SCM, lors des 14èmes rencontres de l'AIRL-SCM à Clermont-Ferrand³ (<https://www.airl-scm.com/lesactesrairl2022>).

Je me suis investi dans l'ouverture de plusieurs formations. J'ai participé au projet d'ouverture de la formation d'ingénieur en alternance GSP en concevant les trois modules de mathématiques. J'ai co-porté le projet d'ouverture en apprentissage de la licence professionnelle LOGIN. Enfin, je participe au projet d'ouverture d'un Master MIAGE (Méthodes Informatiques Appliquées à la Gestion des Entreprises), avec une ouverture souhaitée en septembre 2024.

1.3. ACTIVITES SCIENTIFIQUES

Depuis mon doctorat, toutes mes activités de recherche ont été effectuées au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS, CNRS UMR 6158). Je m'intéresse plus particulièrement aux méthodes d'optimisation à base de métaheuristiques. Mieux connaître et comprendre le comportement des métaheuristiques afin de pouvoir les utiliser de manière efficiente pour résoudre des problèmes d'optimisation combinatoire. Les cas d'application proviennent de l'étude des systèmes de production de biens et de services (problèmes d'ordonnement, de planification, d'équilibrage, ...) ou plus généralement de l'étude de systèmes logistiques (tournee de véhicules, conception de réseaux, ...).

J'ai participé à l'encadrement de 6 thèses soutenues et je co-encadre actuellement 1 thèse. Mes travaux ont donné lieu à l'écriture de **2 ouvrages, 4 chapitres d'ouvrage, 9 articles** dans des revues internationales avec comité de lecture, **3 articles** dans des revues nationales avec comité de lecture. Je suis également co-auteur de **10 conférences internationales** et **5 conférences nationales** avec comité de lecture et publication des actes sur papier complet. Depuis janvier 2021, je m'implique dans la nouvelle structure du GdR ROD (Recherche opérationnelle et Décision), en co-animant l'axe M2HPPC (Méthodes Hybrides, Heuristiques, Programmation Par Contraintes) et en intégrant le conseil scientifique du GdR.

³ "Un entrepôt pédagogique grandeur nature pour les étudiants de Montluçon", Grangeon, N., D. Sarramia, L. Deroussi et A.-C. Palier

"Une pédagogie innovante pour la conception de tableaux de bord logistique", Deroussi, L. et A.-C. Palier

J'appartiens à la communauté STP (Sciences et Techniques de la Production de Biens et de Services) de la SAGIP (Société d'Automatique, de Génie Industriel et de Productique). Je suis membre de plusieurs GT (Origin (Ex Bermudes), GISEH, META, FL)

Je co-anime le GT META (Métaheuristiques et Applications) depuis 2015, un groupe de travail aujourd'hui Comité Technique (CT) de la SAGIP et rattaché au GdR ROD. Dans ce cadre, j'ai participé à l'organisation d'une école de Printemps à Troyes en 2018, animé un module à l'école MACS à Bordeaux en 2019, organisé de nombreuses réunions de groupes (META seul ou en collaboration avec d'autres GT) pour les journées STP, co-proposé des tracks ou des sessions dans des conférences.

J'ai participé à l'organisation des 26èmes journées STP du GdR MACS en 2018.

Je participe à la diffusion et au rayonnement de la recherche au travers de l'évaluation d'articles pour des conférences ou des revues, ou de la participation à des forums de vulgarisation scientifique (Exposcience, Fête de la Science, Orbimob, ...).

1.4. RESPONSABILITES COLLECTIVES

Depuis mon arrivée, je me suis impliqué dans la vie de l'IUT, au niveau de mon département de rattachement (GTE) et du département GLT.

Au fil des ans, j'ai participé à de nombreuses instances (Conseil d'Administration, Commission Scientifique, ...). Je suis actuellement membre élu du comité d'établissement de l'IUT et du conseil de département GTE.

Je me suis investi tout au long du projet A2Ex (Auvergne : Alternance d'Excellence). Ce projet d'un budget de 4.2 M€, a été déposé en 2011 et consistait en la réhabilitation des ateliers du département GEII avec en particulier un entrepôt pédagogique de 300 m². Le nouveau bâtiment a été inauguré en 2017, et les premiers cours dans l'entrepôt (WMS) ont débuté en 2018.

Depuis 2009, je suis co-responsable de la licence professionnelle LOGIN.

Je vais candidater au poste de chef du département MTEE (élection prévue en juin 2023).

2. ACTIVITES PEDAGOGIQUES

2.1. PRESENTATION DE L'ACTIVITE D'ENSEIGNEMENT

Nota bene : Les labels entre crochets se rapportent au tableau de synthèse des enseignements.

A mon arrivée à l'IUT d'Allier en 2003, bien que mon poste fût rattaché au département Génie Thermique et Energie (GTE), j'ai effectué la majeure partie de mon enseignement en Génie Electrique et Informatique Industrielle (GEII). Ce département connaissant des difficultés de recrutement, mon service dans ce département a progressivement diminué jusqu'à ma nomination sur le poste de Maître de Conférences en 2007.

Je suis intervenu en Mathématiques en DUT première année [1a] où on s'est partagé l'enseignement avec un collègue. J'ai assuré les CM sur le semestre 2, et un groupe de TD sur l'année. J'ai monté mes propres supports de cours à partir des documents fournis par le collègue. Nous avons utilisé les fiches de TD déjà existantes.

Le département GEII avait mis en place un dispositif particulier, avec un groupe dit de transition, contenant des étudiants en difficulté pendant la première année et qui suivaient pendant un an des cours adaptés avant de réintégrer la 2^{ème} année du DUT. J'ai eu l'occasion d'enseigner les mathématiques [1b] et la programmation en langage C [1c]. J'ai assuré la responsabilité de ces enseignements.

Je suis intervenu la première année dans la licence professionnelle Responsable de Projet, où j'ai assuré la responsabilité de l'enseignement de base de données [3a].

Je suis également intervenu en mathématiques dans le département Génie Mécanique et Productique (GMP) en DUT 1^{ère} année [1d]. J'avais la charge d'un groupe de TD sur toute

l'année. Le CM et les autres groupes TD étaient assurés par le collègue de mathématiques du département. Les supports étaient fournis. Je ne suis intervenu sur cet enseignement qu'une seule année.

Au sein du département GTE, j'ai pris la responsabilité de l'enseignement de mathématiques au niveau de la deuxième année depuis mon arrivée. Cet enseignement a changé de par son contenu, au gré de l'évolution du Programme Pédagogique National (PPN), et son volume horaire ([2a] en 2003-04, [2b] et [2c] actuellement). A l'origine, j'enseignais les mathématiques de manière classique, avec des séances de CM en classe entière, et des séances de TD. Aujourd'hui, le module de mathématiques est découpé en deux parties bien distinctes. Le premier s'appelle "Mathématiques Appliquées" au semestre 3. L'accent est mis ici sur l'utilisation des mathématiques comme des outils au service de la Physique. Le deuxième s'intitule "Mathématiques pour l'ingénieur" au S4. Il n'est suivi que par le groupe d'étudiants souhaitant faire des poursuites d'études longues. Des notions plus abstraites telles que les espaces vectoriels ou les applications linéaires y sont abordées, avec un public un peu plus aguerri. Les cours magistraux ont disparu de mon enseignement en Mathématiques, car ils ne sont plus du tout adaptés au public que nous recevons (2/3 de bacheliers technologiques). L'objectif visé est que les étudiants soient tous acteurs et acquièrent des savoir-faire, chacun à leur niveau. Même si je ne mentionne pas dans cette synthèse, il m'est arrivé d'intervenir ponctuellement dans le cours de Mathématiques de GTE de première année, pour pallier des absences (congés maternité par exemple)

J'ai eu l'occasion d'intervenir en bureautique au sein du département GTE. D'abord en DUT première année de 2003 à 2007 sur un enseignement standard Word-Excel, réalisé en collaboration avec des collègues, puis en deuxième année depuis 2015 où j'assume la responsabilité d'un module "Informatique" dans lequel les étudiants sont initiés au solveur d'Excel et à VBA. Je travaille en général avec 1 groupe de TP. Les autres groupes sont pris en charge par un jeune collègue (ATER ou contractuel).

De 2007 à 2014, j'ai pris la responsabilité de l'enseignement "Optimisation Logistique" en DUT GLT 2^{ème} année. Les étudiants devaient proposer des solutions à des problèmes de distribution. Depuis l'ouverture de la licence professionnelle LOGIN en 2004, j'ai eu l'occasion d'intervenir dans de nombreux enseignements.

Certaines de ces interventions étaient ponctuelles pour pallier des besoins immédiats. J'ai ainsi pris la responsabilité du cours d'EDI en 2004 (Echange de Données Informatisé), le temps de trouver un intervenant. Je suis ponctuellement intervenu en support de l'enseignant responsable du cours, dans des TP tels que la simulation [3c] ou la planification [3d].

J'ai assuré également des enseignements sur la durée, avec un cours d'optimisation [3e], découpé par la suite en ordonnancement [3f] et localisation [3g]. Ces cours sont l'occasion de mettre à disposition des étudiants de Licence Professionnelle des éléments d'Aide à la Décision, au travers d'études de cas rencontrés dans mes activités de recherche. Ils constituent une version simplifiée de cours de Recherche Opérationnelle classiques et qui sont davantage dispensés à un niveau Master.

Depuis la dernière évolution de la maquette, je me suis investi dans des enseignements nouveaux pour moi, qui m'ont obligé à sortir de ma zone de confort. Ces cours portent sur l'apprentissage du logiciel de gestion d'entrepôt qui est implanté dans l'entrepôt pédagogique (WMS) [3h] et sur les tableaux de bord [3i]. Ces cours ont également été pour moi l'occasion de faire de l'innovation pédagogique comme il sera détaillé dans la suite.

J'interviens régulièrement dans la formation d'ingénieur en alternance "Génie des Systèmes de Production" (GSP) porté par Polytech'Clermont. J'intervenais au début de la formation en 2008 en 1^{ère} année sur le cours de Mathématiques jusqu'au recrutement d'un collègue en mathématiques en 2012. J'ai assuré ensuite pendant deux ans de 2016 à 2018 le module de 3^{ème} année en théorie de graphes (module composé également d'une partie en programmation

linéaire dispensé par une collègue), et depuis 2018 le module de mathématiques de 2^{ème} année en probabilités-statistiques.

Je suis intervenu de 2016 à 2019 dans le M1 d'Informatique [4b] rattaché à l'ISIMA, sur le module "Méthodes approchées", où j'assurai la partie "Méthodes d'optimisation" en liaison avec une collègue qui assurait la partie "Méthodes d'évaluation des performances".

Depuis 2014, je suis coresponsable d'un module de l'Ecole Doctorale Sciences pour l'Ingénieur. "Métaheuristiques pour l'optimisation combinatoire : Évolutions et tendances actuelles" [6a]. Ce module est destiné à faire découvrir le domaine des métaheuristiques et de l'optimisation combinatoire à un public de jeunes chercheurs.

En complément des activités d'enseignements que je viens d'énumérer, j'ai eu l'occasion d'intervenir dans les éléments suivants :

- Encadrements de projets tuteurés réalisés par les LP LOGIN. Depuis plusieurs années et la mise en place d'un calendrier d'alternance, ces projets tuteurés sont réalisés sur des semaines bloqués et permettent de faire travailler les étudiants sur des projets avec des réalisations concrètes (organiser des activités dans le cadre de la Fête de la Science, Faire réaliser l'inventaire de l'entrepôt à des étudiants de 1^{ère} année) ou sur des missions en collaboration avec des entreprises (FPT, ATAC Logistique, ...). Certains de ces projets tuteurés pourront être utilisés comme SAE (Situation d'Apprentissage et d'Evaluation) en BUT MLT3.

- Suivi de stagiaires en entreprise pour leur stage de fin d'étude que ce soit pour le DUT GTE, pour les licences professionnelles LOGIN ou BEPCB (Bâtiments à Energie Positive, Construction Bois), ou pour le master Génie Industriel, à raison de 4 à 6 étudiants par an.

- Tuteur académique d'étudiants en alternance pour la LP LOGIN (2 à 3 par an depuis 2011) ou GSP (1 étudiant tous les 3 ans)

- Depuis 2018, encadrement d'étudiants du master Génie Industriel pour effectuer une recherche bibliographique sur un thème donné (2 par an)

- Depuis 2022, j'interviens dans la matière "Optimisation et aide à la décision" du master Génie Industriel [5b], à hauteur de 6h CM et 4h TD pour une introduction aux métaheuristiques.

2.2. PRESENTATION SYNTHETIQUE DES ENSEIGNEMENTS

Dans cette partie, mes enseignements sont regroupés en trois tableaux dans lesquels vous retrouverez le niveau d'enseignement, l'intitulé du cours et les volumes horaires étudiants.

Du point de vue des effectifs, les promotions de DUT, tout départements confondus, sont d'au plus 56 étudiants avec 2 groupes TD et 4 groupes TP. Le groupe de transition de GEII était composée de moins de 10 étudiants.

Les effectifs de la LP LOGIN sont en moyenne de 20 étudiants.

Pour les formations hors IUT, une promotion de GSP oscille entre 10 et 15 étudiants. Les cours du master Informatique étaient suivis par une trentaine d'étudiants (il s'agit d'un module optionnel dans la formation), et les cours de l'école doctorale sont suivis par 20 à 30 doctorants chaque année.

Dans ce premier tableau, j'ai choisi de classer d'abord les enseignements les plus anciens, qui correspondent aux activités pédagogiques de mon début de carrière (PRCE), essentiellement en DUT 1^{ère} année et en mathématiques.

Niveau	Intitulé du cours	CM	TD	TP
1 ^{ère} année de DUT	[1a] Mathématiques (GEII) <i>Etudes de fonctions, calcul intégral, équations différentielles, polynômes et fractions rationnelles ...</i>	20	60	16
	[1b] Mathématiques (GEII – groupe de transition) <i>Idem [1a]</i>		22,50	

	[1c] Programmation (GEII – groupe de transition) <i>Langage C</i>	9	13,50	25
	[1d] Mathématiques (GMP) <i>Idem [1a]</i>		64	
	[1e] Bureautique (GTE) <i>Word - Excel</i>		4	24
LP Resp. de projet	[3a] Base de données <i>Méthode d'analyse et de conception de base de données (MERISE)</i>	2	3	10

Depuis une quinzaine d'années, j'interviens quasi-exclusivement en DUT 2^{ème} année et en licence professionnelle LOGIN. Sur plusieurs de ces cours, j'ai eu l'occasion de partager les enseignements avec des professionnels, des doctorants, des contractuels ou des collègues de l'IUT.

Niveau	Intitulé du cours	CM	TD	TP
2 ^{ème} année de DUT	[2a] Mathématiques (GTE) <i>Transformées de Laplace, suites et séries, séries de Fourier, Intégrales multiples, algèbre linéaire, ...</i>	24	51,25	6
	[2b] Mathématiques appliquées (GTE) <i>Transformées de Laplace, Opérateurs différentiels, Intégrales curvilignes, calcul matriciel</i>		54	
	[2c] Mathématiques pour l'ingénieur (GTE) <i>Développements limités, Séries de Fourier, Algèbre linéaire</i>	9	18	
	[2d] Informatique (GTE) <i>Solveur linéaire Excel, programmation VBA</i>	4		20
	[2e] Optimisation logistique (GLT2) <i>Etude de cas : Voyageur de commerce, Tournées de véhicules, arbres couvrants de coûts min</i>		10	
LP LOGIN	[3b] Echange de Données Informatisé <i>Traçabilité, norme EDIFACT, ...</i>	10	0	0
	[3c] Simulation <i>Etudes de cas avec le logiciel WITNESS</i>	0	0	20
	[3d] Planification <i>Programmation des méthodes de calculs de taille de lots</i>			10
	[3e] Optimisation <i>Méthodes d'optimisation – application à des problèmes de tournées de véhicules et d'ordonnancement</i>	10	10	20
	[3f] Ordonnancement <i>Initiation avec le jeu "Fourmis Racers", résolution de problèmes d'ordonnancement</i>	5	5	10
	[3g] Localisation <i>Problèmes de localisation, conception de réseaux logistiques</i>	5	5	10
	[3h] WMS <i>Configuration des emplacements, gestion du stockage, réception, préparation de commandes, ...</i>	0	0	20

	[3i] Tableaux de bord <i>Pilotage de l'entreprise par les processus, mesure avec des indicateurs clés, conception de tableaux de bord</i>	4	4	8
--	--	---	---	---

Dans ce dernier tableau sont regroupés les enseignements dispensés à l'extérieur de l'IUT.

Niveau	Intitulé du cours	CM	TD	TP
GSP3 (1 ^{ère} année)	[3j] Mathématiques <i>Séries de Fourier, Algèbre linéaire</i>	20	20	
GSP4 (2 ^{ème} année)	[4a] Mathématiques <i>Combinatoire, Probabilités, Statistiques</i>	20	10	10
GSP5 (3 ^{ème} année)	[5a] Mathématiques <i>Théorie des graphes (plus courts chemins, flots min, arbres couvrants, ...)</i>	10	10	0
M1 Informatique	[4b] Méthodes approchées <i>Optimisation combinatoire, problèmes NP-difficiles, Heuristiques et Métaheuristiques</i>	4	6	4
ED SPI	[6a] Métaheuristiques pour l'optimisation combinatoire <i>Jour 1 : Problèmes d'optimisation et méthodes de résolution</i> <i>Jour 2 : Etudes de cas (réalisé par une collègue)</i> <i>Jour 3 : Hybridation et couplage, optimisation multicritère, application sur le TSP</i>			
M2 Génie Industriel	[5b] Optimisation et aide à la décision <i>Introduction aux métaheuristiques</i>	6	4	

2.3. FORMATIONS SUIVIES CONCERNANT VOS ACTIVITES PEDAGOGIQUES

J'ai suivi plusieurs formations, dans le but de prendre en main les nouveaux outils mis à notre disposition. Parmi les formations organisées par l'Université :

- Février 2014 : Vidéo projecteur interactif
- Octobre 2014 : les bases de la plateforme pédagogique Claroline
- Août 2020 : Prise en main de Teams

Ces formations trouvent leur pleine utilité en cette période de crise sanitaire pour assurer des cours en distanciel dans les meilleures conditions pédagogiques possible. J'utilise par exemple un speechi (écran interactif tactile) pour les cours de mathématiques, Teams et Moodle pour tous les cours en distanciel.

J'ai également suivi une journée de formation sur les blocs de compétences il y a 5-6 ans, qui a eu une influence certaine dans l'évolution de la dernière maquette pédagogique de la LP LOGIN (les 6 modules théoriques qui composent la maquette sont pensés comme des blocs de compétences).

Dans le cadre du projet A2Ex et plus particulièrement dans l'exploitation de l'entrepôt pédagogique, en tant que membre de l'équipe projet, j'ai participé à l'installation du matériel et au paramétrage du WMS maGistor (réunion/formation de deux jours tous les deux mois pendant un an, en 2017-18). Nous avons été trois enseignants-chercheurs à suivre une formation de deux jours tous les deux mois pendant un an. L'objectif était de maîtriser le logiciel pour

pouvoir assurer l'ensemble des activités pédagogiques qui seront proposées dans l'entrepôt. Nous avons depuis lors construit un cours sur l'apprentissage du logiciel et de son paramétrage, qui a débuté en 2018, et proposé plusieurs activités aux étudiants dans le cadre des projets tuteurés.

2.4. RESPONSABILITES PEDAGOGIQUES

En 2007, j'ai participé au **montage de la formation d'ingénieur en alternance GSP** (Génie des Systèmes de Production) de Polytech'Clermont. J'ai notamment écrit les modules Mathématiques 1 et 2 et participé à l'écriture du module Mathématiques 3.

Depuis 2009, j'assure la **co-responsabilité de la Licence Professionnelle LOGIN**. Je participe depuis à l'ensemble des tâches relatives à cette fonction (emplois du temps, relations avec les entreprises, gestion des vacataires, communication et recrutement, conseils de perfectionnement, suivi des diplômés, ...)

J'ai corédigé depuis cette date les dossiers d'autoévaluation (2010), de réhabilitation (2012), d'évaluation (2015) et d'accréditation (2016). Au travers de tous ces dossiers, nous avons régulièrement fait évoluer les objectifs et la maquette de la formation pour toujours mieux répondre aux besoins des entreprises.

En 2011, j'ai travaillé sur **l'ouverture de la Licence professionnelle LOGIN à l'alternance**, en contrat de professionnalisation : construction du calendrier, recherche d'entreprises et de contrat. Ce travail a abouti à une ouverture en 2011 avec 4 alternants, puis une progression jusqu'à l'ouverture en apprentissage en 2016 de la formation. J'ai corédigé le dossier de demande d'ouverture en apprentissage qui a été acceptée à la première demande. Nous accueillons ces dernières années une quinzaine d'apprentis (pour une vingtaine étudiants).

Depuis 2017, j'ai participé au **montage d'un dossier de Soutien à l'Innovation Pédagogique** (SIP) dans le cadre d'un appel à projet de l'Université Clermont Auvergne pour lequel nous avons obtenu 100h eq TD enseignant, 60h d'accompagnement en ingénierie pédagogique et 15h en accompagnement audiovisuel.

Dans le cadre de ce projet, j'ai notamment mis en œuvre une activité de formation des étudiants par les étudiants où les étudiants de LOGIN organisent l'inventaire complet de l'entrepôt, qui est réalisé par les étudiants de 1ère année, dans le cadre de leur cours de gestion d'entrepôt. Une analyse de cette activité et une restitution sont également réalisées en aval.

Je participe actuellement à un projet de **montage d'une formation MIAGE**, pour une ouverture envisagée à la rentrée 2024.

3. ACTIVITES SCIENTIFIQUES

3.1. PRESENTATION SYNTHETIQUE DES THEMATIQUES DE RECHERCHE

Depuis le début de ma carrière, j'effectue ma recherche au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS UMR CNRS 6158). Mes travaux de recherche s'inscrivent dans l'axe ODPS (Outils Décisionnels pour la Production et les Services).

Ces activités de recherche concernent plus spécifiquement le domaine de l'optimisation combinatoire, appliqué aux secteurs industriel et hospitalier. Elles peuvent être regroupées en plusieurs thèmes portant d'une part sur les problèmes étudiés et d'autre part sur les méthodes mises en œuvre pour résoudre ces problèmes.

Le premier thème que j'aborderai porte sur l'étude **des problèmes de tournées de véhicules** :

Le problème du voyageur de commerce est le premier problème d'optimisation combinatoire que j'ai étudié, avant et pendant ma thèse. Depuis, j'ai eu l'occasion d'étudier de nombreux autres

problèmes de tournées de véhicules qui peuvent être présentés selon certaines caractéristiques qu'ils possèdent.

- **la synchronisation des ressources.** Nous avons étudié les systèmes flexibles de production qui combinent un problème d'ordonnancement de type job-shop avec un problème de tournées de véhicules. Dans de tels systèmes, il s'agit de synchroniser les ressources de transport (des véhicules automatiquement guidés) avec les ressources de production.

Dans le cadre du co-encadrement de la thèse de Fabrice Gayraud, nous nous sommes intéressés à l'organisation des soins à domicile, dans lequel il s'agit d'organiser la tournée du personnel soignant. Une collaboration a été initiée avec l'hôpital de Montluçon. La réalisation de certaines activités de soins nécessite la présence de deux soignants, d'où la nécessité de synchroniser les tournées.

- **l'hétérogénéité des ressources de transport.** Dans le cadre du co-encadrement de la thèse de Raïssa Saleu, nous nous plaçons dans le contexte de la logistique du dernier kilomètre et nous nous intéressons à une classe émergente de problèmes de distribution incluant des véhicules classiques et des drones. Il s'agit de décider à quel type de véhicules seront affectés chaque livraison pour optimiser la distribution

Le second thème porte sur l'étude des **problèmes d'atelier (ordonnancement et équilibrage):**

Nous avons travaillé sur le problème du flow-shop de permutation. Pour ce problème, qui est un classique, nous avons proposé des méthodes de résolution basées sur un voisinage innovant à base de recherche arborescente, inspiré du voisinage Lin & Kernighan pour le voyageur de commerce.

Dans le cadre du co-encadrement de la thèse d'Arnaud Laurent, nous nous sommes intéressés à un problème de gestion de projet avec contraintes de ressources (RCPS) en contexte Multi-Site. Des modélisations mathématiques et plusieurs approches basées sur des métaheuristiques ont été proposées et comparées.

Dans le cadre de la thèse de Youssef Lahrichi, nous nous plaçons dans le contexte de l'usine du futur. Nous avons étudié les lignes de production reconfigurables et robotiques. L'originalité du travail réalisé tenait en la prise en compte de temps de réglage dépendant de la séquence, nécessitant une optimisation de la séquence des opérations affectées à chaque poste de charge.

Le troisième thème porte sur l'étude du comportement **des métaheuristiques :**

Comprendre le bon fonctionnement des métaheuristiques pique ma curiosité scientifique depuis mes travaux de thèse. Nous avons alors proposé d'évaluer la performance d'un système de voisinage en mesurant deux valeurs que nous avons appelées la combinatoire et la profondeur. Ces valeurs permettaient de quantifier l'équilibre entre la vitesse à laquelle on obtient une solution, et la qualité de celle-ci.

Dans le cadre des travaux de thèse de Sylvie Huet, nous avons étudié les modèles de diffusion de l'innovation. Même si le contexte de l'étude semblait *a priori* éloignée des métaheuristiques, nous voulions comprendre en quoi de tels modèles pourraient être utilisés pour diffuser la découverte d'une bonne solution au sein d'un algorithme à essais particuliers.

A la suite de ces travaux, et dans le cadre d'un appel à projet du Réseau National des Systèmes Complexes (RNSC) et des travaux de thèse de Benjamin Louvet, nous avons travaillé sur l'étude des échanges d'information dans la méthode d'Optimisation par Essaim Particulaire et l'application à la génération *in silico* d'organes humains.

Le quatrième et dernier thème porte sur les **types d'approches de résolution** dans le cadre de problèmes de décision multiple. Parmi les quelques problèmes que nous avons énumérés, nous

retrouvons plusieurs niveaux de décision. Le problème étudié par Youssef Lahrichi consiste à équilibrer les postes de charges et à séquencer les opérations. Dans le cas de Raïssa Saleu, le problème consiste à répartir les clients à livrer et à les séquencer sur les véhicules.

Selon que l'on cherche à résoudre l'ensemble des décisions de manière globale, ou à les traiter de manière séquentielle (Séquencement d'abord, répartition ensuite ou répartition d'abord, séquencement ensuite), nous allons proposer des méthodes de résolution extrêmement différentes. C'est ainsi que dans ces deux travaux de thèse, des méthodes originales ont été proposées, qui hybrides des composant venant de plusieurs domaines : métaheuristiques, méthodes exactes, programmation dynamique. Malgré des problématiques très différentes, ces travaux ont conduit à la proposition de méthodes similaires, à la fois efficaces et originales, basées sur le séquencement des clients ou des opérations puis une répartition optimale sur les ressources de transport ou de production avec un algorithme polynomial de type SPLIT.

3.2. ENCADREMENT DOCTORAL ET SCIENTIFIQUE

Encadrement de thèses soutenues

1. **Sylvie Huet** : de septembre 2008 à janvier 2013

« Individual based models of social systems: data driven hybrid micro-models of rural development and collective dynamics of filtering or rejecting messages»,

Co-encadrement avec Guillaume Deffuant (33%) et Michel Gourmand (33%).

Autofinancement : S. Huet était salariée par l'IRSTEA sur un poste de technicienne

Soutenance le 15 janvier 2013

Sylvie a continué à travailler à l'IRSTEA et a pu s'impliquer davantage dans les travaux de recherche

2. **Benjamin Louvet** : de octobre 2009 à juillet 2014

« Automates cellulaires et métaheuristiques pour la modélisation multi-échelle de systèmes biologiques: application en radiobiologie»,

co-encadrement avec Michel Gourmand (20%) et David Sarramia (40%)

Financement Conseil Général de l'Allier - Participation à la rédaction du dossier de demande de bourse au Conseil Général de l'Allier.

Soutenance le 11 juillet 2014

Benjamin est retourné dans le privé, dans le secteur de la bio-informatique

3. **Fabrice Gayraud** de janvier 2012 à juillet 2015

« Problème de tournées de véhicules avec contraintes de synchronisation dans le cadre de structures de maintien à domicile »

Co-encadrement avec Nathalie Grangeon 33%, Sylvie Norre 33%

Financement Conseil Général de l'Allier- Participation à la rédaction du dossier de demande de bourse au Conseil Général de l'Allier.

Soutenance le 13 juillet 2015

Ces travaux ont donné lieu à 8 publications. Fabrice est ingénieur Supply Chain chez Airbus.

4. **Arnaud Laurent** de octobre 2013 à novembre 2017

« Proposition et étude d'une extension du RCPSP pour la mutualisation entre plusieurs sites : définition, formalisation, méthodes exactes et méta-heuristiques »

Co-encadrement avec Nathalie Grangeon 33%, Sylvie Norre 33%

Financement : Bourse MENRT

Soutenance le 10 octobre 2017

Ces travaux ont donné lieu à 10 publications. Arnaud est Maître de Conférences à l'Université de Nantes.

5. **Raïssa Mbiadou Saleu** de Octobre 2016 à Janvier 2021

« Méthodes exactes et heuristiques pour des systèmes de livraison composés de véhicules et de drones sans synchronisation »

Co-encadrement avec Dominique Feillet 25%, Nathalie Grangeon 25%, Alain Quilliot 25%

Financement : Bourse Laboratoire d'Excellence Mobilité Innovante Imobs3

Ces travaux ont donné lieu à 6 publications. Raïssa a obtenu un contrat postdoctoral en apprentissage artificiel au LIMOS.

6. **Youssef Lahrichi** de Octobre 2017 à Janvier 2021

« Méthodes d'optimisation pour la gestion des lignes de production flexibles et reconfigurables »

Co-encadrement avec Nathalie Grangeon 33%, Sylvie Norre 33%

Financement : Allocation doctorale I-Site – Cap 2025 – Systèmes et services intelligents pour la production et les transports – Participation à la rédaction du dossier de réponse à l'appel à projets

Ces travaux ont donné lieu à 6 publications. Youssef est en contrat postdoctoral à l'Ecole de Management de Strasbourg.

Encadrement de thèses en cours

7. **Arnauld Tuyaba** à partir de Septembre 2022

« Problèmes de conception de réseaux et d'optimisation de tournées autour de la mise en place de circuits courts alimentaires de proximité.»

Co-encadrement avec Nathalie Grangeon 33%, Sylvie Norre 33%

Financement Conseil Général de l'Allier- Participation à la rédaction du dossier de demande de bourse au Conseil Général de l'Allier.

Encadrement de stages

Hector Gatt, Post-doctorat, début en juin 2023

« Aide à la conception d'une usine du futur (ultra-flexible) : dimensionnement d'une flotte d'AGV »

Ce post-doctorat est financé par l'IRC ITPS (International Research Center Innovation Transportation and Production Systems), dans le cadre du Factolab (laboratoire public-privé UCA, CNRS, Michelin).

3.3. DIFFUSION ET RAYONNEMENT

- J'appartiens à la communauté STP (Sciences et Techniques de la Production de Biens et de Services) de la SAGIP (Société d'Automatique, de Génie Industriel et de Productique).
- **Relecture d'articles** pour des revues (EAAI, DAM, CAIE, IJPR, JOH, RAIRO, ANOR, 4Or, Discrete Dynamics in Nature and Society, IEEE Transactions, IJSI, SOCO) et des conférences (EA, CIE, GECCO, ICSI, IESM, META, MIC, MOSIM, ROADEF, ...)
- **Examineur des thèses de :**

- Abbas El Dor, « *Perfectionnement des algorithmes d'optimisation par essaim particulière : applications en segmentation d'images et en électronique* », dirigée par Patrick Siarry et Maurice Clerc, soutenue le 05 décembre 2012 à l'Université Paris-Est

- Nizar Triki, « *Planification des soins à domicile* », dirigée par Xiaolan Xie et Thierry Garaix, soutenue le 06 octobre 2016 à l'école des Mines de Saint-Etienne.

- **Co-animateur de l'axe MH2PPC** (Méthodes Hybrides, Heuristiques, Programmation Par Contraintes) depuis janvier 2021. Cette responsabilité s'inscrit dans la restructuration du **GdR ROD**. Notre travail consistera à animer la communauté en soutenant d'une part

les activités des GdT existants ("META" et "RO et Contraintes" et en structurant d'autres part les communautés métaheuristiques et programmation par contraintes. J'intègre dans ce cadre le conseil scientifique du GdR ROD.

- **Membre de comité scientifique**

MOSIM 2020, CoDit 2023.

- **Membre de comité d'organisation**

- 14èmes rencontres de l'AIIRL-SCM en 2022 à Clermont-Ferrand
- Journées STP du GdR MACS en 2018 à Clermont-Ferrand
- Congrès ROADEF'2008 à Clermont-Ferrand.

- **Organisation de sessions dans des conférences**

- Conférence MOSIM 2010, Session « Modelling and Optimisation in bio integrative research » avec Michel Gourgand, Nathalie Grangeon et David Sarramia
- ICSI 2011, Session « Swarm Intelligence for Biological Applications », avec David Sarramia
- IFAC 2016, « Hyper-Heuristics: new trends in theoretical and practical approaches », avec G. Ochoa et P. Siarry

Dans le cadre de mes activités au sein du GdT META, nous organisons des sessions ou des tracks sur les métaheuristiques pour chaque conférence META, MOSIM et ROADEF

- **Co-animateur du GdT META** (Métaheuristiques et Applications), groupe de travail rattaché aux GdR ROD et Comité Technique de la SAGIP, depuis 2015. J'ai été invité à rejoindre l'animation du groupe pour travailler au développement des métaheuristiques pour les jeunes chercheurs. Dans ce cadre, j'ai participé à l'organisation d'une école de Printemps à Troyes en 2018, animé un module à l'école MACS à Bordeaux en 2019, organisé de nombreuses réunions de groupe pour les journées STP, co-proposé des tracks ou des sessions dans des conférences. J'ai également participé à la rédaction des rapports d'activité du groupe, et à de nombreuses réunions intra-GdT ou inter GdT pour présenter le bilan des activités au GdR MACS par exemple. J'administre également le site internet du GdT.

4. RESPONSABILITES COLLECTIVES

4.1. PRESENTATION GENERALE DES RESPONSABILITES

Depuis ma nomination, je me suis investi dans de nombreuses tâches administratives. Les circonstances locales ont fait que je me suis davantage impliqué au niveau du département GLT qu'à mon département de rattachement GTE, mais j'ai à cœur de représenter GTE dans toutes les instances auxquelles j'ai participé.

Je participe depuis ma nomination à la gestion **du recrutement** en GTE et depuis 2009 à la gestion du recrutement en LP LOGIN.

En LP LOGIN, je m'occupe de la gestion **des absences**, des **modalités de contrôle des connaissances**, des **notes**, des **emplois du temps**, des **vacataires**.

Je participe régulièrement à des **forums** où je représente le département GTE, et j'ai fait de multiples présentations de la LP LOGIN pour des formations de BTS ou DUT.

Je suis investi dans la vie de mon établissement, en représentant le département GTE auprès des différents **conseils** (conseil de département, comité d'établissement, conseil d'IUT, commission recherche)

Je suis **coresponsable de la licence professionnelle LOGIN** depuis 2009.

Je me suis également fortement impliqué dans les projets et manifestations au niveau de l'IUT : le **projet A2Ex**, la **Fête de la Science** ou plus récemment **Orbimob** (association qui a pour vocation de regrouper les acteurs territoriaux autour des enjeux de la mobilité en Auvergne), avec l'organisation d'activités et la présentation de plusieurs conférences.

4.2. RESPONSABILITES ADMINISTRATIVES

Missions et gestion de projets de l'établissement

Depuis 2009 : coresponsabilité de la LP LOGIN :

- En complément des tâches de gestion décrites ci-dessus, j'ai coréalisé les tâches qui incombent à un responsable de formation : accueil et suivi des étudiants, recherche de vacataires, communication (site web, plaquettes, affiches, ...), organisation des évaluations des enseignements et des conseils de perfectionnement.
- J'ai co-rédigé les **dossiers d'autoévaluation** de 2010, de **réhabilitation** de 2012, **d'évaluation** de 2015 et **d'accréditation** de 2016 et 2019.
- J'ai travaillé à l'ouverture à l'alternance de la formation avec une première phase en contrat de professionnalisation puis en contrat d'apprentissage. J'ai co-rédigé le dossier de **demande d'ouverture en apprentissage** en 2016. Cette ouverture a été acceptée à la première demande.
- J'ai participé entre 2015 et 2019 à des réunions de pilotage des licences professionnelles, qui avaient pour but d'harmoniser les pratiques des différentes LP de l'IUT et de proposer collectivement des solutions pour mettre en œuvre les politiques d'économies demandées par l'Université.

En dehors de cette tâche administrative, je me suis impliqué dans des projets d'envergure, qui avaient pour objectif de promouvoir les formations de l'IUT et les activités scientifiques qui y sont menées. Je présente trois projets sur lesquels je me suis impliqué fortement.

J'ai participé de manière très active au **projet A2EX**, qui visait à réhabiliter les ateliers du département GEII. Ce projet d'envergure, d'un budget de 4.2 M€ et qui a duré plus de 6 ans, a permis d'impliquer 4 départements autour du thème de l'économie verte.

Le projet a été déposé en 2011. La première pierre a été posée en septembre 2015 et le bâtiment a été inauguré en février 2017.

Je me suis investi au niveau du département GLT en participant :

- à la rédaction du pré-projet sous la forme d'un entrepôt mutualisé.
- en co-animant le groupe Formation, qui avait pour but de recenser les besoins des formations en termes de plateaux techniques et du matériel pédagogique en vue du développement de l'alternance
- en participant au comité de pilotage du projet
- en intégrant l'équipe en charge de l'installation du matériel dans l'entrepôt et de l'intégration du logiciel de pilotage (WMS). Pour la petite histoire, l'inauguration du bâtiment s'est déroulée dans l'entrepôt pédagogique, qui était alors complètement vide. Nous avons été formés et avons pu commencer les activités pédagogiques en mars 2018 avec la LP LOGIN

Je participe également activement à la **Fête de la Science** depuis son lancement à l'IUT en 2013. L'objectif est d'accueillir dans nos locaux des collégiens ou lycéens, et de leur faire découvrir de manière ludique les activités scientifiques que nous menons.

- J'ai présenté en 2013 une conférence sur le thème de la chaîne logistique du lait (pour le département GLT)

- Depuis 2013, les étudiants de LOGIN animent un atelier autour du BeerGame où ils font jouer les élèves. Cette activité est préparée dans le cadre des projets tuteurés.

Enfin, je me suis récemment investi dans le projet **Orbimob'** qui vise à faire de l'Université un acteur incontournable sur le thème de la mobilité durable au cœur de l'Auvergne.

Dans ce cadre, nous avons réalisé un poster résumant nos activités de recherche sur le sujet, et nous souhaitons présenter les travaux de Raïssa Saleu de la livraison par drones.

La première journée devait se dérouler en novembre 2020 et les manifestations que nous avions prévues ont dû malheureusement être annulées.

4.3. RESPONSABILITES ET MANDATS LOCAUX OU REGIONAUX

J'ai participé depuis mon arrivée aux différentes instances qui régissent la vie de mon établissement.

- Membre élu du **conseil du département** Génie Thermique et Energie de l'IUT d'Allier de septembre 2004 à septembre 2007 et depuis 2020.

- Membre élu au **Conseil de l'IUT d'Allier**, en tant que représentant du collège des Maîtres de Conférences, de 2010 à 2014

- Membre élu du **Comité d'Etablissement** de l'IUT d'Allier de 2020 à 2022.

- Membre de la **Commission Recherche** de l'IUT de 2010 à 2016.

4.4. ORGANISATION DE CONGRES

Membre de comité scientifique de MOSIM 2020

Membre de comité d'organisation

2022 : 14èmes rencontres de l'AIRL-SCM à Clermont-Ferrand

2018 : Journées STP du GdR MACS en 2018 à Clermont-Ferrand

2008 : Congrès ROADEF'2008 à Clermont-Ferrand.

5. LISTE DES TRAVAUX ET PUBLICATIONS

5.1. SYNTHÈSE DES PUBLICATIONS

Type de publication	Nombre	Lieu de publication
Ouvrage	1	Wiley & Sons (2016)
Chapitre d'ouvrage	3	2014 (2), 2016
Articles dans une revue	11	EJOR (2021), JoH (2021), IJPR (2008, 2021, 2022), Networks (2018), CIE (2017), RAIRO (2015), NC (2014), IJAMC (2011), JESA (2006)
Communications dans un congrès	15	
Rapport interne LIMOS	1	

5.2. OUVRAGE

(1) Laurent Deroussi. *Metaheuristics for Logistics*. John Wiley Sons, Inc., March 2016.
doi: 10.1002/9781119136583. URL <https://hal.uca.fr/hal-02059623>.

5.3. CHAPITRES D'OUVRAGE

(2) Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Optimization of logistics systems using metaheuristic based hybridization techniques. In *Metaheuristics*. 2016.
URL <https://hal.archives-ouvertes.fr/hal-02023692>

(3) Laurent Deroussi and Michel Gourgand. A scheduling Approach for the design of Flexible Manufacturing Systems. In *Heuristics: Theory and Applications*, page Chapter 9. Nova science publishers, 2014.
URL <https://hal.uca.fr/hal-03263543>.

(4) Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Techniques d'hybridation à base de métaheuristiques pour optimiser des systèmes logistiques. In *Métaheuristiques pour l'optimisation difficile*. 2014.
URL <https://hal.archives-ouvertes.fr/hal-02023663>

5.4. ARTICLES DANS UNE REVUE

(5) Youssef Lahrichi, David Damand, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Investigating two variants of the sequence-dependent robotic assembly line balancing problem by means of a split-based approach. *International Journal of Production Research*, April 2022.
doi: 10.1080/00207543.2022.2062266.

(6) Raïssa G Mbiadou Saleu, Laurent Deroussi, Dominique Feillet, Nathalie Grangeon, and Alain Quilliot. The parallel drone scheduling problem with multiple drones and vehicles. *European Journal of Operational Research*, September 2021.
doi: 10.1016/j.ejor.2021.08.014.
URL <https://hal.archives-ouvertes.fr/hal-03448521>.

(7) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. A Balance-First Sequence-Last Algorithm to design RMS A Matheuristic with performance guaranty to balance Reconfigurable Manufacturing Systems. *Journal of Heuristics*, 27:107–132, March 2021a.
doi: 10.1007/s10732-021-09473-1.
URL <https://hal.uca.fr/hal-03263525>.

(8) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. A new split-based hybrid metaheuristic for the Reconfigurable Transfer Line Balancing Problem. *International Journal of Production Research*, 2021b.
Doi: 10.1080/00207543.2020.1720929
URL <https://hal.archives-ouvertes.fr/hal-03165254>.

(9) Raïssa G Mbiadou Saleu, Laurent Deroussi, Nathalie Grangeon, Alain Quilliot, and Dominique Feillet. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, 72(4):459–474, December 2018.
doi: 10.1002/net.21846.
URL <https://hal.archives-ouvertes.fr/hal-01952007>

(10) A. Laurent, L. Deroussi, Sylvie Norre, and Nathalie Grangeon. A new extension of the RCPSp in a multi-site context: Mathematical model and metaheuristics. *Computers Industrial Engineering*, 112:634–644, 2017

doi: 10.1016/j.cie.2017.07.028.

URL <https://hal.archives-ouvertes.fr/hal-01952840>

(11) Fabrice Gayraud, Nathalie Grangeon, Laurent Deroussi, and Sylvie Norre. Notation and classification for logistic network design models. *RAIRO - Operations Research*, 49(1):195–214, January 2015.

URL <https://hal.archives-ouvertes.fr/hal-01952920>.

(12) Abbas El Dor, David Lemoine, Laurent Deroussi, Michel Gourgand, Patrick Siarry, and Maurice Clerc. Dynamic cluster in particle swarm optimization algorithm. *Natural Computing*, pages 1–18, 2014.

URL <https://hal.archives-ouvertes.fr/hal-01155162>.

(13) Laurent Deroussi and David Lemoine. Discrete Particle Swarm Optimization for the Multi-Level LotSizing Problem. *International Journal of Applied Metaheuristic Computing*, 2(1):44–57, 2011.

doi: 10.4018/jamc.2011010104.

URL <https://hal.archives-ouvertes.fr/hal-00860165>.

(14) Laurent Deroussi, Michel Gourgand, and Nikolay Tchernev. A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, 46(8):2143–2164, February 2008.

doi: 10.1080/00207540600818286.

URL <https://hal.uca.fr/hal-03263531>.

(15) Laurent Deroussi, Michel Gourgand, and N. Tchernev. Un système de voisinage efficace pour le problème du job-shop avec transport. *Journal Européen des Systèmes Automatisés (JESA)*, page 10 pages, 2006.

URL <https://hal.archives-ouvertes.fr/hal-00119869>.

5.5. COMMUNICATION DANS UN CONGRES

(16) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Multi-Objective Robotic Assembly Line Balancing Problem: A NSGA-II Approach Using Multi-Objective Shortest Path Decoders. In *PMS2020: 17th International Conference on Project Management and Scheduling*, Toulouse, France, April 2021.

URL <https://hal.archives-ouvertes.fr/hal-03165239>.

(17) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. A min-max path approach for balancing robotic assembly lines with sequence-dependant setup times. In *13^eme Conférence internationale de modélisation, optimisation et simulation (MOSIM2020)*, 12-14 Nov 2020, Agadir, Maroc, Agadir, Morocco, November 2020.

URL <https://hal.archives-ouvertes.fr/hal-03178123>.

(18) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. A Polynomial Algorithm For Balancing a Sequence of Operations in Reconfigurable Transfer Lines. In *MIM 2019 (Manufacturing Modeling, Management and Control)*, Berlin, Germany, August 2019.

URL <https://hal.archives-ouvertes.fr/hal-03165250>.

(19) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Reconfigurable transfer line balancing problem : deterministic and stochastic context. In *EURO 2018 29th European Conference On Operational Research.*, Valencia, Spain, July 2018a.

URL <https://hal.archives-ouvertes.fr/hal-02061416>.

- (20) Youssef Lahrichi, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Reconfigurable transfer line balancing problem: A new MIP approach and approximation hybrid algorithm. In MOSIM 2018 (Modélisation et Simulation), Toulouse, France, June 2018b.
URL <https://hal.archives-ouvertes.fr/hal-02060595>.
- (21) Arnaud Laurent, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Modèles mathématiques à base de flots pour le RCPSP multi-site. In ROADEF 2018, Lorient, France, February 2018.
URL <https://hal.archives-ouvertes.fr/hal-01954119>.
- (22) Raïssa Saleu, Laurent Deroussi, Dominique Feillet, Nathalie Grangeon, and Alain Quilliot. Optimization of urban delivery systems with drones. In VeRoLog (Annual Workshop of the EURO Working Group on Vehicle Routing and Logistics optimization), Amsterdam, Netherlands, July 2017.
URL <https://hal.archives-ouvertes.fr/hal-02083195>.
- (23) Arnaud Laurent, Laurent Deroussi, Sylvie Norre, and Nathalie Grangeon. Comparison of Three Solution Encodings and Schedule Generation Scheme for the Multi-Site RCPSP. In IFAC 2017, Toulouse, France, 2017.
URL <https://hal.archives-ouvertes.fr/hal-02023706>.
- (24) Arnaud Laurent, Nathalie Grangeon, Laurent Deroussi, and Sylvie Norre. Mathematical formulation and approximate methods for an extension of the RCPSP: The multi-site RCPSP. In CIE 45 45th International Conference on Computers Industrial Engineering, 45th International Conference on Computers Industrial Engineering 2015 (CIE45), Metz, France, 2015.
URL <https://hal.archives-ouvertes.fr/hal-02023753>.
- (25) Arnaud Laurent, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Une extension du RCPSP pour la mutualisation de ressources entre plusieurs sites : le multi-location RCPSP. In MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nancy, France, November 2014a.
URL <https://hal.archives-ouvertes.fr/hal-01166609>. Colloque avec actes et comité de lecture internationale.
- (26) Fabrice Gayraud, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Problème de tournées de véhicules avec synchronisation pour l'organisation de soins à domicile. In MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nancy, France, November 2014a.
URL <https://hal.archives-ouvertes.fr/hal-01166617>. Colloque avec actes et comité de lecture internationale.
- (27) Arnaud Laurent, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. RCPSP multi-sites - Application à une CHT. In ROADEF – 15ème congrès annuel de la société française de recherche opérationnelle et d'aide à la décision, Bordeaux, France, February 2014b. Société française de recherche opérationnelle et d'aide à la décision.
URL <https://hal.archives-ouvertes.fr/hal-00946309>.
- (28) Fabrice Gayraud, Laurent Deroussi, Nathalie Grangeon, and Sylvie Norre. Modélisation mathématique et couplage métaheuristique-simulation pour un problème de tournée de véhicules. In ROADEF - 15ème congrès annuel de la société française de recherche opérationnelle et d'aide à la décision, Bordeaux, France, February 2014b. Société française de recherche opérationnelle et d'aide à la décision.
URL <https://hal.archives-ouvertes.fr/hal-00946433>.

(29) Laurent Deroussi and David Lemoine. A particle swarm approach for the MLLP. In CIE'39, pages 12–17, Troyes, France, July 2009.
doi: 10.1109/ICCIE.2009.5223751.
URL <https://hal.archives-ouvertes.fr/hal-00860252>.

(30) Laurent Deroussi, Gérard Fleury, Michel Gourgand, and Philippe Lacomme. Coupling Heuristics and Stochastic Algorithms for the Traveling Salesman Problem. In International Conference on Operations Research, Zurich, Switzerland, 1998.
URL <https://hal.archives-ouvertes.fr/hal-02082023>

5.6. RAPPORT

(31) Laurent Deroussi, Michel Gourgand, and Sylvie Norre. New effective neighborhoods for the permutation flow shop problem. Technical report, November 2006.
URL <https://hal.archives-ouvertes.fr/hal-00678053>.

Mémoire

TABLES DES MATIERES

Tables des matières	23
Introduction	25
1. Fondements des problèmes et algorithmes	29
Calculabilité, décidabilité et complexité	29
1.1. Théorie de la complexité, problèmes et algorithmes	30
1.2. Problèmes d'optimisation combinatoire à travers l'histoire	31
1.2.1. La brouette de Monge	31
1.2.2. Problème de Fermat-Torricelli.....	32
1.2.3. Ponts de Königsberg et postier chinois.....	33
1.2.4. Icosian game et voyageur de commerce	34
1.3. Théorème "No Free Lunch" (NFL)	35
2. métaheuristiques : Prémices et concepts	37
2.1. Quelques définitions de base	37
2.1.1. Définitions et notations	37
2.1.2. Le problème du sac-à-dos.....	38
2.1.3. Le problème du voyageur de commerce	38
2.2. Une première approche des métaheuristiques	39
2.3. Espaces des solutions et des représentations	40
2.3.1. Définition et vocabulaire	40
2.3.2. Exemples.....	41
2.3.3. Espace des représentations pour quelques métaheuristiques	42
2.4. Mécanismes de construction de représentations	44
2.4.1. Construction d'une représentation	44
2.4.2. Construction par modification d'une représentation.....	47
2.4.3. Construction par combinaison de plusieurs représentations.....	51
2.5. Notions d'exploitation et d'exploration	53
2.5.1. Définition des concepts d'exploitation et d'exploration	53
2.5.2. Guider une métaheuristique avec l'utilisation de mémoires	55
2.5.3. Guider une métaheuristique avec l'utilisation de voisinages.....	57
3. Métaheuristiques : avancées récentes et tendances actuelles	63
3.1. Quoi de neuf du côté des métaheuristiques ?	64
3.2. Méthodes de recherche locale	67
3.2.1. Cas des recherches locales déterministes	68
3.2.2. Cas des recherches locales stochastiques	68
3.3. Méthodes hybrides	70
3.3.1. Définition et classifications.....	70
3.3.2. Un cas particulier important : l'hybridation métaheuristiques, recherche locale.....	71
3.3.3. Hybridation entre une métaheuristique et une méthode exacte	72
3.4. Sources bibliographiques	74

4. Exemples d'application des métaheuristiques	77
4.1. Le flow-shop de permutation	77
4.1.1. Présentation du problème.....	77
4.1.2. Proposition de mouvements élaborés.....	78
4.1.3. Synthèse des résultats.....	80
4.1.4. Retour sur expérience pour le flow-shop de permutation.....	81
4.2. Les systèmes flexibles de production	82
4.2.1. Le problème du job-shop avec transport	83
4.2.2. Le problème du job-shop flexible avec transport.....	86
4.2.3. Le problème de reconfiguration d'un atelier flexible de production	86
4.2.4. Retour sur expérience pour les systèmes flexibles de production.....	89
4.3. Les approches par décomposition d'un problème	90
4.3.1. Problèmes de distribution par drones.....	90
4.3.2. Problèmes d'équilibrage de lignes d'assemblage	93
4.3.3. Retour sur expérience pour les approche par décomposition	97
4.4. Résoudre les problèmes de demain	98
5. Conclusion générale	101
6. Bibliographie	105

INTRODUCTION

L'optimisation combinatoire peut être vu comme un voyage aux confins de la finitude, là où les évidences peuvent très vite se distordre en paradoxes. Ainsi, derrière la recherche apparemment facile d'un meilleur élément parmi un ensemble fini et préordonné d'éléments (Figure 1), qui ne demande que de les parcourir, se cache un des problèmes ouverts en mathématiques les plus énigmatiques et fondamentaux, $P \stackrel{?}{\neq} NP$. Si ce résultat s'avère vrai, il signifie que toute une classe de problèmes (ceux qui appartiennent à la classe NP) ne pourront sans doute jamais être résolus, sauf pour quelques instances de petite taille. Les méthodes approchées, qui se contentent de rechercher des solutions de bonne qualité, sont alors l'ultime recours. Parmi ces méthodes approchées, les métaheuristiques sont connues et reconnues depuis plus de cinquante ans comme étant les plus efficaces. Elles ont l'avantage d'être génériques - une métaheuristique est un algorithme de principe qui peut s'appliquer à de nombreux problèmes d'optimisation -, faciles à programmer, flexibles, *etc.*

Soit R une relation binaire sur un ensemble E .

On dit que R est une **relation d'ordre** si elle est *réflexive*, *antisymétrique* et *transitive*.

Réflexivité : $\forall x \in E, xRx$

Antisymétrie : $\forall (x, y) \in E \times E, xRy \text{ et } yRx \Rightarrow x = y$

Transitivité : $\forall (x, y, z) \in E \times E \times E, xRy \text{ et } yRz \Rightarrow xRz$

E est **totalemment ordonné** si tous ces éléments sont comparables.

Totalité : $\forall (x, y) \in E \times E, xRy \text{ ou } yRx$

On montre que dans un ensemble fini totalement ordonné, toute partie non vide possède un plus petit élément, qui est unique.

R est une **relation de préordre** si elle est *réflexive* et *transitive*. La perte de la transitivité permet les *ex aequo*, et donc le fait qu'il puisse exister plusieurs optima.

Figure 1. Quelques rappels sur les relations d'ordre et de préordre.

Traditionnellement, les métaheuristiques sont présentées en les structurant selon qu'elles sont inspirées par la nature ou non, qu'elles sont à population ou à individus. On retrouve des sous-groupes particulièrement importants comme les algorithmes évolutionnaires, les algorithmes en essaim ou les méthodes à trajectoire. Les métaheuristiques qui sont mises en avant dans ces différentes classifications sont très souvent les mêmes (algorithmes génétiques, recuit simulé, algorithme tabou, colonies de fourmis, ...), malgré le fait qu'il en existe des centaines.

Le principe général d'une métaheuristique peut se décrire en une dizaine de lignes tout au plus. La principale difficulté réside dans le passage de l'algorithme de principe à l'implémentation pour un problème donné.

Les métaheuristiques sont pourtant (nous l'avons proclamé) faciles à programmer. Cela n'est pas contradictoire. Il est très facile de programmer une métaheuristique en appliquant des composants simples, ce qui permettra d'obtenir des résultats déjà satisfaisants. Ce n'est que lorsqu'on recherche la performance qu'il devient nécessaire d'utiliser des composants plus sophistiqués, et de les coder de manière efficace.

Dans ce mémoire, le focus ne sera pas mis sur les métaheuristiques mais sur les concepts qui font leur efficacité. Les métaheuristiques que nous avons déjà mentionnées, et quelques autres, ne seront pas décrites explicitement. Par contre, les idées, les mécanismes, les concepts qu'elles véhiculent seront bien présents. Plutôt que de les emprisonner dans un nom, nous préférons voir

une métaheuristique comme un puzzle dont les pièces sont des composantes qui peuvent venir de différentes métaheuristiques, voire d'autres domaines, et qu'il faudra assembler intelligemment.

Ce mémoire est structuré en quatre chapitres. Les trois premiers sont pleinement consacrés aux métaheuristiques. Ils répondent de manière schématique aux questions suivantes : à quoi elles servent ? comment elles fonctionnent ? et vers quoi elles évoluent ? Le quatrième chapitre est consacré à des cas d'application des métaheuristiques, que nous avons mis en œuvre sur quelques problèmes.

Le premier chapitre s'appelle "*Fondements des problèmes et algorithmes*". Il justifie l'intérêt des métaheuristiques. Comme toute méthode approchée, leur zone d'utilisation commence là où se termine celle des méthodes exactes, c'est-à-dire pour résoudre des instances pour lesquelles les méthodes exactes échouent à trouver une solution optimale. Les métaheuristiques sont utilisées pour résoudre des problèmes d'optimisation difficiles. Pour l'optimisation combinatoire, la théorie de la complexité fournit un cadre théorique qui permet d'identifier les problèmes pour lesquels l'utilisation de métaheuristiques se justifie. Dans ce chapitre, nous rappellerons les fondements de cette théorie. Nous ferons également un historique de quelques problèmes d'optimisation combinatoire. Nous parlerons enfin du théorème No Free Lunch qui montre qu'en moyenne, tous les algorithmes se valent.

Le deuxième chapitre s'intéresse aux métaheuristiques vues par les concepts. Après avoir rappelés des éléments de vocabulaire et la définition d'une métaheuristique, nous proposons de travailler avec deux espaces de recherche. Le premier dédié au problème d'optimisation et qui sera l'espace des solutions, et le deuxième dédié aux métaheuristiques et qui sera l'espace des représentations. Il faudra piloter la métaheuristique dans l'espace des représentations, de manière à se déplacer intelligemment dans l'espace des solutions. Au travers de l'illustration du problème du voyageur de commerce (Traveling Salesman Problem ou TSP), nous listerons un certain nombre de mécanismes (issus d'heuristiques dédiées au TSP et de métaheuristiques) qui permettent de générer des représentations nouvelles, puis nous discuterons sur les moyens de guider efficacement la métaheuristique dans cet espace, via les notions d'exploration et d'exploitation et en utilisant des voisinages et des mémoires.

Dans le troisième chapitre, nous aborderons les métaheuristiques telles qu'elles existent aujourd'hui, avec leurs travers et leurs forces. Nous parlerons tout d'abord d'une partie de la recherche qui se consacre à la proposition de nouvelles métaheuristiques. L'objectif poursuivi est de trouver de nouveaux systèmes à modéliser (biologiques, physiques, sociaux, ou autres) de manière à définir de nouvelles règles d'évolution dans l'espace de recherche. La limite de cette démarche est que malgré la proposition de plusieurs centaines de nouvelles métaheuristiques, aucune n'a réellement proposé quelque chose de novateur pour le domaine. Nous détaillons ensuite la recherche locale vue comme une boîte noire permettant de mettre en œuvre l'exploitation, puis des techniques d'hybridation à base de métaheuristiques.

Le dernier chapitre liste trois exemples d'application que nous avons mis en œuvre sur des problèmes variés, tout au long de ma carrière. Ces exemples sont choisis car ils montrent différentes facettes dans l'utilisation des métaheuristiques, et qu'ils nous donnent des connaissances particulières pour affronter les thématiques de recherche que nous commençons à investiguer. Ces exemples d'application sont décrits tels que nous les avons traités, mais un retour sur expérience sera fait à la lumière des concepts présentés dans les chapitres 2 et 3. Le premier exemple décrit un mécanisme d'exploitation très pointu. Le deuxième exemple met en

avant l'importance du choix de la représentation d'une solution et traite la synchronisation des ressources de production et de transport. Le troisième exemple décrit une approche de décomposition d'un problème en sous-problèmes, avec la mise en œuvre de techniques d'hybridation.

1. FONDEMENTS DES PROBLEMES ET ALGORITHMES

La Recherche Opérationnelle est une science jeune dont certains attribuent la naissance pendant la deuxième guerre mondiale. Des scientifiques ont optimisé la localisation des antennes radars, une invention récente, pour améliorer le système de défense britannique face aux assauts de la Luftwaffe. Dans la continuité, des groupes ont été formés comme l'ASWORG (Anti-Submarine Warfare Operations Research Group), dont le but était de déterminer les meilleures stratégies à suivre, en modélisant les déplacements des navires, afin de contrer le blocus maritime de l'île. Un autre groupe, SCOOP (Scientific Computation Of Optimum Programs), a continué après-guerre à conseiller les stratèges américains. SCOOP comptait dans ses rangs un certain Georges Dantzig, l'inventeur du Simplexe.

L'invention des ordinateurs et la naissance de l'informatique ont mis un coup d'accélérateur à cette discipline et ont permis de formaliser un certain nombre de concepts.

L'objectif de ce chapitre est de poser les fondements du contexte théorique qui ont conduit aux métaheuristiques. Dans un premier temps, les notions de décidabilité et de complexité d'un problème sont abordées. Nous verrons ensuite comment les problèmes d'optimisation combinatoire sont apparus au fil de l'histoire. Ce chapitre se terminera avec le théorème No Free Lunch, qui pose question sur l'efficacité des méthodes de résolution de problèmes.

CALCULABILITE, DECIDABILITE ET COMPLEXITE

La calculabilité est une théorie qui s'intéresse à la caractérisation des fonctions qui peuvent être calculées à l'aide d'un algorithme. La perception de ce qui est calculable et de ce qui ne l'est pas permet de comprendre quels problèmes pourront être résolus par un ordinateur.

Une fonction f sera calculable s'il existe un processus mécanique qui, étant donné x retourne toujours la valeur $f(x)$ en un temps fini.

La thèse de Church-Turing⁴ affirme que tout processus mécanique (et par conséquent tout programme informatique) peut être traduit en une machine de Turing.

L'exemple classique de fonction incalculable est celui du problème de l'arrêt. Il n'existe pas de programme universel capable, en temps fini, de répondre "oui" si l'exécution d'un programme se terminera et "non" dans le cas contraire.

Bien avant les problèmes du millénaire (nous y reviendrons plus tard), David Hilbert énonça, lors du congrès international des mathématiciens qui se tint à Paris en 1900, une liste de 23 problèmes. Parmi eux le dixième (voir encadré) qui pose la question de l'existence d'un algorithme capable de décider, pour une équation diophantienne⁵ quelconque, si celle-ci possède des solutions entières. C'est à notre connaissance la première fois que se pose la question de la décidabilité

"On se donne une équation diophantienne à un nombre quelconque d'inconnues et à coefficients entiers rationnels : on demande de trouver une méthode par laquelle, au moyen d'un nombre fini d'opérations, on pourra distinguer si l'équation est résoluble en nombre entiers rationnels" (dixième problème de Hilbert)

En 1970, un nouveau résultat en logique mathématique, connu sous le nom de Théorème de Matiyassevitch, le résout par la négative.

A titre illustratif, l'équation $x^2+y^2-1=0$ admet 2 solutions $(x,y) = (0,1)$ ou $(1,0)$. L'équation $x^2-991y^2-1=0$ admet quant à elle des solutions beaucoup moins évidentes à trouver (Delahaye, 1995)

$x=379\ 516\ 400\ 906\ 811\ 930\ 638\ 014\ 896\ 080$ et
 $y=12\ 055\ 735\ 790\ 331\ 359\ 447\ 442\ 538\ 767$

⁴ La thèse de Church-Turing fait référence aux travaux de Alonzo Church et d'Alan Turing, pour répondre au problème de la décision de Hilbert (voir https://ladoc.fii.fr/Temoignage_Bournez.pdf).

⁵ De Diophante d'Alexandrie, mathématicien grec du IIIe siècle. Equations polynomiales à une ou plusieurs inconnues, à coefficients entiers, et dont les solutions sont aussi des entiers.

d'un problème. Cette notion fait référence à celle de la calculabilité en ceci qu'elle recherche l'existence d'un algorithme qui s'arrête en un temps fini, et qui répond à la question posée par "oui" ou par "non".

Ces questions ont été à la base des travaux d'Alan Turing, dont la machine éponyme a servi de cadre formel pour établir l'indécidabilité du problème de l'arrêt (Turing, 1936).

Dans le cadre de ce mémoire, nous nous intéresserons exclusivement aux problèmes décidables. Ce sont les problèmes pour lesquels il existe un processus mécanique, donc un algorithme, qui termine en un temps fini, et qui les décide, en répondant par oui ou par non à la question posée. Dire qu'un problème est décidable ne préfigure en aucune manière qu'il soit facile à résoudre, simplement qu'il existe un algorithme qui le résout finiment. C'est la théorie de la complexité qui s'intéresse et apporte des réponses quant à l'efficacité de cette résolution.

1.1. THEORIE DE LA COMPLEXITE, PROBLEMES ET ALGORITHMES

La théorie de la complexité cherche à savoir si un problème décidable peut être résolu efficacement, ou au contraire si la réponse n'est pas accessible en pratique. Si au sens large la mesure de l'efficacité implique toutes les ressources nécessaires pour exécuter un algorithme, par algorithme le plus efficace on entend généralement le plus rapide en temps.

La complexité d'un problème sera définie par la complexité du meilleur algorithme le résolvant. Ainsi, un problème sera dit polynomial s'il existe un algorithme le résolvant dans le pire des cas en un temps polynomial en la taille de l'instance. Dans le cas contraire, le problème sera dit exponentiel ou NP-complet.

Les problèmes NP-complets ont ceci de particulier, qu'il a été montré qu'ou bien aucun d'entre eux, ou bien tous, sont solubles en un temps polynomial. Aujourd'hui, aucun algorithme polynomial permettant de résoudre un problème NP-complet n'a été découvert. On ne sait pas si un tel algorithme existe. Derrière cette question se trouve le problème ouvert $P = NP$, un des sept problèmes du prix du millénaire⁶ posés par l'Institut de Mathématiques Clay en 2000, dotés d'un prix de 1 million de dollars. La communauté pense majoritairement que $P \neq NP$. Le Tableau 1 est une capture d'écran qui détaille les résultats d'un sondage posé en 2002, 2012 et 2019 à un échantillon de spécialistes du problème (Gasarch, 2019; Ind = Independant, DC = Don't Care, DK = Don't Know).

Tableau 1. Résultats du sondage $P=NP$ auprès de spécialistes du problème, (W. I. Gasarch, 2019, <https://www.cs.umd.edu/users/gasarch/BLOGPAPERS/pollpaper3.pdf>)

	$P \neq NP$	$P = NP$	Ind	DC	DK	DK and DC	other
2002	61 (61%)	9 (9%)	4 (4%)	1 (1%)	22 (22%)	0 (0%)	3 (3%)
2012	126 (83%)	12 (9%)	5 (3%)	5 (3%)	1 (0.66%)	1 (0.66%)	1 (0.66%)
2019	109 (88%)	15 (12%)	0	0	0	0	0

Selon (Schirmer, 1995), les problèmes sont caractérisés par des variables, chacune étant associée à un ensemble ou domaine d'objets finis. Une instanciation du problème revient à spécifier pour chaque variable un objet dans son domaine.

L'auteur définit quatre types de problèmes : les problèmes de recherche, d'optimisation, de décision et de faisabilité.

- Un **problème de recherche** est un problème pour lequel on doit retrouver, parmi une collection d'objets X , ceux considérés comme possédant une propriété particulière

⁶ Ces défis ont été proposés dans le même esprit que les 23 problèmes listés par Hilbert en 1900.

(désignés comme solutions du problème). Un algorithme résout le problème s'il répond « Echech » dans le cas où aucun objet n'est solution, et retourne une solution sinon.

- Un **problème d'optimisation**, dans lequel une fonction objectif permet de comparer les objets. Résoudre un tel problème, c'est trouver un objet qui optimise cette fonction.
- Un **problème de décision**, qui sont les problèmes dont la réponse est oui ou non.
- Un **problème de faisabilité** est un problème de décision pour lequel la réponse est oui si un des objets de la collection est faisable.

Ces problèmes sont fortement reliés les uns aux autres comme le montre la Figure 2.

Dans la suite, les problèmes rencontrés seront essentiellement des problèmes d'optimisation, mais lors de la mise en œuvre de certaines méthodes de résolution, nous pourrons être amenés à considérer d'autres types de problèmes, en particulier des problèmes de faisabilité.

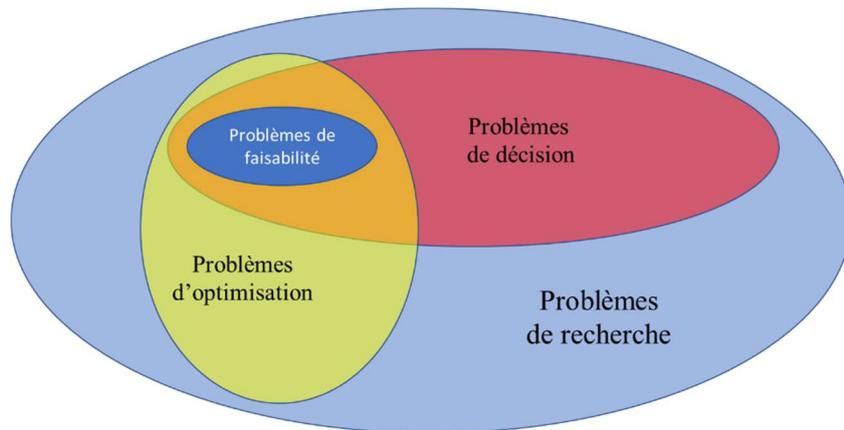


Figure 2. Relation entre les 4 types de problèmes (d'après Schirmer, 1995).

1.2. PROBLEMES D'OPTIMISATION COMBINATOIRE A TRAVERS L'HISTOIRE

Si le cadre théorique qui classe les problèmes selon leur complexité est apparu dans la deuxième moitié du XX^{ème} siècle, ceux-ci sont mentionnés bien avant dans la littérature scientifique. Nous proposons d'ici de montrer comment quelques-uns de ces problèmes sont apparus. Cela nous permettra aussi de montrer les liens parfois étroits qui peuvent relier des problèmes discrets *versus* continus, et de classe P *versus* NP.

1.2.1. LA BROUETTE DE MONGE

Le problème connu sous le nom de la "brouette de Monge" est l'un des premiers exemples de recherche opérationnelle. Il a été introduit en 1781 par un mathématicien français, Gaspard Monge, dans son *Mémoire sur la théorie des déblais et des remblais* (Monge, 1781). Le problème peut se résumer ainsi. Nous disposons d'un tas de sable (le déblai) et d'un trou à combler (le remblai), dont les volumes coïncident. Le déplacement d'un grain de sable, de sa position initiale i vers une position du trou j nécessite une énergie $f(i, j)$. Comment déplacer chaque grain de sable du déblai vers le remblai, de manière à minimiser l'énergie totale dépensée ?

Le problème ainsi défini est un problème d'optimisation continu (déterminer la fonction de transport T qui minimise la fonction intégrale $\int_X f(x, T(x)) d\mu$ où μ désigne la mesure du déblai). Il en découle deux problèmes d'optimisation combinatoire très connus : le problème de transport et le problème d'affectation.

Supposons que nous avons plusieurs déblais et plusieurs remblais, et que l'énergie pour transporter un grain de sable d'un déblai donné vers un remblai donné est constant. Un coût par

unité de volume transportée peut alors être défini entre chaque paire (déblai, remblai) et on obtient alors le problème de transport.

En rajoutant l'hypothèse que les volumes des déblais et remblais sont tous identiques (que l'on peut supposer unitaires). Le problème résultant est alors celui d'affectation.

Pour l'anecdote, G. Monge a proposé, pour résoudre son problème continu, de le discrétiser et de s'intéresser au déplacement des grains de sable du déblai vers le remblai. Ce faisant, il a défini un problème d'affectation de très grande taille. Le premier problème d'optimisation combinatoire ?

1.2.2. PROBLEME DE FERMAT-TORRICELLI

En 1636, Pierre de Fermat énonce en ces termes un problème d'optimisation continu : "étant donné trois points en trouver un quatrième tel que la somme de ses distances aux trois points donnés soit minima". Quatre ans plus tard, Torricelli a proposé une résolution graphique pour ce problème, qui a été finalement démontrée en 1659 par Viviani. Le point de Fermat, concourant des droites (AA') , (BB') et (CC') , s'il appartient à l'intérieur du triangle⁷, et celui qui répond à la question (Figure 3).

Ce problème a une application évidente en logistique, qui est de localiser le meilleur emplacement d'un entrepôt, destiné à distribuer trois villes. La croyance populaire est souvent tenace, et elle veut encore aujourd'hui que la réponse à ce problème soit le centre de gravité du triangle (la méthode dite du barycentre figure dans tous les manuels de logistique).

Il est avisé de conseiller d'utiliser des méthodes approchées, dès lors que les mathématiques ne nous permettent pas de résoudre un problème de manière exacte. Ce problème en est un contre-exemple. On sait résoudre ce problème depuis près de 4 siècles, et avec une construction simple. Pourtant, une méthode approchée lui est préférée.

Ce problème est lié à la famille de problèmes dits de localisation de sites (Plant Location Problems), qui sont de classe NP.

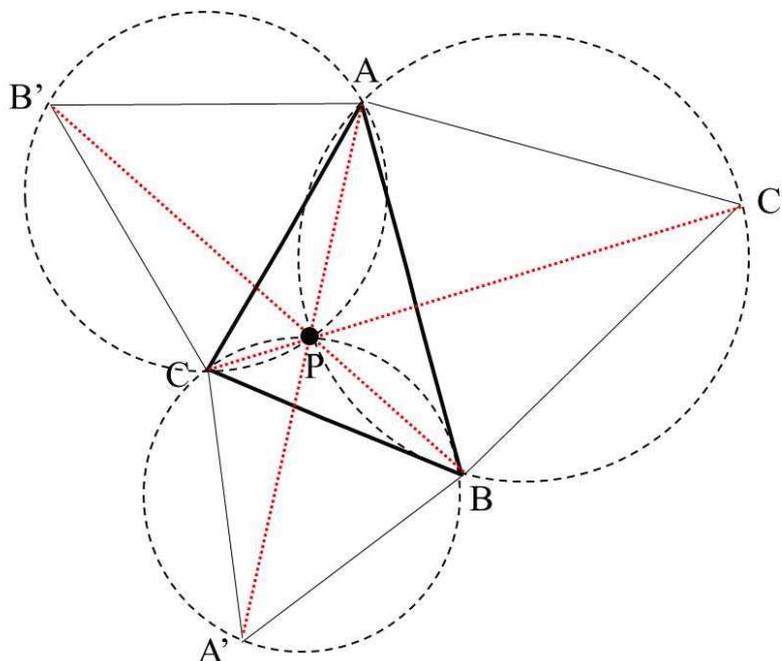


Figure 3. Le point de Fermat-Torricelli. Le point P est celui qui minimise l'expression $MA + MB + MC$.

⁷ Si un des angles, par exemple \hat{A} , est supérieur à 120° , alors le point de Fermat est à l'extérieur du triangle. Le point recherché dans le problème de Fermat est alors le sommet A.

Les deux problèmes qui vont suivre sont deux problèmes de faisabilité célèbres. Leurs énoncés respectifs en théorie des graphes sont similaires. Ils sont reliés à deux problèmes d'optimisation eux aussi très connus, un de classe P, l'autre de classe NP.

1.2.3. PONTS DE KÖNIGSBERG ET POSTIER CHINOIS

Cet historique nous amène maintenant autour des travaux de Leonhard Euler et Sir William Rowan Hamilton (section suivante), qui ont donné naissance en théorie des graphes aux cycles eulériens et hamiltoniens.

Le problème des sept ponts de Königsberg est bien connu en mathématiques. La question qui alimentait les réflexions d'Euler lors de ses promenades dans la ville était de savoir s'il pourrait regagner sa maison en empruntant chaque pont de la ville exactement une fois. La ville de Königsberg (aujourd'hui Kaliningrad) se représente de manière schématique sous la forme d'un graphe dont les sommets modélisent les quatre quartiers de la ville, et les arcs modélisent les ponts permettant de passer d'un quartier à un autre. Le mathématicien a publié un document en 1736 (dont est extrait le dessin de la Figure 4), dans lequel il affirme sans le prouver que le problème n'a pas de solution. En théorie des graphes, ce problème se ramène à la recherche d'un cycle eulérien (cycle qui passe par toutes les arêtes du graphe). Un tel cycle existe si et seulement si tous les sommets du graphe sont de degré pair. Tout graphe possédant cette propriété est dit eulérien.

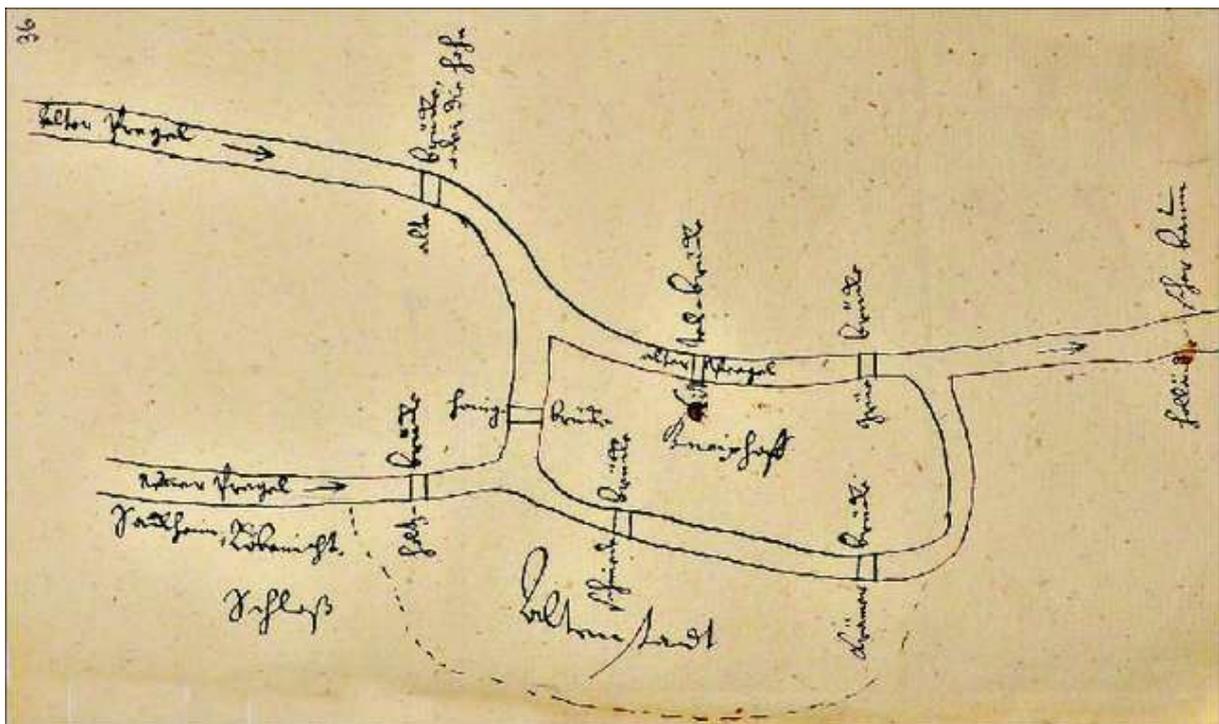


Figure 4. Le problème des 7 ponts de Königsberg (Euler, 1736 ©BY-NC-ND 4.0).

Ce problème est à l'origine du problème du postier chinois, formalisé par le mathématicien chinois Meigu Guan en 1962, et qui peut s'énoncer ainsi : dans un graphe donné, déterminer le cycle passant au moins une fois par chaque arête du graphe, et qui est de longueur minimale. (Edmonds & Johnson, 1973) ont montré que ce problème est de classe P. Il peut être réduit à la recherche d'un couplage maximum de poids minimal, qui permet de relier tous les sommets de degré impair afin de rendre le graphe eulérien.

1.2.4. ICOSIAN GAME ET VOYAGEUR DE COMMERCE

Le jeu connu sous l'intitulé "*Icosian Game*"⁸ a été inventé en 1857 par Hamilton. Le plateau de jeu est composé de trous reliés entre eux par des segments. Ce casse-tête consiste à placer des pions numérotés de 1 à 20 dans les trous, de manière à ce que deux pions consécutifs soient reliés par une arête, ainsi que les pions numéro 20 et 1. Le problème équivaut à trouver un "cycle hamiltonien", c'est-à-dire un chemin qui part et arrive au même point, et qui passe par tous les autres points. Résoudre "*Icosian Game*", c'est démontrer l'existence d'un cycle hamiltonien. Quelques années plus tard, la recherche du meilleur cycle hamiltonien aboutira sur la définition d'un des problèmes emblématiques de la discipline, à savoir le problème du Voyageur de Commerce (Traveling Salesman Problem ou TSP)

Selon (Schrijver, 2005), la première trace de ce problème remonte à 1832 dans un guide pour un voyageur de commerce prospère, "*Le voyageur de commerce – ce qu'il doit être et ce qu'il doit faire pour obtenir des commandes et prospérer dans ses affaires – par un voyageur de commerce expérimenté*"⁹ (traduction de l'allemand). Loin de toutes considérations mathématiques, ce guide proposait cependant plusieurs tournées pour des villes allemandes. La Figure 5 permet de comparer une de ces tournées avec une solution optimale, obtenue *a posteriori*.

La première définition formelle du problème du messenger (aujourd'hui notre TSP) a été proposée par Karl Menger en 1930. Nous donnons ci-dessous une traduction du texte d'origine en allemand. Il est intéressant de noter qu'en plus de la définition, Menger conjecture sur la complexité de ce problème et introduit l'heuristique Plus Proche Voisin.

« Nous appelons problème du messenger (parce que cette question doit être résolue dans la pratique par chaque facteur, et d'ailleurs aussi par de nombreux voyageurs) le problème consistant à trouver le chemin le plus court reliant un nombre fini de points dont les distances deux à deux sont connues. Ce problème peut bien sûr toujours être résolu par un nombre fini d'essais. On ne connaît pas de règles qui permettraient d'abaisser le nombre d'essais en dessous du nombre de permutations des points donnés. La règle selon laquelle il faut d'abord aller du point de départ au point le plus proche, puis au point le plus proche, et ainsi de suite, ne fournit généralement pas le chemin le plus court » (Karl Menger, 1930).

Notons, au travers de la présentation de ces quelques problèmes historiques, la proximité qui existe entre les problèmes d'optimisation continue et combinatoire. La première résolution du problème continu des déblais et des remblais est basée sur la discrétisation des déblais. De plus, ce problème, ainsi que celui de Fermat, sont apparentés à des problèmes d'optimisation combinatoire.

Bien qu'il semble que $P \neq NP$, la frontière entre ces deux classes de problèmes est à la fois infranchissable et ténue. Des problèmes qui sont proches dans leur formulation peuvent être de classe différente comme le postier chinois et le voyageur de commerce.

D'autres exemples viendront illustrer ces deux points. Des méthodes conçues pour travailler sur des espaces continus (resp. combinatoires) peuvent être utilisées pour résoudre des problèmes combinatoires (resp. continus). Des problèmes de classe NP peuvent être relaxés en problèmes de classe P pour construire des solutions approchées.

⁸ Un plateau du jeu est en couverture du livre (Lawler *et al.*, 1985) consacré au TSP.

⁹ *Der Handlungsreisende — wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein — von einem alten Commis-Voyageur*

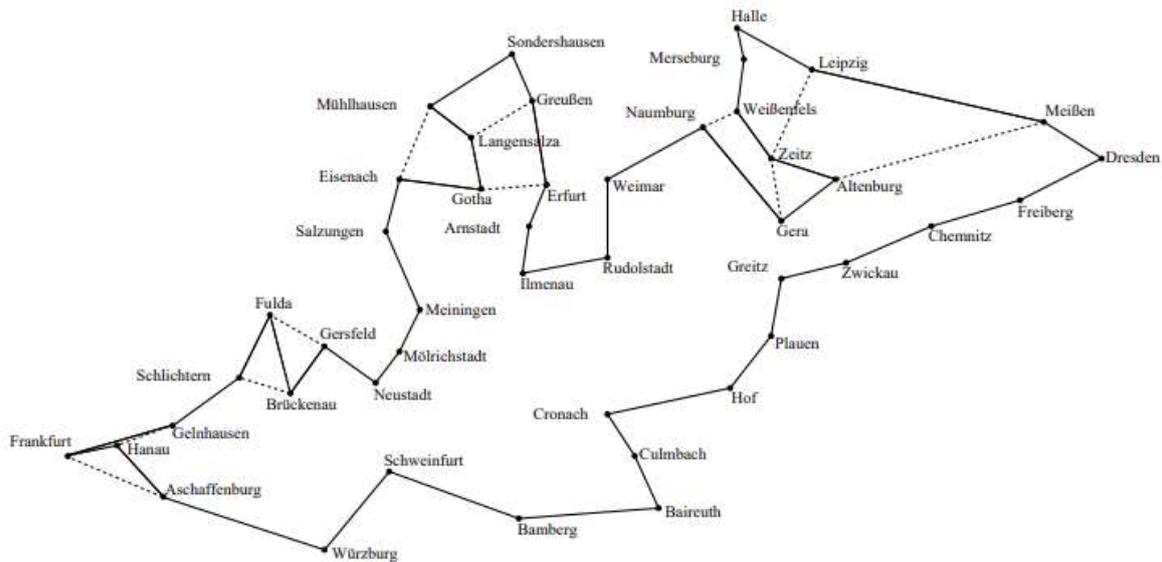


Figure 5. Solution d'un problème de voyageur de commerce avec 45 villes allemandes, issu du manuel de 1832 (en trait plein). Cette solution a un coût de 1285 km. La solution optimale (en pointillés) a un coût de 1248 km.

1.3. THEOREME "NO FREE LUNCH" (NFL)

Le nom de No Free Lunch vient de l'adage « *there ain't no such thing as a free lunch* » dont l'équivalent français pourrait être la locution « *on a rien sans rien* ».

Dans le domaine de l'optimisation combinatoire, (Wolpert & Macready, 1997) indiquent que deux algorithmes quelconques qui recherchent l'extremum d'une fonction de coût sont équivalents en performance quand celle-ci est moyennée sur l'ensemble de toutes les fonctions coûts possibles.

En d'autres termes, si l'algorithme A est meilleur que l'algorithme B pour certaines fonctions coûts, alors il en existera d'autres pour lesquelles B surclassera A.

Ce théorème peut être reformulé de manière équivalente ainsi :

Soit S l'espace des solutions (fini) et soit V un ensemble de nombres réels (fini). Supposons que $f : S \rightarrow V$ est choisi aléatoirement selon une loi de distribution uniforme parmi l'ensemble de toutes les fonctions possibles de S dans V . Alors pour résoudre le problème d'optimisation $\min_{x \in S} f(x)$, aucun algorithme ne sera plus performant que la recherche aveugle.

La recherche aveugle (blind search) consiste, à une itération donnée, à choisir aléatoirement selon une distribution uniforme un élément de S parmi ceux qui n'ont pas encore été choisis.

La conclusion de ce théorème pourrait apparaître comme très pessimiste, et l'on pourrait se demander s'il y a un intérêt de rechercher des algorithmes performants. Fort heureusement, nous ferons une conjecture pour l'ensemble des problèmes d'optimisation que nous serons amenés à étudier. Dans le cadre de nos travaux en Recherche Opérationnelle, les solutions proposées devront répondre à des exigences métiers et auront pour finalité d'aider les décideurs à faire leur choix (voir encadré). Des prises de décisions sensiblement égales ne doivent pas avoir de répercussions importantes sur les objectifs à atteindre. Ceci revient à supposer qu'il y a une forte corrélation entre l'espace des solutions S et l'ensemble des valeurs V . Deux solutions structurellement proches dans S auront des coûts relativement proches.

Sans jamais le mentionner de manière explicite, nous ne nous intéressons donc qu'à une classe très restreinte de fonctions coûts, celles qui sont fortement dépendantes de la structure d'une solution.

La ROADEF (Société Française de la Recherche Opérationnelle et d'Aide à la Décision) définit la Recherche Opérationnelle comme étant :

« la mise en œuvre de méthodes scientifiques, essentiellement mathématiques et algorithmiques, en vue de prendre la meilleure décision possible. La RO fournit des outils pour rationaliser, simuler et optimiser l'architecture et le fonctionnement des systèmes industriels et économiques. Elle propose des modèles pour analyser des situations complexes et permet aux décideurs de faire des choix efficaces et robustes »

Le livre blanc de la recherche opérationnelle en France, ROADEF, 2019.

Les problèmes d'optimisation combinatoire que nous pouvons rencontrer dans la vie courante (par exemple au hasard d'une promenade comme Euler !), et *a fortiori* dans les entreprises, appartiennent la plupart du temps à la classe NP. Il en résulte que les décisions optimales à prendre, qui reposent sur la résolution du problème, ne seront pas toujours accessibles. Dans ce cas, il faudra se satisfaire de prendre une bonne décision, la meilleure possible, en cherchant une solution approchée.

Parmi les méthodes qui permettent d'obtenir des solutions approchées, les métaheuristiques font partie des plus populaires et des plus utilisées, et ce depuis largement plus d'un demi-siècle. Les deux prochains chapitres sont consacrés à percer leurs secrets. Le contraposée du théorème NFL nous donne déjà quelques pistes de réflexion : une heuristique, pour être performante, devra nécessairement *exploiter* la structure d'un problème.

2. METAHEURISTIQUES : PREMICES ET CONCEPTS

Les métaheuristiques sont utilisées pour résoudre des problèmes d'optimisation combinatoire depuis très longtemps. D'un abord très simple, elles sont apparues naturellement avec l'essor des ordinateurs. Elles permettaient, après avoir construit une solution, de chercher progressivement à l'améliorer. Ce n'est que plus tard, au début des années 80s, que des principes algorithmiques sont apparus, portant des noms tels que "*algorithmes génétiques*" ou "*recuit simulé*". Il faudra attendre encore quelques années pour que ces principes algorithmiques deviennent enfin des métaheuristiques, et que des concepts généraux soient posés.

Après avoir rappelé et introduit les termes clés, et défini ce qu'est une métaheuristique, nous poursuivrons ce chapitre avec la proposition du concept d'espace des représentations. Cet espace, dans lequel évoluent les métaheuristiques, est détaché de l'espace des solutions qui, lui, caractérise le problème. Nous listerons alors quelques mécanismes qui permettent de se mouvoir dans cet espace des représentations. Nous terminerons par aborder les concepts clés d'exploration (ou diversification) et d'exploitation (ou intensification).

2.1. QUELQUES DEFINITIONS DE BASE

Avant de voyager plus avant dans le domaine des métaheuristiques, nous rappelons d'abord les définitions de base avant de les illustrer sur deux problèmes classiques : le problème du sac-à-dos (Knapsack Problem ou KP) et le problème du voyageur de commerce (Traveling Salesman Problem ou TSP), tous deux de classe NP.

2.1.1. DEFINITIONS ET NOTATIONS

Un **problème d'optimisation** consiste à rechercher une solution $s = (x_1, x_2, \dots, x_n)$ parmi un ensemble S de solutions réalisables, de telle sorte que toutes les contraintes soient respectées et que la fonction objectif $f : S \rightarrow \mathbb{R}$ soit de valeur optimale. Nous nous intéresserons principalement aux problèmes d'optimisation combinatoire. Sans perte de généralité et sauf précision contraire, nous considérerons dans la suite de ce mémoire des problèmes de minimisation.

Toute solution $s^* \in S$ qui satisfait $\forall s \in S, f(s^*) \leq f(s)$ est appelée un **minimum global**. Résoudre un problème d'optimisation consiste à rechercher l'optimum de la fonction objectif, c'est-à-dire la valeur $f(s^*)$, et non d'explicitier le sous-ensemble $S^* \subset S$ des solutions optimales.

Un **système de voisinage**¹⁰ est formellement défini comme étant une fonction $\mathcal{N} : S \rightarrow \mathcal{P}(S)$, qui associe à chaque solution $s \in S$ un ensemble de solutions $\mathcal{N}(s) \subset S$ dites voisines.

A partir de cette notion de système de voisinage, il est maintenant possible de définir la notion de minimum local. Toute solution $s' \in S$ qui satisfait $\forall s \in \mathcal{N}(s'), f(s') \leq f(s)$ est appelée un **minimum local**, pour le voisinage \mathcal{N} . Nous désignerons par $S_{\mathcal{N}} \subset S$ l'ensemble des minima locaux pour le voisinage \mathcal{N} .

¹⁰ Nous reviendrons un peu plus loin sur cette définition, que nous retrouvons dans la plupart des articles portant sur l'optimisation combinatoire. Nous apporterons des précisions et essaierons de faire le lien avec la notion de voisinages topologiques, telle qu'ils sont définis dans des espaces topologiques.

Une méthode de **Recherche Locale** (Local Search ou LS) est un algorithme permettant d'associer à toute solution $s \in S$ un minimum local $s' \in S_{\mathcal{N}}$. Son principe est le suivant. Partant d'une solution de référence s , on choisit parmi l'ensemble $\mathcal{N}(s)$ des solutions dites "voisines" de s , une solution qui améliore la fonction objectif et devient la nouvelle solution de référence. On itère ce principe jusqu'à avoir obtenu un minimum local, pour lequel il n'existe pas de solution meilleure dans son voisinage. En supposant pour simplifier que la recherche locale est déterministe, la recherche locale peut être vue comme une boîte noire qui, en réponse à la même entrée $s \in S$, fournira toujours en sortie le même minimum local $s' \in S_{\mathcal{N}}$.

Nous noterons $LS: S \rightarrow S_{\mathcal{N}}$ cette application.

Les recherches locales sont décrites de manière plus détaillée dans la section 3.2 qui leur est consacrée.

Le **bassin d'attraction** d'un minimum local $s' \in S_{\mathcal{N}}$ désigne l'ensemble des solutions qui, si elles sont fournies en entrée de l'application LS , donne s' en sortie.

$$LS^{-1}(s') = \{s \in S, LS(s) = s'\}, s' \in S_{\mathcal{N}}.$$

Dans la réalité, une recherche locale peut être stochastique (on choisit au hasard un voisin parmi ceux qui sont améliorants) ou utiliser des mécanismes d'interdiction dynamiques (on choisit le meilleur voisin parmi ceux qui ne sont pas interdits). LS n'est alors plus une application. Nous continuerons par abus de langage à utiliser les termes de minima locaux ou de bassins d'attraction.

2.1.2. LE PROBLEME DU SAC-A-DOS

Un alpiniste prépare un trek de plusieurs jours en montagne. Il dispose d'un sac-à-dos d'une capacité C exprimée en kg, et d'un ensemble d'objets $I = \{1, \dots, n\}$, chacun caractérisé par sa masse m_i et sa valeur v_i .

Le problème du sac-à-dos consiste à choisir les objets à placer dans le sac, de manière à respecter la contrainte de capacité du sac, et à maximiser la somme des valeurs des objets emportés.

Ce problème se caractérise par ses variables (x_1, \dots, x_n) , avec $x_i \in D_i = \{0, 1\}$ (1 si l'objet i est dans le sac, 0 sinon). La seule contrainte à satisfaire est de respecter la capacité du sac.

L'ensemble des solutions réalisables est alors $S = \left\{ s \in D_1 \times \dots \times D_n, \sum_{i \in I} m_i \cdot x_i \leq C \right\}$. La fonction

à minimiser est $f(s) = \sum_{i \in I} v_i \cdot x_i$.

Pour ce problème, une solution est donnée par un vecteur de booléens, et un système de voisinage peut être défini en procédant comme suit :

1. On choisit un objet $i \in I$ au hasard, et on change la valeur de la variable ($x_i \leftarrow 1 - x_i$).
2. Si $x_i = 0$ on rajoute des objets au hasard (en respectant la contrainte de capacité) sinon on enlève des objets au hasard (jusqu'à respecter à nouveau la contrainte de capacité)

2.1.3. LE PROBLEME DU VOYAGEUR DE COMMERCE

Un voyageur de commerce cherche à visiter un ensemble de villes $I = \{1, \dots, n\}$ avant de revenir à son point de départ, en minimisant la distance totale parcourue.

Une manière de modéliser ce problème est de considérer les variables de décision

$$x_{i,j} = \begin{cases} 1 & \text{si la ville } i \text{ précède la ville } j \\ 0 & \text{sinon} \end{cases}, \forall (i,j) \in I \times I.$$

L'objectif à minimiser est la somme des distances parcourues $\sum_{i \in I} \sum_{j \in I} d_{i,j} \cdot x_{i,j}$, où $d_{i,j}$

représente la distance entre les villes i et j .

Dans un cycle, chacune des villes précède et succède à une autre ville d'où les contraintes suivantes à satisfaire :

$$\sum_{j \in I} x_{i,j} = 1, \forall i \in I \text{ et } \sum_{i \in I} x_{i,j} = 1, \forall j \in I.$$

Ces contraintes seules ne suffisent cependant pas à caractériser un cycle hamiltonien. Il faut de plus prévenir l'apparition de sous cycles en rajoutant des contraintes pour les éliminer

$$\sum_{i \in I} \sum_{j \in I} x_{i,j} \leq |S| - 1, \forall S \subset I, 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor.$$

L'ensemble des solutions réalisables est donc égal à l'ensemble des cycles hamiltoniens qui peuvent être décrits (de manière non unique) par un ordre de parcours des villes (v_1, \dots, v_n) . v_i désignant la $i^{\text{ème}}$ ville visitée en partant de v_1 .

Des exemples de systèmes de voisinage peuvent consister à échanger la position de deux villes, ou à enlever une ville pour la repositionner ailleurs. Un autre système de voisinage est décrit au tout début de la section suivante.

2.2. UNE PREMIERE APPROCHE DES METAHEURISTIQUES

Il est difficile de retrouver avec certitude les prémices des métaheuristiques dans la littérature scientifique. (Croes, 1958) s'intéresse par exemple à un problème du voyageur de commerce symétrique de 20 villes. Il propose une méthode d'optimisation en deux phases : la première consiste à construire une solution initiale de manière heuristique à partir de la formulation de (Dantzig *et al.*, 1954). La deuxième phase améliore itérativement une solution en introduisant et en appliquant un mouvement d'inversion (bientôt appelé 2-opt dans le vocabulaire propre au TSP). Bien que la résolution proposée soit une mise en œuvre papier, nous retrouvons le principe général d'une recherche locale, et que d'aucuns assimilent à une métaheuristique. Nous voulons ici insister sur le fait que des métaheuristiques ont été appliquées très naturellement bien avant que le mot lui-même ne soit d'abord inventé, puis diffusé dans la communauté scientifique.

Venons-en maintenant à la définition d'une métaheuristique. Sur le plan purement étymologique, le terme métaheuristique est la concaténation de deux mots grecs méta ($\mu\epsilon\tau\acute{\alpha}$) qui signifie au-delà et heuristique ($\epsilon\upsilon\pi\acute{\iota}\sigma\kappa\epsilon\upsilon\nu$) qui signifie trouver. Les métaheuristiques sont souvent présentées comme des méthodes générales, capables de résoudre une large gamme de problèmes. C'est donc le caractère générique de ces méthodes qui est mis en avant.

Pourtant, pour sa première apparition dans la littérature, dont la paternité est attribuée à (Glover, 1986), il a un tout autre sens. L'auteur dit ceci : « *Tabu search may be viewed as a meta-heuristic superimposed on another heuristic* ». Glover voit dans les métaheuristiques un moyen d'introduire un mécanisme permettant de ne pas se faire piéger dans des optima locaux (en l'occurrence la liste taboue¹¹ qui a pour objectif d'interdire à la méthode de revenir sur ses pas). (Osman & Laporte, 1996) proposent la définition suivante, qui complète la définition originelle, et que nous prendrons comme référence.

¹¹ "tabou" est un adjectif d'origine polynésienne qui reviendra à de nombreuses reprises dans ce mémoire. Il peut être invariable mais, comme le préconise le Larousse, nous faisons le choix de l'accorder en genre et en nombre.

"Une métaheuristique est formellement définie comme un processus de génération itératif qui guide une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions quasi-optimales."

Cette définition met en exergue des concepts d'exploitation -qui cherchent à attirer la méthode vers des minima locaux- et des concepts d'exploration -qui cherchent au contraire à s'en extraire-. Toute la force de la métaheuristique sera d'arbitrer entre ces deux forces *a priori* antagonistes (c'est une première approche très schématique des concepts d'exploration et d'exploitation. Nous aurons l'occasion de préciser ces concepts, beaucoup plus élaborés qu'il n'y paraît, au fil des pages).

Nous terminons cette introduction aux métaheuristicques en notant le lien très fort qui unit ce mot au domaine de l'Intelligence Artificielle. Ce n'est pas un hasard si le titre de l'article dans lequel le mot "*meta-heuristic*" est apparu contient "Intelligence Artificielle". Ce n'est pas un hasard non plus si la définition d'Osman contient les termes "guide", "intelligemment", "stratégies d'apprentissage".

Notre univers contiendrait autour de 10^{82} atomes. Cela correspond à peu de choses près au nombre de solutions d'un problème du voyageur de commerce de 61 villes à peine (on résout aujourd'hui des problèmes de plusieurs milliers voire dizaines de milliers de villes. Le solveur Concorde a résolu une instance de 85900 villes). Résoudre un problème de cette taille revient donc à rechercher un atome dans notre univers. Les métaheuristicques devront donc mettre en œuvre des "mécanismes intelligents", afin d'être capables de trouver des solutions proches de l'optimum, en prélevant un échantillon de taille infinitésimale en comparaison de la taille de l'espace de recherche.

2.3. ESPACES DES SOLUTIONS ET DES REPRESENTATIONS

2.3.1. DEFINITION ET VOCABULAIRE

Les métaheuristicques sont avant reconnues pour leur caractère générique. Une métaheuristique doit pouvoir s'appliquer sur tous types de problèmes. Les concepts importants doivent pouvoir être présentés indépendamment des problèmes étudiés. Pour cela, nous proposons que les métaheuristicques évoluent dans un espace particulier, que nous appelons l'espace des représentations et que nous notons Ω (Figure 6).

Chaque métaheuristique s'appuie sur ses mécanismes internes pour générer un échantillon de représentations, dans le but de rechercher une solution qui soit la meilleure possible.

Une fois défini cet espace des représentations, il suffit de lui adjoindre une application de décodage $\mathcal{D}:\Omega\rightarrow S$ dont l'objectif est d'associer à chaque représentation une solution $s\in S$. L'évaluation $f(s)$ de s pourra ensuite être utilisée pour orienter les futures recherches.

Nous dirons que l'espace des représentations est **direct** lorsqu'on pourra instancier directement certaines variables du problème à partir d'une représentation. Il sera **indirect** sinon.

Nous dirons que l'espace des représentations est **complet** lorsqu'il est direct et qu'il existe une formalisation du problème telle qu'une représentation permet d'instancier toutes les variables du problème.

Nous dirons enfin que le problème est **intègre** lorsque $\mathcal{D}(\Omega)\cap S^*\neq\emptyset$. Autrement dit, il existe au moins une représentation qui se décode en une solution optimale. Lorsque cette propriété d'intégrité ne peut être établie, il ne sera pas garanti que l'ensemble des solutions optimales soit atteignable par la métaheuristique.

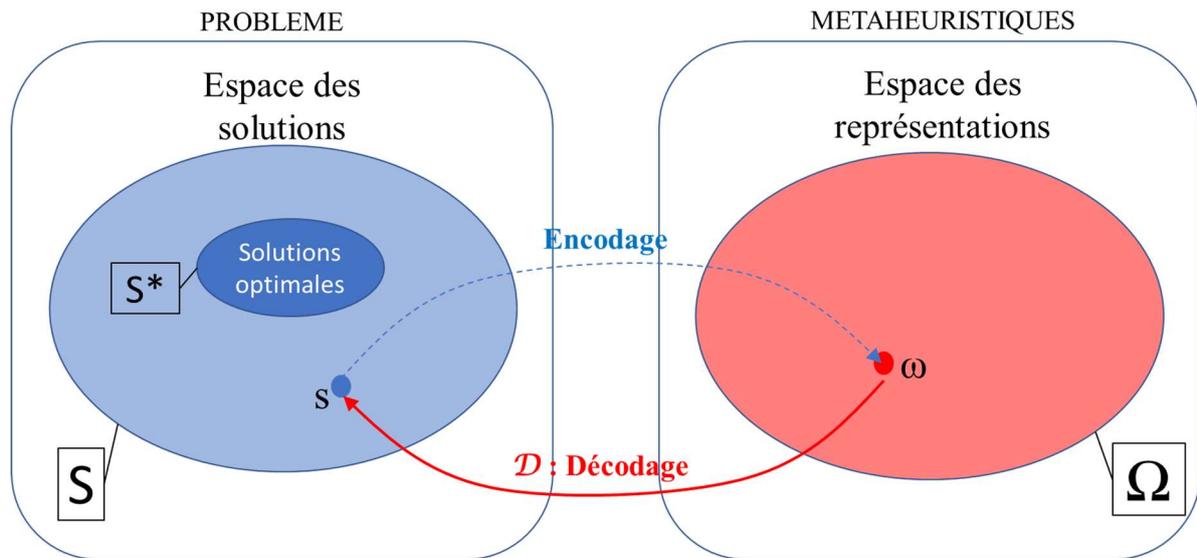


Figure 6. Espace des solutions versus espace des représentations.

2.3.2. EXEMPLES

Pour illustrer ces concepts, nous allons montrer comment les solutions des problèmes du sac-à-dos et du TSP peuvent être encodées par le même espace des permutations.

Exemple 1 : le problème du sac-à-dos.

L'espace des permutations est indirect pour le problème du sac-à-dos. Considérons la permutation $\sigma = (\sigma_1, \dots, \sigma_n)$. σ_i représente l'indice de l'article qui sera examiné en position i . nous déterminerons la valeur de la variable x_{σ_i} en observant si l'article σ_i rentre dans le sac ou non (en tenant compte de ce qui a été décidé pour les articles précédents). Nous posons $x_{\sigma_i} = 1$ si $c^{(i-1)} + m_{\sigma_i} \leq C$ et $x_{\sigma_i} = 0$ sinon, avec $c^{(i)} = c^{(i-1)} + m_{\sigma_i} \cdot x_{\sigma_i}$ pour $i \in \{1, \dots, n\}$ et $c^{(0)} = 0$ ($c^{(i)}$ représente la masse occupée dans le sac après avoir décidé si l'on amenait les i premiers articles selon l'ordre σ).

Remarquons que :

- L'espace des permutations est beaucoup plus vaste que l'espace des solutions. $|\Omega| = n!$ alors que $|S| = 2^n$.
- Toute solution est encodée par les permutations qui satisfont la propriété suivante : $x_i < x_j \Rightarrow \sigma^{-1}(j) < \sigma^{-1}(i)$ où $\sigma^{-1}(j)$ désigne l'indice k pour lequel $\sigma_k = j$ (pour une solution donnée, convient toute permutation dans laquelle les produits qui sont dans le sac sont placés avant ceux qui n'y sont pas). Ceci prouve qu'il existe au moins une permutation qui se décode en une solution optimale et donc que le problème est intègre.
- L'espace des représentations $\{0,1\}^n$ est direct et complet pour le problème du sac-à-dos. Soit $\omega = (\omega_i, i=1, \dots, n)$ avec $\omega_i \in \{0,1\}$ une représentation. On a clairement $x_i = \omega_i, \forall i \in \{1, \dots, n\}$. C'est un choix plus naturel (et sans doute plus pertinent !) pour encoder une solution.

La Figure 7 contient les algorithmes de principe des procédures d'encodage d'une solution (version qui génère aléatoirement une représentation) et de décodage d'une représentation. Avec le décodage proposé, si un objet est trop lourd pour être placé dans le sac, les objets

suiuants dans la permutation seront malgré tout testés. La solution ainsi construite est saturée. Cela signifie qu'aucun des objets non choisis ne peut rentrer dans le sac. Si tous les objets ont une valeur positive il est facile de montrer que le sous-ensemble des solutions saturées contient une solution optimale.

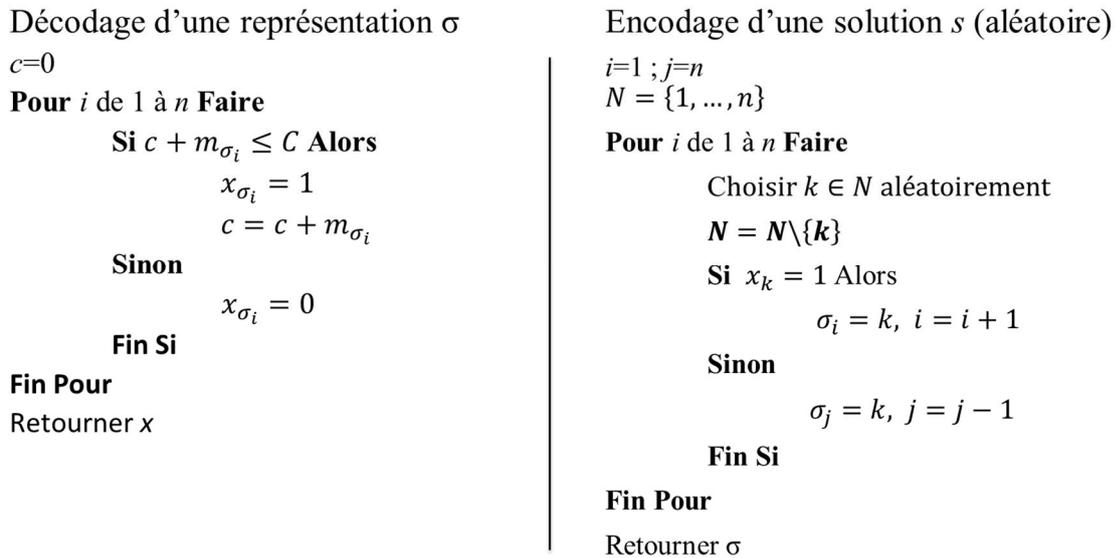


Figure 7. Schémas d'encodage et de décodage pour le problème du sac-à-dos

Exemple 2 : le problème du voyageur de commerce.

Une permutation est une description naturelle d'une solution du TSP, qui indique l'ordre de parcours des villes. L'espace des permutations est intègre. En effet, étant donné une permutation $\sigma = (\sigma_1, \dots, \sigma_n)$, il s'ensuit que $x_{i,i+1} = 1$ et $x_{n,1} = 1$ (avec la notation simplifiée $x_{i,j} = x_{\sigma_i, \sigma_j}$), les autres variables sont de valeur nulle. De plus, de par la structure d'une permutation, ces variables satisfont bien l'ensemble des contraintes définies pour le voyageur de commerce. Remarquons que pour le TSP, les espaces des solutions et des représentations ne sont pas reliés par une bijection. En effet, plusieurs représentations encodent la même solution, à un cycle près, et au sens de parcours près si le TSP est symétrique¹² (par exemple, les représentations $(3, 6, 1, 5, 4, 2)$ et $(5, 1, 6, 3, 2, 4)$ encodent la même solution pour le TSP symétrique).

L'espace des représentations peut être utilisé pour encoder des solutions correspondant à des problèmes très différents, renforçant le côté générique des métaheuristiques. C'est la procédure de décodage qui permet de faire le lien entre la métaheuristique et le problème. En plus du TSP ou du sac-à-dos, une permutation pourra également représenter des problèmes d'atelier (flow-job ou job-shop), de bin-packing, ou autres.

Dans la suite de ce chapitre, nous continuerons à illustrer notre propos en nous appuyant sur l'espace des permutations. Nous utiliserons la lettre $\sigma = (\sigma_1, \dots, \sigma_n)$ pour désigner une permutation.

2.3.3. ESPACE DES REPRESENTATIONS POUR QUELQUES METAHEURISTIQUES

¹² Le TSP est symétrique si la distance séparant une ville quelconque d'une autre ville quelconque ne dépend pas du sens de parcours. S'il existe une paire de villes ne satisfaisant pas cette propriété, le TSP est dit asymétrique.

Certaines métaheuristiques s'appuient sur des mécanismes particuliers. Le fait qu'elles évoluent dans un espace de recherche spécifique peut donc s'avérer intéressant, puisqu'il suffira d'adapter la métaheuristique au problème étudié (*via* son opérateur de décodage).

Nous allons illustrer l'intérêt de l'espace des représentations pour deux métaheuristiques emblématiques : les algorithmes génétiques et l'optimisation par essaim particulière.

Les algorithmes génétiques (Genetic Algorithms ou GA ; Holland, 1975) cherchent à simuler la théorie de l'évolution. Dans GA, une population d'individus, représentés par leur chromosome, évolue de génération en génération. Chaque individu s'efforce d'améliorer son fitness, à savoir sa capacité à se reproduire, à survivre et à transmettre son patrimoine génétique à la génération suivante.

Pour cela sont appliqués des opérateurs de croisement et de mutation. A partir du chromosome de deux parents choisis aléatoirement, les opérateurs de croisement permettent de générer de nouveaux individus appelés enfants. La Figure 8 montre quelques opérateurs classiques, illustrés à partir de la représentation standard d'un chromosome sous la forme d'une chaîne de bits (ici symbolisée par les caractères A et B). Chaque enfant a une possibilité de subir une mutation, qui se traduit généralement par une modification aléatoire d'un gène.

Parmi cette population où se mélangent parents et enfants, certains des individus, davantage ceux qui ont un meilleur fitness, pourront survivre et se retrouver à la génération suivante.

Bien que d'autres types de codage puissent être définis, l'utilisation du codage binaire permet de standardiser les opérateurs utilisés, et facilite ainsi la mise en œuvre de cette métaheuristique. Nous laissons le lecteur vérifier qu'appliquer un des opérateurs de croisement sur deux permutations ne conduit généralement pas à une permutation. Le chromosome obtenu nécessite donc d'être réparé.

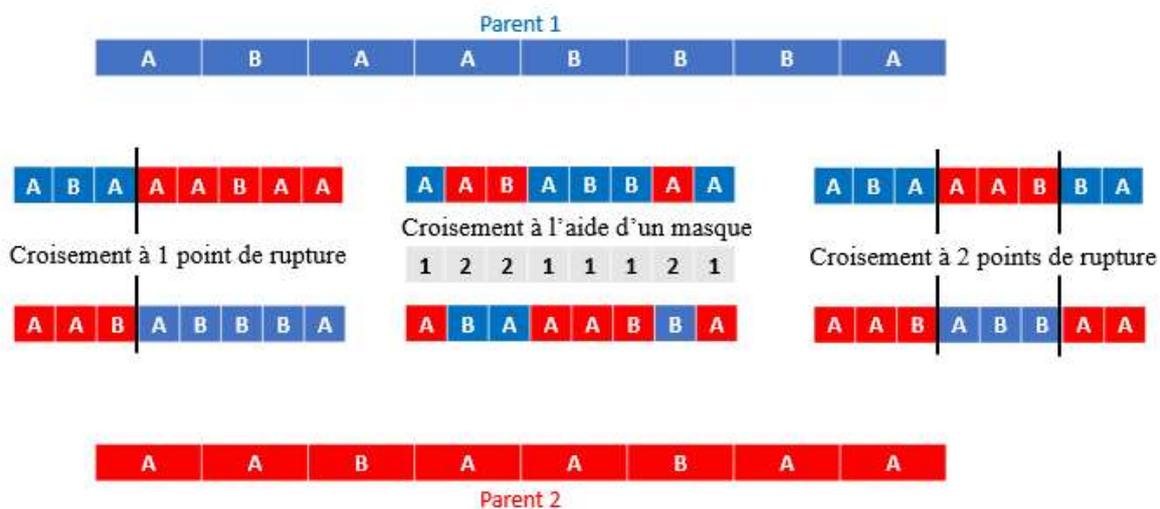


Figure 8. Illustration de croisements classiques dans les Algorithmes Génétiques.

L'Optimisation par Essaim Particulaire (Particle Swarm Optimization = PSO ; Kennedy & Eberhardt, 1995) est une métaheuristique qui cherche à reproduire le déplacement d'une nuée d'oiseaux ou d'un banc de poissons. Des particules se déplacent dans un espace en interagissant les unes avec les autres. Le déplacement d'une particule est obtenu en effectuant une combinaison linéaire de trois vecteurs, traduisant chacune une tendance propre de la particule (Clerc & Siarry, 2004) :

- La tendance *aventureuse* qui est l'aptitude d'une particule à suivre sa propre voie.
- La tendance *conservatrice* par laquelle une particule se souvient et cherche à revenir à la meilleure position qu'elle a occupée dans le passé.

- La tendance *panurgienne* qui traduit le fait qu'une particule communique avec d'autres et va suivre la meilleure information qu'elle a pu obtenir.

PSO est une métaheuristique qui a été conçue pour travailler dans l'espace vectoriel \mathbb{R}^n où n est le nombre de variables. (Tasgetiren *et al.*, 2004) ont appliqué PSO pour résoudre un problème de flow shop de permutation¹³. Les auteurs proposent d'utiliser une règle appelée Smallest Position Value (SPV) pour décoder un n-uplet $(\alpha_1, \dots, \alpha_n)$ et obtenir une solution (sous la forme d'une permutation). SPV peut se résumer ainsi : $(\alpha_i < \alpha_j \Rightarrow i < j)$, où " $<$ " est l'opérateur de précédence, i est avant j). Par exemple le vecteur $(-0.13; 1.20; -0.72; 0.25; 1.12)$ correspond à la permutation $(3, 1, 4, 5, 2)$. La troisième coordonnée de ce vecteur (-0.72) est la plus petite de ses composantes, donc la pièce 3 précède toutes les autres et est en première position dans la permutation.

Notons qu'il est tout à fait possible d'appliquer PSO sur un espace des représentations discret. (Clerc, 2005) montre comment les équations qui modélisent les déplacements des particules peuvent être adaptées pour travailler sur l'espace des permutations.

Maintenant que l'espace des représentations est introduit, nous allons voir quelques moyens dont disposent les métaheuristicques pour générer de nouvelles représentations.

2.4. MECANISMES DE CONSTRUCTION DE REPRESENTATIONS

« ... processus de génération itératif qui guide une heuristique subordonnée en combinant intelligemment différents concepts ... » (extrait de la définition d'une métaheuristique – Osman & Laporte, 1996)

Une métaheuristique doit pouvoir évoluer dans son espace des représentations. Cela signifie qu'elle doit posséder des mécanismes lui permettant de générer de nouvelles représentations.

Dans les sections qui suivent, vont être listés un certain nombre de mécanismes qui permettent de construire des représentations à partir de 0 (construction complète), 1 (altération d'une représentation) ou plusieurs autres représentations (combinaison de représentations). Ces mécanismes seront illustrés à partir du problème du voyageur de commerce et de l'espace des permutations. Ces mécanismes pourront être généralisés pour s'appliquer à d'autres problèmes ou d'autres représentations.

2.4.1. CONSTRUCTION D'UNE REPRESENTATION

De nombreux mécanismes permettent de construire une représentation, qui sont souvent apparentés à des heuristiques de construction. Deux grandes stratégies se dégagent : la première stratégie regroupe les heuristiques qui visent à construire une représentation progressivement, en partant d'une représentation partielle (voire vide) et en intégrant des éléments les uns après les autres. Des choix permettant de décider quel élément va être intégré, et à quel endroit, sont souvent faits selon des critères précis (optimisation locale). Si ces choix ne sont plus remis en question par la suite, on dit que ces heuristiques sont gloutonnes (greedy heuristics). La deuxième stratégie consiste à travailler sur la nature ou la structure du problème P étudié, par exemple en se ramenant à un problème P' jugé plus simple, et dont les solutions seront exploitées pour générer une représentation du problème initial P .

¹³ Un flow-shop de permutation est un problème d'atelier dans lequel des pièces sont à produire sur une ligne de production. Une permutation des pièces en amont de la ligne détermine l'ordre de passage des pièces pour toutes les machines qui composent la ligne.

Heuristiques gloutonnes pour le TSP

L'heuristique originelle proposée par Menger en 1930, aujourd'hui connu sous le nom de NN (Nearest Neighbor ou Plus Proche Voisin en français), fait partie de la catégorie des heuristiques gloutonnes. Le choix qui est fait de s'orienter vers la ville non encore visitée la plus proche (élément de W – voir l'encadré) est intuitif et a pour objectif de générer rapidement une permutation qui correspond à une solution de bonne qualité. A l'opposé, si le choix de la prochaine ville dans W est motivé uniquement par le hasard, cela revient à générer une permutation aléatoire. En répétant cette construction, on réalise une marche aléatoire dans l'espace des représentations.

Algorithme de principe des heuristiques gloutonnes de type Nearest Neighbor

```

Choisir une ville de départ  $i$ 
 $\sigma \leftarrow (i)$ 
 $W = \{1, \dots, n\} \setminus \{i\}$ 
Tant que  $W \neq \emptyset$  Faire
    Choisir  $j \in W$ 
     $\sigma \leftarrow \sigma + (j)$  (concaténation)
     $W = W \setminus \{j\}$ 
Fin Tant Que
Retourner la permutation  $\sigma$ 
    
```

Tout choix intermédiaire réalisera un compromis entre la recherche d'une bonne solution et la construction d'une solution aléatoire. On peut ainsi imaginer une version randomisée de NN, dans laquelle le choix de la prochaine ville se fait au hasard parmi un sous-ensemble restreint de villes (S), que nous appellerons villes candidates, par exemple celles situées à proximité de la ville courante. La ville suivante peut être choisie uniformément parmi l'ensemble des villes candidates, ou avec une pondération donnant plus de poids aux villes estimées les plus intéressantes. Le principe de la sélection par roulette, cher aux algorithmes génétiques, est souvent utilisé. La probabilité de sélectionner l'individu j parmi une population P d'individus

est donné par $p(j) = \frac{f_j}{\sum_{k \in P} f_k}$, où f_j représente le fitness de l'individu i . La Figure 9 illustre ce

principe avec une population de 6 individus en posant $f_j = j$.

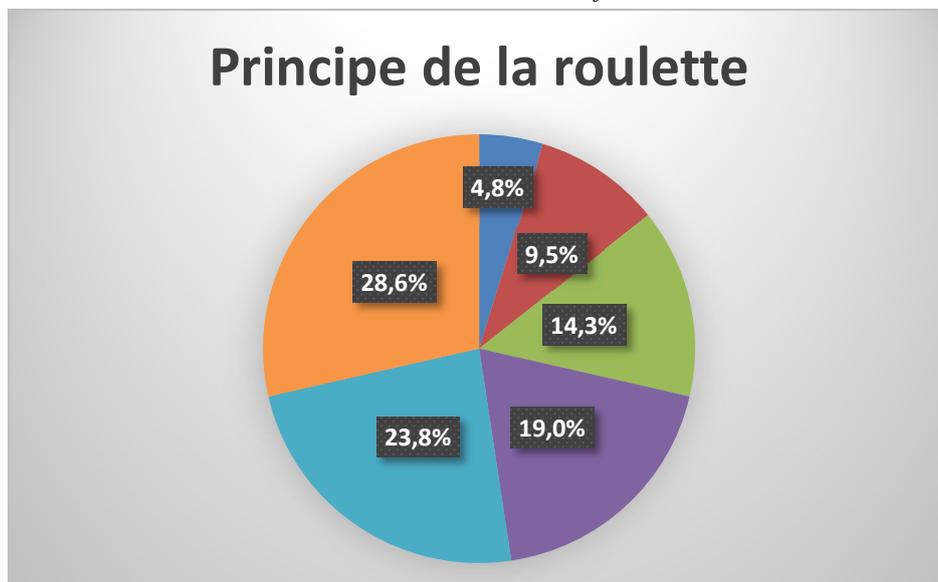


Figure 9. Illustration du principe de la roulette avec une population de 6 individus et $f_j = j$.

L'individu n°6 (repéré par le secteur le plus grand), a une probabilité d'être choisi égale à $6/21$ (environ 28.6%). Pour l'individu 1 (secteur le plus petit), cette probabilité chute à $1/21$ (4,8%).

De nombreuses métaheuristiques exploitent un tel mécanisme (Hao & Solnon, 2014).

- GRASP utilise la notion de Restricted Candidate List (RCL), qui peut évoluer pendant le déroulement de la recherche. Si i est la ville courante, les villes candidates pour lui succéder peuvent être choisies en donnant plus de chance aux villes les plus proches de i , par exemple en posant $f_j = \eta(i, j) = \frac{1}{d_{i,j}}$, pour tout $j \in RCL(i) \cap W$ $j \in RCL(i) \cap W$ (pour rappel $d_{i,j}$ est la distance entre les villes i et j).
- Dans EDA (Estimation of Distribution Algorithms), une métaheuristique de la famille des algorithmes évolutionnaires, on pose $f_j = f^{(p)}(i, j)$ où $f^{(p)}(i, j)$ est la fréquence d'apparition de l'arc (i, j) dans la population P . Partant de la ville i , EDA cherche donc à choisir la ville suivante en estimant l'effet qu'aura ce choix sur la qualité de la solution globale, et plus seulement en s'appuyant sur des considérations locales (les villes les plus proches de i). Cette idée est également présente, sous une forme différente, dans le mécanisme suivant.
- Dans ACS (Ant Colony System), on a $f_j = \eta(i, j)^\alpha \tau(i, j)^\beta$ où $\eta(i, j) = \frac{1}{d_{i,j}}$ représente le facteur d'attractivité basé sur les distances et $\tau(i, j)$ est le facteur de désirabilité d'un arc, basé sur le taux de phéromones. Le premier facteur permet de tenir compte des villes qui sont potentiellement les plus intéressantes, tandis que la désirabilité permet d'apprendre l'intérêt d'un arc. Les coefficients alpha et beta permettent de donner plus ou moins de poids à l'attractivité par rapport à la désirabilité. Les règles de dépôt et d'évaporation des phéromones s'apparentent à une technique d'apprentissage par renforcement. Chaque fourmi dépose des phéromones sur les arcs empruntés (règle locale). La solution record est utilisée pour renforcer les arcs (règle globale).

Il existe bien d'autres heuristiques gloutonnes pour le TSP, comme par exemple les heuristiques d'insertion. Partant d'un sous-cycle de trois villes (une seule solution pour le TSP symétrique), ces heuristiques consistent à choisir à chaque itération la prochaine ville à insérer (la plus proche du sous-cycle, la plus éloignée, au hasard, ...) et à l'insérer à la meilleure position possible, jusqu'à avoir inséré toutes les villes. De nombreuses stratégies sont par exemple décrites, testées et comparées dans (Reinelt, 1994).

Heuristiques de construction globale

Le Tour de Cristofides (Cristofides, 1976) est une heuristique de construction célèbre pour le TSP. Cette heuristique n'est sans doute pas la plus facile à implémenter, et la solution obtenue est de qualité relativement pauvre. Par contre, elle possède une bonne structure globale, et elle offre un excellent point de départ pour des méthodes de recherche locale.

Le point de départ consiste à considérer qu'un tour est un arbre couvrant particulier, auquel on rajoute une arête pour reconstituer un cycle, et tel que tous ses sommets sont de degré deux. Or, le problème de l'arbre couvrant de coût minimum est un problème de classe P, qui peut être résolu avec l'algorithme de Prim ou de Kruskal.

Fort de ce constat, (Cristofides, 1976) propose une heuristique en 4 étapes, illustrée par la Figure 10. La première étape consiste à construire un arbre couvrant de coût minimum. Des arêtes sont ensuite rajoutées entre les sommets de degré impair (couplage de poids minimum – problème de classe P) pour obtenir un graphe eulérien (une variante existe qui consiste simplement à doubler toutes les arêtes de l'arbre couvrant). De ce graphe est extrait un cycle eulérien. La

quatrième et dernière étape se résume à parcourir les villes de ce cycle eulérien, et de supprimer les doublons pour obtenir un cycle hamiltonien.

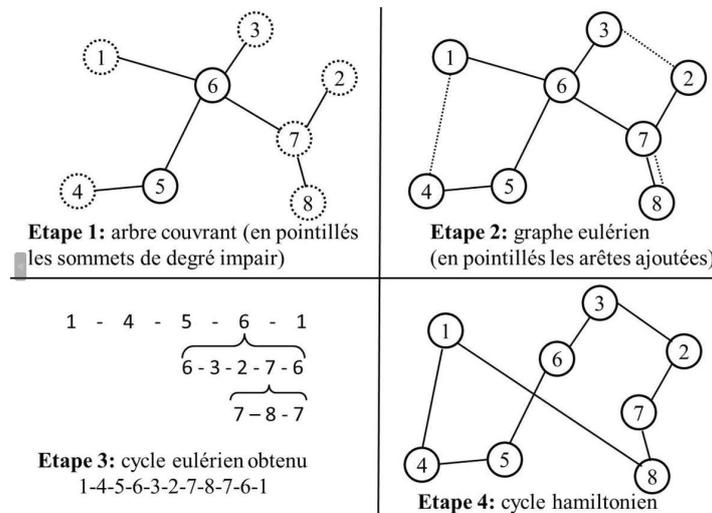


Figure 10. Illustration du tour de Cristofides.

Il existe une multitude d'autres heuristiques pour le TSP.

Les heuristiques **Savings** consistent à fusionner progressivement des sous-tours entre eux jusqu'à ce qu'il n'en reste plus qu'un. Le point de départ peut être que chaque ville est son propre sous-tour, ou les sous-tours obtenus en résolvant le problème d'affectation (qui est une relaxation du TSP, de classe P). La fusion de deux sous-tours est en général réalisée de manière gloutonne.

Les heuristiques **Sweep** consistent, partant par exemple de la position centrale des villes, à balayer le plan euclidien à 360° autour de ce point, et à constituer un cycle en relevant les villes dans l'ordre où elles sont rencontrées.

Les heuristiques s'appuient sur la connaissance du problème pour construire des solutions. Certaines d'entre elles peuvent intégrer des décisions aléatoires permettant de randomiser la construction et par là-même de générer un grand nombre de représentations. Il est également possible de mettre en place des techniques de listes restreintes de candidats pour guider ces choix. Certaines métaheuristiques comme GRASP ou ACS utiliseront ce type de mécanismes afin d'exploiter ou d'explorer l'espace de recherche. Ces notions seront abordées un peu plus loin.

2.4.2. CONSTRUCTION PAR MODIFICATION D'UNE REPRÉSENTATION

Les méthodes à trajectoire

Cette section met en avant les représentations obtenues à partir d'autres représentations. Un mécanisme est appliqué à une représentation, et permet d'obtenir une nouvelle représentation. Ces mécanismes sont extrêmement importants pour les métaheuristiques, car un des objectifs poursuivis est, en altérant légèrement une représentation, d'obtenir une représentation structurellement proche, et donc, l'espère-t-on, qualitativement proche. Ces mécanismes sont souvent appelés des mouvements et permettent de reconstruire de manière *ad hoc* le voisinage d'une représentation quelconque. C'est une façon de contrer le théorème NFL en exploitant la corrélation entre la structure des solutions d'un problème (via leur représentation) et leur coût. Il serait cependant faux de limiter l'usage des mouvements à cette seule finalité. Appliqués de

manière itérative à partir d'une représentation, ils dessinent une trajectoire¹⁴ dans l'espace de recherche. Les mouvements sont de fait un procédé mécanique qui permet à la métaheuristique de se mouvoir à l'envi dans l'espace de recherche.

Soit x la représentation courante, et soit $y \in M(x)$ ¹⁵ la représentation obtenue en appliquant un mouvement M . La fonction $AccepteTransition(x, y)$ indique la règle utilisée pour déterminer si la transition de x vers y est acceptée ou non. Cette règle dépend de la métaheuristique utilisée.

- $AccepteTransition(x, y) = y$: le mouvement est toujours appliqué, même s'il conduit à l'obtention d'une solution moins bonne. Cette règle décrit le comportement de la recherche aveugle. Cette stratégie consiste à errer au hasard dans l'espace des représentations, sans chercher à obtenir de bonnes solutions.
- $AccepteTransition(x, y) = \underset{a \in \{x, y\}}{argmin} (f \circ \mathcal{D}(a))$: la représentation courante sera la meilleure des deux (celle qui, une fois décodée, conduit à la solution de meilleur coût). Cette règle est appliquée dans la recherche locale. Son application conduit dans un minimum local
- $AccepteTransition(x, y) = y$ si :

$$f \circ \mathcal{D}(y) \leq f \circ \mathcal{D}(x), \text{ ou}$$

$$\alpha \leq \exp\left(-\frac{f \circ \mathcal{D}(y) - f \circ \mathcal{D}(x)}{T}\right) \text{ avec } \alpha \rightarrow U(0,1).$$

La transition de x vers y est acceptée si y est meilleure que x , ou avec une certaine probabilité si elle est moins bonne. Cette règle de transition est utilisée dans l'algorithme du recuit simulé. C'est une stratégie intermédiaire qui permet, avec une bonne gestion du paramètre T , de ne pas se faire piéger trop rapidement dans un minimum local.

Une métaheuristique utilise souvent plusieurs règles de transition. A titre d'illustration, la métaheuristique ILS utilise la deuxième règle dans la phase de recherche locale, la première règle dans la phase de perturbation et une règle quelconque pour le critère d'acceptation du minimum local (Les trois règles énoncées correspondent respectivement, lorsque x et y sont deux minima locaux, aux politiques *RW*, *Better*, *LSMC* définies dans (Lourenço et al., 2003) pour le choix du critère d'acceptation).

Dans la suite, quelques mouvements portant sur les permutations sont présentés, en allant des plus simples aux plus élaborés.

Mouvements de base

Pour une métaheuristique parcourant l'espace des représentations, trois mouvements de base (basic moves) sont régulièrement présentés dans la littérature. Les mouvements d'échange (exchange move) consistent à échanger deux éléments de la permutation. Pour l'insertion (insertion move), un élément est enlevé de la permutation pour être inséré à une autre position. Quant aux mouvements d'inversion (inversion move), ils consistent à inverser l'ordre de lecture

¹⁴ Le terme de trajectory methods est parfois utilisé dans la littérature scientifique pour désigner des métaheuristicques à individu (recherche taboue, recuit simulé, recherche locale itérée, ...). L'individu décrit une trajectoire dans l'espace des représentations.

¹⁵ A ce stade, nous aurions pu utiliser le mot voisinage. Nous préférons cependant le réserver pour l'espace des solutions, et utiliser le mot mouvement pour l'espace des représentation (voir section 2.5.3)

entre deux éléments. La Figure 11 illustre chacun de ces trois mouvements appliqués aux éléments situés en 3^{ème} et 6^{ème} position.

Comme vu précédemment, lorsqu'un mouvement est appliqué, se posera la question de valider ou non son exécution. Il suffit d'appliquer le même mouvement en intervertissant les paramètres pour annuler son exécution.

$$\text{Move}(\sigma, i, j) \circ \text{Move}(\sigma, i, j) = \sigma, \text{ avec Move} = \text{Echange, Insertion ou Inversion} ;$$

Notons également que les mouvements d'échange et d'inversion sont symétriques :

$$\text{Echange}(\sigma, i, j) = \text{Echange}(\sigma, j, i) \text{ et } \text{Inversion}(\sigma, i, j) = \text{Inversion}(\sigma, j, i)$$

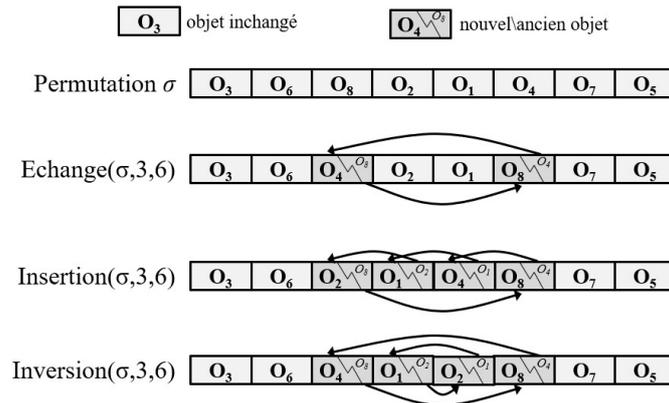


Figure 11. Illustration des mouvements de base d'une permutation : Echange, Insertion et Inversion.

Mouvements LK (Lin & Kernighan, 1973)

Les mouvements que nous allons présenter dans cette partie sont dédiés au problème du TSP. Ils mettent en œuvre des techniques de guidage qui utilisent les propriétés du problème.

Observons plus attentivement la solution proposée pour notre voyageur de commerce allemand (Figure 5). Il existe beaucoup de similitudes entre une solution de bonne qualité et une solution optimale (11 arêtes de différence sur 45, soit un taux de similitude de 76%).

La motivation première des auteurs était justement d'identifier ces 11 arêtes à enlever. Pour cela, ils ont imaginé un mouvement k -opt où k est déterminé dynamiquement pendant la construction du mouvement.

Le principe est le suivant. Supposons que la solution courante T (non optimale) se différencie d'une solution optimale T^* par les arêtes $x_1, \dots, x_k \in T$ qu'il faut remplacer par les arêtes $y_1, \dots, y_k \notin T$. La difficulté pour rendre le tour T optimal est donc d'identifier ce nombre k ainsi que les arêtes x_1, \dots, x_k et y_1, \dots, y_k . L'idée est d'essayer de déterminer ces éléments progressivement, en commençant par une première paire d'arêtes (x_1, y_1) , qui conduit à un gain local $g_1 = d(y_1) - d(x_1) > 0$ (où $d(a)$ est le coût de l'arête a), puis une deuxième paire d'arête (x_2, y_2) tel que $g_1 + g_2 > 0$, etc.

Pour cela, les auteurs introduisent la notion de transformation séquentielle. Il est possible de transformer séquentiellement une solution T en une solution T' s'il est possible de numérotter les arêtes enlevées x_1, \dots, x_k et les arêtes ajoutées y_1, \dots, y_k de telle manière que les arêtes x_i et y_i aient un sommet en commun, de même que les arêtes y_i et x_{i+1} (en posant $x_{k+1} = x_1$).

On pourra alors exhiber une liste de sommets $t = (t_1, \dots, t_{2k})$ qui satisfait $x_i = (t_{2i-1}, t_{2i})$ et $y_i = (t_{2i}, t_{2i+1})$ pour tout $i = 1, \dots, k$.

Pour illustrer ce mouvement et en montrer l'intérêt, nous allons essayer d'améliorer localement le problème de notre TSP allemand (Figure 12). Les villes sont notées par leur initiale. Les arêtes enlevées sont (N, W) , (L, Z) , (Z, G) et (A, M) ; celles ajoutées sont (N, G) , (A, Z) , (Z, W) et (L, M) . Il est possible de transformer la solution à gauche sur la figure, en celle de droite par une transformation séquentielle. Une liste de sommets possible est $t = (N, W, Z, L, M, A, Z, G)$, correspondant aux arêtes $x_1 = (N, W)$, $y_1 = (W, Z)$, $x_2 = (Z, L)$, $y_2 = (L, M)$, $x_3 = (M, A)$, $y_3 = (A, Z)$, $x_4 = (Z, G)$ et $y_4 = (G, N)$.

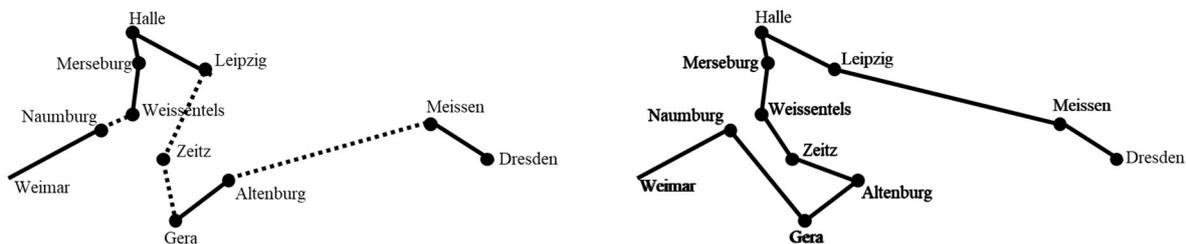


Figure 12. Zoom sur les solutions du TSP pour le problème des 45 villes allemandes.

A gauche la solution proposée en 1830. A droite la solution optimale.

Un mouvement 4-opt séquentiel permet de passer de la solution de gauche à celle de droite.

Chaînes d'éjection (Glover, 1982)

Cette technique manipule des structures appelées "Stem-and-Cycle" (cycle et tige) composées d'un cycle (r, s_1, \dots, s_2, r) et d'une tige (r, \dots, t) . La ville r , située à l'intersection du cycle et de la tige, constitue la racine de cette structure. s_1 et s_2 sont les sous-racines et t est l'extrémité de la tige (Figure 13). En supprimant l'arête (r, s_i) reliant la racine à une des deux sous-racines, et en rajoutant l'arête (s_i, t) , on peut reconstituer les deux solutions qui sont représentées sur la partie droite de la figure.

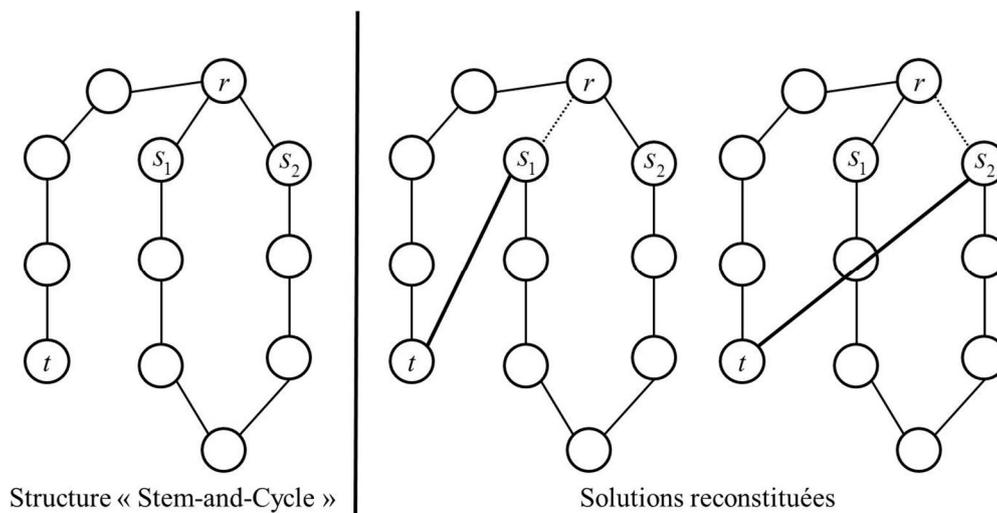


Figure 13. Exemple d'une structure "Stem-and-Cycle" avec les solutions reconstituées.

Partant de la solution courante T , on commence par ajouter une arête $(r, s_1) \notin T$ et supprimer une des deux arêtes $(s_1, t) \in T$ incidentes à s_1 . On obtient la structure Stem-and-Cycle initiale.

Dans son article, l'auteur énonce des règles qui permettent de transformer une structure "Stem-and-Cycle" en une nouvelle structure "Stem-and-Cycle". A chaque structure créée, les deux solutions sont considérées et évaluées dans le but d'améliorer la solution courante.

Le mouvement ainsi créé est très similaire au mouvement LK. Pour les chaînes d'éjection, le mouvement est initié en ajoutant une arête à la solution courante, alors que pour les mouvements LK on commence par supprimer une arête.

Ces mouvements élaborés ont montré leur efficacité sur le TSP. Aujourd'hui encore, les mouvements de type LK, dans une version améliorée par (Helsgaun, 2000), font office de référence.

2.4.3. CONSTRUCTION PAR COMBINAISON DE PLUSIEURS REPRESENTATIONS

De nombreuses métaheuristiques utilisent des mécanismes faisant intervenir au moins deux permutations pour en concevoir de nouvelles. Les opérateurs de croisement, path relinking et les opérateurs de déplacement dans l'algorithme par essaim particulière seront mis en avant dans cette section.

L'opérateur de croisement

L'opérateur de croisement provient des algorithmes génétiques. Son objectif est d'utiliser le patrimoine génétique des parents, de manière à créer des enfants qui vont à la fois posséder de bonnes caractéristiques héritées de leurs parents, et aussi leurs propres caractéristiques, afin de garantir une certaine diversité.

Un des opérateurs régulièrement utilisé est le croisement à un point de rupture. Nous l'avons déjà décrit, appliqué à des vecteurs binaires. Il convient de faire quelques adaptations pour l'appliquer sur des permutations.

Soit deux parents P1 et P2 représentés sur la Figure 14, avec un point de rupture situé au milieu du chromosome. En combinant la première moitié du chromosome du Parent P1, avec la deuxième moitié du chromosome du parent P2, le chromosome obtenu serait $(5, 7, 2, 6, 1, 9, 7, 8, 5, 2)$. Ce chromosome ne correspond pas à une permutation en raison de la répétition des villes 5, 7 et 2. L'idée pour obtenir une permutation est de prendre effectivement la première moitié du chromosome 1 $(5, 7, 2, 6, 1)$ et de le compléter avec les villes manquantes, d'après l'ordre de lecture induit par le chromosome du parent P2 $(3, 6, 10, 1, 4, 9, 7, 8, 5, 2)$ $(3, 6, 10, 1, 4, 9, 7, 8, 5, 2)$. Les villes déjà présentes sont rayées. Il reste les villes $(3, 10, 4, 9, 8)$ qui forment la deuxième moitié du chromosome de l'enfant 1. En procédant de manière analogue en inversant le rôle des deux parent, le chromosome de l'enfant 2 est reconstitué.

Path Relinking

Ce mécanisme est une nouvelle fois issu des travaux de (Glover *et al.*, 2000) dans le contexte de scatter search. Le but de Path Relinking est, partant d'une permutation donnée (l'origine), d'aller vers une permutation destination (le modèle), en introduisant progressivement des attributs du modèle dans l'origine. Les permutations ainsi générées définissent un chemin de l'origine vers le modèle.

Dans l'espace des permutations, un moyen simple de générer ce chemin est par exemple d'appliquer une succession de mouvements de base (l'échange de deux villes est souvent utilisée). Le lecteur pourra vérifier qu'en appliquant successivement les mouvements d'échange $(5, 3), (7, 6), (2, 10), (1, 7), (4, 7), (7, 9), (2, 7), (2, 5)$, on construit un chemin dans l'espace des

permutations du parent P1 de la Figure 14 vers le parent P2. Les attributs qui sont introduits sont le nombre d'éléments situés dans la même position que la permutation modèle P2.

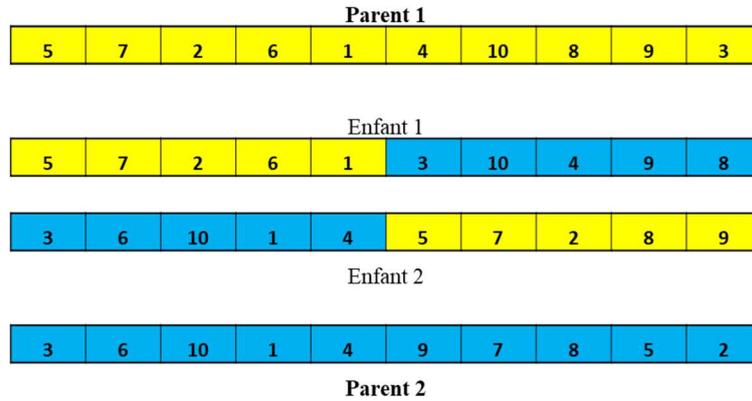


Figure 14. Croisement à 1 point de rupture appliqué à des permutations

Dans le contexte du TSP, les attributs qui servent à guider la définition du chemin sont généralement les arêtes. On peut déterminer une succession de mouvements 2-opt de telle manière que le nombre d'arêtes commune avec la permutation modèle soit strictement croissant.

Path Relinking est un mécanisme qui a été largement utilisé dans d'autres métaheuristiques, parfois sans y faire explicitement référence. Nous pouvons mentionner par exemple l'utilisation de path relinking pour construire des opérateurs de croisement d'un algorithme génétique (Reeves & Yamada, 1998), ou pour discrétiser les opérateurs de mouvements dans les essaims particuliers (Clerc, 2004). Ce dernier mécanisme est détaillé ci-après.

Mouvements de particules dans l'Optimisation par Essaim Particulaire

PSO (Particle Swarm Optimization) est une métaheuristique conçue pour travailler dans un espace continu. Des particules sont injectées dans cet espace et à tout instant t , la particule i est caractérisée par sa position $p_{i,t}$ et sa vitesse $v_{i,t}$. Si aucune interaction ne vient perturber sa trajectoire, à l'instant $t + 1$, la particule se trouvera à la position $p_{i,t+1} = p_{i,t} + v_{i,t}$.

Les particules gardent la mémoire de la meilleure position qu'elles ont trouvée depuis le début de la recherche $p_{i,t'}$.

Chaque particule communique avec d'autres particules et est en mesure de connaître la meilleure position obtenue à l'intérieur de son réseau d'informatrices $p_{j,t}$.

Lorsqu'elle se déplace, la particule dévie de sa trajectoire initiale pour prendre en compte les informations dont elle dispose. Sa nouvelle vitesse sera calculée grâce à la relation :

$$v_{i,t+1} = \lambda_1 v_{i,t} + \lambda_2 \overline{p_{i,t} p_{i,t'}} + \lambda_3 \overline{p_{i,t} p_{j,t}}$$

Les trois termes de cette somme correspondent aux tendances aventurière, conservatrice et panurgienne que nous avons déjà mentionnées. Les coefficients λ_1 , λ_2 et λ_3 déterminent les poids relatifs donnés à chaque tendance.

On déduit alors la position de la particule à l'itération suivante.

$$p_{i,t+1} = p_{i,t} + v_{i,t+1}$$

Dans l'espace des permutations, (Clerc, 2004) propose de définir le vecteur vitesse comme la différence entre deux permutations, et plus précisément comme une liste de mouvements qui

permet de passer de la permutation origine à la permutation destination (on retrouve le path relinking). A partir de ce postulat, il reconstruit une algèbre qui lui permet d'appliquer les équations de mouvements des particules sur l'espace des permutations.

Les quelques mécanismes décrits ici, plus de nombreux autres, permettent aux métaheuristiques de générer de nouvelles représentations. Dans la section suivante, nous allons voir dans quel but elles vont les utiliser, pour exploiter et explorer l'espace de recherche.

2.5. NOTIONS D'EXPLOITATION ET D'EXPLORATION

« ... Toute métaheuristique doit combiner intelligemment différents concepts pour explorer et exploiter l'espace de recherche... » (extrait de la définition d'une métaheuristique – Osman & Laporte, 1996)

2.5.1. DEFINITION DES CONCEPTS D'EXPLOITATION ET D'EXPLORATION

Les notions d'exploration et d'exploitation sont les notions clés des métaheuristiques, qui vont nous aider à parcourir l'espace des représentations. Naïvement, l'exploitation consiste à concentrer les efforts à proximité des bonnes représentations (Glover parle d'un sous-ensemble de solutions élites), tandis que l'exploration permet d'aller librement dans l'espace de recherche. Il est communément admis par la communauté que pour être performante, une métaheuristique doit réaliser un bon équilibre entre l'exploration et l'exploitation, sans que l'on sache précisément dire ce qu'est un bon équilibre.

Le comportement d'une métaheuristique est directement relié aux notions d'exploitation et d'exploration. Une métaheuristique qui privilégie trop l'exploitation convergera rapidement mais se retrouvera piégée dans un optimum local, tandis qu'une métaheuristique qui privilégie trop l'exploration aura toutes les chances de trouver un optimum global, mais avec une convergence très lente. Le fameux équilibre recherché consiste donc à trouver le meilleur compromis entre la qualité de la solution obtenue et le temps mis pour l'obtenir.

Aux termes d'exploitation et d'exploration, sont parfois préférés les termes d'intensification et de diversification. Même si aujourd'hui, les concepts qu'ils désignent sont souvent confondus, ce n'était pas le cas à l'origine.

L'exploration et l'exploitation proviennent du champ métaphorique des algorithmes génétiques. Ils cherchent à prévenir l'algorithme génétique du phénomène de convergence prématurée¹⁶. Ces notions sont moins triviales qu'il n'y paraît, et sont parfois interprétées de différentes manières. Ainsi, (Eiben & Schippers, 1998) notent que beaucoup d'utilisateurs des AG pensent que l'exploitation et exploration sont des forces opposées. L'exploration est réalisée par les opérateurs de croisement et de mutation. De nouvelles solutions sont alors générées dans l'espace de recherche (d'où l'exploration). L'exploitation est à la charge de l'opérateur de sélection, qui choisit quelles solutions conserver, parmi un panel de solutions existantes. (Crepinsek *et al.*, 2013) dépeignent une autre vision, selon laquelle les opérateurs de sélection, de mutation et de croisement contribuent tous à l'exploration et à l'exploitation *via* des mécanismes de contrôle (paramètres). Considérons par exemple l'opérateur de sélection, qui consiste à déterminer, parmi les individus d'une génération, lesquels vont survivre à la génération suivante. La stratégie purement élitiste qui consiste à conserver les meilleurs individus est un facteur d'exploitation. Celle aléatoire qui consiste à garder les individus au hasard est un facteur d'exploration. Toute stratégie hybride qui consiste à garder avec une proportion τ les meilleurs individus et à compléter avec des individus choisis au hasard contribue à l'exploitation et à l'exploration.

¹⁶ Dans les algorithmes évolutionnaires, et par extension dans les métaheuristiques à population, la convergence prématurée signifie que la population converge trop vite vers une zone de l'espace de recherche qui ne contient pas d'optimum global. La population d'individus se retrouvent alors "piégée" dans cette zone.

Les concepts d'intensification et de diversification ont été introduits dans le contexte de la recherche taboue. (Glover & Laguna, 1998) les présentent comme des composants importants de la recherche taboue. Les auteurs définissent l'intensification comme une recherche focalisée sur l'étude du voisinage¹⁷ des solutions élites ; le reste étant de la diversification. Peu à peu, cette définition s'est imposée dans la littérature. L'intensification ou exploitation désignent l'aptitude d'une métaheuristique à orienter la recherche dans les zones prometteuses de l'espace de recherche. La diversification ou exploration caractérisent l'aptitude à prospecter l'espace de recherche pour trouver de nouvelles zones prometteuses.

Pour désigner ces concepts, nous utiliserons les termes d'exploitation et d'exploration, ceux-là même qui sont utilisés dans la définition d'Osman & Laporte. Leur signification sera par contre celle définie dans la recherche taboue, qui correspond aujourd'hui au sens usuel. Les métaheuristiques possèdent des mécanismes qui contribuent à l'exploitation ou à l'exploration. Il est essentiel de noter que contrairement à certaines idées reçues, ces notions ne s'opposent pas. Un même mécanisme peut contribuer aux deux concepts.

Le critère d'acceptation de la solution voisine dans le recuit simulé illustre parfaitement ce dernier point. Considérons l'espace des solutions et soit x la solution courante. Une solution voisine $y \in \mathcal{N}(x)$ choisie aléatoirement sera substituée à x si :

- elle est améliorante ($f(y) \leq f(x)$ pour un problème de minimisation
- elle n'est pas améliorante, avec une probabilité qui dépend de la détérioration $\Delta f = f(y) - f(x)$ et d'un paramètre T appelé température (souvent la probabilité

$$P(\Delta f, T) = \exp\left(-\frac{\Delta f}{T}\right) \text{ est utilisée.})$$

Lorsque la température est élevée ($T \gg \Delta f$), cette probabilité est proche de 1. Le recuit simulé se comporte alors comme une recherche aveugle, et ce mécanisme contribue à l'exploration.

Lorsque la température est proche de 0, cette probabilité est également proche de 0. Le recuit simulé se comporte alors comme une recherche locale¹⁸, et ce mécanisme contribue à l'exploitation.

Une des forces du recuit simulé est qu'en diminuant progressivement la température, l'équilibre qui au départ favorise l'exploration, va peu à peu laisser sa place à une stratégie qui favorise l'exploitation.

(Blum & Roli, 2003) ont proposé un cadre pour unifier l'effet des mécanismes sur l'exploration et l'exploitation. L'espace des mécanismes est représenté sous la forme d'un triangle dont les sommets vont regrouper des mécanismes qui ont des comportements particuliers (Figure 15). Le coin R regroupe les mécanismes d'exploitation et d'exploration uniquement basés sur le hasard, OG les mécanismes guidés uniquement par la fonction objectif et NOG les mécanismes guidés par d'autres fonctions que la fonction objectif.

En reprenant l'exemple du recuit simulé, les auteurs le positionnent à l'intérieur du triangle, entre les coins R, OG et NOG. Le critère d'acceptation du recuit dépend à la fois de la fonction objectif (OG) et du hasard, puisqu'il dépend d'une probabilité (R). La température est un paramètre qui, en baissant, va contribuer à piloter la recherche de la diversification vers l'intensification (NOG).

¹⁷ *Voisinage* a ici un sens plus large que celui du système de voisinage utilisé dans les recherches locales.

¹⁸ Plus précisément une descente stochastique (section 3.2.2)

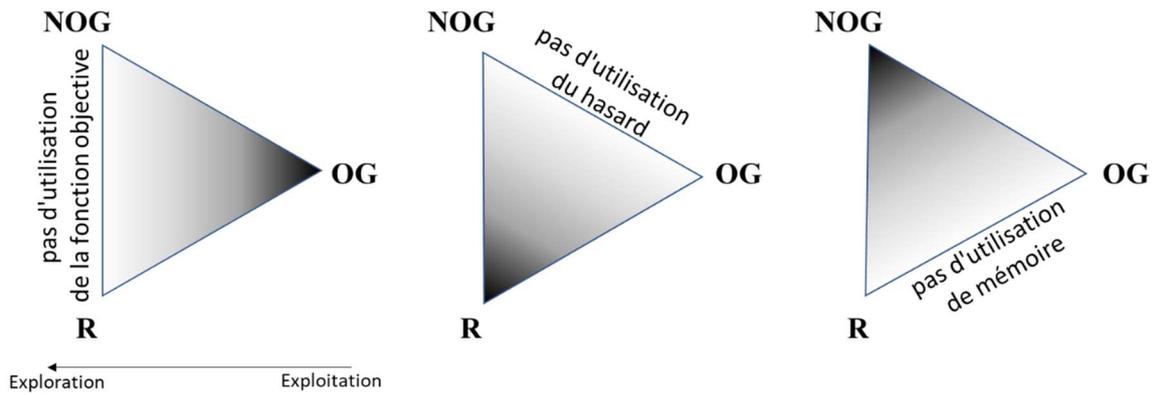


Figure 15. Vue unifiée des concepts d'exploitation et d'exploration pour les métaheuristiques (traduit et adapté de Blum & Roli, 2003)

Au cours des deux sections suivantes, nous allons nous intéresser à l'usage des mémoires et des voisinages, pour guider une métaheuristique et élaborer une stratégie de recherche.

2.5.2. GUIDER UNE METAHEURISTIQUE AVEC L'UTILISATION DE MEMOIRES

L'utilisation des mémoires pour guider la recherche est apparue progressivement avec le développement de la recherche taboue. La relecture de quelques articles de F. Glover (Glover, 1986 ; 1989 ; 1997 ; Glover & Laguna 1998) autorise le constat suivant. Les objectifs poursuivis dans les premiers articles, structurés autour de l'intégration des techniques d'intelligence artificielle pour guider la recherche, cèdent progressivement la place aux moyens pour y parvenir : la mémoire.

Une présentation de mémoire à moyen et long terme commence à apparaître dès l'article de 1989.

- La mémoire à moyen terme se concentre sur le stockage de caractéristiques communes dans les solutions de bonne qualité, obtenues pendant une période donnée de la recherche. Ces informations sont ensuite exploitées pour générer de nouvelles solutions possédant les caractéristiques observées.
- Le rôle de la mémoire à long terme est au contraire de diversifier la recherche en l'orientant dans des régions de l'espace différentes de celles déjà examinées, par exemple en pénalisant les caractéristiques communes (par exemple le coût des arcs les plus fréquemment rencontrés pour le TSP).

C'est dans les papiers plus récents que s'est formalisé le concept de mémoires. La mémoire tient en effet un rôle central autour duquel s'articulent les articles de 1997 et 1998.

"Over a wide range of problem settings, however, strategic use of memory can make dramatic differences in the ability to solve problems." (Glover, 1997)

Dans ces articles, la mémoire à moyen terme a disparu, au profit de mémoires à court terme et à long terme. Si on essaye de faire l'analogie avec le fonctionnement de la mémoire humaine (Figure 16), la mémoire à court terme s'apparente effectivement à une mémoire de travail, dans laquelle on va stocker de l'information sous des formes qui peuvent être très variées, et qui vont nous aider à prendre des décisions. L'objectif de cette mémoire est principalement d'appuyer l'exploitation. Il s'agit d'être le plus agressif possible dans la recherche de solutions de bonne qualité. La mémoire à long terme aura pour rôle de mettre en place une stratégie de recherche, en étant capable de prévoir les zones dans lesquelles il serait judicieux d'arrêter, de poursuivre ou de commencer la recherche.

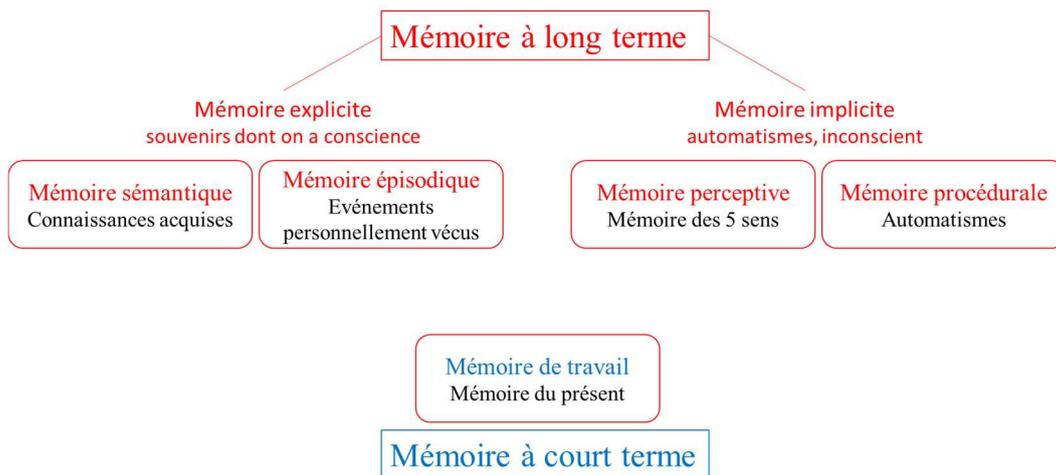


Figure 16. Cartographie des cinq types de mémoire humaine réalisée à partir du site de l'Inserm (<https://www.inserm.fr/dossier/memoire>), et du site d'Amélie Wallyn, Ergothérapeute (<https://autonome-a-domicile.com/>).

Nous proposons de réorganiser les travaux de (Glover & Laguna, 1998) sur la mémoire en disant qu'un mécanisme mémoriel repose sur trois dimensions : sa persistance, sa nature et sa structure.

- La persistance désigne la durée de vie pendant laquelle une information est stockée. La persistance peut être très courte, interne à la construction d'une solution voisine (par exemple, dans la construction d'un mouvement L&K pour le TSP, une arête qui a déjà été enlevée ou ajoutée est stockée et n'a plus le droit de changer de nouveau d'état jusqu'à la fin du mouvement). D'autres informations vont être stockées pendant quelques itérations (classiquement une liste taboue, où une information est stockée le temps de permettre à la recherche de s'extraire d'un minimum local, les auteurs utilisent le terme de "*tenure value*" pour désigner cette durée). D'autres enfin vont être stockées très longtemps, voire de manière permanente ; la solution record par exemple.
- La nature de la mémoire donne des informations générales sur l'élément mémorisé. Cet élément est-il dans l'espace des représentations ou dans l'espace des solutions ? S'intéresse-t-il à une représentation ou une solution dans son intégralité, ou à des attributs ? Ce dernier point correspond à ce que les auteurs décrivent comme la mémoire explicite ou attributive.
- Enfin les auteurs découpent la structure de la mémoire selon quatre grandeurs : la fréquence, la récence, la qualité et l'influence. Les deux premières sont complémentaires et regroupent des informations relatives à des événements qui apparaissent souvent ou qui sont apparues dans un passé récent. Les données relatives à la qualité permettent de différencier les bonnes des mauvaises solutions. Les auteurs précisent par exemple que cette dimension permet de mettre en place des mécanismes d'apprentissage par renforcement, en favorisant les attributs des bonnes solutions et en pénalisant ceux des mauvaises solutions. Enfin l'influence correspond à l'impact que peut avoir une décision sur la suite de la recherche.

Le succès pour une métaheuristique efficace résidera en partie dans la définition et l'utilisation de ces mémoires.

2.5.3. GUIDER UNE METAHEURISTIQUE AVEC L'UTILISATION DE VOISINAGES

Mouvements ou voisinages

Dans la section 2.1.1, une définition des systèmes de voisinage est proposée qui est relativement consensuelle. Elle est par exemple présente dans (Savage, 1975) ou plus récemment (Blum & Roli, 2003). Elle permet d'introduire les notions de minima locaux sur lesquels reposent les méthodes de recherche locale.

Cette définition s'appuie clairement sur l'espace des solutions, alors que les métaheuristiques ou les techniques de recherche locale qui sont conçues explorent l'espace des représentations. A partir de maintenant, le terme de "mouvements"¹⁹ sera utilisé pour désigner des mécanismes qui permettent de passer d'une représentation à une autre (donc qui œuvrent dans l'espace des représentations). Le terme de "voisinage" sera quant à lui réservé pour l'espace des solutions.

Comparaison des mouvements de base appliqués au TSP

Des exemples de mouvements ont déjà été présentés dans la section 2.4.2. Dans l'espace des permutations, des mouvements simples (échange, insertion et inversion) sont décrits. Ils ont l'avantage d'être génériques, en ce sens qu'ils ne dépendent pas du problème, à condition que les solutions de ce problème soient encodées par une permutation.

La Figure 17 montre l'impact du choix du mouvement sur l'exploitation, mais aussi l'intérêt de l'utilisation des mouvements pour guider efficacement la recherche vers des solutions de bonne qualité. L'expérience porte sur l'étude de l'instance "CAR 54" du TSP. C'est un problème historique de 33 villes. Nous avons prélevé un échantillon de 1000 permutations (ce qui est très peu pour un espace de $33! \approx 8.7 \cdot 10^{36}$ permutations). Les résultats présentés sont moyennés sur 10 exécutions pour réduire les effets du hasard. Les courbes montrent l'évolution de la meilleure permutation rencontrée (sauvegarde du record) selon quatre modes opératoires. Ceux-ci sont décrits, du moins bon vers le meilleur.

- Le premier est une marche aléatoire. On prélève 1000 permutations aléatoirement et on garde la meilleure rencontrée. Cette mémoire persistante permet de mettre en place un facteur d'exploitation. Ce processus converge en probabilité vers un optimum global, mais la convergence est extrêmement lente comme en atteste la courbe. La meilleure permutation (ou solution) obtenue a un coût qui est de l'ordre du double de celui obtenu avec les autres modes opératoires.
- Le second mode opératoire consiste à générer une permutation en appliquant un mouvement d'échange à partir de la permutation record. La courbe montre clairement le bénéfice de rechercher des solutions à proximité de la solution record.
- Les deux modes opératoires suivants sont similaires mais en appliquant des mouvements d'insertion pour le troisième, d'inversion pour le quatrième. Les courbes obtenues ressemblent à la précédente, mais avec une différence significative quant à la meilleure permutation obtenue au terme des 1000 prélèvements (autour de 20000 avec le mouvement d'échange, de 18000 pour l'insertion et de 15000 pour l'inversion). Aussi, nous pouvons remarquer que le mouvement d'inversion permet d'obtenir une permutation de qualité 20000 après 250 itérations environ, soit trois fois plus rapidement que pour le mouvement d'échange. L'insertion est environ deux fois plus rapide que l'échange pour atteindre ce même palier.

¹⁹ Les termes mouvements (moves) ou mouvements de base (basic moves) sont régulièrement utilisés dans la littérature.

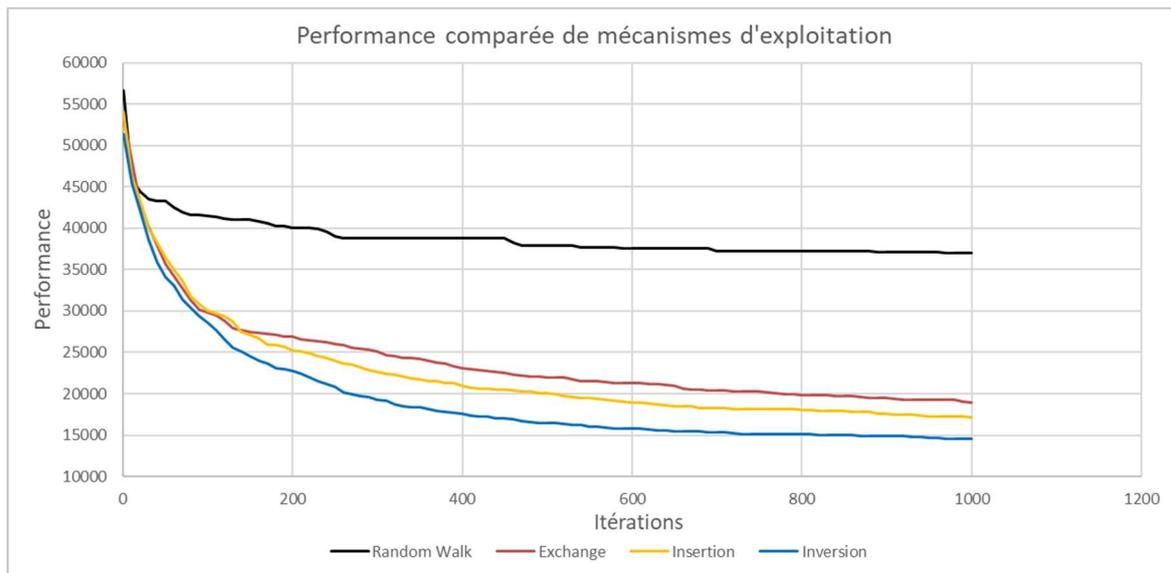


Figure 17. Influence du choix du mouvement pour l'exploitation.

Analyse des résultats, corrélation entre solutions voisines

Pour comprendre ces différences, il faut revenir à l'effet que produisent ces mouvements dans l'espace des solutions, et donc revenir au problème étudié. Le mouvement d'échange est un voisinage 4-opt particulier. Une solution diffère donc de sa solution voisine de quatre arêtes, soit de huit variables (x_{ij} qui passent de 1 à 0, et qui correspondent aux arêtes enlevées, et quatre variables x_{ij} qui passent de 0 à 1, et qui correspondent aux arêtes ajoutées). Cette différence est de trois arêtes (6 variables) pour les mouvements d'insertion et de deux arêtes (4 variables) pour les mouvements d'inversion, qui sont respectivement des voisinages 3-opt et 2-opt.

Cette petite expérience montre clairement que moins un voisinage est perturbateur (au sens du nombre de variables qu'il modifie), et plus il est efficace dans sa capacité à exploiter la recherche autour des bonnes solutions. En appliquant le mouvement d'inversion, la corrélation entre une solution et ses solutions voisines est plus forte qu'avec les autres mouvements, et *a fortiori* avec la marche aléatoire. Donc meilleure est la corrélation entre la structure et le coût d'une solution, et plus efficace sera l'exploitation. Le Tableau 2 donne l'estimation des coefficients de corrélation entre le coût de la solution associée à une permutation générée aléatoirement (série de valeurs X) et celui de la solution associée à une permutation obtenue aléatoirement en appliquant un des mouvements Random, Echange, Insertion ou Inversion (série de valeurs Y). Cette estimation est basée sur un échantillon de 1000 valeurs. Les résultats obtenus corroborent les propos tenus dans ce paragraphe. On vérifie bien que pour le mouvement random (la série Y est composée de permutations générées aléatoirement), le coefficient de corrélation est proche de 0, montrant aucun lien linéaire entre la structure et le coût). Pour les autres mouvements, moins le voisinage induit est perturbateur et plus le coefficient de corrélation obtenu est proche de 1 (dans l'ordre Echange, Insertion et Inversion). L'exploitation mise en œuvre a donc une qualité différente selon le mouvement utilisé et illustre une conséquence de la contraposée du théorème NFL (section 1.3). Un algorithme est plus performant s'il tire meilleure partie du lien qui existe entre la structure et le coût des solutions. Le coefficient de corrélation est calculé sur l'espace des solutions. Donc, lorsqu'une permutation encode une solution d'un autre problème (sac-à-dos, flow-shop, ...), les mouvements de base peuvent être classés dans un autre ordre d'efficacité.

Tableau 2. Estimation du coefficient de corrélation entre une série de solutions X générées aléatoirement et une série de solutions Y choisies aléatoirement dans le voisinage de X sur un échantillon de 1000 solutions du TSP.

Voisinage Ω	Random	Echange	Insertion	Inversion
$\rho(X,Y)$	-0,019	0,882	0,895	0,933

Analogies discret-continu

Pour terminer cette discussion, nous essayons de lier la notion de voisinage dans un espace topologique et la notion de voisinage au sens où il est défini dans l'espace discret des solutions. Un espace topologique est un couple (E, T) où E désigne un ensemble et T une topologie sur E . Une topologie T est un ensemble de parties de E définie par les axiomes suivants :

- i) $\emptyset \in T$ et $E \in T$
- ii) Si $(O_i)_{i \in I}$ est une famille quelconque d'éléments de T , alors $\bigcup_{i \in I} O_i \in T$
- iii) Si $(O_i)_{i \in I}$ est une famille finie d'éléments de T , alors $\bigcap_{i \in I} O_i \in T$

Les éléments de T sont appelés les ouverts de E . Les fermés de E sont construits par complémentation des ouverts.

Soit $a \in E$, on appelle voisinage de a toute partie de E qui inclut un ouvert contenant a . Intuitivement, le concept de voisinage signifie entourer, être à proximité du point a autant qu'on peut l'imaginer.

Les espaces topologiques permettent d'introduire de manière élégante les notions de continuité et de limites au point a . Soit $f : E \rightarrow F$ une application entre deux espaces topologiques E et F . f est continue en $a \in E$ si l'image réciproque de tout voisinage de $f(a)$ est un voisinage de a . Par exemple, dans la topologie usuelle de \mathbb{R} , cette définition devient :

$$f \text{ est continue en } a \Leftrightarrow \forall \varepsilon > 0, \exists \mu > 0, |x - a| < \mu \Rightarrow |f(x) - f(a)| < \varepsilon$$

Par suite, dans la topologie usuelle de \mathbb{R}^n , nous en déduisons les notions de dérivées partielles

$$\frac{\delta f}{\delta x_i} = \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_n)}{h},$$

puis de gradient $\bar{\nabla} f = \left(\frac{\delta f}{\delta x_1}, \dots, \frac{\delta f}{\delta x_n} \right)^t$ duquel découle l'algorithme de descente du gradient.

Dans un espace discret, on ne pourra jamais se rapprocher autant que nous le souhaitons d'une solution. Cependant, pour le TSP, on a vu que les mouvements d'inversion se traduisait par une différence pour quatre variables dans l'espace des solutions. Il n'est pas possible de trouver une distance plus petite qui sépare deux solutions distinctes. Ce voisinage contient donc toutes les solutions équidistantes de quatre variables d'une solution donnée x . Il peut être interprété comme contenant la plus petite boule ouverte (non réduite à x) dans l'espace des solutions.

Par ailleurs, soit Δf la différence de coût entre une solution et sa solution voisine (coût des arêtes ajoutées – coût des arêtes enlevées). Une direction de descente sera donc obtenue lorsque $\Delta f < 0$. En explorant l'ensemble du voisinage pour déterminer la solution qui donne la plus grande pente (i.e. qui minimise $\Delta f < 0$), on construit une méthode de recherche locale qui peut être perçue comme une version discrétisée de l'algorithme de descente du gradient.

Une première technique de guidage

Toujours dans le contexte du TSP et du mouvement d'inversion (voisinage 2-opt). Une solution de bonne qualité sera composée majoritairement d'arêtes associées à des coûts faibles. Si on choisit au hasard les deux arêtes à enlever, il y a de fortes chances que les deux arêtes qui devront être rajoutées pour reconstituer le tour seront de coût plus élevé, en particulier si les arêtes relient des villes éloignées géographiquement. Par exemple, la Figure 18 montre le cas où les arêtes enlevées sont (F,G) et (Z,C) . Pour reconstituer un tour, il faudrait rajouter les arêtes (F,C) et (G,Z) , qui sont clairement de coût beaucoup plus élevé.

Une idée pour accélérer serait d'éviter de considérer de manière aveugle des arêtes qui sont géographiquement éloignées l'une de l'autre, pour ne se concentrer que sur des mouvements prometteurs. Pour cela, nous pouvons focaliser sur les solutions dont au moins une des arêtes ajoutées est parmi une liste d'arêtes présélectionnées, par exemple en mémorisant une liste des villes les plus proches de chaque ville. Sur notre illustration, la liste des villes proches de F sont H, G, S, B, A et W . Ainsi, après avoir enlevé l'arête (F,G) , nous devons rajouter une arête incidente à F , dont l'autre extrémité sera parmi la liste précédente. Supposons que le choix se porte sur l'arête (F,A) . La deuxième arête à supprimer est l'arête (A,H) , pour casser le sous-cycle $A-F-H-A$ qui est apparu. Il suffit de reconstituer le tour en ajoutant l'arête (H,G) ²⁰.

Partant de la ville F , il est facile de tester les cinq autres possibilités. L'utilisation de cette liste restreinte de candidats permet donc d'explorer le voisinage d'une solution en un temps linéaire plutôt que quadratique, et donc d'accélérer la recherche locale. Le risque est bien entendu de manquer des améliorations possibles, et donc d'obtenir des minima locaux de moins bonne qualité.

(Reinelt, 1994) appelle cette stratégie Fast 2-opt. Il utilise la triangulation de Delaunay pour construire les listes de villes candidates.

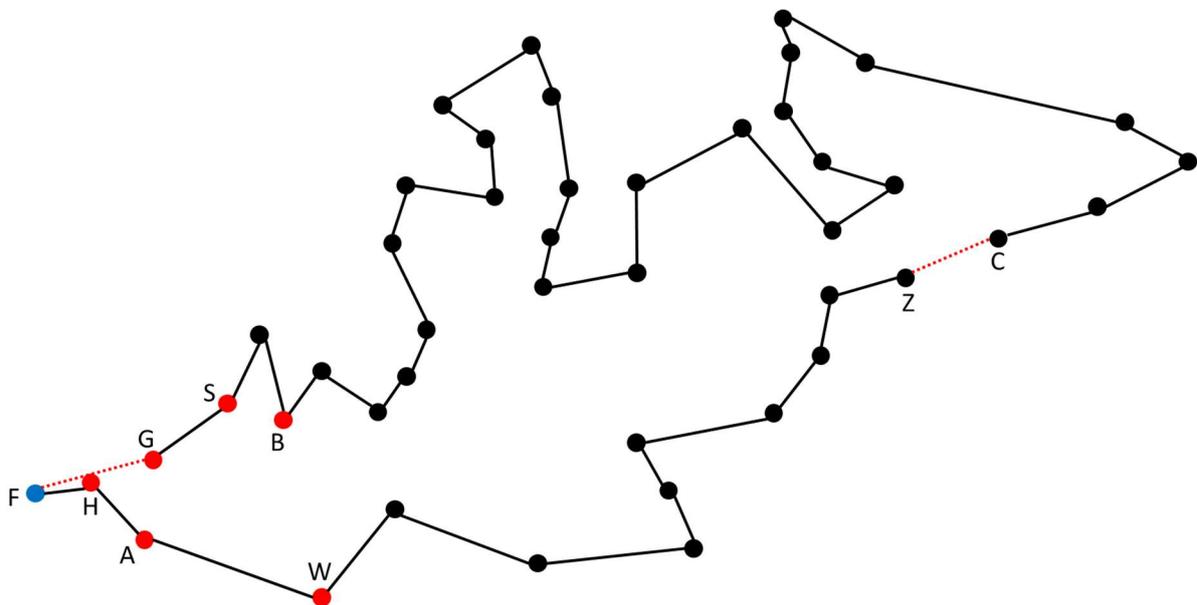


Figure 18. Illustration d'une technique de guidage pour le voisinage 2-opt.

²⁰ On pourra vérifier que ce mouvement est améliorant. On retrouve la séquence de villes (S,G,H,F,A,W) qui correspond à la solution optimale (voir Figure 5)

Reprenons l'exemple précédent. Au moment de rajouter l'arête (H, G) , si l'on constate que le tour obtenu est moins bon, on peut chercher à rajouter une autre arête, incidente à H , parmi les villes proches de H . Ce faisant, nous pourrions construire un mouvement 3-opt. La généralisation de ce processus conduit au mouvement LK.

Dans ce chapitre, nous avons proposé de dissocier l'espace des solutions, qui caractérise le problème d'optimisation, de l'espace des représentations dans lequel évoluent les métaheuristiques.

Nous avons vu plusieurs mécanismes sur lesquels les métaheuristiques peuvent s'appuyer pour guider la recherche dans l'espace des représentations. Ces mécanismes permettent d'exploiter et d'explorer l'espace de recherche, en utilisant des mémoires à court et à long terme.

Parfois, l'exploration est considérée à tort comme le complément de l'exploitation. Tout ce qui n'est pas relatif à de l'exploitation est lié à l'exploration. Pourtant lorsqu'on revient à la source de ces notions, on arrive à des définitions assez similaires. L'exploration et l'exploitation sont liées à des mécanismes dont le but profond est de rechercher des solutions de bonne qualité. L'exploitation recherche ces solutions à proximité de solutions élites, tandis que l'exploration les recherche dans des zones de l'espace encore inconnues.

Les solutions élites ne sont pas nécessairement les meilleures solutions rencontrées, mais les solutions autour desquelles on juge intéressant de faire de l'exploitation. Le rôle de l'exploration est plutôt de renouveler cet ensemble de solutions, dans une démarche prospective.

Les effets de l'exploration se font sentir sur le long terme. C'est peut-être une raison qui explique qu'elle est parfois mésestimée. Le rôle de l'exploitation est au contraire bien visible. Les chercheurs focalisent souvent dessus car une exploitation qualitative permettra d'obtenir de bons résultats rapidement, et sera souvent suffisante pour obtenir des résultats publiables.

Les concepts généraux que nous avons exposés datent d'une vingtaine d'années pour les plus récents, et n'ont que peu évolué depuis. Dans le chapitre suivant, nous aborderons les métaheuristiques en étudiant les progrès mais aussi les dérives du domaine.

3. METAHEURISTIQUES : AVANCEES RECENTES ET TENDANCES ACTUELLES

Même si le socle conceptuel sur lequel se sont construites les métaheuristiques est le même depuis longtemps, les métaheuristiques n'ont jamais cessé et continuent à être très utilisées. Elles restent parmi les approches les plus efficaces pour résoudre des problèmes d'optimisation difficiles. Il y a de nombreuses raisons à cela, qui sont maintenant bien connues. Parmi quelques-uns de leurs avantages, elles sont simples à mettre en œuvre, capables de traiter des instances de grande taille, efficaces et flexibles.

Nous allons structurer ce chapitre en commençant par nous intéresser d'une part à la proposition de nouveaux mécanismes proposés pour guider les métaheuristiques, d'autre part à l'utilisation de mécanismes existants, combinés avec d'autres mécanismes issus de domaines autres que les métaheuristiques.

Le premier point concerne plus précisément la proposition de nouvelles métaheuristiques. Nous allons voir que cet aspect est foisonnant de par le nombre de nouvelles métaheuristiques proposées chaque année. Pourtant, ces recherches entraînent de vives critiques dans la communauté. Nous lui consacrons la première section de ce chapitre.

En lien avec le deuxième point, les sections suivantes sont consacrées à différentes techniques d'hybridation. Les méthodes hybrides se sont fortement développées ces dernières années avec, parmi les formes les plus anciennes, l'hybridation entre les métaheuristiques classiques et la recherche locale. L'idée de départ est d'appliquer les schémas classiques des métaheuristiques (algorithmes génétiques, recuit simulé, algorithmes par essaim particulaire, colonies de fourmis, ...) sur le sous-espace restreint des minima locaux.

La section 3.2 sera consacrée à la recherche locale qui, vue comme une boîte noire, est une brique essentielle pour de nombreuses formes d'hybridation. Ces méthodes sont ensuite abordées dans la section 3.3.

Nota Bene : un certain nombre de références citées dans la section 3.1 sont indiquées en italique. Ces références figurent à la fin de ce chapitre, dans la section 3.4, et non dans la bibliographie générale (section 6).



Figure 19. Trouver le point commun entre ces images.

3.1. QUOI DE NEUF DU COTE DES METAHEURISTIQUES ?

Mais quel est donc le point commun entre des cuisiniers, le tiki-taka, le modèle de développement open source, les bernard-l'hermite et une fanfare (Figure 19) ?

Les métaheuristiques sont des méthodes qui permettent d'explorer et d'exploiter l'espace dans lequel elles évoluent. Outre leur efficacité, elles doivent en partie leur popularité du fait que beaucoup de métaheuristiques "historiques" sont inspirées de la nature (algorithmes génétiques, recuit simulé, colonies de fourmis, optimisation par essaim particulière, ...). L'idée était de simuler des phénomènes naturels afin d'en déduire des règles pour équilibrer l'exploitation et l'exploration.

D'autres se sont plutôt inspirées par la volonté de mettre en place des stratégies de recherche appropriées pour définir une métaheuristique. C'est par exemple le cas de la recherche taboue qui, pour s'extraire des minima locaux, interdit via une liste de mouvements tabous de revenir sur ses pas. D'autres stratégies sont mises en place dans VNS, VND (Variable Neighborhood Search ou Descent) ou GLS (Guided Local Search) où, lorsqu'on est bloqué dans un minimum local, on change de voisinage ou de fonction objectif. L'algorithme du kangourou (voir encadré), ILS ou GRASP ont pour principe d'enchaîner des recherches locales, en activant des mécanismes mémoriels. Ces mécanismes, qui servent à générer un nouveau point de départ entre deux recherches locales successives, ont des noms différents selon la métaheuristique. Si ce mécanisme est un peu différent pour GRASP (application d'une heuristique gloutonne randomisée), les autres métaheuristiques citées déploient un mécanisme similaire qui consiste à appliquer un ou une succession de mouvements à un minimum local. On parle ainsi de "sauts" lorsqu'on est dans le contexte de l'algorithme du Kangourou, de "perturbations" (perturbation) pour ILS, de "secousses" (shaking) pour VNS, ...

Chaque métaheuristique s'accompagne d'un contexte métaphorique, dans lequel s'inscrivent les mécanismes d'exploration et d'exploitation qu'elle orchestre. Ainsi, dans le contexte des algorithmes génétiques, on parlera de loci, d'allèles, de chromosomes, de croisements, de mutations, de fitness²¹ ... Le recuit simulé introduit les termes de température, de schéma de refroidissement (cooling schedule), d'états gelés (frozen states), ... Les algorithmes à base de fourmis mettent en place des mécanismes pour déposer ou faire évaporer des phéromones (substance chimique émise par un organisme et qui va interagir avec d'autres organismes de la même espèce).

L'algorithme du Kangourou (Kangaroo Algorithm ou KA) a été inventé par (Fleury, 1993). KA est défini par son concepteur comme un algorithme stochastique de type recuit simulé. C'est une succession de recuits simulés de température nulle (ou descentes stochastiques), séparés par des sauts (d'où le nom de cette métaheuristique). L'idée était, pour Fleury, de "*ne pas perdre l'information relative à la qualité des états proches de l'optimum*".

Comme le recuit simulé, c'est un algorithme : il existe un théorème de convergence en probabilité vers un optimum global.

L'algorithme du kangourou est une recherche locale itérée (Iterated Local Search ou ILS), dans laquelle la recherche locale est une descente stochastique et le critère d'acceptation est *Random Walk (RW)*.

Aujourd'hui encore, nous sommes quelques-uns à être des héritiers de cette métaheuristique. Si nous écrivons dans des articles avoir mis en œuvre ILS, nous disons entre nous avoir programmé un kangourou. La seule évolution est le critère d'acceptation. Nous utilisons plutôt *Better* ou *LSMC* (Lourenço *et al.*, 2002). On perd la convergence théorique, mais on gagne en performance (à l'instar d'ILS).

²¹ Fitness, (ou valeur adaptative ou sélective.) Dans la théorie de l'évolution, capacité d'un individu à produire des descendants mûres (viabiles et reproducteurs), relativement aux autres individus de la même population et au même moment

Les métaheuristiques que nous venons de mentionner sont peu ou prou les métaheuristiques historiques. Elles sont apparues dans les décennies 1980 et 1990. A partir de l'an 2000, et particulièrement avec le succès de PSO et ACO, fers de lance de l'intelligence en essaim, de nombreuses métaheuristiques ont été imaginées.

Dans son état de l'art, sur les nouvelles inspirations dans l'intelligence en essaim (*Parpinelli, 2011*) mentionne par exemple des métaheuristiques basées sur le comportement des abeilles, des bactéries *Escherichia Coli*, des lucioles, des vers luisants, des libellules, de la moisissure (amibes), des blattes, des cafards, des moustiques et des chauves-souris.

Plus récemment, (*Alorf, 2023*) recense quelques 62 nouvelles métaheuristiques, publiées dans les seules années 2020 et 2021 dans des revues ou des conférences. 23 d'entre elles sont des algorithmes évolutionnaires ou en essaim ; 4 sont fondées sur le monde des jeux ou du sport ; 19 sont issus de phénomènes physiques ou des mathématiques ; 10 sur des modèles d'interactions sociales et 6 sur d'autres phénomènes divers et inclassables.

Cette course à l'échalote a très rapidement attiré les critiques d'une partie de la communauté scientifique. L'innovation scientifique s'arrête souvent au simple cadre métaphorique, sans apport au domaine des métaheuristiques. Des mécanismes somme toute assez similaires se retrouvent noyés sous des dizaines de termes différents. Une question vient alors, quelle est la dernière innovation scientifique majeure dans le microcosme des métaheuristiques ?

Si l'on en juge par les avancées des vingt dernières années, la recherche de nouvelles métaheuristiques ne semble pas forcément constituer une piste de recherche prometteuse. Les métaheuristiques sont avant toute chose des concepts, prétendre qu'une métaheuristique est meilleure qu'une autre n'a pas réellement de sens. La recette pour une implémentation réussie s'articule autour de deux ingrédients principaux, que nous imagerons au travers des courses automobiles (Figure 20.). Ces ingrédients sont un concepteur expérimenté et une bonne connaissance du problème.



Figure 20. Dans le monde des métaheuristiques, ce qui compte, ce n'est pas seulement la carrosserie (à gauche), c'est aussi le pilote ! (l'image de droite est extraite d'une présentation de K. Sørensen).

L'expérience permet d'arrêter le choix d'une métaheuristique et d'optimiser son comportement par l'intermédiaire du réglage de ses paramètres. Cela correspond au réglage du bolide et à son pilotage. C'est ce qui va permettre de tirer le meilleur profit de la métaheuristique et de trouver le meilleur équilibre entre l'exploration et l'exploitation.

La "connaissance spécifique du problème" est un terme que l'on rencontre souvent, et qui regroupe la définition de mécanismes dédiés à un problème. Par exemple, si une permutation encode une solution du TSP, nous privilégierons l'utilisation de mouvements de type k -opt, alors que les mouvements de type k -insertion seront préférés pour des solutions du flow-shop. Ces mécanismes sont indépendants de la métaheuristique choisie et vont permettre de gagner en efficacité dans la recherche de solutions de bonne qualité et donc de "mieux exploiter" la

recherche. Cet ingrédient est représenté par la qualité du pilote, qui saura intégrer les éléments extérieurs (météo, connaissance du parcours, ...) pour adapter son pilotage.

L'âge d'or des métaheuristiques est situé entre les années 1980 et 2000. C'est au cours de cette période que sont apparues les métaheuristiques historiques. C'est également au cours de cette période que des recherches plus fondamentales, comme les questions relatives à la convergence des métaheuristiques vers l'ensemble des optima globaux, ont été menées. Sous certaines conditions, les algorithmes génétiques, le recuit simulé, le kangourou, la recherche taboue, ... convergent vers une solution optimale. Si ces résultats théoriques ont assez peu d'impacts en pratique, ils sont importants car ils permettent de légitimer leur utilisation, en particulier vis-à-vis des méthodes exactes.

Il y a aussi eu tout un travail pour construire un cadre formel aux métaheuristiques, pour définir les concepts d'exploration et d'exploitation ainsi que les moyens de les mettre en œuvre. Ce travail a permis d'asseoir toutes les connaissances acquises avant les années 80, où on était plus dans un schéma de résolution en deux phases avec l'utilisation d'heuristique de construction pour déterminer une solution initiale, suivie d'une heuristique d'amélioration itérative (ou de descente).

Depuis, les métaheuristiques continuent à jouer un rôle essentiel. Elles sont toujours autant utiles et utilisées pour résoudre des problèmes d'optimisation. Elles sont de plus en plus souvent combinées avec d'autres composantes issues du domaine des métaheuristiques ou d'autres domaines de Recherche Opérationnelle ou de l'Intelligence Artificielle (programmation mathématique, programmation par contraintes, techniques d'apprentissage, ...). Des termes comme *méthodes hybrides*, *matheuristiques*, *hyperheuristiques* sont apparues depuis les années 2000 et se sont progressivement imposés dans la littérature scientifique. Un rapide historique sur l'évolution des métaheuristiques est représenté sur une frise chronologique (Figure 21)

Les méthodes hybrides ont permis de traiter des problèmes d'optimisation deviennent de plus en plus compliqués. Cette difficulté provient de la taille des instances considérées, mais aussi et surtout de par le fait que les problèmes sont combinés plutôt que d'être étudiés séparément. Prenons l'exemple de la logistique, qui est un de nos domaines d'expertise. Il n'est pas utile d'optimiser la production, si l'approvisionnement et/ou la distribution ne sont pas coordonnés et engendrent pénuries ou blocages. Ce n'est plus un seul, mais plusieurs problèmes difficiles qui doivent être synchronisés. Pour contrer cette complexité grandissante, des techniques de guidage de plus en plus agressives ont dues être mises en œuvre, par exemple en décomposant le problème en sous-problèmes.



Figure 21. Les métaheuristiques au fil du temps.

L'analyse que nous présentons est pleinement en accord avec la vision de (Sörensen, 2015), qui conclut son pamphlet *The metaphor exposed*, en appelant les "métaheuristiciens" à investiguer du côté des méthodes hybrides à base de métaheuristiques, et d'éviter de tomber dans le piège des nouvelles métaheuristiques, source de discrédit pour la communauté.

Revenons à la question posée en préambule. Le lecteur attentif aura bien évidemment compris que chaque image de la Figure 19 symbolise une métaheuristique. Pour chaque métaheuristique, nous donnons de manière très synthétique le principe général, en donnant lorsque cela nous semble pertinent, une traduction des termes utilisés.

La fanfare avait pour unique but de mettre sur la voie. Elle fait référence à la recherche harmonique (*Geem et al, 2001*), qui est relativement bien connue (le 16/03/2023, google scholar recensait 6876 citations de l'article historique), et aussi une des premières à avoir attiré les critiques de (Sörensen, 2015) (citée en exemple car elle introduit un vocabulaire propre, mais finalement aucun concept nouveau). Le principe général de la recherche harmonique est de modifier aléatoirement une note (valeur d'une variable) pour improviser une nouvelle harmonie (solution). Les harmonies les plus musicales étant conservées d'une itération à l'autre, en espérant atteindre l'harmonie fantastique (optimum global).

Le tiki-taka, tactique bien connue des amoureux du ballon rond en général, et ceux du FC Barcelone de Johann Cruyff, puis de Pep Guardiola, en particulier, a également l'honneur de posséder sa métaheuristique. Les joueurs représentent des solutions du problème. A chaque itération, le joueur en possession du ballon fait la passe au joueur le plus proche, puis se déplace en tenant compte de la position du ballon et d'un joueur clé (choisi parmi un ensemble de joueurs clés – des solutions élites). Le monde du sport a d'ailleurs inspiré plusieurs autres métaheuristicques (*Alatas, 2019*).

(*Hajipour et al., 2012*) ont proposé de modéliser le processus de développement d'un logiciel open-source. Les logiciels (softwares) représentent les solutions. A chaque itération, la phase d'évolution consiste en trois tâches : sélectionner un logiciel leader (solutions élites), faire évoluer les logiciels leader et initialiser de nouveaux logiciels à partir des logiciels leaders.

Je possède sur mon disque dur un dossier appelé "*Métaheuristicques*", dans lequel je dois l'avouer se trouve un autre dossier appelé "*Bestiaire*". Il faut dire que peu d'espèces animales ont échappé à cette foire métaphorique. Par exemple, le Bernard-L'Hermite à la quête de la coquille de ses rêves n'a pas échappé à la règle (*Tafokkori and Tavakkoli-Moghaddam, 2022, Sharma et al., 2022*).

Les cuisiniers sont les seuls intrus puisque, à notre connaissance, le monde de la gastronomie a été pour le moment épargné. Leur présence est un clin d'œil à l'exemple introductif décrit dans (Sörensen, 2015), où l'auteur explique non sans ironie une théorie de physique corpusculaire à l'aide de la métaphore culinaire...

Nous allons maintenant consacrer la prochaine section à une des composantes clés des métaheuristicques, la recherche locale.

3.2. METHODES DE RECHERCHE LOCALE

La recherche locale (Local search ou *LS*) est un des mécanismes les plus simples pour mettre en œuvre une exploitation efficiente. Une méthode de recherche locale utilise un mouvement pour modifier une représentation afin d'obtenir une nouvelle représentation que nous espérons proche de l'initiale. Cette représentation est décodée pour obtenir une solution, puis évaluée pour déterminer son coût. Un bon mouvement permettra de guider la recherche dans des zones prometteuse en s'appuyant sur la corrélation entre la structure et le coût de deux solutions voisines.

Dans la suite de cette section, afin de ne pas alourdir inutilement la rédaction, nous nous placerons dans le cas d'un espace des représentations complet et nous assimilerons l'espace des représentations à celui des solutions.

3.2.1. CAS DES RECHERCHES LOCALES DETERMINISTES

Partons de la définition générale d'une recherche locale, telle qu'elle est donnée dans (Blum & Roli, 2003 – voir encadré). Nous retrouvons la phase de construction d'une solution initiale, et la phase d'amélioration qui est itérée dans la boucle "Répéter". A chaque itération, le voisinage de s est scruté. Si un voisin améliorant est trouvé, celui-ci devient la nouvelle solution de référence. Sinon l'algorithme s'arrête et un minimum local est obtenu.

Algorithme de principe de la recherche locale
 $s \leftarrow \text{GenerateInitialSolution}(\)$
 Répéter
 $s \leftarrow \text{Improve}(\mathcal{N}(s))$
 Jusqu'à ce qu'aucune amélioration soit possible

Il existe de nombreuses variantes de LS, selon la manière dont le voisinage est exploré et que le voisin améliorant est choisi.

$\mathcal{N}(s)$ peut être parcouru de manière déterministe, en énumérant les uns après les autres tous les voisins de s . Deux politiques sont généralement utilisées FI (*First Improvement*) ou BI (*Best Improvement*) (Blum & Roli, 2003): dans FI, le parcours du voisinage s'arrête dès qu'un voisin améliorant est trouvé. BI consiste à scruter tous les voisins, et à choisir parmi les voisins améliorants, le meilleur d'entre eux. Ces deux politiques conduisent à des descentes déterministes, à savoir la même solution initiale (sous réserve d'une règle déterministe pour départager d'éventuels *ex aequo*) conduira au même minimum local. LS est alors une boîte noire

qui associe à chaque solution s un minimum local s^* ($LS: \begin{matrix} S \rightarrow S^* \\ x \mapsto x^* \end{matrix}$). Les bassins d'attraction

des minima locaux forment alors une partition de l'espace des solutions (en particulier, $S = \bigcup_{s^* \in S^*} LS^{-1}(s^*)$).

3.2.2. CAS DES RECHERCHES LOCALES STOCHASTIQUES

En plus de FI et BI, d'autres politiques peuvent être imaginées. Parmi elles, RI (*Random Improvement*) qui consiste à choisir parmi tous les voisins améliorants, l'un d'entre eux aléatoirement. Si le choix du voisin est fait en suivant une loi de distribution uniforme, cette stratégie correspond à un algorithme de recherche locale stochastique appelé *Iterative Improvement* par (Hoos & Stützle, 2004). Partant de la même solution initiale, plusieurs minima locaux différents peuvent être obtenus en fonction des choix faits lors des transitions d'une solution vers sa solution voisine améliorante. Les bassins d'attraction forment alors un recouvrement de l'espace des solutions. Cette politique nécessite, à chaque itération, d'énumérer complètement l'ensemble des voisins de la solution de référence.

Pour éviter cet écueil, une solution consiste à piocher les solutions candidates au hasard parmi les solutions voisines (toujours avec une distribution de probabilité uniforme). Le fait de garder la mémoire des solutions voisines déjà évaluées permet de générer une permutation aléatoire Π des éléments de $\mathcal{N}(s)$. La politique RI correspond alors à la politique FI appliquée à Π (arrêt de l'exploration du voisinage dès qu'un voisin améliorant est détecté ; les voisins étant parcourus dans un ordre aléatoire). L'utilisation de cette mémoire est malheureusement difficile à mettre en place en pratique. En la désactivant, on obtient alors la descente stochastique (voir encadré ; celle qui est utilisée dans l'algorithme du kangourou, décrit en début de chapitre).

Sous cette forme, la descente stochastique est très simple à programmer. La difficulté est par contre reportée au niveau du choix de la condition d'arrêt. En piochant au hasard et avec remise

les solutions candidates dans le voisinage de s , si l'on considère X la variable aléatoire qui compte le nombre de tentatives avant de trouver une solution améliorante, alors X suit une loi géométrique de paramètre $p = \frac{|I(s)|}{|\mathcal{N}(s)|}$ avec $I(s)$ l'ensemble des solutions améliorantes. Si

$p > 0$ (ou de manière équivalente s n'est pas un minimum local), il convient de fixer un seuil maximal de tentatives A , au-delà duquel on considère être dans un minimum local et arrêter la recherche.

Si l'on souhaite que $P(X \leq A) \geq \rho$, où ρ correspond au seuil d'erreur souhaité (on décide d'arrêter la recherche alors qu'il existe des solutions améliorantes), une résolution simple nous conduit à $A \geq \frac{\ln(1-\rho)}{\ln(1-p)}$. En posant $|I(s)|=1$ (cas le plus défavorable) et en substituant

$\ln(1-p)$ par son équivalent $-p$ en 0, on arrive à $A \geq -\ln(1-\rho) \times |\mathcal{N}(s)|$.

Algorithme de principe de la descente stochastique

$s \leftarrow \text{GenerateInitialSolution}()$

Répéter

 Choisir $s' \in \mathcal{N}(s)$ aléatoirement

 Si $f(s') \leq f(s)$ Alors

$s \leftarrow s'$

 Fin Si

Jusqu'à ce qu'une condition d'arrêt soit obtenue

L'utilisation de ce palier nécessite de déclarer un compteur d'échecs successifs, initialisé à 0. Dans la boucle, ce compteur sera incrémenté en cas d'échec ($f(s') \leq f(s)$) et remis à 0 sinon.

L'obtention d'une solution voisine de même coût est considérée comme un échec.

Dans la pratique, le seuil d'erreur $\rho = 0.5$ est souvent recommandé (au moment où l'on décide d'arrêter, on a 50% de chances d'être effectivement dans un minimum local), ce qui correspond à un palier proche de $A \approx 0.7 \times |\mathcal{N}(s)|$.

Il est important de noter qu'à l'arrêt d'une descente stochastique, on ne peut pas prouver que la solution obtenue est un minimum local. Par abus de langage, nous continuerons à parler de minima locaux pour désigner les solutions obtenues par une recherche locale, même dans le cas d'une descente stochastique.

Dans de nombreux cas, plusieurs mouvements de base N_i peuvent être utilisés. Dans la métaheuristique VND (Variable Neighborhood Descent), chacun des mouvements est utilisé à tour de rôle, et dès qu'un minimum local est obtenu par rapport à un mouvement, celui-ci cède la place au mouvement suivant. Ce n'est que lorsque la solution obtenue est un minimum local par rapport à l'ensemble des mouvements considérés que l'algorithme s'arrête.

VND décrit en fait une règle de gestion des mouvements avec une politique déterministe. Dans le cadre d'une descente stochastique, on peut par exemple affecter une probabilité $p_i = \frac{|N_i|}{\sum_k |N_k|}$

d'utiliser le mouvement i . Ainsi, pour générer une solution voisine, on décide d'abord du mouvement selon cette probabilité. Chaque solution voisine a ainsi la même probabilité d'être considérée, indépendamment du mouvement dont elle provient (ce calcul s'appuie sur le fait

que les mouvements sont disjoints, ce qui n'est généralement pas le cas. Inverser ou échanger des éléments consécutifs dans une permutation conduit par exemple à la même permutation voisine).

A l'instar de (Lourenço *et al.*, 2003), nous adoptons la vision que les recherches locales sont une boîte noire, qui prennent en entrée la représentation d'une solution, et fournissent en sortie un minimum local. Sous cette forme, les recherches locales sont un composant dont la finalité est d'exploiter l'espace. Il est alors naturel d'utiliser la recherche locale pour la combiner avec des stratégies d'exploration, qui permettent de prospecter l'espace de recherche, et de déceler les zones prometteuses à exploiter. De telles stratégies conduisent aux méthodes hybrides.

3.3. METHODES HYBRIDES

Comme de nombreux concepts sur les métaheuristiques, les méthodes hybrides ne font pas l'objet d'une définition qui fait consensus. Nous proposons dans un premier temps de présenter des travaux généraux sur les techniques d'hybridation avant de focaliser sur deux classes d'hybridation que nous utilisons régulièrement : l'hybridation entre une métaheuristique et une recherche locale, et l'hybridation entre une métaheuristique et une méthode exacte.

3.3.1. DEFINITION ET CLASSIFICATIONS

D'après (Raidl *et al.*, 2019), un grand nombre de méthodes ne suivent plus strictement les concepts d'une unique métaheuristique traditionnelle, mais combinent des concepts algorithmiques provenant de différentes branches de l'optimisation. C'est ainsi qu'ils introduisent les métaheuristiques hybrides.

La motivation derrière l'hybridation est généralement d'obtenir des méthodes plus performantes qui exploitent et unifient les avantages des stratégies individuelles pures. Les auteurs précisent notamment le fait que, de nos jours, c'est la combinaison adéquate de concepts algorithmiques multiples qui est la clé pour obtenir des méthodes performantes.

(Talbi, 2002) propose une taxonomie des métaheuristiques hybrides. L'auteur décrit une classification bi-niveaux.

Le premier niveau concerne les hybridations haut-niveau versus bas-niveau. Dans une hybridation bas-niveau, une métaheuristique est encapsulée dans une autre métaheuristique, avec pour but de répondre à une fonction précise. Au contraire, dans une hybridation haut-niveau, les deux métaheuristiques fonctionnent de manière complètement autonome.

Le deuxième niveau est la collaboration en relai versus en équipe. Dans une collaboration en relai, les deux métaheuristiques se déroulent l'une à la suite de l'autre, les sorties de l'une étant utilisées comme entrées pour l'autre. Dans la collaboration en équipe, chaque métaheuristique travaille en parallèle dans l'espace de recherche de manière coopérative.

(Raidl, 2006) étend cette classification en précisant d'une part les différents types d'hybridation à base de métaheuristiques :

- Hybridation de métaheuristiques entre elles,
- Hybridation métaheuristique / algorithme dédié au problème (par exemple un modèle d'évaluation de performance)
- Hybridation métaheuristique / technique provenant d'un autre domaine de la Recherche Opérationnelle ou de l'Intelligence Artificielle

Ils rajoutent également la notion de stratégie de contrôle, qui peut prendre deux aspects : intégrée ou collaborative.

Dans les approches intégrées, une composante est encapsulée dans une autre. C'est par exemple le cas des méthodes hybrides de type métaheuristiques / recherche locale que nous développerons dans la section suivante.

Dans les approches collaboratives, les composantes échangent de l'information mais sans que l'une soit une partie de l'autre. Un cas important est la décomposition d'un problème de grande taille en sous-problèmes. Les composantes travaillent chacune sur un sous-espace bien défini. Nous aurons l'occasion de donner plusieurs exemples illustrant ce type d'hybridation dans le chapitre suivant.

3.3.2. UN CAS PARTICULIER IMPORTANT : L'HYBRIDATION METAHEURISTIQUES, RECHERCHE LOCALE

L'hybridation "métaheuristique / recherche locale" est certainement la forme d'hybridation la plus répandue depuis plus de 25 ans. L'idée est de combiner le pouvoir exploratoire des métaheuristiques (en particulier celles qui travaillent sur une population) avec la grande capacité des méthodes de recherche locale à exploiter l'espace de recherche.

De manière générale, une métaheuristique peut se résumer en une première phase pendant laquelle elle va générer une ou plusieurs représentations (par exemple un voisin de la représentation courante pour le recuit simulé ; des enfants *via* les opérateurs de sélection, croisement et mutation pour les algorithmes génétiques). Cette phase sera suivie d'une phase de consolidation durant laquelle elle va prendre des décisions selon la qualité des solutions obtenues et ses mécanismes internes (par exemple accepter ou refuser ce voisin pour le recuit simulé, préparer la population de la génération suivante pour les algorithmes génétiques).

Le principe général de l'hybridation "Métaheuristique / Recherche Locale" est illustré sur la Figure 22. A chaque fois qu'une nouvelle représentation x est générée, celle-ci est transmise comme représentation initiale à la recherche locale, qui détermine en retour un minimum local x^* . La métaheuristique prend les décisions à partir de x^* . Dans l'absolu, cette technique revient à appliquer le schéma général de la métaheuristique sur l'espace des minima locaux plutôt que sur l'espace de recherche tout entier. Les résultats obtenus sont généralement de bien meilleure qualité qu'en utilisant la métaheuristique seule.

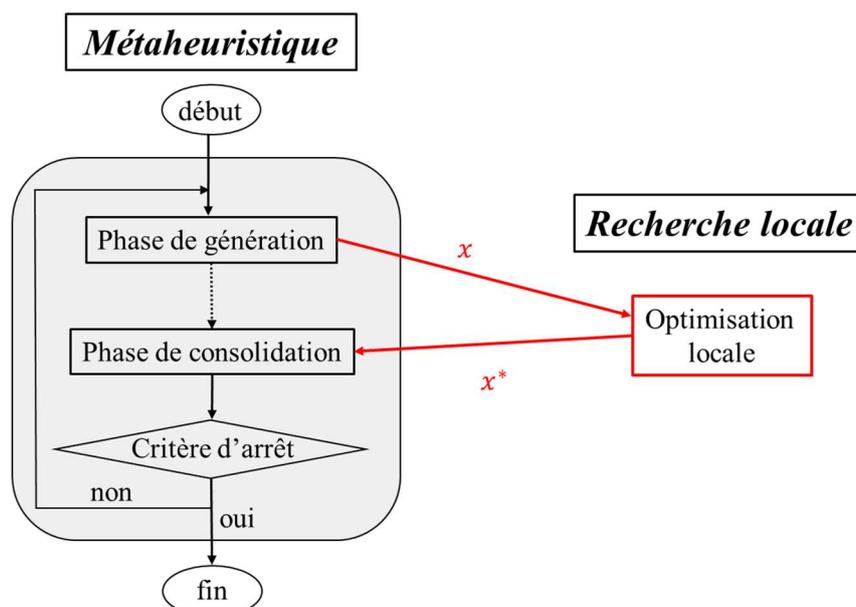


Figure 22. Schéma de principe de l'hybridation métaheuristique / Recherche locale.

Cette technique offre donc le double avantage d'être efficace et facile à mettre en œuvre. C'est ce qui explique son succès dès les années 1990s. Ainsi, l'hybridation Recuit Simulé / Recherche locale a été utilisée par (Martin et al., 1996). L'hybridation des algorithmes génétiques avec la recherche locale a été initiée par (Moscato, 1989). Elle a donné lieu à une classe d'algorithmes évolutionnaires qui s'appellent les algorithmes mémétiques (Memetic Algorithms). L'hybridation entre les algorithmes de colonies de fourmis et la recherche locale a été utilisée par (Dorigo & Gambardella, 1997) ou (Stützle & Hoos, 1997) pour le problème du voyageur de commerce. (Clerc, 2004) a imaginé un cadre général pour appliquer les Algorithmes par Essaim Particulaires (Discrete PSO ou DPSO) à des problèmes d'optimisation combinatoire. Nous avons utilisé ce cadre pour proposer une approche hybride (DPSO-LS) pour le problème du flow-shop de permutation (Deroussi *et al.*, 2006).

Cette forme d'hybridation est une des plus courantes et la littérature abonde en applications. Nous l'utilisons très régulièrement, avec une descente stochastique. Le réglage du palier est un moyen simple pour contrôler le taux d'exploitation que l'on souhaite incorporer dans la métaheuristique par l'hybridation.

La recherche locale peut être remplacée par un recuit simulé ou par n'importe quelle autre métaheuristique, donnant naissance à de nombreuses formes d'hybridation bas niveaux entre deux métaheuristicques.

3.3.3. HYBRIDATION ENTRE UNE METAHEURISTIQUE ET UNE METHODE EXACTE

Comme la précédente technique d'hybridation, celle qui relie une métaheuristique et une méthode exacte (programmation dynamique, méthodes de séparation-évaluation, ...) est particulièrement intéressante. Ces deux techniques sont des approches très différentes, mais sont reconnues pour leur efficacité à résoudre des problèmes d'optimisation. L'idée de concevoir des méthodes de résolution qui mixent les idées de ces deux domaines, est de combiner leurs avantages respectifs.

Les avantages des méthodes exactes sont :

- de trouver une solution optimale (en prouvant l'optimalité), lorsque c'est possible,
- de fournir un encadrement de la solution optimale, à défaut,
- de tronquer l'espace de recherche dans laquelle se situe une solution optimale.

Pour les métaheuristicques, les avantages sont :

- de pouvoir trouver des solutions de bonne qualité sur des instances de grande taille,
- d'évaluer un grand nombre de solutions en peu de temps,
- de pouvoir intégrer facilement de nouvelles variantes d'un problème et d'être flexibles,
- d'être basées sur un principe simple et d'être faciles à implémenter.

Dans un des premiers états de l'art qui leur est consacré, (Dimitrescu & Stützle, 2003) proposent de classer ces méthodes hybrides en cinq groupes.

- **Explorer des grands voisinages** : être capable d'explorer des voisinages de grande taille en guidant la recherche avec une méthode exacte. Par exemple, pour le TSP, enlever directement k arêtes de la solution courante et utiliser des méthodes exactes pour reconnecter les k blocs de manière optimale. Cette stratégie fait partie des méthodes de type LNS (Large Neighborhood Search) ou ALNS (Adaptive Large Neighborhood Search), qui intègrent des mécanismes de destruction et de réparation. Dans le cas présent, la réparation est confiée à des méthodes exactes.

- **L'exploitation de la structure de bonnes solutions** : ce type d'approche fonctionne en deux phases. Une première phase consiste à obtenir un ensemble de bonnes solutions, par exemple en appliquant une recherche locale itérée de type multi-start, et en conservant les minima locaux ainsi obtenus. La deuxième phase consiste à exploiter ce pool de solutions pour construire un sous-problème qui peut être résolu avec une méthode exacte. Elle est par exemple utilisée par (Applegate *et al.*, 1999) pour le TSP. Les auteurs expliquent qu'en procédant ainsi, ils ont réussi à retrouver une solution optimale pour l'instance "fl1577", à partir d'un échantillon de 25 solutions obtenues avec une heuristique de type Lin & Kernighan et dans lequel ne figurait aucune solution optimale. Cette instance est issue de la TSPLIB et possède 1577 villes. Le sous-problème construit consiste à rechercher une solution du voyageur de commerce dans un sous-graphe dont les sommets sont les 1577 villes, et dont les arêtes sont uniquement celles présentes dans une des 25 solutions de l'échantillon (en l'occurrence 2391 arêtes).

Malgré le mot "exploiter" qui est indiqué dans le titre de ce paragraphe, c'est plutôt un mécanisme d'exploration qui est mis en œuvre dans cette classe de méthodes hybrides. L'exploitation a été utilisée pour déterminer le pool de solutions élites. Ce pool sert à définir une stratégie d'exploration.

- **Utiliser des bornes inférieures dans des heuristiques gloutonnes** : les heuristiques gloutonnes construisent des solutions en partant d'une solution partielle vide, et en rajoutant progressivement des composants de la solution jusqu'à obtenir une solution complète. Les composants sont ajoutés généralement sur la base de la fonction objectif, en choisissant celui qui conduit à la meilleure évaluation, éventuellement de manière probabiliste.

Une autre approche peut être suivie, en calculant à chaque étape une borne inférieure et en basant le choix du composant à intégrer sur ce critère. Cela permet d'estimer, au moment du choix, l'impact des composants qui ne sont pas encore intégrés dans la solution partielle.

Dès lors que des algorithmes de calculs de bornes inférieures sont disponibles, cette approche peut être facilement intégrée dans des métaheuristiques telles que les systèmes de colonies de fourmis ou GRASP.

Cette classe de méthodes hybride illustre un cas d'utilisation d'une mémoire qui utiliserait comme structure la grandeur d'influence.

- **Orienter la recherche dans des zones de l'espace prometteuses** : ces techniques reposent sur des relaxations du problème linéaire qui peuvent être résolues de manière optimale. L'exemple qui sert d'illustration vient des travaux de (Vazquez & Hao, 2001) sur le problème du sac-à-dos multi-dimensionnel. Ce problème est identique au sac-à-dos (section 2.1.2), à ceci près que la contrainte de capacité $\sum_{i \in I} m_i \cdot x_i \leq C$ est généralisée par les contraintes

$$\sum_{i \in I} m_{ij} \times x_i \leq C_j \text{ où } j \text{ représente la dimension.}$$

Les auteurs proposent de résoudre le problème en relâchant les contraintes d'intégrité, et en rajoutant des contraintes additionnelles $\sum_{i \in I} x_i = k$ pour certaines valeurs de k judicieusement choisies. Les solutions optimales s_k^* ainsi obtenues servent à délimiter une zone de l'espace de recherche, contenant toutes les solutions situées à une distance de Hamming de s_k^* inférieure à un rayon fixé, et contenant exactement k objets (phase d'exploration).

Une recherche taboue est ensuite appliquée sur chacun des sous-espaces, en limitant sa recherche dans ce sous-espace (phase d'exploitation).

- **Améliorer les métaheuristiques** : La dernière catégorie d'hybridation est d'utiliser une méthode exacte comme composant d'une métaheuristique, afin d'améliorer son comportement. Par exemple, ce type d'hybridation peut être mis en œuvre au niveau de la perturbation d'ILS pour améliorer la qualité de l'exploration. Dans la solution qui doit être perturbée, un certain nombre de composants sont fixés et les autres sont laissés libres. Une méthode exacte est appliquée pour résoudre les parties laissées libres.

Pour preuve de l'intérêt suscité par l'hybridation "métaheuristique / méthode exacte", plusieurs auteurs ont proposé des synthèses (Puchinger & Raidl, 2005 ; Jourdan *et al.*, 2009). Le terme de mathheuristique s'est progressivement imposé et est maintenant largement utilisé.

Dans ce chapitre, deux techniques d'hybridation à base de métaheuristiques ont été plus particulièrement mises en avant. L'hybridation des métaheuristiques avec d'une part des techniques de recherches locales, et d'autre part des méthodes exactes. Des applications de ces techniques seront décrites dans le chapitre suivant, entièrement consacré à la résolution de cas concrets.

Pour autant, ce ne sont pas les seules hybridations possibles. (Talbi, 2013) reprend sa taxonomie et l'illustre par des exemples d'hybridation entre des métaheuristiques et de la programmation par contraintes, ainsi que du machine learning.

Afin d'aider les métaheuristiques à prélever un échantillon de solutions (*via* les représentations) avec efficacité, un large panel de composants sont disponibles, qu'ils proviennent du domaine de la Recherche Opérationnelle ou de l'IA.

Une partie importante de la communauté travaille depuis plus de vingt ans, sur des techniques de sélection automatique des composants, dans le but de résoudre des problèmes d'optimisation combinatoire. Le mot-clé d'hyper-heuristique²² caractérise ce domaine de recherche. Les lectures de (Burke *et al.*, 2013) et (Drake *et al.*, 2020) permettent d'avoir une première vision d'ensemble des travaux et résultats obtenus.

Intégrer la connaissance spécifique du problème est un facteur primordial pour concevoir une méthode de recherche efficace. Le développement des hyper-heuristiques, renforcé par les progrès des techniques d'intelligence artificielle, permettront peut-être, face à un nouveau problème, d'automatiser ce recueil de la connaissance et de l'intégrer pour générer des méthodes efficaces.

3.4. SOURCES BIBLIOGRAPHIQUES

Nous indiquons ici les sources bibliographiques qui nous ont permis d'illustrer les dérives liées à la proposition de nouvelles métaheuristiques.

Alatas, B. (2019). Sports inspired computational intelligence algorithms for global optimization. *Artificial Intelligence Review*, 52, 1579-1627.

Alorf, A. (2023). A survey of recently developed metaheuristics and their comparative analysis. *Engineering Applications of Artificial Intelligence*, 117, 105622.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60-68.

Hajipour, H., Rostami, H., BehzadiKhourmuji, H., & Oskouei, R. J. (2012, November). ODMA: A new metaheuristic optimization algorithm based on open source development model. In 2012 12th international conference on intelligent systems design and applications (ISDA) (pp. 758-763). IEEE.

Parpinelli, R. S., & Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation*, 3(1), 1-16.

²² Une hyper-heuristique est définie comme étant une méthode de recherche ou un mécanisme d'apprentissage pour sélectionner ou générer des heuristiques pour résoudre des problèmes de recherche (Burke *et al.*, 2013).

Ab. Rashid, M. F. F. (2021). Tiki-taka algorithm: a novel metaheuristic inspired by football playing style. *Engineering Computations*, 38(1), 313-343.

Sharma, A., Sharma, N., & Sharma, H. (2022). Hermit crab shell exchange algorithm: a new metaheuristic. *Evolutionary Intelligence*, 1-27.

Tafakkori, K., & Tavakkoli-Moghaddam, R. (2022). Hermit crab optimizer (HCO): a novel meta-heuristic algorithm. *IFAC-PapersOnLine*, 55(10), 702-707.

4. EXEMPLES D'APPLICATION DES METAHEURISTIQUES

Les chapitres précédents ont permis de poser les principaux concepts des métaheuristiques, et d'avancer sur le chemin de la compréhension de leur comportement. Après ces considérations théoriques, nous allons maintenant mettre le focus sur quelques travaux que nous avons menés avec, pour chaque exemple, un retour sur expérience à travers le prisme des métaheuristiques que nous avons essayé d'échafauder.

Le premier problème mis en avant est le flow shop de permutation. A l'instar du TSP, c'est l'un des problèmes académiques les plus simples et aussi les plus étudiés. En mettant en œuvre des techniques élaborées, nous avons conçu un mouvement très efficace qui a permis d'obtenir des résultats probants. Nous parlerons ensuite des travaux réalisés sur les systèmes flexibles de production. Ceux-ci ont été initiés pendant ma thèse, et se sont poursuivis pendant quelques années. En dehors des résultats obtenus, ils ont aussi la vertu de nous donner une expertise sur la gestion d'une flotte de véhicules automatiquement guidés, qui pourrait être très utile dans les projets à venir où les enjeux de mobilité et de flexibilité sont très importants. Le troisième point que je souhaite développer concerne les approches par décomposition, qui permettent de contrer la difficulté d'un problème en hiérarchisant les décisions pour permettre de mieux nous guider vers des bonnes solutions. Ce point se décline par la présentation de deux problèmes très différents de par leur domaine d'application et les enjeux qu'ils représentent, et pourtant très similaires dans leur structure et dans la manière dont nous les avons décomposés et résolus. Le focus sur ces quelques problèmes nous amènera à conclure en présentant quelques projets de recherche sur lesquels nous commençons à travailler à court, moyen et long terme.

4.1. LE FLOW-SHOP DE PERMUTATION

4.1.1. PRESENTATION DU PROBLEME

Le problème du flow-shop de permutation (Permutation Flow-Shop Problem ou PFSP) fait partie des problèmes d'atelier, dont il est un des membres les plus étudiés.

Un ensemble $I = \{1, \dots, n\}$ de n pièces indépendantes doivent être exécutées sur un ensemble $J = \{1, \dots, m\}$ de m machines dans l'ordre d'indexation des machines.

Le temps d'exécution de la pièce i sur la machine j est noté p_{ij} .

L'objectif est de déterminer une permutation des pièces $\sigma = \{\sigma_1, \dots, \sigma_n\}$, de manière à minimiser un critère donné, par exemple le makespan (C_{max}). L'ordre des pièces est donné en amont de l'atelier, et est le même pour toutes les machines de l'atelier.

Si t_{ij} est la date de fin d'exécution de la pièce i sur la machine j , le makespan, pour une permutation σ donnée, peut être calculé de la manière suivante :

$$\begin{aligned}t_{\sigma_1 1} &= d_{\sigma_1 1} \\t_{\sigma_i 1} &= t_{\sigma_{i-1} 1} + d_{\sigma_i 1}, \text{ pour } i = 2, \dots, n \\t_{\sigma_i j} &= t_{\sigma_{i-1}, j-1} + d_{\sigma_i j}, \text{ pour } j = 2, \dots, m \\t_{\sigma_i j} &= \max(t_{\sigma_{i-1} j}, t_{\sigma_{i-1}, j-1}) + d_{\sigma_i j}, \text{ pour } i = 2, \dots, n \text{ et } j = 2, \dots, m \\C_{max} &= t_{\sigma_n m}\end{aligned}$$

Sous cette forme, l'ensemble des solutions est bijectif à l'ensemble des permutations. L'espace des représentations est direct et complet.

4.1.2. PROPOSITION DE MOUVEMENTS ELABORES

La motivation de ce travail est de reprendre et d'adapter les idées développées pour les mouvements de type LK, qui proviennent du monde du TSP (et que nous avons décrites en détail dans les chapitres précédents), pour le PFSP. Alors que les mouvements de type LK sont basés sur les mouvements de base "Inversion" pour guider un voisinage k -opt, les mouvements que nous proposons s'appuient sur les mouvements de base "Insertion". Par analogie, nous avons appelés ces mouvements k -I pour " k -Insertion".

Afin d'implémenter ces mouvements de manière efficace, nous avons dû dans un premier temps reprendre puis étendre les travaux de Taillard.

Ces mouvements ont ensuite été intégrés dans des métaheuristiques simples de type ILS. Les résultats obtenus ont permis de trouver de nouvelles bornes supérieures sur deux instances de la littérature. Ce travail est consultable à (Deroussi *et al.*, 2006) et est fourni dans les annexes du tome 2 du mémoire.

Extension de l'implémentation de Taillard

Nous avons d'abord proposé une extension des travaux de (Taillard, 1990). L'implémentation de Taillard (Algorithme 1) est une astuce algorithmique qui permet de déterminer efficacement la meilleure position d'insertion d'une pièce dans une séquence partielle de pièces. Pour rappel, $d_{i,j}$ représente la durée opératoire de la pièce i sur la machine j . L'exploration complète de tous les voisins d'une permutation donnée peut se faire en $O(n^2m)$ au lieu d'être en $O(n^3m)$ ($O(nm)$ pour le calcul d'un makespan, et $O(n^2)$ pour le nombre de permutations voisines avec un mouvement d'insertion).

Nous avons étendu cette implémentation pour la suppression d'un job quelconque dans une permutation (Algorithme 2). Nous évaluons les n makespans obtenus après avoir supprimé la $k^{\text{ème}}$ pièce de la permutation en $O(nm)$ au lieu de $O(n^2m)$.

Algorithme 1. L'implémentation de Taillard pour un mouvement d'insertion rapide.

La pièce k est insérée dans une séquence partielle de $k-1$ pièces (Taillard, 1990).

1. Déterminer les dates de fin d'exécution au plus tôt e_{ij} de la $i^{\text{ème}}$ pièce sur la $j^{\text{ème}}$ machine ; la date de début de la première pièce sur la première machine est 0.
 $e_{0j} = 0$; $e_{i0} = 0$; $e_{ij} = \max(e_{i,j-1}, e_{i-1,j}) + d_{ij}$ pour $i = 1, \dots, k-1$, $j = 1, \dots, m$
2. Déterminer la queue q_{ij} , i.e. c'est-à-dire la durée entre le début d'exécution de la $i^{\text{ème}}$ pièce sur la $j^{\text{ème}}$ machine et la fin des opérations.
 $q_{kj} = 0$; $q_{i,m+1} = 0$; $q_{ij} = \max(q_{i,j+1}, q_{i+1,j}) + d_{ij}$ pour $i = k-1, \dots, 1$, $j = m, \dots, 1$
3. Déterminer la date relative de fin d'exécution au plus tôt f_{ij} sur la $j^{\text{ème}}$ machine lorsque la pièce k est insérée en $i^{\text{ème}}$ position.
 $f_{i0} = 0$; $f_{ij} = \max(f_{i,j-1}, e_{i-1,j}) + d_{kj}$ pour $i = 1, \dots, k$, $j = 1, \dots, m$
4. La valeur du makespan partiel M_i quand la pièce k est insérée en $i^{\text{ème}}$ position est :

$$M_i = \max_{j=1, \dots, m} (f_{ij} + q_{ij})$$
 pour $i = 1, \dots, k$

Algorithme 2. L'implémentation de Taillard pour un mouvement de suppression rapide dans une séquence de k pièces.

1. Déterminer les dates de fin d'exécution au plus tôt e_{ij} de la $i^{\text{ème}}$ pièce sur la $j^{\text{ème}}$ machine ; la date de début de la première pièce sur la première machine est 0.
 $e_{0j} = 0$; $e_{i0} = 0$; $e_{ij} = \max(e_{i,j-1}, e_{i-1,j}) + d_{ij}$ pour $i = 1, \dots, k$, $j = 1, \dots, m$
2. Déterminer la queue q_{ij} , i.e. c'est-à-dire la durée entre le début d'exécution de la $i^{\text{ème}}$ pièce sur la $j^{\text{ème}}$ machine et la fin des opérations.
 $q_{kj} = 0$; $q_{i,m+1} = 0$; $q_{ij} = \max(q_{i,j+1}, q_{i+1,j}) + d_{ij}$ pour $i = k, \dots, 1$, $j = m, \dots, 1$
3. La valeur du makespan partiel M'_i après avoir supprimé la pièce en $i^{\text{ème}}$ position est :
 $M'_i = \max_{j=1, \dots, m} (e_{i-1,j} + q_{i+1,j})$ pour $i = 1, \dots, k$

Proposition de nouveaux mouvements élaborés pour le flow-shop de permutation

Nous avons ensuite proposé six nouveaux mouvements pour le PFSP. Parmi eux, nous allons décrire ici le mouvement k -I (k -insertion), qui s'est avéré le plus efficace.

Ce principe général du mouvement k -I est décrit dans l'Algorithme 3. La pièce i située en position p est enlevée (1) (le nombre de mouvements possibles est égale à n) et est déclarée taboue (2). Nous obtenons une permutation incomplète de $n-1$ pièces. A partir de cette permutation incomplète, un processus est itéré jusqu'à trouver une permutation meilleure que l'initiale (réussite) ou avoir fait un nombre d'essais jugé suffisant (échec).

Algorithme 3. Principe général du mouvement k -I

```

// k-I move build from the initial position p
found ← false
k ← 0
M ← makespan(σ)
i ← remove(σ, p) (1)
TabuJobs ← {i} (2)
While (not found) and (k < kmax) do
    best_insert(σ, i, {p_i ≠ p}) (algorithme 1) (3)
    M' ← makespan(σ)
    if M' < M then
        found ← true
    else
        k ← k + 1
        i, p ← best_remove(σ, p, {i ∉ TabuJobs}) (algorithme 2) (4)
        TabuJobs ← TabuJobs ∪ {i} (5)
    endif
end

```

Ce processus consiste à réinsérer la pièce i à la meilleure position possible (3), en imposant que cette position soit différente de celle d'où elle provient ($p_i \neq p$). Si la permutation ainsi obtenue est meilleure, le mouvement s'arrête. Dans le cas contraire, une nouvelle pièce non taboue est enlevée (nécessairement différentes des pièces préalablement choisie) pour retrouver une permutation incomplète (4). Cette nouvelle pièce i est rajoutée à la liste taboue (5).

Les procédures *Best_insert* (3) et *Best_remove* (4) utilisent respectivement l'Algorithme 1 et l'Algorithme 2. Ces procédures sont appelées au plus $kmax$ fois où $kmax$ est un paramètre.

La liste taboue a pour objectif d'interdire de choisir une même pièce plusieurs fois pendant la construction du mouvement, afin de prévenir des éventuels cycles dans le choix des pièces.

Afin de guider la construction de ce mouvement, la pièce qui est enlevée dans la procédure *Best_remove* est celle qui est considérée comme la "moins bien insérée" dans la permutation (parmi les pièces non taboues). Le critère retenu pour estimer le degré d'insertion d'une pièce dans une permutation est une mesure relative obtenue en divisant le gain de makespan obtenu en enlevant la pièce, par la durée totale de fabrication de la pièce. Ce critère donne de meilleurs résultats que le critère de gain absolu de makespan.

4.1.3. SYNTHÈSE DES RESULTATS

Le Tableau 3 synthétise les résultats obtenus sur les instances de Taillard de taille 50 pièces et 20 machines. Parmi cette classe de dix instances (dénommées TA051 à TA060), 5 sont encore ouvertes actuellement.

Dans ce tableau, nous indiquons le meilleur encadrement connu en Avril 2005 (date à laquelle ces travaux ont été initiés), les résultats obtenus avec le mouvement k -I (meilleur résultat obtenu sur 10 runs de 3600s) et le meilleur encadrement connu aujourd'hui, obtenu à partir des travaux de (Gmys *et al.*, 2020) et (Gmys, 2022), et disponibles sur internet (Gmys, 2021). Les encadrements sont donnés avec la meilleure borne inférieure connue (LB) et la meilleure borne supérieure connue (UB) aux dates indiquées. Pour ILS, nous indiquons le makespan obtenu, ainsi que l'écart relatif en pourcentage à la meilleure solution connue (colonne UB 2005)

Tableau 3. Synthèse des résultats obtenus sur des instances de Taillard difficiles

		(Avril 2005)		(Deroussi, 2006)		(Avril 2023*)	
Name		LB	UB	ILS		LB	UB
50 pièces, 20 machines	TA051	3771	3850	3863	0,34%	3771 (3784)	3846
	TA052	3668	3704	3708	0,11%	3699	3699
	TA053	3591	3640	3641	0,03%	3640	3640
	TA054	3635	3723	3724	0,03%	3635 (3649)	3719
	TA055	3553	3611	3610	-0,03%	3553 (3569)	3610
	TA056	3667	3681	3687	0,16%	3679	3679
	TA057	3672	3704	3711	0,19%	3704	3704
	TA058	3627	3691	3699	0,22%	3691	3691
	TA059	3645	3743	3747	0,11%	3645 (3657)	3741
	TA060	3696	3756	3767	0,29%	3696 (3704)	3755

* Entre parenthèses, les bornes inférieures réactualisées par Jan Gmys (communication privée), qui n'étaient pas encore mises à jour sur son site au moment du bouclage de ce mémoire.

Nous avons choisi ILS pour sa simplicité. Notre souhait était de concentrer l'étude autour de l'efficacité des mouvements proposés. Pour ces instances difficiles, nous avons obtenu d'excellents résultats. Pour l'instance TA054, nous avons été les premiers à trouver une solution de coût 3720 dans des tests complémentaires mettant en œuvre trois voisinages (incluant k -I) selon une stratégie VND. Depuis, une solution de coût 3719 a été trouvée. Pour l'instance TA055, nous avons obtenu une solution de coût 3610 (au lieu de 3611), qui est encore la

meilleure solution connue à ce jour (l'instance TA055 est toujours ouverte). Les résultats obtenus ont montré que, parmi tous ceux proposés, k -I était le mouvement le plus performant.

4.1.4. RETOUR SUR EXPERIENCE POUR LE FLOW-SHOP DE PERMUTATION

Espace des représentations

La présentation du flow-shop de permutation qui est donnée est tout à fait classique. L'espace des solutions est réduit aux permutations et l'algorithme de calcul du critère, qui permet d'évaluer une permutation quelconque, repose sur l'hypothèse de la construction d'un ordonnancement semi-actif.

Pour autant, cette description ne correspond pas à un problème d'optimisation tel que nous l'avons défini. Ni les variables ni les contraintes du problème ne sont explicitement données. L'Algorithme 4 montre un modèle mathématique du PFSP, sous la forme d'un programme linéaire mixte. Ce modèle fait intervenir les variables $x_{i,k}$ qui, avec les contraintes d'affectation ((2) et (3)), correspondent effectivement à une permutation. Nous retrouvons également les variables continues $I_{i,k}$ et $W_{i,k}$, qui servent à définir les règles du processus de fabrication et la fonction à minimiser. Cet espace de solutions correspond à un ordonnancement quelconque (plus nécessairement semi-actif) et est donc beaucoup plus vaste que celui induit par la description usuelle du problème. Avec cette définition, nous aurions dit que l'espace des représentations était direct (mais plus complet). Par contre le problème est intègre car tout ordonnancement optimal peut se ramener à un ordonnancement semi-actif optimal (en calant les activités à gauche). On peut donc légitimement focaliser la recherche dans l'espace des solutions semi-actifs, sans risquer de manquer l'optimum global.

Retour sur le mouvement k -I

Pour concevoir ce mouvement, nous avons mis en œuvre une mémoire sous la forme d'une liste taboue. La persistance est limitée à la construction du mouvement. Sa nature porte sur des attributs, en l'occurrence sur les pièces qui ont déjà été déplacées depuis le début du mouvement, et qu'on s'interdit de choisir à nouveau. Sa structure a pour dimension la récurrence puisqu'on interdit de choisir les pièces qu'on vient de déplacer.

Nous avons noté que les résultats obtenus étaient très bons jusqu'à cette taille d'instances (50x20). Les résultats obtenus pour les instances plus grandes (100x20) et (200x20) étaient en retrait. La performance de ILS- k I est respectivement de 0.46% et 0.70% pour ces classes. Nous pensons aujourd'hui que ce n'est pas la qualité du mouvement qui est en cause. Ses aptitudes permettent de réaliser une exploitation de très bonne qualité, quelle que soit la classe d'instances. Pour améliorer les résultats, la piste à privilégier serait plutôt d'améliorer la qualité de l'exploration, en étant plus agressif dans la recherche de nouvelles zones de l'espace à prospecter. Deux idées nous semblent intéressantes à creuser :

- en randomisant le mouvement k -I. La fonction *best_remove* peut être modifiée en *random_remove*, une nouvelle fonction dans laquelle la pièce supprimée serait choisie au hasard parmi celles non taboues. Pour ce faire, nous conseillons de privilégier les pièces les moins bien insérées, en utilisant par exemple les critères définis dans la section 2.4.1 (un choix déterministe de la position qui minimise ce critère, ou un choix randomisé avec par exemple le principe de la roulette),
- en mettant en place une véritable stratégie d'exploration, par exemple en utilisant un ensemble de permutations élites, ou en essayant d'identifier des sous-séquences de pièces qui se retrouvent fréquemment dans les très bonnes solutions.

Pour finir sur le problème du flow-shop, nous mentionnons que dans (Gmys *et al.*, 2020), le mouvement k -I est utilisé pour obtenir des bornes supérieures de bonne qualité.

Données :

$I = \{1, \dots, n\}$: ensemble des pièces

$J = \{1, \dots, m\}$: ensemble des machines

$p_{i,j}, \forall (i, j) \in I \times J$: temps de fabrication de la pièce i sur la machine j

Variables :

$x_{i,k} = \begin{cases} 1 & \text{si la pièce } i \text{ est située en position } k \\ 0 & \text{sinon} \end{cases}, \forall (i, k) \in I \times I$

$I_{j,k} \geq 0, \forall (j, k) \in J \times I$: période d'inactivité de la machine j entre les $k^{\text{ème}}$ et $k+1^{\text{ème}}$ pièces

$W_{k,j} \geq 0, \forall (k, j) \in I \times J$: temps d'attente de la $k^{\text{ème}}$ pièce dans le stock situé entre les machines j et $j+1$

Minimiser :

$$\sum_{j=1}^{m-1} \sum_{i=1}^n p_{i,j} x_{i,1} + \sum_{k=1}^{n-1} I_{m,k} \quad (1)$$

Sous les contraintes

$$\sum_{i \in I} x_{i,k} = 1, \forall k \in I \quad (2)$$

$$\sum_{k \in I} x_{i,k} = 1, \forall i \in I \quad (3)$$

$$W_{k,j} + \sum_{i \in I} p_{i,j+1} x_{i,k} + I_{j+1,k} = I_{j,k} + \sum_{i \in I} p_{i,j} x_{i,k+1} + W_{k+1,j} \\ \forall k \in \{1, \dots, n-1\}, \forall j \in \{1, \dots, m-1\} \quad (4)$$

4.2. LES SYSTEMES FLEXIBLES DE PRODUCTION

Les systèmes flexibles de production (Flexible Manufacturing Systems ou FMS) sont des systèmes entièrement automatisés, composés de plusieurs cellules de production (que nous appellerons plus simplement des stations). Ces stations sont reliées les unes aux autres par un réseau de transport sur lequel circule une flotte de véhicules autonomes (Automated Guided Vehicles ou AGV). Les AGVs transportent les pièces d'une station à l'autre.

Les FMS ont fait l'objet de nombreuses études en RO, pour résoudre des problèmes liés à des décisions stratégiques, tactiques ou opérationnels : Conception des sites (problèmes de layout), conception du réseau de transport et dimensionnement de la flotte de véhicules, ordonnancement hors-ligne (pas de gestion de trafic, le trajet pour aller d'un point A au point B est statique), ordonnancement en ligne (gestion du trafic, le trajet pour aller d'un point A au point B est dynamique).

La difficulté de ces problèmes vient du fait qu'il faut synchroniser les ressources de transport (AGVs) et les ressources de production (stations). Les problèmes sont la combinaison de deux problèmes de classes NP : le Vehicle Scheduling Problem et un problème d'atelier de type job-shop ou job-shop flexible.

Dans les différents problèmes que nous avons étudiés, nous nous positionnons au niveau de l'ordonnancement hors-ligne. Dans un premier temps, nous présentons de manière plus détaillée les travaux effectués sur le problème du job-shop avec transport. L'article (Deroussi *et al.*, 2008), consacré à ce problème, est consultable dans les annexes. Deux extensions sont ensuite décrites de manière plus synthétique : le problème du job-shop flexible avec transport, dans

lequel une opération peut être exécutée sur un sous-ensemble de stations, et le problème de la reconfiguration de l'atelier dans lequel nous cherchons à relocaliser les stations à l'intérieur de l'atelier.

4.2.1. LE PROBLEME DU JOB-SHOP AVEC TRANSPORT

Description du problème

Dans le problème du job-shop (JSP pour Job Shop Problem), un ensemble de pièces $J = \{J_1, \dots, J_n\}$ doivent être usinées sur un ensemble de stations $M = \{M_0, M_1, \dots, M_m\}$. La station M_0 représente la station de chargement / déchargement des pièces et fait l'interface entre l'atelier et le monde extérieur. Les autres stations sont des postes dédiés à la production des pièces. Les stocks amont et aval sur les stations sont supposés de capacité infinie.

Chaque pièce i possède sa propre gamme opératoire $O_i = \{o_{ij}, j = 1, \dots, n_i\}$, qui consiste en une séquence d'opérations à exécuter dans l'ordre spécifié. Une opération o_{ij} est caractérisée par la station $\mu_{ij} \in \{M_1, \dots, M_m\}$ sur laquelle elle doit être exécutée et sa durée opératoire p_{ij} .

$O = \bigcup_{i=1}^n O_i$ désigne l'ensemble des opérations.

Une flotte $V = \{V_1, \dots, V_k\}$ de k véhicules (AGV) permet de déplacer les pièces d'une station à une autre. Les temps de déplacement pour aller d'une station à l'autre sont déterminés *a priori* en prenant le plus court chemin dans le graphe issu du réseau de transport. Ils ne dépendent pas de l'état du système (prise en compte du trafic : congestions, blocages, ...). $t(M_i, M_j)$ et $t'(M_i, M_j)$ désignent respectivement les temps de déplacement à charge et à vide pour aller de la station $M_i \in M$ à la station $M_j \in M$. Les réseaux de transport comprenant de nombreuses voies à sens unique, les matrices de déplacement ne sont généralement pas symétriques.

L'objectif est d'ordonnancer les opérations de production sur les stations ainsi que le déplacement des pièces dans l'atelier de manière à minimiser le makespan (date de fin de production de la dernière pièce) et la date de sortie de la dernière pièce de l'atelier. Dans les articles de la littérature antérieurs à nos travaux, seul le makespan était considéré. Nous avons proposé ce deuxième critère d'une part car il nous semblait aussi pertinent que le makespan, voire davantage (en ne considérant que le makespan, il n'est pas utile de se soucier de faire sortir les pièces de l'atelier), d'autre part car il suffit de compléter la gamme opératoire de chaque pièce i en rajoutant une opération fictive o_{i, n_i+1} avec $\mu_{i, n_i+1} = M_0$ et $p_{i, n_i+1} = 0$, pour intégrer la gestion de la sortie des pièces dans le problème. C'est donc le même problème, mais avec quelques opérations supplémentaires à traiter, et donc une combinatoire plus élevée.

(Bilge & Ulusoy, 1995) ont proposé une formalisation mathématique de ce problème sous la forme d'un programme non linéaire mixte. Ils décomposent leur modèle en deux parties : une représentant le problème du job-shop et l'autre le problème du Vehicle Scheduling Problem. Des contraintes d'encadrement lient entre elles les variables associées à chacune des parties.

L'espace S des solutions est donc défini à partir des variables de ce modèle.

Après avoir décrit ce problème nous allons maintenant synthétiser les travaux que nous avons publiés dans (Deroussi *et al.*, 2008)

Représentation et évaluation d'une solution

A chaque opération o_{ij} de production, nous avons associé une opération de transport \tilde{o}_{ij} , définie comme suit :

- Si o_{ij} est une opération de début de gamme ($j = 1$), la pièce correspondante devra être transportée de la station de chargement M_0 jusqu'à la station μ_{i1} .
- Sinon (pour $2 \leq j \leq n_i$), la pièce correspondante devra être transportée de la station μ_{ij-1} jusqu'à la station μ_{ij} pour y être déchargée.
- Pour l'étude du deuxième critère, une opération de transport supplémentaire est ajoutée pour chaque pièce, pour faire sortir la pièce de l'atelier. La pièce correspondante devra être transportée de la station μ_{in_i} jusqu'à la station de déchargement M_0 .

La Figure 23 illustre le diagramme de Gantt obtenu sur un petit exemple avec deux pièces P_1 et P_2 . Les gammes opératoires sont composées des opérations de production $o_{1,1} = (M_2, 4)$, $o_{1,2} = (M_1, 8)$ et $o_{1,3} = (M_0, 0)$ pour la pièce P_1 , et $o_{2,1} = (M_1, 7)$, $o_{2,2} = (M_2, 6)$ et $o_{2,3} = (M_0, 0)$ pour la pièce P_2 . A l'opération de production $o_{i,j}$ correspond l'opération de transport $\tilde{o}_{i,j}$.

La représentation d'une solution est la liste des opérations de transport affectées à chaque AGV. La représentation illustrée sur le diagramme de Gantt de la Figure 23 est $(\tilde{o}_{1,1}, \tilde{o}_{2,2}, \tilde{o}_{1,3})$ pour l'AGV1 et $(\tilde{o}_{2,1}, \tilde{o}_{1,2}, \tilde{o}_{2,3})$ pour l'AGV2. La matrice des temps de déplacement est indiquée sur la figure.

Le diagramme de Gantt montre le déroulé des opérations de production (avec l'opération et entre parenthèses la durée correspondante) et de transport (avec le déplacement effectué et entre parenthèses l'opération de transport correspondante). Les déplacements à charge sont indiqués en couleurs, contrairement aux déplacements à vide. Les périodes d'inactivité des ressources (de production ou de transport) sont indiquées en noir. Les AGVs sont situés à la station de chargement (M_0) au début de la simulation. Les pièces quittent toutes les deux l'atelier à la date 30. Le makespan est égal à 25.

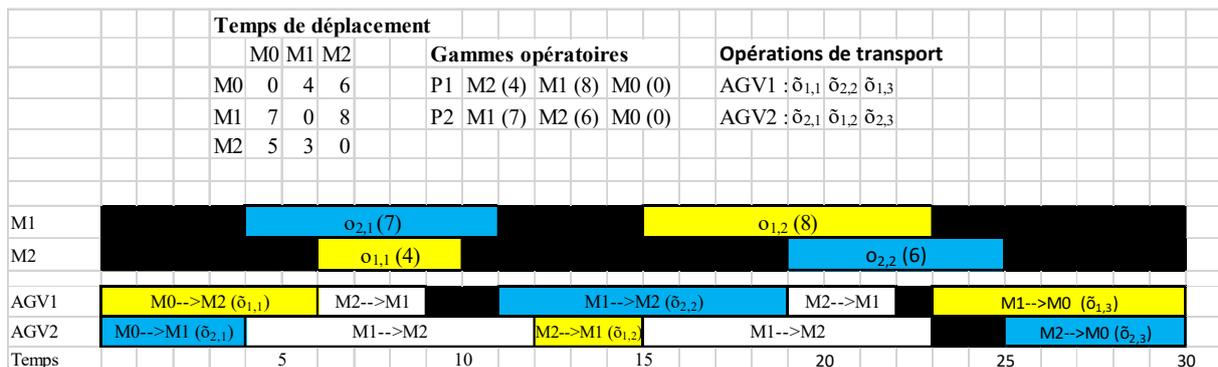


Figure 23. Exemple de représentation avec 2 véhicules, 2 pièces et 6 opérations de transport

A partir d'une représentation donnée, on reconstitue l'ordonnancement des pièces sur les stations en faisant l'hypothèse suivante : sur une station donnée, les pièces seront traitées selon une gestion FIFO, en fonction de leur date d'arrivée dans le stock amont de la station. On est alors en mesure d'évaluer cette représentation, quel que soit le critère retenu.

Cette représentation est directe, car la séquence sur les véhicules permet de fixer un certain nombre de variables. C'est la reconstitution du diagramme de Gantt qui permettra de déterminer les valeurs des autres variables. La représentation est donc incomplète. Ce problème n'est pas intègre. Le fait de faire appel à la règle de gestion FIFO dans l'évaluation de la représentation ne permet pas de garantir l'optimalité.

Mouvements et méthode de résolution

Nous avons proposé deux types de mouvements : échanger deux opérations de transport, et insérer une opération de transport après une autre opération de transport. Ces deux mouvements sont complémentaires car le premier conserve le nombre d'opérations affectées à chaque véhicule, tandis que le deuxième est susceptible de le modifier. Pour éviter de considérer des voisins inutilement, nous utilisons une liste restreinte de candidats qui est gérée dynamiquement. Une opération de transport \tilde{o}_{ij} ne pourra être échangée ou insérée qu'avec une opération de transport dont la date de réalisation est comprise dans une fenêtre de temps déterminée par la réalisation des opérations précédentes \tilde{o}_{ij-1} et suivantes \tilde{o}_{ij+1} , lorsqu'elles existent.

Une heuristique gloutonne randomisée, et les métaheuristiques ILS (Iterated Local Search), SA (Simulated Annealing) et leur hybridation (SALS) sont ensuite mise en œuvre pour ce problème.

Synthèse des résultats et analyse

Les résultats obtenus sont indiqués dans le Tableau 4. Le jeu d'essai est composé à partir de 10 jeux de pièces (jobsets) et 4 configurations d'atelier (layouts). Le numéro de l'instance est obtenu en concaténant le préfixe Ex avec l'indice du jobset et l'indice du layout. Les colonnes B&U reprennent le meilleur résultat proposé dans (Bilge & Ulusoy, 95) et (Ulusoy *et al.*, 1997). Les colonnes GAA présentent les résultats publiés dans (Abdelmaguid *et al.*, 2004) avec un algorithme génétique. Pour chaque méthode sont donnés le makespan et l'écart relatif à la meilleure solution connue.

La comparaison avec les méthodes de la littérature a montré la supériorité de l'approche proposée. Sur les quarante instances du jeu d'essai, la meilleure solution connue est au minima retrouvée, et des nouvelles bornes supérieures sont trouvées pour 11 instances (représées par une astérisque).

Les instances sont de petite taille (entre 13 et 21 opérations). Grace au deuxième critère, nous avons pu proposer des résultats sur des instances un peu plus difficiles (entre 18 et 27 opérations).

Tableau 4. Résultats obtenus pour le problème du job-shop avec transport

Inst.	B&U		GAA		SALS		Inst.	B&U		GAA		SALS	
Ex11	96	0.00	96	0.00	96	0.00	Ex61	121	2.54	118	0.00	118	0.00
Ex12	82	0.00	82	0.00	82	0.00	Ex62	98	0.00	98	0.00	98	0.00
Ex13	84	0.00	84	0.00	84	0.00	Ex63	104	0.00	104	0.00	103*	-0.96
Ex14	103	0.00	103	0.00	103	0.00	Ex64	123	2.50	120	0.00	120	0.00
Ex21	104	1.96	102	0.00	100*	-1.96	Ex71	118	2.61	115	0.00	111*	-3.48
Ex22	76	0.00	76	0.00	76	0.00	Ex72	85	7.59	79	0.00	79	0.00
Ex23	86	0.00	86	0.00	86	0.00	Ex73	88	2.33	86	0.00	83*	-3.49
Ex24	113	4.63	108	0.00	108	0.00	Ex74	128	0.79	127	0.00	126*	-0.79
Ex31	105	6.06	99	0.00	99	0.00	Ex81	161	0.00	161	0.00	161	0.00
Ex32	85	0.00	85	0.00	85	0.00	Ex82	151	0.00	151	0.00	151	0.00
Ex33	86	0.00	86	0.00	86	0.00	Ex83	153	0.00	153	0.00	153	0.00
Ex34	113	1.80	111	0.00	111	0.00	Ex84	163	0.00	163	0.00	163	0.00
Ex41	116	3.57	112	0.00	112	0.00	Ex91	117	0.00	118	0.85	116*	-0.85
Ex42	88	0.00	88	0.00	87*	-1.14	Ex92	102	0.00	104	1.96	102	0.00
Ex43	91	2.25	89	0.00	89	0.00	Ex93	105	0.00	106	0.95	105	0.00
Ex44	126	0.00	126	0.00	121*	-3.97	Ex94	123	0.82	122	0.00	120*	-1.64
Ex51	87	0.00	87	0.00	87	0.00	Ex101	150	2.04	147	0.00	147	0.00
Ex52	69	0.00	69	0.00	69	0.00	Ex102	137	0.74	136	0.00	135*	-0.74
Ex53	75	1.35	74	0.00	74	0.00	Ex103	143	1.42	141	0.00	138*	-2.13
Ex54	97	1.04	96	0.00	96	0.00	Ex104	164	3.14	159	0.00	159	0.00

4.2.2. LE PROBLEME DU JOB-SHOP FLEXIBLE AVEC TRANSPORT

Ce problème est similaire au précédent, à ceci près que pour chaque opération o_{ij} , est associé maintenant un sous-ensemble de stations $\Lambda_{ij} \subset M$ susceptibles de la traiter, avec un temps de traitement qui peut dépendre de la station.

Nous avons fait une simple adaptation de nos travaux sur le job-shop avec transport. Les véhicules, lorsqu'ils effectuent leur opération de transport, n'ont plus maintenant une destination connue μ_{ij} , mais un ensemble de destinations potentielles. Pour chaque station de Λ_{ij} , nous sommes en mesure de connaître, lorsque nous prendrons la pièce en charge, la date à laquelle cette pièce terminera son usinage (ce calcul prend en considération le temps de transport de la station courante vers la station potentielle, et la date à laquelle cette station potentielle sera en mesure de commencer l'exécution de l'opération). Nous choisissons de manière gloutonne, la station qui minimise cette date de fin d'exécution. Il est tout à fait possible qu'entre le moment où cette décision est prise, et le moment où la pièce arrive physiquement dans la station, d'autres pièces aient pu s'intercaler, retardant ainsi la date initialement prévue²³.

Ce problème n'avait jamais été traité auparavant. Pour tester cette approche, nous avons adapté les instances du job-shop avec transport en reprenant les 10 jobsets (ensemble de pièces) et en choisissant une des quatre configurations d'atelier (Figure 24, à gauche). Le nombre de stations a été doublé de manière à ce que deux stations soient en concurrence pour réaliser chaque opération (configuration à droite sur la figure). Les temps de production ont également été doublés pour conserver le ratio entre l'utilisation des ressources de production et des ressources de transport. Les résultats ont été présentés lors d'une conférence (Deroussi & Norre, 2010).

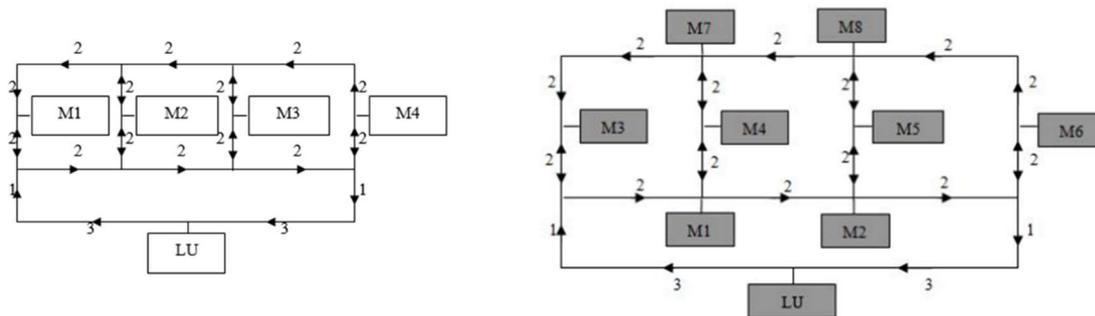


Figure 24. Proposition d'un jeu d'essai pour le job-shop flexible avec transport

Nous n'avons pas investigué davantage ce problème, mais la bibliothèque d'instances que nous avons proposée a été reprise par la suite, et utilisée dans la majorité des travaux portant sur le job-shop flexible avec transport comme le montre le Tableau 5. Dans ce tableau, nous indiquons dans la colonne S-J-0-V respectivement le nombre de stations, de jobs, d'opérations et de véhicules (AGVs). Les colonnes MILP et LAHC montrent les résultats obtenus par (Homayouni & Fontes, 2021) avec une méthode exacte et une heuristique (recherche locale itérée), ainsi que les résultats obtenus par (Zhang et al., 2012 avec une hybridation entre un algorithme génétique et une recherche taboue (GATS). Ces instances, à l'exception du FJSP7, sont aujourd'hui résolues. Nous ne présentons pas les résultats que nous avons obtenus car il y a des incohérences et des investigations seraient nécessaires pour en connaître la cause.

4.2.3. LE PROBLEME DE RECONFIGURATION D'UN ATELIER FLEXIBLE DE PRODUCTION

Dans ce problème, nous considérons que l'agencement de l'atelier reste la même. En particulier le réseau de guidage est inchangé, ainsi que les positions de chargement/déchargement des

²³ Les stocks amont sur les stations sont supposés de capacité infinie et gérés selon une politique FIFO.

pièces entre AGVs et stations. Le problème considéré est celui de la réaffectation des stations (Machine Layout Problem) dans cet atelier. En reprenant l'atelier de la Figure 24 (à gauche), les stations LU et M1 à M4 sont maintenant considérés comme des emplacements (L1 à L5) dans lesquels nous pouvons repositionner les différentes stations. L'objectif de ce problème est de pouvoir adapter la configuration de l'atelier aux évolutions de la demande (en conservant son agencement), de manière à optimiser les flux dans l'atelier, par exemple à des fréquences de décision trimestrielle ou semestrielle. Ces travaux sont présentés dans (Deroussi & Gourmand, 2014).

Tableau 5. Quelques résultats obtenus sur la bibliothèque d'instances proposée pour le job-shop flexible avec transport

Instances		MILP (Hom21)	LAHC (Hom21)	GATS (Zha12)
Name	S-J-O-V	Cmax*		
fjsp1	8-7-19-2	134	138	144
fjsp2	8-6-15-2	114	114	118
fjsp3	8-6-16-2	120	120	124
fjsp4	8-5-19-2	114	114	124
fjsp5	8-5-13-2	94	94	94
fjsp6	8-6-18-2	138	138	144
fjsp7	8-8-19-2	[?-110]	112	124
fjsp8	8-6-20-2	178	178	180
fjsp9	8-5-17-2	144	144	150
fjsp10	8-6-21-2	174	174	178

Classiquement, ce problème d'affectation des stations se modélise comme un problème d'affectation quadratique. Nous montrons dans l'Algorithme 5 le modèle simplifié pour les ateliers de type job shop. C'est un cas particulier du modèle que nous avons considéré dans nos travaux, qui intègre la notion de types de machines, et qui permet de prendre en compte des ateliers de type job-shop flexibles.

La résolution de ce problème permet d'obtenir une affectation optimale par rapport au flux des pièces dans l'atelier, c'est-à-dire en ne prenant en considération que les temps de déplacement à charge des véhicules.

Algorithme 5. Problème d'affectation quadratique

Minimise	$\sum_{o_{ij} \in O} \sum_{l \in M} \sum_{l' \in M} t_{ll'} \times x_{\mu_{i,j-1},l} \times x_{\mu_{ij},l'} \quad (1)$
Subject to	$\sum_{m \in M} x_{ml} = 1, l \in M \quad (2)$
	$\sum_{l \in M} x_{ml} = 1, m \in M \quad (3)$
Avec :	
M : ensemble des indices pour les machines / emplacements	
$t_{ll'}$: temps de déplacement de l'emplacement l vers l'emplacement l' .	
μ_{ij} : station sur laquelle l'opération o_{ij} doit être exécutée.	
x_{ml} : variable binaire qui vaut 1 si la station m est affectée à l'emplacement l .	
La fonction quadratique (1) à minimiser est la somme des temps de déplacement.	
Les contraintes (2) et (3) permettent d'affecter les stations sur les emplacements	

Nous souhaitons pouvoir intégrer les temps de déplacement à vide dans cette décision. Comme le soulignent certains auteurs dans le domaine des AGVs, il est important de réduire les temps de déplacement à vide, car ils entraînent des pertes de productivité, que ce soit au niveau des temps de production, ou pour la maintenance des véhicules. Pour faire cela, nous avons simplement réutilisé les méthodes précédentes développées pour le job-shop avec transport et pour le job-shop flexible avec transport. Ces méthodes sont utilisées comme une boîte noire, et permettent d'évaluer le makespan pour une reconfiguration d'atelier donnée.

Nous avons pu montrer que, même si les instances ne possèdent que cinq emplacements (soit une combinatoire de $5!=120$ reconfigurations d'atelier), la solution proposée par le modèle d'affectation quadratique (dénommée solution QAP) n'était pas nécessairement optimale lorsque les temps de déplacement des véhicules à vide sont intégrés dans le critère. Sur les 40 instances qui composent la bibliothèque standard du job-shop avec transport, la solution QAP est battue pour 17 instances, soit près d'une fois sur deux.

Le Tableau 6 détaille les résultats obtenus sur ces 17 instances. La colonne C_{best} indique le makespan obtenu pour la meilleure des 120 reconfigurations possibles (énumération exhaustive). Chacune des 120 configurations possibles est évaluée en utilisant la méthode mise au point pour le job-shop avec transport. L'évaluation étant heuristique, l'optimalité n'est pas garantie par cette énumération exhaustive. Les résultats indiqués dans le tableau sont l'écart relatif par rapport à solution reportée dans la colonne C_{best} . Une solution de coût 137 au lieu de 138 a été obtenue sur l'instance Ex74.

La colonne C_{QAP} donne le makespan correspondant à la solution QAP. Les cinq colonnes suivantes correspondent à différentes stratégies d'optimisation mises en œuvre pour optimiser la solution QAP. Dans les trois premières, la solution QAP est choisie comme solution initiale et est optimisée à l'aide d'une recherche locale basée sur des mouvements de type Echange (colonne EXC), Insertion (colonne (INS) ou Inversion (colonne 2-opt). Dans la colonne RND, des reconfigurations sont choisies de manière aléatoire puis évaluées. Enfin, dans la colonne ACS, le paradigme des colonies de fourmis est utilisé pour générer des reconfigurations randomisées. La solution QAP est utilisée de manière à guider les fourmis à choisir des affectations "proches" de celle proposée par QAP.

Tableau 6. Comparaison des stratégies d'optimisation de l'affectation proposée par QAP.

Instances	C_{best}	C_{QAP}	EXC	INS	2-opt	RND	ACS	Time (s)
Ex12	82	84	0,00	2,44	0,00	0,00	0,24	250
Ex13	91	92	1,10	0,00	1,10	0,00	0,11	250
Ex14	114	119	0,00	0,35	0,00	0,53	0,26	250
Ex21	106	116	3,96	0,19	3,21	0,75	0,00	430
Ex22	82	86	4,88	0,00	4,88	2,93	4,88	430
Ex24	118	120	1,69	1,69	1,69	1,86	1,69	430
Ex31	114	117	0,00	0,00	0,00	0,70	0,35	420
Ex44	140	142	1,43	1,43	1,43	1,00	0,00	420
Ex51	94	97	0,32	0,00	0,00	0,53	0,00	250
Ex52	73	74	1,37	1,37	1,37	0,96	1,10	250
Ex61	123	129	4,88	1,30	4,88	1,46	0,81	500
Ex64	132	135	2,27	0,23	2,27	0,53	0,83	500
Ex74*	138	141	3,77	2,46	2,75	2,46	-0,07	680
Ex92	97	99	2,06	2,06	2,06	0,82	0,00	330
Ex94	125	126	0,80	0,80	0,80	0,80	0,88	330
Ex101	148	153	2,70	0,95	2,70	1,22	1,49	570
Ex104	158	159	0,63	0,63	0,63	0,57	0,63	570
Gap		2,87	1,87	0,94	1,75	1,01	0,78	

* : une évaluation record a été obtenue par la politique dite ACS, qui n'avait pas été obtenue lors de l'évaluation exhaustive de toutes les configurations.

Toutes ces stratégies sont limitées à 1200 appels de la boîte noire pour évaluer une solution. Les temps de calcul en secondes sont indiqués dans la colonne "Time".

Ces résultats montrent que la politique dite ACS donne de bons résultats. Cette stratégie peut être vue comme un GRASP qui utilise ACS comme heuristique de construction randomisée, ou comme une méthode hybride ACS/LS. L'autre point intéressant est que la solution QAP fournit des informations pertinentes pour initialiser le paramétrage de ACS.

Avec les mouvements de base Echange, Insertion ou Inversion, la méthode réalise une recherche locale dans l'espace des 120 configurations, et peut se retrouver bloquer dans un minimum local. Cela peut expliquer que les résultats obtenus avec RND et ACS sont globalement de meilleure qualité.

4.2.4. RETOUR SUR EXPERIENCE POUR LES SYSTEMES FLEXIBLES DE PRODUCTION

Job shop avec transport

Jusqu'à nos travaux, ce problème avait été traité dans la littérature (depuis l'article historique de (Bilge & Ulusoy, 1995) et les suivants) comme un problème de job-shop classique en rajoutant des règles de gestion de type "premier véhicule disponible" pour gérer les véhicules. Autrement dit, la méthode de décomposition utilisée est similaire à la nôtre, mais en s'appuyant sur les ressources de production, plutôt que sur les ressources de transport. En observant les diagrammes de Gantt des solutions obtenues, on constate effectivement que pour les instances traitées, les ressources de transport sont largement critiques. Le fait de représenter les mouvements des véhicules est donc un avantage indéniable et explique *a posteriori* la qualité des résultats présentés dans (Deroussi *et al.*, 2008).

Cela signifie que sur d'autres classes d'instances, pour lesquelles ce seraient les ressources de production qui seraient critiques, notre méthode serait certainement moins performante. Il serait intéressant de pouvoir mettre en place une méthode qui serait capable, en apprenant de l'instance étudiée, de switcher d'un mode de représentation à l'autre (orientée transport ou production), et de rester performante quelle que soit l'instance étudiée.

En général, lorsque nous utilisons une approche par décomposition, c'est que le problème complet est trop complexe pour pouvoir le traiter de manière efficace dans son intégralité. La décomposition est en fait une technique qui permet de travailler dans un espace plus restreint, et de guider, via l'utilisation de règles, pour aller rapidement vers des zones intéressantes. C'est ce que nous constatons au travers de cette étude. Malgré le fait que l'optimum puisse être hors d'atteinte, la méthode proposée reste efficace.

Le choix judicieux d'une représentation est important, mais ne garantit pas à lui seul d'obtenir de bons résultats. Dans ce travail, les mouvements jugés peu intéressants ne sont pas considérés. Pour cela, une liste restreinte et adaptative de candidats est mise en place. L'idée est de déplacer une opération de transport à l'intérieur de la fenêtre de temps définie par l'opération qui la précède et celle qui la suit. De plus, lors de l'évaluation d'une représentation voisine, le diagramme de Gantt n'est recalculé qu'à partir du moment où une différence est constatée avec la représentation courante.

Job shop flexible avec transport

Concernant le job-shop flexible avec transport, nous avons pour ambition de construire un vrai jeu d'essai à partir du benchmark complet des instances du job-shop avec transport. Ces recherches n'ont finalement pas été poursuivies, mais la conception d'une librairie plus complète et avec des instances de plus grande taille est toujours pertinente.

Le choix de la machine qui va exécuter une opération est fait au moment où l'opération de transport commence. Ce choix est fait de manière gloutonne en tenant compte uniquement de la date de fin d'exécution de cette opération, c'est-à-dire un critère de production. D'autres critères comme les temps de déplacement à vide des véhicules auraient pu être utilisés.

Reconfiguration d'atelier flexible de production

La problématique de reconfiguration d'atelier est intéressante en ceci qu'elle fait appel à un type de décomposition différent. Ce problème se positionne au niveau tactique, en supposant que la configuration de l'atelier et du réseau de guidage sont données (niveau stratégique). L'idée est de pouvoir utiliser un certain degré de flexibilité pour pouvoir déplacer les stations d'une cellule de production à une autre. Cela permet notamment d'intégrer des évolutions au niveau des quantités de produits, ou des effets de saisonnalité, pour que le système reste le plus efficient possible. Nous avons montré que le fait de considérer le problème opérationnel (ordonnancement hors-ligne) permet d'affiner la décision au niveau tactique. Ce concept de synchronisation verticale a été discuté par David Lemoine dans ses travaux de thèse (Lemoine, 2008), pour relier la planification et l'ordonnancement. Au contraire, la synchronisation, dont nous avons déjà discuté, entre les ressources de transport et de production, concerne le même niveau décisionnel.

Nous resterons dans les problèmes de décomposition en abordant dans la section suivante deux problèmes, qui ont été résolus avec une approche originale.

4.3. LES APPROCHES PAR DECOMPOSITION D'UN PROBLEME

Les deux derniers problèmes abordés sont *a priori* très différents. Le premier est un problème de distribution de colis avec des véhicules classiques et des drones. Le deuxième est un problème d'équilibrage de lignes d'assemblage.

Nous les avons abordés de manière très similaire, en décomposant le problème global en deux sous-problèmes, et en résolvant le deuxième avec une procédure d'éclatement optimale (split). Les articles (Saleu *et al.*, 2018) et (Lahrichi *et al.*, 2021b), présentant les approches par split pour la distribution par drones et pour l'équilibrage de lignes reconfigurables, sont fournies en annexes.

4.3.1. PROBLEMES DE DISTRIBUTION PAR DRONES

La livraison de colis connaît une croissance d'environ 6% par an, hausse qui a encore été accentuée par l'effet COVID, avec un pic de 12% en 2020.

Livrer toujours davantage, raccourcir les délais pour satisfaire les exigences des clients, répondre aux enjeux environnementaux, telles sont les problématiques auxquelles doit faire face la logistique du dernier kilomètre.

Afin d'y répondre, des solutions innovantes sont régulièrement mises en place. La livraison par drone est une de ces solutions.

Le premier projet, Amazon PrimeAir, a été lancé en 2013, rapidement suivi par de nombreux autres projets. Le monde académique s'est mis au diapason, avec une littérature foisonnante depuis l'article historique de (Murray & Chu, 2015). Le grand mérite des auteurs a été de formaliser deux nouveaux problèmes de tournées mettant en jeu des livraisons avec des véhicules classiques et des drones.

- Dans le premier problème, le drone est lancé depuis un véhicule, alors que celui-ci effectue sa tournée, et est récupéré un peu plus loin. Il est nécessaire de synchroniser le véhicule et le drone.
- Pour le deuxième problème, les deux types de véhicules livrent leurs clients de manière indépendante, depuis le dépôt.

Nous nous sommes intéressés à cette problématique au travers de la thèse de Raissa Mbiadou Saleu, soutenue en 2021, et qui a porté essentiellement sur le deuxième problème (véhicules indépendants). Ces travaux ont fait l'objet de deux publications dans des revues (Saleu *et al.*, 2018, 2022). Le premier article traite de la résolution du problème avec un seul véhicule et

plusieurs drones. Le travail a été élargi au cas plusieurs véhicules dans le deuxième article. Nous indiquons ci-après le principe de résolution pour le cas avec un véhicule.

Présentation du problème

Ce problème utilise les notations suivantes.

K : ensemble de véhicules

M : ensemble de drones.

N : ensemble des n clients à livrer.

$N_d \subset N$: le sous-ensemble des clients pouvant être livrés par un drone.

$\{0\}$: le dépôt

A : ensemble des arcs ; $A = \{(i, j) : i, j \in N \cup \{0\}, i \neq j\}$

t_{ij} : temps de déplacement d'un véhicule pour aller de i vers j , $i, j \in N \cup \{0\}$

\hat{t}_i : temps mis par le drone pour faire l'aller-retour entre le dépôt et le client i , $i \in N_d$

Le problème consiste à construire les tournées de chaque véhicule de telle manière que chaque client soit livré. Dans cette étude, l'objectif est de minimiser la date de retour du dernier véhicule au dépôt. Ce problème s'appelle le Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP).

Approche de résolution

Les grandes étapes de la méthode de résolution sont résumées dans l'Algorithme 6.

- Le premier tour géant (étape 1) est construit avec l'heuristique Plus Proche Voisin. Il n'est pas nécessaire d'avoir un tour géant de très bonne qualité. Cet ordre va juste servir de base pour répartir les clients aux véhicules. Une post-optimisation de chaque tournée aura lieu à ce moment.
- Les principes de la phase d'éclatement du tour géant (Split – étape 2) sont détaillés ci-après. C'est le point fort de l'algorithme, qui permet d'"éclater" les clients en les répartissant de manière optimale entre le véhicule et les drones.
- La phase de post optimisation (étape 3) consiste d'une part à résoudre le problème du TSP pour optimiser la tournée du véhicule. Ceci est fait en utilisant l'heuristique LKH (Helsgaun, 2000). D'autre part, s'il y a plusieurs drones, à répartir les clients affectés aux drones (de manière globale) à chaque drone en particulier. Ce deuxième problème est connu sous le nom de Parallel Machine Scheduling (PMS), un problème d'ordonnancement de la classe NP. Ce problème est résolu de manière heuristique avec une règle de type LPT (Longest Processing Time).
- La réinitialisation (étape 4) consiste à réinsérer les clients affectés aux drones dans la tournée du véhicule avec une stratégie de type "Best Insertion" pour reconstituer un nouveau tour géant.

Algorithme 6. Principes généraux de l'approche de résolution

-
- Etape 1 : **(Initialisation)** Construire un tour géant (qui contient tous les clients)
 - Etape 2 : **(Split)** Appliquer la procédure Split sur le tour géant. Cette procédure détermine les clients affectés au véhicule et les clients affectés aux drones
 - Etape 3 : **(Post-optimisation)** Optimiser la tournée du véhicule (Traveling Salesman Problem) et l'affectation des drones (Parallel Machine Scheduling)
 - Etape 4 : **(Réinitialisation)** Construire un nouveau tour géant et revenir à l'étape 2.
-

Illustration de la procédure Split

Pour expliquer le fonctionnement de Split, nous allons introduire le graphe $G^\tau = (X^\tau, A^\tau)$, construit à partir du tour géant τ de la manière suivante :

- $X^\tau = \{0\} \cup A \cup \{n+1\}$. Le sommet $n+1$ est une copie du dépôt qui représentera le puits du graphe auxiliaire.
- A^τ est l'ensemble des arcs (i, j) définis par les deux conditions suivantes
 - $i = 0$ ou $j = n+1$ ou i est situé avant j dans le tour géant τ
 - Tous les clients situés entre i et j dans τ sont dans N_d

Emprunter l'arc $(i, j) \in A^\tau$ signifie que le véhicule va livrer le client j immédiatement après le client i . Tous les clients intermédiaires seront livrés par un drone. Les arcs $(i, j) \in A^\tau$ sont valués par le couple $\left(t_{ij}, \sum_{k=i+1}^{j-1} \hat{t}_k\right)$ représentant les coûts de livraison correspondant à l'arc (le véhicule pour l'arc (i, j) , les drones pour les clients intermédiaires). S'il n'y a aucun sommet entre i et j , la deuxième valeur est nulle.

Avec l'aide de la Figure 25, nous proposons d'illustrer la construction de ce graphe. Cet exemple comprend 5 clients dont 3 sont livrables par drone (les clients 1, 3 et 5). Le tour géant considéré est $\tau = (1, 2, 3, 4, 5)$.

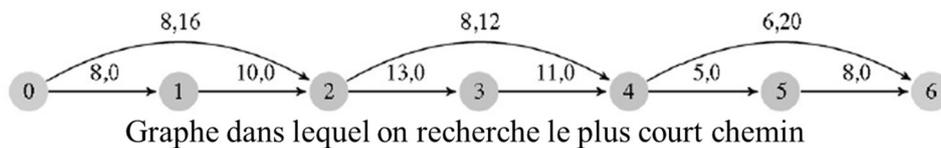
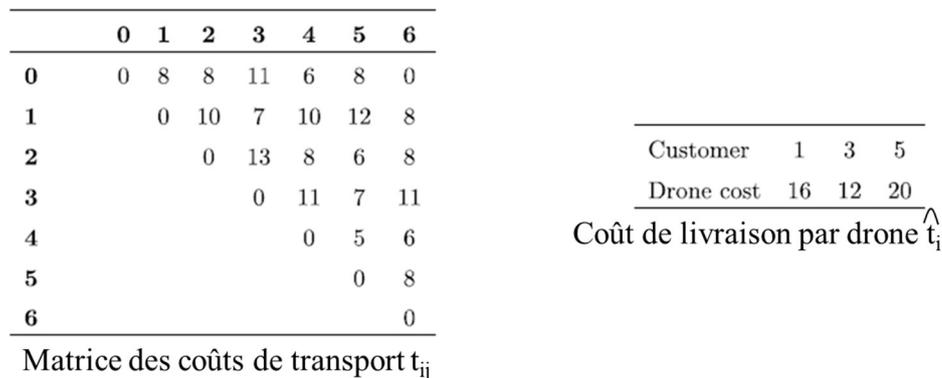


Figure 25. Illustration du split (1) : matrices des coûts et construction du graphe auxiliaire.

Une procédure de programmation dynamique est ensuite mise en œuvre, au cours de laquelle, en partant du sommet initial 0 auquel est affecté le label $(0, 0)$, on propage ce label sur tous ses successeurs en rajoutant le coût de l'arc emprunté selon les règles usuelles dans l'espace vectoriel \mathbb{R}^2 . On passe alors au sommet suivant dans le tour géant, et on recommence jusqu'à avoir propagé tous les labels au sommet final 6. Les labels barrés sont des labels dominés par d'autres labels du même sommet, au sens de Pareto. Par exemple, pour le sommet 4, le label $(32, 16)$ est dominé par $(26, 12)$. Les labels dominés n'ont aucune chance de se retrouver sur un chemin optimal. Il n'est donc pas utile de les propager.

Une fois obtenus tous les labels sur le sommet final, on repère le meilleur (celui qui minimise le max des deux valeurs) qui est représenté par un fond en couleur, et on reconstitue le chemin qui a permis d'obtenir ce label. Le chemin optimal est donc (0, 2, 4, 5, 6) qui induit la tournée affectée au véhicule. Les clients 1 et 3 sont livrés par le drone.

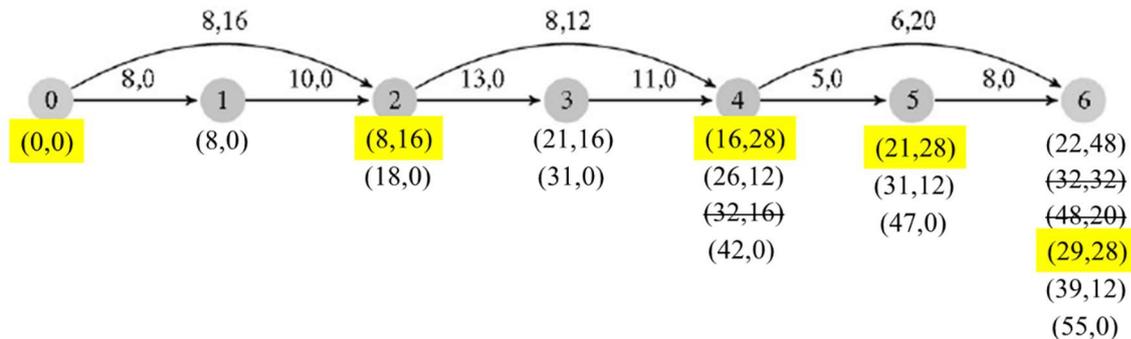


Figure 26. Illustration du split (2) : propagation des labels et reconstitution du plus court chemin.

Conclusion

Cette méthode s'est montrée très efficace. Des résultats probants ont été obtenus sur les instances de Murray & Chu (qui étaient de petite taille). En parallèle, nous avons proposé une nouvelle bibliothèque d'instances basée sur la TSPLIB²⁴, comprenant entre 48 et 229 clients. La méthode a été étendue pour passer d'un à plusieurs véhicules. Si le principe reste le même, la définition des labels ainsi que les règles de propagation sont nettement plus compliquées. Un label est maintenant un 4-uplet qui contient les informations suivantes :

- L'indice du véhicule courant
- Le coût de la plus longue tournée déjà construite
- Le coût de la tournée actuelle (en cours de construction)
- L'estimation du coût des tournées affectées aux drones

Que ce soit avec l'une ou l'autre des méthodes, le nombre de labels explose avec la taille du graphe auxiliaire. Lorsqu'il devient impossible de tous les générer, il est nécessaire de restreindre leur nombre et de ne travailler qu'avec les labels qui semblent les plus intéressants. La répartition des clients n'est alors plus optimale (pour le tour géant considéré). Le sous problème de clustering est résolu de manière heuristique.

4.3.2. PROBLEMES D'EQUILIBRAGE DE LIGNES D'ASSEMBLAGE

Le problème d'équilibrage de lignes d'assemblage simple (Simple Assembly Line Balancing Problem ou SALBP) est un problème d'atelier très étudié, qui consiste à équilibrer la charge des ressources de production (opérateurs ou machines) pour faire correspondre la cadence de production au temps de cycle.

Présentation du problème

Nous nous sommes intéressés, lors des travaux de thèse de Youssef Lahrichi, à l'étude d'une variante appelée le Sequence-Dependent SALBP (SDSALBP). Dans cette variante, des temps de réglage sont pris en compte qui dépendent de l'ordre dans lequel les opérations sont exécutées.

²⁴ Bibliothèques d'instances pour le problème du TSP, provenant de différentes sources et de différents types, créée par G. Reinelt et accessible à l'adresse <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>.

Les données du problème sont les suivantes :

- un ensemble $N = \{1, \dots, n\}$ de n opérations, reliées par un ensemble P de relations de précédence, doivent être réparties sur un ensemble $S = \{1, \dots, s_{\max}\}$ de stations,
- l'opération $i \in N$ requiert un temps d'exécution d_i ,
- si l'opération i est effectuée juste avant l'opération j dans une station, un temps de réglage égal à $t_{i,j}$ est requis entre l'exécution des deux opérations,
- un temps de cycle C indique la cadence de la ligne.

En plus de la répartition des opérations sur les stations, la prise en compte des temps de réglage impose de fixer le séquençement des opérations. Ce niveau de décision supplémentaire rajoute à la difficulté du problème.

Les contraintes à respecter sont :

- **les contraintes de précédence** : Etant données deux opérations i et j telles que $(i, j) \in P$ (on dira que i précède j ou $i < j$), alors ou bien i devra être placée sur une station située en amont de celle de j , ou bien i et j sont affectées à la même station, mais i est placée avant j dans la séquence des opérations,
- **les contraintes de temps de cycle** : à chaque station, la somme des temps d'exécution des opérations affectées, plus les temps de réglage induits par la séquence des opérations, doit être inférieure au temps de cycle.

L'objectif, dans le problème considéré, est de minimiser le nombre s_{\max} de stations (une autre variante consiste, étant donné le nombre de stations, à minimiser le temps de cycle)

La modélisation mathématique en PLNE

Le problème d'optimisation se modélise comme suit :

Les variables binaires sont :

- $x_{i,s,j} = \begin{cases} 1 & \text{si l'opération } i \text{ est affectée à la station } s \text{ en position } j \\ 0 & \text{sinon} \end{cases}$
- $y_s = \begin{cases} 1 & \text{si une opération au moins est affectée à la station } s \\ 0 & \text{sinon} \end{cases}$
- $z_{i,i',s} = \begin{cases} 1 & \text{si l'opération } i \text{ est affectée juste avant } i' \text{ sur la station } s \\ 0 & \text{sinon} \end{cases}$
- $w_{i,s} = \begin{cases} 1 & \text{si l'opération } i \text{ est affectée à la station } s \text{ en dernière position} \\ 0 & \text{sinon} \end{cases}$

La fonction à minimiser est :

$$\text{Min} \sum_{s=1}^{s_{\max}} y_s \quad (1)$$

Sous les contraintes

$$\sum_{s=1}^{s_{\max}} \sum_{j=1}^n x_{i,s,j} = 1, \forall i \in N \quad (2)$$

$$\sum_{i=1}^n x_{i,s,j} \leq 1, \forall s \in S, \forall j \in N \quad (3)$$

$$\sum_{i=1}^n x_{i,s,j+1} \leq \sum_{i=1}^n x_{i,s,j}, \forall s \in S, \forall j \in N \setminus \{n\} \quad (4)$$

$$y_{s+1} \leq y_s, \forall s \in S \setminus \{s_{\max}\} \quad (5)$$

$$\sum_{s=1}^{s_{\max}} \sum_{j=1}^n (n(s-1) + j) \times x_{i,s,j} \leq \sum_{s=1}^{s_{\max}} \sum_{j=1}^n (n(s-1) + j) \times x_{i',s,j}, \forall (i, i') \in P \quad (6)$$

$$\sum_{i=1}^n \sum_{j=1}^n d_i \times x_{i,s,j} + \sum_{i=1}^n \sum_{j=1}^n t_{i,j} \times z_{i,j,s} \leq C \times y_s, \forall s \in S \quad (7)$$

$$x_{i,s,j} + x_{i',s,j+1} \leq 1 + z_{i,i',s}, \forall (i, i') \in N^2, i \neq i', \forall j \in N \setminus \{n\}, \forall s \in S \quad (8)$$

$$x_{i,s,j} - \sum_{i' \in N, i' \neq i} x_{i',s,j+1} \leq w_{i,s}, \forall i \in N, \forall j \in N \setminus \{n\}, \forall s \in S \quad (9)$$

$$x_{i,s,n} \leq w_{i,s}, \forall i \in N, \forall s \in S \quad (10)$$

$$w_{i,s} + x_{i',s,1} \leq 1 + z_{i,i',s}, \forall (i, i') \in N^2, i \neq i', \forall s \in S \quad (11)$$

Les contraintes (2) garantissent que chaque opération est affectée à une station. Les contraintes (3) expriment que chaque station a au plus une opération affectée à chaque position de la séquence. Les contraintes (4) (resp. (5)) interdisent qu'une position (resp. station) soit occupée alors que la précédente ne l'est pas. Les contraintes de précédence (resp. temps de cycle) sont garanties par (6) (resp. (7)). Les contraintes (8) lient entre elles les variables $x_{i,s,j}$ et $z_{i,i',s}$. Les contraintes (9) et (10) force la variable $w_{i,s}$ à prendre la valeur 1 pour la dernière opération de la séquence. La contrainte (11) garantit que le temps de réglage entre la pièce en dernière position et celle en première position sur chaque station sera bien pris en compte, de manière cyclique, dans les contraintes (7).

Les différentes approches de résolution

Le modèle précédent, par l'intermédiaire des variables et des contraintes, définit l'espace des solutions de notre problème.

Le fait de travailler sur un espace des représentations formé, pour chaque station, d'une séquence d'opérations, permet bien d'instancier toutes les variables du problème. Cet espace des représentations est donc direct et complet.

Nous pouvons d'ailleurs remarquer que la connaissance des seules variables x permet d'instancier toutes les autres variables du modèle. Ces variables intègrent à la fois les décisions au niveau de l'équilibrage (répartition des opérations sur les stations) et du séquençement.

Pendant ces travaux de thèse, Des approches par décomposition ont été proposées en hiérarchisant ses deux types de décision.

Nous avons d'abord conçu une mathéuristique en décomposant le problème avec une approche de type équilibrage puis séquençement (Balance First – Sequence Last). Le fait de relaxer le séquençement permet de simplifier le modèle mathématique. En intégrant une estimation des temps de réglage et en résolvant de manière exacte le sous-problème d'équilibrage, une première solution est obtenue qui est ensuite optimisée avec une métaheuristique de type recuit simulé.

Nous avons ensuite proposé une approche de type séquençement puis équilibrage (Sequence First – Balance Last). A partir d'une séquence géante formée par toutes les opérations, le problème d'équilibrage est résolu avec une méthode de type Split. C'est cette approche que nous allons maintenant détailler.

Illustration de la procédure Split

Cette approche est très similaire à celle décrite pour le problème de livraison par drone. Pour autant, si l'approche est connue depuis (Beasley, 1983), et reconnue depuis (Prins, 2004) dans

le contexte des problèmes de tournées de véhicules, elle n'avait jamais été mise en œuvre sur des problèmes d'équilibrage.

La Figure 27 illustre la démarche sur un exemple à 5 opérations. Les temps de réglages inter-opération et les temps de production sont donnés. Un graphe auxiliaire $G^r = (X^r, A^r)$ est ensuite construit avec :

- $X^r = \{0\} \cup N$, la liste des opérations plus un sommet fictif 0.
- A^r est l'ensemble des arcs (i, j) tels que $\sum_{k=i+1}^j p_k + \sum_{k=i+1}^{j-1} t_{k,k+1} + t_{j,i+1} \leq C$. L'arc $(i, j) \in A^r$ indique que les opérations situées entre $i+1$ et j sont situées sur la même machine.

La valeur indiquée sur les arcs est donnée à titre indicatif. Elle indique la charge de la station dans le cas où cet arc est emprunté. La charge de la station est égale à la somme des temps d'exécution et des temps de réglage des opérations qui lui sont affectées. Cette valeur doit être inférieure au temps de cycle (ici $C=8$).

Le nombre optimal de stations correspond à la résolution du problème de plus court chemin (en termes de nombre d'arcs empruntés) entre les sommets 0 et 5. Le plus court chemin est ici de longueur 2 (0-3-5). Il correspond à l'équilibrage suivant. Les opérations 1, 2 et 3 sont affectées à la station 1. Les opérations 4 et 5 sont affectés à la station 2.

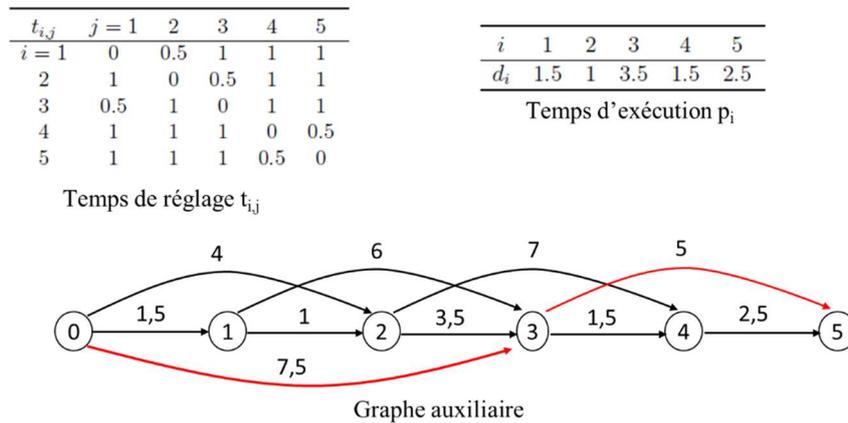


Figure 27. Illustration du split : matrices des temps et construction du graphe avec $C=8$.

Application à la résolution de problèmes d'équilibrage

Pendant ces travaux de thèse, nous avons établi que plusieurs variantes du problème d'équilibrage pouvaient être résolues en temps polynomial, en se ramenant à des recherches de chemins dans le graphe auxiliaire. Le SDSALBP que nous avons présenté ici ne figure d'ailleurs pas dans les variantes proposées dans la thèse (c'est un cas particulier de la variante appelée RTLBP).

Sans rentrer dans les détails de ces variantes, nous montrons dans le Tableau 7 un récapitulatif des résultats obtenus. Ce qui est important de noter ici, c'est le caractère générique de la démarche, qui montre que la procédure Split peut être appliquée dans de nombreux cas. Certaines de ces variantes seulement ont été mises en œuvre, offrant l'opportunité de nombreuses perspectives.

Tableau 7. Récapitulatif de quelques variantes du SALBP pouvant être résolues avec l'approche SPLIT

Problem	Underlying graph problem	Time complexity
SALBP-2	Constrained min-max path	$O(n^4)$
RTLBP	Constrained shortest path	$O(n^4)$
RALBP-1/SDRALBP-1	Shortest path	$O(n^2)$
RALBP-2/SDRALBP-2	Constrained min-max path	$O(n^4)$

Ces travaux ont fait l'objet de trois publications dans des revues internationales (Lahrichi et al., 2021a, 2021b et 2023)

4.3.3. RETOUR SUR EXPERIENCE POUR LES APPROCHE PAR DECOMPOSITION

Les deux problèmes que nous venons de présenter sont similaires dans leur structure, en ceci qu'ils contiennent plusieurs types de variables de décision.

Si l'on reprend le modèle mathématique du PDSTSP, proposé par (Murray & Chu, 2015), il comprend des variables d'affectation au véhicules (\hat{x}_{ij} : le véhicule emprunte l'arc (i, j) pour desservir le client j et y_{iv} : le client i est affecté au drone v) et des variables \hat{u}_i , liées aux variables \hat{x}_{ij} , et qui définissent le séquençement des clients pour le véhicule. On retrouve pour ce problème :

- des approches de type Sequence First – Cluster Second, qui consistent à définir un tour (ou séquence) géant, qui contient l'ensemble des clients à livrer, puis à le décomposer pour affecter ces clients aux véhicules,
- des approches de type Cluster First – Sequence second, qui consistent à faire d'abord une répartition des clients aux véhicules, puis à optimiser les tournées.

Concernant le problème d'équilibrage de lignes d'assemblage avec temps de réglage dépendant de la séquence, nous avons également deux types de décisions concernant la répartition (ou équilibrage) des opérations sur les postes de charge (balance) et le séquençement des opérations sur les postes (sequence), donnant lieu aux approches Balance First – Sequence Second ou Sequence First – Balance Second.

Pour ces deux problèmes, les approches de type Sequence First ne sont pas les plus naturelles, mais elles sont ici très efficaces car le sous problème d'éclatement (des clients aux véhicules ou des opérations aux postes de charge) peut se résoudre de manière optimale grâce à la programmation dynamique et la recherche de plus courts chemins dans les graphes auxiliaires (SPLIT). En quelques sortes, nous travaillons sur un espace des représentations qui est une séquence (et donc une permutation), et SPLIT agit comme un décodeur qui reconstruit la solution et l'évalue. Cependant, la recherche de ce plus court chemin nécessite de gérer des labels dont le nombre explose avec la taille du problème. Des techniques de troncatures peuvent alors être mises en place, en limitant le nombre de labels affiliés à chaque sommet, mais avec comme conséquence la perte de l'optimalité.

Que ce soit dans les travaux de thèse de Raïssa ou de Youssef, nous avons peu investigué ces techniques de troncature, et leur impact sur la qualité de la solution. Cette piste mériterait d'être approfondie.

Un autre point éveille notre curiosité. Une fois l'éclatement réalisé, le schéma de la métaheuristique requiert de reconstituer une ou des représentations, donc de nouvelles séquences géantes. Ceci est fait par l'intermédiaire de mécanismes relativement arbitraires. Il est difficile de déterminer des critères qui indiquent a priori que des séquences géantes sont de plus ou moins bonne qualité.

Enfin, nous avons pu montrer, par l'intermédiaire de ces travaux, que SPLIT pouvait éclater des séquences géantes et répondre à des problématiques très diverses. Pour les problèmes de livraison, nous avons étendu la technique classique du VRP pour inclure la gestion des drones, puis passer à plusieurs véhicules. Pour les problèmes d'équilibrages, nous avons montré que de nombreuses variantes pouvaient être résolues par ce type d'approche. Il est vraisemblable de pouvoir étendre ces résultats à d'autres problèmes dès lors qu'une des décisions à prendre porte sur un séquençement d'éléments.

4.4. RESOUDRE LES PROBLEMES DE DEMAIN

Au travers des deux derniers problèmes que nous venons d'évoquer, nous avons abordé deux problématiques importantes, la logistique du dernier kilomètre par le biais de l'étude de problèmes de livraison, et l'usine 4.0 avec l'étude de systèmes flexibles et reconfigurables.

Ces problématiques s'inscrivent dans un cadre plus large, dans lequel les futures décisions à prendre devront permettre aux femmes et aux hommes de répondre à des enjeux sociétaux, environnementaux et économiques.

Au travers de quelques exemples sur lesquels nous avons travaillé ces dernières années dans l'axe ODPS (Outils Décisionnels pour la Production des Systèmes) du LIMOS, nous tenterons d'illustrer ces enjeux et d'introduire quelques perspectives de recherche à court ou moyen terme.

- Le premier exemple est les Circuits Courts Alimentaires de Proximité (CCAP). Les Français sont de plus en plus sensibles à la provenance des produits alimentaires qu'ils achètent. Pourtant, contrairement à ce que nous pourrions croire, il est parfois moins coûteux de faire venir un aliment de l'autre bout du monde, que de la même région. Des réseaux de distribution efficaces doivent être conçus, dimensionnés puis pilotés afin de répondre à ce défi. Nous avons commencé à collaborer sur cette problématique en partenariat avec le département de l'Allier et avec le laboratoire Clermont Recherche Management (CleRMa). Une thèse a commencé en octobre 2022 sur cette thématique. Notre objectif est d'imaginer un certain nombre de scénarii, intégrant des stratégies de mutualisation des transports pour aller des producteurs vers les consommateurs, et des dépôts intermédiaires, et d'en mesurer les performances. Les problèmes que nous pourrions être amenés à résoudre se ramènent à des problèmes de location routing problem. Ils intègrent des problématiques de conception du réseau de distribution, de dimensionnement et de pilotage.
- Il devient urgent d'agir pour mieux maîtriser notre consommation d'énergie. Alors que des échéances sont fixées en matière de décarbonation (fin des moteurs thermiques dans l'industrie automobile en 2035, objectifs de neutralité carbone en France à l'horizon 2050), la transition énergétique doit s'accélérer et les industriels en seront des acteurs majeurs. Des solutions devront être apportées au niveau de la maîtrise de consommation énergétique. Nous avons commencé à travailler sur des problèmes de minimisation des

pics de consommation dans des problèmes d'équilibrages de lignes d'assemblage. Ces problèmes sont intéressants en ceci que dans l'horizon de production, l'objectif est de répartir la charge des opérations à planifier pour lisser la consommation énergétique. Cela va à l'encontre des techniques classiques que nous avons pu mettre en œuvre pour résoudre les problèmes d'ordonnancement, et qui consistent à débiter l'exécution d'une opération dès lors que toutes les ressources nécessaires sont disponibles (ordonnancement semi-actifs). Pour chaque opération, le début d'exécution doit être inclus dans une fenêtre de temps. Nous envisageons de mettre en place des méthodes hybrides métaheuristiques / programmation par contraintes.

- Un des enjeux de l'usine 4.0 est de concevoir des outils de production ayant une productivité, une flexibilité et une agilité accrue. L'ensemble des travaux que nous avons pu mener sur les Systèmes Flexibles de Production et plus récemment la thèse de Youssef Lahrichi s'inscrivent dans ce thème. Ces travaux ont permis d'étudier des lignes d'assemblages flexibles et robotisées. A l'avenir, en nous impliquant dans le laboratoire Factolab, né de l'association du LabEx IMobS3 et du groupe Michelin, nous souhaiterions travailler sur les problématiques de l'usine du futur. Nous avons obtenu un financement pour un post-doctorat qui va débiter au mois de juin 2023. Ce travail est réalisé en collaboration avec les usines Michelin. L'objectif est de concevoir sur un horizon de 10 ans, une usine ultra-flexible, capable de réaliser une production de masse.

Nous l'avons mentionné lors du dernier point. Le Laboratoire d'Excellence IMobS3 offre l'opportunité aux chercheurs clermontois de pouvoir s'impliquer dans ces problématiques au travers d'appels à projets qui permettent d'obtenir des financements des travaux (stages, post-doctorats, thèses). Nous avons bénéficié ces dernières années des financements pour les thèses de Raïssa Saleu (sur le thème de la mobilité), de Youssef Lahrichi (sur le thème de l'usine 4.0), de sujets de stage de master (Safa Mrad en 2021 et Arnaud Tuyaba en 2022 sur des problèmes de lignes d'assemblage) et un sujet de post-doc en 2023 (Hector Gatt sur le thème de l'usine 4.0).

J'ai eu l'occasion de présenter en 2022 les travaux de thèse de Youssef Lahrichi, et en 2023 ceux de Raïssa Saleu, lors du séminaire annuel du Centre Internationale de Recherche "Innovative Transportation and Production Systems" (CIR ITPS).

Toutes ces problématiques, ont permis de faire émerger des problèmes d'optimisation originaux et difficiles, pour lesquels il reste encore beaucoup de recherches à accomplir, et beaucoup de méthodes d'optimisation à concevoir.

5. CONCLUSION GENERALE ET PERSPECTIVES

Au travers de ce mémoire, nous avons présenté les métaheuristiques en portant notre attention non pas sur le système biologique, physique ou social qu'elles cherchent à simuler, mais sur les mécanismes et concepts généraux qu'elles utilisent pour s'orienter efficacement dans l'espace de recherche.

Le système réel introduit un champ métaphorique. Ainsi, chaque métaheuristique possède son propre vocabulaire pour définir les éléments du système et leurs interactions. Ces éléments sont modélisés par des lois et des paramètres, qui relient le métaheuristicien et la métaheuristique. De son expertise dépendra grandement la qualité du paramétrage de la métaheuristique, ainsi que de l'efficacité de son comportement. Ce travail dépend beaucoup de la métaheuristique et moins des problèmes. L'objectif recherché à ce niveau est de définir une stratégie de pilotage dans l'espace de recherche et définir un équilibre entre l'exploration et l'exploitation.

Maintes et maintes fois, nous avons pu lire dans des articles que telle métaheuristique était particulièrement efficace pour tel problème, mais cela a-t-il réellement un sens ? Y a-t-il un lien direct entre l'implémentation d'une métaheuristique et son principe général ? Nous avons pris le parti de croire que pour qu'une métaheuristique soit efficace, il faut nécessairement qu'elle soit bien réglée (ce que sait faire tout spécialiste), mais c'est en aucun cas suffisant. Deux autres ingrédients au moins seront nécessaires :

- s'appuyer sur les propriétés du problème pour intégrer des connaissances spécifiques, sous forme d'algorithmes performants,
- implémenter ces algorithmes de manière efficace.

Le deuxième point a été abordé de manière très sporadique dans ce mémoire, lors de la description de nos travaux sur le flow-shop de permutation. L'implémentation de Taillard permet de considérer très rapidement le voisinage d'une permutation et la performance d'une métaheuristique, quelle qu'elle soit, s'en trouve grandement améliorée. Ces aspects sont très souvent passés sous silence dans les articles scientifiques où l'on reste en général au niveau des algorithmes de principe, mais ils sont indispensables si on espère obtenir des résultats compétitifs. Par exemple, lors de l'évaluation d'une solution d'un problème d'ordonnancement, on peut gagner beaucoup de temps en ne recalculant que partiellement les dates de début et/ou de fin d'exécution des tâches, à partir du moment où la solution voisine commence à différer de la solution courante.

Nous avons davantage insisté sur le premier point. Le problème du voyageur de commerce a été souvent utilisé pour illustrer nos propos et nous avons pu montrer à de multiples reprises comment l'utilisation de connaissances sur des attributs (par exemple les arêtes les moins coûteuses ou les plus *désirables*), pouvait permettre d'améliorer la qualité de l'exploration et l'exploitation de l'espace de recherche.

L'exploration et l'exploitation viennent presque inmanquablement dans la discussion. Si nous devons les résumer, nous dirions que ces notions caractérisent toutes deux le fait de chercher des bonnes solutions ; à proximité d'autres bonnes solutions pour l'exploitation, dans des zones de l'espace inexplorées pour l'exploration. Loin d'être opposées, ces notions sont complémentaires et des mécanismes peuvent contribuer aux deux. C'est ainsi que la qualité de l'exploration ou de l'exploitation peuvent se définir, au travers de la qualité des solutions que les mécanismes les mettant en œuvre permettent d'obtenir.

Au cours des différents chapitres, nous avons essayé de comprendre comment ces notions pouvaient se mettre en place.

Le premier chapitre a simplement permis de poser le contexte de l'optimisation combinatoire, et de rappeler que les métaheuristiques sont là pour résoudre des problèmes d'optimisation difficiles. Le théorème *No Free Lunch* fournit un premier renseignement de choix. Pour conclure ce chapitre, nous disions qu' "*une heuristique, pour être performante, devra nécessairement exploiter la structure d'un problème*". Nous avons pu voir par la suite qu'un choix judicieux de mouvements permettait en effet de construire un bon voisinage dans l'espace des solutions, et par suite de mettre en place une stratégie d'exploitation efficace.

Dans le deuxième chapitre, les concepts des métaheuristiques sont abordés. Nous avons proposé de dissocier l'espace des solutions de l'espace des représentations. Ce découpage permet de clairement distinguer les mécanismes liés au pilotage de la métaheuristique, qui ne portent que sur les représentations, des mécanismes liés au problème. Nous avons également décrit au travers d'exemples concrets, comment générer de nouvelles représentations, à proximité d'autres représentations (mouvements de base ou élaborés, mise en place de l'exploitation) ou avec d'autres techniques (exploration). C'est l'ensemble de ces techniques, combinées à l'utilisation de mémoires, qui permettent de mettre en œuvre une stratégie de pilotage dans l'espace de recherche. L'exploitation est associée avec une stratégie à court terme avec une recherche intensive à proximité de solutions élites, tandis que l'exploration est une stratégie sur le long terme, avec la recherche de nouvelles zones prometteuses.

Les exemples de mécanismes qui ont servi à illustrer les concepts appartiennent aux métaheuristiques que nous appelons historiques, qui ont été développées dans les années 1980s et 1990s. Le troisième chapitre est consacré aux principales avancées dans les deux dernières décennies. Un pan important de la recherche est consacré au développement de nouvelles métaheuristiques, mais peu d'entre elles apportent de réelles innovations. Plus prometteuses, sont les différentes formes d'hybridation, qui ont permis de concevoir des méthodes utilisant des idées provenant d'autres domaines de la Recherche Opérationnelle ou de l'Intelligence Artificielle. Ces méthodes ont permis de travailler sur des problèmes beaucoup plus difficiles. Deux formes d'hybridation sont détaillées, qui correspondent aux applications que nous avons eu l'occasion de mettre en œuvre : l'hybridation des métaheuristiques avec la recherche locale et avec des méthodes exactes.

Le dernier chapitre propose une synthèse de quelques-uns des travaux que nous avons menés. Les thèmes choisis sont :

- le flow-shop de permutation : pour ce problème, nous avons proposé des mouvements élaborés et une technique d'implémentation efficace qui ont permis d'obtenir d'excellents résultats sur des instances très étudiées.
- les systèmes flexibles de production : une série de trois problèmes, demandant une synchronisation des ressources de production (atelier de type job-shop ou job-shop flexible) et des ressources de transport (AGV) sont décrits. Nous avons traité des instances académiques de petite taille avec de très bons résultats.
- les approches par décomposition : nous avons mis en œuvre deux approches de résolution similaire, en construisant d'abord une séquence géante qui est ensuite décomposée à l'aide d'une approche de programmation dynamique (SPLIT). Ces approches ont été appliquées pour résoudre un problème de distribution de colis avec des véhicules et des drones, et un problème d'équilibrage de lignes d'assemblage avec des temps de réglages dépendant de la séquence. Ce type d'approche est classique pour

les problèmes de tournées de véhicules. Nous l'avons étendu avec succès pour des problèmes d'ateliers.

Les réflexions que nous avons pu mener ouvrent de nombreuses perspectives, à la fois sur les méthodes et sur les problèmes.

Concernant les métaheuristiques, nous souhaitons continuer à monter en compétences sur l'ouverture à de nouveaux domaines pour concevoir de nouvelles formes d'hybridation.

- Métaheuristiques / PPC : Des réflexions ont commencé pour des hybridations de métaheuristiques avec la programmation par contraintes, qui permettraient de travailler de manière similaire à une hybridation métaheuristique / programmation linéaire, en remplaçant le solveur linéaire (pour résoudre un sous-problème d'optimisation) par un solveur SAT (pour résoudre un sous-problème de faisabilité).
- Métaheuristiques / Machine Learning : Utiliser une bibliothèque d'instances test comme données d'entrées pour adapter par exemple le paramétrage ou le choix de composants à appliquer sur de nouvelles instances.

Nous avons également négligé dans beaucoup de nos précédents travaux le fait de faire une exploration de bonne qualité. A notre décharge, le fait de travailler sur des applications industrielles, souvent difficiles en raison de leur taille et de leur structure, limite le nombre d'itérations dont dispose la métaheuristique dans un temps raisonnable. L'exploration, qui est une stratégie de long terme, n'a pas toujours suffisamment de temps pour pouvoir se mettre en place.

Concernant les problèmes, nous avons eu l'occasion de travailler récemment sur des problématiques qui correspondent à des enjeux sociétaux et environnementaux importants. La logistique du dernier kilomètre avec la livraison par drone, et la production sous contrainte énergétique en considérant la minimisation de pics de consommation dans des lignes d'assemblage. Ces travaux ont été menés dans le cadre du projet CAP 20-25 du I-Site Clermont, au niveau du CIR ITPS (Centres internationaux de recherche Systèmes Innovants pour les transports et la Production) et du LabEx Imobs3 (Laboratoire d'Excellence Mobilité Innovante). Deux autres projets importants sont en phase de lancement sur les thèmes suivants :

- L'industrie du futur : imaginer une usine du futur, qui permettrait de transformer une production du paradigme taylorien vers une usine ultra flexible. Ce projet concerne les phases de conception, de dimensionnement et de pilotage d'une nouvelle usine. Il se déroule dans le cadre du laboratoire Factolab, un laboratoire public-privé créé par le groupe Michelin et le LabEx Imobs3 et devrait financer des post-docs et des thèses sur les dix prochaines années.
- Les Circuits Courts Alimentaires de Proximité (CCAP) : Une thèse a débuté autour de la mise en place des circuits courts. Nous réfléchissons actuellement à la proposition de schémas organisationnels pour pouvoir les comparer. Ces travaux sont menés en collaboration avec le laboratoire CLerMa (Clermont Recherche Management) et le département de l'Allier.

6. BIBLIOGRAPHIE

- Abdelmaguid, T. F., Nassef, A. O., Kamal, B. A., & Hassan, M. F. (2004). A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *International journal of production research*, 42(2), 267-281.
- Applegate, D., Bixby, R., Chvatal, V., & Cook, W. (1999). Finding tours in the TSP. Preliminary version of a book chapter available via <https://scholarship.rice.edu/bitstream/handle/1911/101911/TR99-05.pdf?sequence=1>, 2000.
- Beasley, J. E. (1983). Route first—cluster second methods for vehicle routing. *Omega*, 11(4), 403-408.
- Bilge, Ü., & Ulusoy, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research*, 43(6), 1058-1070.
- Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied soft computing*, 11(6), 4135-4151.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3), 268-308.
- Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., & Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, 64, 1695-1724.
- Clerc, M. (2004). Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem. In G.C. Onwobulu, B.V. Babu (eds) *New Optimization Techniques in Engineering*, 219-239, Springer, Berlin.
- Clerc, M., & Siarry, P. (2004). Une nouvelle métaheuristique pour l'optimisation difficile: la méthode des essaims particulaires. *J3eA* 3, 007.
- Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, 45(3), 1-33.
- Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group.
- Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, 6(6), 791-812.
- Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America* 2(4), 393-410.
- Delahaye, J. P. (1995). *Logique, informatique et paradoxes*. Pour la science.
- Deroussi, L., Gourgand, M., Kemmoe, S., & Quilliot, A. (2006a). Discrete particle swarm optimization for the permutation flow-shop problem. Technical report, LIMOS CNRS UMR 6158.
- Deroussi, L., Gourgand, Norre, S., (2006b). New effective neighborhoods for the permutation flow shop problem. 2006. hal-00678053.
- Deroussi, L., Gourgand, M., & Tchernev, N. (2008). A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, 46(8), 2143-2164.

- Deroussi, L., & Norre, S. (2010). Simultaneous scheduling of machines and vehicles for the flexible job shop problem. In *International conference on metaheuristics and nature inspired computing* (pp. 1-2). Tunisia: Djerba Island.
- Deroussi, L., Gourgand, M. (2014) A scheduling Approach for the design of Flexible Manufacturing Systems. *Heuristics: Theory and Applications*, Nova science publishers, Chapter 9, 2014. (hal-03263543)
- Deroussi, L. (2016). *Metaheuristics for logistics*. John Wiley & Sons.
- Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation* 1 (1), 53-66.
- Drake, J. H., Kheiri, A., Özcan, E., & Burke, E. K. (2020). Recent advances in selection hyperheuristics. *European Journal of Operational Research*, 285(2), 405-428.
- Dumitrescu, I., & Stützle, T. (2003). Combinations of local search and exact algorithms. In *Applications of Evolutionary Computing: EvoWorkshops 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM* Essex, UK, April 14–16, 2003 *Proceedings* (pp. 211-223). Springer Berlin Heidelberg.
- Edmonds, J., & Johnson, E. L. (1973). Matching, Euler tours and the Chinese postman. *Mathematical programming*, 5, 88-124.
- Eiben, A. E., & Schippers, C. A. (1998). On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4), 35-50.
- Fleury, G. (1993). *Méthodes stochastiques et déterministes pour les problèmes NP-difficiles*, these de doctorat, Université Blaise Pascal, Clermont 2).
- Gasarch, W. I. (2019). Guest column: The third P=? NP poll. *ACM SIGACT News*, 50(1), 38-59.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5), 533-549.
- Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, 1(3), 190-206.
- Glover, F. (1992). New ejection chain and alternating path methods for traveling salesman problems. In *Computer science and operations research* (pp. 491-509). Pergamon.
- Glover, F. (1997). Tabu search and adaptive memory programming—advances, applications and challenges. *Interfaces in computer science and operations research: Advances in metaheuristics, optimization, and stochastic modeling technologies*, 1-75.
- Glover, F., & Laguna, M. (1998). *Tabu search* (pp. 2093-2229). Springer US.
- Glover, F., Laguna, M., & Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29(3), 653-684.
- Gmys, J., Mezmaç, M., Melab, N., & Tuytens, D. (2020). A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem. *European Journal of Operational Research*, 284(3), 814-833.
- Gmys, Jan. (2021). *Permutation Flow-shop : best-known makespans and schedules for Taillard and VFR benchmarks [Data set]*. Zenodo. <https://doi.org/10.5281/zenodo.4542886>
- Gmys, J. (2022). Exactly solving hard permutation flowshop scheduling problems on peta-scale GPU-Accelerated supercomputers. *INFORMS Journal on Computing*, 34(5), 2502-2522.

- Hao, J. K., & Solnon, C. (2014). *Méta-heuristiques et intelligence artificielle*.
- Helsgaun, K. (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European journal of operational research*, 126(1), 106-130.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*: MIT press.
- Homayouni, S. M., & Fontes, D. B. (2021). Production and transport scheduling in flexible job shop manufacturing systems. *Journal of Global Optimization*, 79(2), 463-502.
- Hoos, H. H., & Stützle, T. (2004). *Stochastic local search: Foundations and applications*. Elsevier.
- Jourdan, L., Basseur, M., & Talbi, E. G. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, 199(3), 620-629.
- Lahrichi, Y., Deroussi, L., Grangeon, N., & Norre, S. (2021a). A balance-first sequence-last algorithm to design RMS: a matheuristic with performance guaranty to balance reconfigurable manufacturing systems. *Journal of Heuristics*, 27, 107-132.
- Lahrichi, Y., Grangeon, N., Deroussi, L., & Norre, S. (2021b). A new split-based hybrid metaheuristic for the reconfigurable transfer line balancing problem. *International Journal of Production Research*, 59(4), 1127-1144.
- Lahrichi, Y., Damand, D., Deroussi, L., Grangeon, N., & Norre, S. (2023). Investigating two variants of the sequence-dependent robotic assembly line balancing problem by means of a split-based approach. *International Journal of Production Research*, 61(7), 2322-2338.
- Lawler, E. L. (1985). *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics.
- Lemoine, D. (2008). *Modèles génériques et méthodes de résolution pour la planification tactique mono-site et multi-site* (Doctoral dissertation, Université Blaise Pascal-Clermont-Ferrand II).
- Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2), 498-516.
- Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search (pp. 320-353). Springer US.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
- Martin, O. C., & Otto, S. W. (1996). Combining simulated annealing with local search heuristics. *Annals of operations research*, 63(1), 57-75.
- Monge, G. (1781). *Mémoire sur la théorie des déblais et des remblais*. *Mem. Math. Phys. Acad. Royale Sci.*, 666-704.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826, 37.
- Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86-109.
- Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations research*, 63, 511-623.

- Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & operations research*, 31(12), 1985-2002.
- Puchinger, J., & Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach: First International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2005, Las Palmas, Canary Islands, Spain, June 15-18, 2005, Proceedings, Part II 1* (pp. 41-53). Springer Berlin Heidelberg.
- Raidl, G. R. (2006). A unified view on hybrid metaheuristics. In *Hybrid Metaheuristics: Third International Workshop, HM 2006 Gran Canaria, Spain, October 13-14, 2006 Proceedings 3* (pp. 1-12). Springer Berlin Heidelberg.
- Raidl, G. R., Puchinger, J., & Blum, C. (2019). Metaheuristic hybrids. *Handbook of metaheuristics*, 385-417.
- Reinelt, G. (1994). *The traveling salesman: computational solutions for TSP applications*.
- Saleu, R. G. M., Deroussi, L., Feillet, D., Grangeon, N., & Quilliot, A. (2018). An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, 72(4), 459-474.
- Saleu, R. G. M., Deroussi, L., Feillet, D., Grangeon, N., & Quilliot, A. (2022). The parallel drone scheduling problem with multiple drones and vehicles. *European Journal of Operational Research*, 300(2), 571-589.
- Savage, S. L. (1976). Some theoretical implications of local optimization. *Mathematical Programming*, 10, 354-366.
- Schirmer, A. (1995). *A guide to complexity theory in operations research* (No. 381). Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.
- Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., & Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist?. *Swarm and Evolutionary Computation*, 54, 100671.
- Reeves, C. R., & Yamada, T. (1998). Genetic algorithms, path relinking, and the flowshop sequencing problem. *Evolutionary computation*, 6(1), 45-60.
- Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). *Handbooks in operations research and management science*, 12, 1-68.
- Sörensen, K. (2015). Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1), 3-18.
- Stutzle, T., & Hoos, H. (1997, April). MAX-MIN ant system and local search for the traveling salesman problem. In *Proceedings of 1997 IEEE international conference on evolutionary computation (ICEC'97)* (pp. 309-314).
- Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47, 67-74.
- Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. *Journal of heuristics*, 8, 541-564.
- Talbi, E. G. (2013). A unified taxonomy of hybrid metaheuristics with mathematical programming, constraint programming and machine learning. *Hybrid Metaheuristics*, 3-76.
- Tasgetiren, M. F., Sevkli, M., Liang, Y. C., & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for permutation flowshop sequencing problem. In *Ant Colony*

Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004, Brussels, Belgium, September 5-8, 2004. Proceedings 4 (pp. 382-389). Springer Berlin Heidelberg.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *J. of Math*, 58(345-363), 5.

Ulusoy, G., Sivrikaya-Şerifoğlu, F., & Bilge, Ü. (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, 24(4), 335-351.

Vasquez, M., & Hao, J. K. (2001, August). A hybrid approach for the 0-1 multidimensional knapsack problem. In *IJCAI* (pp. 328-333).

Zhang, Q., Manier, H., & Manier, M. A. (2012). A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. *Computers & Operations Research*, 39(7), 1713-1723.

TABLE DES FIGURES

Figure 1. Quelques rappels sur les relations d'ordre et de préordre.	25
Figure 2. Relation entre les 4 types de problèmes (d'après Schirmer, 1995).	31
Figure 3. Le point de Fermat-Torricelli. Le point P est celui qui minimise l'expression $MA + MB + MC$	32
Figure 4. Le problème des 7 ponts de Königsberg (Euler, 1736 ©BY-NC-ND 4.0).....	33
Figure 5. Solution d'un problème de voyageur de commerce avec 45 villes allemandes, issu du manuel de 1832 (en trait plein). Cette solution a un coût de 1285 km. La solution optimale (en pointillés) a un coût de 1248 km.	35
Figure 6. Espace des solutions versus espace des représentations.	41
Figure 7. Schémas d'encodage et de décodage pour le problème du sac-à-dos.....	42
Figure 8. Illustration de croisements classiques dans les Algorithmes Génétiques.	43
Figure 9. Illustration du principe de la roulette avec une population de 6 individus et $f_j = j$. L'individu n°6 (repéré par le secteur le plus grand), a une probabilité d'être choisi égale à 6/21 (environ 28.6%). Pour l'individu 1 (secteur le plus petit), cette probabilité chute à 1/21 (4,8%).	45
Figure 10. Illustration du tour de Cristofides.	47
Figure 11. Illustration des mouvements de base d'une permutation : Echange, Insertion et Inversion.	49
Figure 12. Zoom sur les solutions du TSP pour le problème des 45 villes allemandes. A gauche la solution proposée en 1830. A droite la solution optimale. Un mouvement 4-opt séquentiel permet de passer de la solution de gauche à celle de droite.	50
Figure 13. Exemple d'une structure "Stem-and-Cycle" avec les solutions reconstituées.	50
Figure 14. Croisement à 1 point de rupture appliqué à des permutations.....	52
Figure 15. Vue unifiée des concepts d'exploitation et d'exploration pour les métaheuristiques (traduit et adapté de Blum & Roli, 2003).....	55
Figure 16. Cartographie des cinq types de mémoire humaine réalisée à partir du site de l'Inserm (https://www.inserm.fr/dossier/memoire),	56
Figure 17. Influence du choix du mouvement pour l'exploitation.	58
Figure 18. Illustration d'une technique de guidage pour le voisinage 2-opt.	60
Figure 19. Trouver le point commun entre ces images.	63
Figure 20. Dans le monde des métaheuristiques, ce qui compte, ce n'est pas seulement la carrosserie (à gauche), c'est aussi le pilote ! (l'image de droite est extraite d'une présentation de K. Sörensen).	65
Figure 21. Les métaheuristiques au fil du temps.....	66
Figure 22. Schéma de principe de l'hybridation métaheuristique / Recherche locale.....	71
Figure 23. Exemple de représentation avec 2 véhicules, 2 pièces et 6 opérations de transport	84
Figure 24. Proposition d'un jeu d'essai pour le job-shop flexible avec transport.....	86
Figure 25. Illustration du split (1) : matrices des coûts et construction du graphe auxiliaire. .	92
Figure 26. Illustration du split (2) : propagation des labels et reconstitution du plus court chemin.	93
Figure 27. Illustration du split : matrices des temps et construction du graphe avec $C=8$	96

